Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The CWI is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O).

CWI Tract 70

Twenty-five years of operations research in the Netherlands: Papers dedicated to Gijs de Leve

edited by
Jan Karel Lenstra
Henk Tijms
Ton Volgenant

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

1980 Mathematics Subject Classification: 90Bxx, 90Cxx.
ISBN 90 6196 385 0
NUGI-code: 811

Preface

In the past twenty-five years much has changed. Not the weather; that is still beyond our control. But the scientific climate did change, particularly in the field of operations research. On January 1, 1965, Gijs de Leve was appointed Professor in the Mathematics of Operations Research at the University of Amsterdam, in addition to his position at the Mathematical Centre in Amsterdam. At that time operations research was hardly recognized as a discipline in the Netherlands, neither in academia nor in industry. Twenty-five years later, the situation is quite different. Dutch researchers are now preeminent, both in stochastic optimization and in mathematical programming. Many of them originate from the Amsterdam school of Gijs de Leve.

The reputation of a scientist is not always determined by his number of publications and the degree of mathematical sophistication achieved therein. Although creativity and sparkling ideas are characteristic of Gijs de Leve, he prefers to put others on the scent of new and exciting research. Eight dissertations with a broad spectrum of topics were completed under his supervision. Not only did he create a most stimulating research environment, but he has also invested much energy in the popularization of operations research in Dutch government and industry. Among his initiatives were the so-called 'working-weeks', where people from practice learned about new developments in the area. He also took the lead in establishing the national Research Community in the Mathematics of Operations Research and the annual Lunteren Conference, where Dutch OR workers meet each other informally. Binding people and groups together is the trademark of Gijs de Leve. It is with good reason that he is called the godfather of operations research in the Netherlands.

The initiative to compile a Liber Amicorum for Gijs de Leve on the occasion of his silver jubilee as Professor at the University of Amsterdam met with an enthusiastic response from his (ex-)associates and students. Our editorial tasks were further alleviated by the willingness of the CWI to publish this book, by the indefatigable efforts of Wim Aspers and Yvonne Samseer in producing it, and by the help of the CWI library staff in compiling the list of publications.

Amsterdam/Eindhoven
December 1989

Jan Karel Lenstra
Henk Tijms
Ton Volgenant

Gijs de Leve

Publications of Gijs de Leve

1. Publications of the Mathematical Centre, Amsterdam

1.1. Mathematical Centre Tracts

G. de Leve (1964). *Generalized Markovian decision processes, part I; model and method*, Mathematical Centre Tracts 3. [Part 1 of dissertation, defended at the University of Amsterdam, May 20, 1964.]

G. de Leve (1964). *Generalized Markovian decision processes, part II; probabilistic background*, Mathematical Centre Tracts 4. [Part 2 of dissertation, defended at the University of Amsterdam, May 20, 1964.]

G. de Leve, H.C. Tijms, P.J. Weeda (1970). *Generalized Markovian decision processes; applications*, Mathematical Centre Tracts 5.

1.2. MC Syllabi and CWI Syllabi

G. de Leve, W. Molenaar (1966). *Leergang besliskunde, deel 4; Markovketens en wachttijden*, MC Syllabus 1.4.

J. Kriens, G. de Leve (1966). *Leergang besliskunde, deel 5; inleiding tot de mathematische besliskunde*, MC Syllabus 1.5.

G. de Leve (1968). *Leergang besliskunde, deel 7a; dynamische programmering 1*, MC Syllabus 1.7a.

G. de Leve, H.C. Tijms (1970). *Leergang besliskunde, deel 7b; dynamische programmering 2*, MC Syllabus 1.7b.

G. de Leve, H.C. Tijms (1971). *Leergang besliskunde, deel 7c; dynamische programmering 3*, MC Syllabus 1.7c.

G. de Leve (1984). Mathematische besliskunde: theorie en praktijk. *Vakantiecursus 1984; Hewet - plus wiskunde*, CWI Syllabus 1, pp. 1-13.

1.3. Reports

G. de Leve, J. Kriens (1954). Overzicht van een aantal artikelen over 'operations research', [1]. Report S-167.

G. de Leve (1956). Enige statistische aspecten van de factoranalyse. Report S-210.

G. de Leve (1958). Dynamische voorraadproblemen. Report S-230, ZW-7.

G. de Leve (1961). Dynamische programmering. Report S-280.

G. de Leve (1961). Meerstapsbeslissingsproblemen; lezing Studiekring Economische Sectie V.V.S., Den Haag. Report S-291.

G. de Leve (1962). Steeds weer meten en weten. Report S-297.

G. de Leve (1962). Stochastic ∞-stage decision problems. Report S-302.

G. de Leve (1965). Markov-programmering. Report S-341.

G. de Leve, H.C. Tijms (1973). A general Markov programming method, with applications. Report BD-2.

G. de Leve (1973). Besliskunde, wat is dat? Report BD-3.

G. de Leve, H.C. Tijms (1974). A general Markov decision method, with applications to controlled queueing systems. Report BN-24.

1.4. Course notes

J. Kriens, G. de Leve, P. de Wolff (1956). Operations research; 1955/56. Report SC-9.

J. Kriens, G. de Leve (1957). Voorraadproblemen. Report SC-10.

J. Kriens, G. de Leve, P. de Wolff (1958). Operations research; 1957/58. Report SC-9a.

J. Kriens, G. de Leve (1960). Operations research; 1959/60. Report SC-9b.

A.R. Bloemena, A.H. Haitsma, J. Kriens, G. de Leve (1960). Cursus operations research II. Report SC-11.

J. Kriens, G. de Leve (1960). Leergang besliskunde, hoofdstuk VII; algemene voorbeelden. Report SC-13h.

G. de Leve, F. Göbel (1960). Leergang besliskunde, hoofdstuk IX; caput uit de waarschijnlijkheidsrekening. Report SC-13j.

G. de Leve (1960). Leergang besliskunde, hoofdstuk XI; voorraadproblemen. Report SC-13l.

G. de Leve (1960). Leergang besliskunde, hoofdstuk XIV; dynamische programmering. Report SC-13p.

G. de Leve, J. van de Lune (1960). Leergang besliskunde, hoofdstuk XIVa; eenstaps- en meerstaps beslissingsproblemen. Report SC-13q.

G. de Leve, F. Göbel (1960). Leergang besliskunde, hoofdstuk XV; wachttijden. Report SC-13r.

G. de Leve, F. Göbel (1961). Leergang besliskunde, hoofdstuk XVII; vervangingsproblemen. Report SC-13w.

G. de Leve, J. Kriens, J. van de Lune, C.J. van Tooren (1962). Oriënterende cursus mathematische besliskunde. Report SC-14.

G. de Leve, J. Kriens, J. van de Lune, C.J. van Tooren (1965). Oriënterende cursus mathematische besliskunde; herziene uitgave. Report SC-14a.

G. de Leve (1965). Inleiding tot de besliskunde. Report SC-15.

G. de Leve, H.C. Tijms (1966). Inleiding tot de mathematische programmering, 1; lineaire programmering. Report SC-16a.

G. de Leve, H.C. Tijms (1966). Inleiding tot de mathematische programmering, 2; discrete en kwadratische programmering. Report SC-16b.

G. de Leve, J.C. van Dalen (1970). Inleiding in de besliskunde, 1. Report BC-2.1.

G. de Leve, J.C. van Dalen (1970). Inleiding in de besliskunde, 2. Report BC-2.2.

2. Publications of the University of Amsterdam

J.A. van der Velde, R. Jonker, G. de Leve, A. Volgenant (1979). Lower bounds for traveling salesman problems with known relative order for a part of the cities. Report AE 3/79, Faculty of Actuarial Science and Econometrics.

3. Papers in journals

G. de Leve (1962). Decision rules for adjusting Markovian processes. *Bulletin de l'Institut International de Statistique 39*, pp. 423-433.

G. de Leve (1962). Stochastische ∞-staps beslissingsproblemen. *Statistica Neerlandica 16*, pp. 433-448.

G. de Leve (1966). Wie beslist er eigenlijk? *Statistica Neerlandica 20*, pp. 155-170. [Text of inaugural lecture, given at the University of Amsterdam, February 21, 1966.]

G. de Leve, P.J. Weeda (1968). Driving with Markov programming. *Astin Bulletin 5*.1, pp. 62-86.

G. de Leve, A. Federgruen, H.C. Tijms (1977). A general Markov decision method I: model and techniques. *Advances in Applied Probability 9*, pp. 296-315.

G. de Leve, A. Federgruen, H.C. Tijms (1977). A general Markov decision method II: applications. *Advances in Applied Probability 9*, pp. 316-335.

J.A. van der Velde, R. Kaas, G. de Leve (1980). Stronger Gomory cuts by multidimensional knapsack problems. *Mathematische Operationsforschung und Statistik Series Optimization 11*, pp. 249-259.

R. Jonker, G. de Leve, J.A. van der Velde, A. Volgenant (1980). Rounding symmetric traveling salesman problems with an asymmetric assignment problem. *Operations Research 28*, pp. 623-627. [In the title, read 'Bounding' instead of 'Rounding'.]

G.A.P. Kindervater, A. Volgenant, G. de Leve, V. van Gijlswijk (1985). On dual solutions of the linear assignment problem. *European Journal of Operational Research 19*, pp. 76-81

4. Miscellaneous

G. de Leve (1972). Besliskunde. *Grote Winkler Prins, zevende druk, deel 3*, Elsevier, Amsterdam, pp. 654-655.

G. de Leve (1976). Besliskunde. *Winkler Prins Technische Encyclopedie, deel 2*, Elsevier, Amsterdam, pp. 77-79.

Dissertations supervised by Gijs de Leve

Analysis of (s, S) inventory models
Henk Tijms, April 19, 1972

Machine scheduling problems: classification, complexity and computations
Alexander Rinnooy Kan, January 28, 1976

Sequencing by enumerative methods
Jan Karel Lenstra, February 25, 1976

Markovian control problems, functional equations and algorithms
Awi Federgruen, May 15, 1978

Finite generalized Markov programming
Piet Weeda, May 31, 1978

Location problems on trees and in the rectilinear plane
Antoon Kolen, March 3, 1982

Traveling salesman and assignment algorithms: design and implementation
Roy Jonker, November 12, 1986

Contributions to the solution of the traveling salesman problem and related problems
Ton Volgenant, February 18, 1987

Table of contents

Decision support: challenges, opportunities, and limitations of operations research

J.Chr. van Dalen
Open University of the Netherlands
Heerlen

1. Introduction

It is quite a difference between the very first developments of linear programming to the state of the art of operations research nowadays. On the one hand it took a short period of time, compared with developments in other disciplines. On the other hand the development path shows a variety of directions. Methodologically one can identify several theoretical and computational varieties. Applications are elaborated in several fields, due to differences in problems to be solved and sectors of industry to be served. Even in the non-profit sector applications of operations research (henceforth abbreviated as OR) are more or less normal practice. With the rise of other disciplines, especially in the field of information and communication technology, the use of OR techniques is emerging in managerial life more and more.

This aspect, more formally called 'the contribution of OR to managerial decision making', will be the main issue in this contribution. As a consequence its scope is broader than - mathematical - problem solving itself, and covers the process of decision making and its organization too. It is interesting to see how OR matches with common decision culture of management in organizations of different origins, while it is as interesting to conclude that OR - as a way of problem solving - has its beneficial impact on decision making itself. The idea of optimality e.g. has spread through whole decision making life. Whether it concerns investment, bargaining or political decision making, most participants in decision making have a notion of being constrained by conditions that prevent them from being greedy to the limit. They have learned to navigate in the feasible space between Scilla and Charibdis of e.g. resource limitations. So we may conclude that OR has not only contributed to the solution of managerial problems themselves, but also to the decision making attitude of managers (and non-managers) engaged in problem solving.

2. Decision making in organizations

Decision making, especially managerial decision making, can be distinguished in a variety of ways. The reader can be given a lot of references to the abundance of publications on decision making (e.g., Cyert & March, 1962; Hanken, 1981; Mintzberg et al., 1976; Saaty, 1981; Simon, 1960, 1979) out of the rich history of decision making research. We will follow some of the recognized distinctions to clarify the role of decision making in organization and management. The first distinction to be mentioned is between levels in decision making:

- *Strategic* decision making, regularly connected with long terms, broad references, goal setting activities, etc. Strategic decision making mostly implies high levels of uncertainty, speculations about future developments in the system considered. It asks for an intensive thinking process of participants and takes quite a long period of preparation. Complexity is challenging the decision maker. One can think of long term investments in Research and Development of new products for certain (new) markets as an example of this kind of decision making.
- *Tactical* decision making is constrained by the outcomes of a strategic decision process. The decision making freedom is less. In a certain sense tactical decision making is the execution of strategic decision making. The decision horizon (like a planning horizon) is shorter than with strategic decisions. The tactical decision making asks for less - cognitive - effort of the participating decision makers. Corresponding to the example mentioned before, the execution of the R&D investment program by means of the procurement of laboratory equipment can be added as an example of the tactical continuation of the decision making process.
- *Operational* decision making is seen as the lowest level of decision making, constrained by tactical decision making. Mostly procedures and rituals are leading the decision maker. The decision horizon is short, and feed back from factual execution of the decisions is direct. The use of the laboratory equipment is an example of this kind of decision making.

Before suggesting some typical application modes of OR with respect to these kinds of decision making, we will treat another distinction of decision making in two levels:

- individual decision making;
- organizational decision making.

Individual decision making refers to one actor decision models, the actor being a person. Especially when dealing with managerial decision making it is of interest to know e.g. how individual managers make their decisions, what information they use, how they treat the acquired information, how their interaction with their environments is taking place etc. All these aspects of individual decision making have to do with psychological factors, with sociological factors, with economical, informational and technical factors, etc. Important is the cognitive treatment of information. The cognitive abilities of managers determine largely the outcomes of the individual decision process. For every decision to be made it is necessary that the decision maker can rely on an adequate knowledge base and on some heuristic. Well known heuristics encompass rules like the minimax, the maximin, the lexicographic, the minimal loss and the equal probability rule. The application of a heuristic is related to an already existing goal. At this point difficulties may arise. E.g., what has to be done in the case of more than one goal. The famous multicriteria problem interfering with many well founded solution procedures emerges. OR theorists and practioners have struggled with the problem for years, and found - among others - solutions like conjunctive and disjunctive rules.

Mostly individual decision making is supposed to be subject to conditions of rationality. Traditionally these conditions refer to (1) the possession of perfect information on choice alternatives and their consequences, (2) the sensitivity of the decision maker for the slightest differences in relevant variables, (3) coherence of the preference ordering of decision outcomes and (4) the constant search for maximal utility. When studying practical decision making it is useful not to take into account one form of rationality but to discriminate between some more. We distinguish as follows (see Vlek & Michon, 1980; Van Dalen, 1987):

- *Representational rationality*, referring to the ability of decision makers to design proper models of the decision problems to be solved. The individual decision maker will need an adequate image of his environment.
- *Goal rationality*, referring to a logical coherent system of values underpinning the decision makers' decision goals. Most individual decision makers do rely on a hierarchical system of values and goals, guiding their decision making.
- *Methodological rationality*, referring to the solution methods for decision problems. Some decision makers will try to build extensive heuristics, and will try for a long time to improve their solutions. Sometimes it will be apparent that decision makers construct intelligent solution procedures, without succeeding in the proper modelling of decision problems.
- *Metarationality*, referring to the policy used by the individual decision maker with respect to the other three rationalities. A well balanced appeal to each of the three rationalities leads to a high valued meta rationality. Overestimation of one of the rationalities leads to a lesser value and mostly correspondingly to a less valuable decision outcome. One can imagine that over-emphasizing e.g. the representational rationality leads to a splendid model of the decision problem, without being capable of solving it. Or, overemphasis of the methodological rationality will possibly lead to very sophisticated solution procedures without having a proper goal structure.

So, one has to compromise with respect to the three rationalities in a clever and balanced way for the best achievable solution.

So far we dealt with cognitive aspects of individual decision making only. Two other aspects of individual decision making are of interest for us too: the affective and the active aspects. For decision making and for its effectivity it is necessary that it is performed in concordance with the proper know-how of decision problems and decision making. However, it is as important that decisions to be made and to be executed are affectively valued positively and are experienced as executable. Often one is confronted in organizations with decisions that are raising resistance and are never executed in the intended way (if executed at all).

The other level of decision making, the organizational level, is characterized by other factors. However, scanning the literature, it seems that organizational decision making is supposed to be just of the same type as individual decision making. The same process is supposed, the same rationalities, the same way of handling information, etc. As Feldman & March (1981) observe: 'The classic representation of organizational choice is a simple extension of decision theory visions of individual choice. In particular, decisions are seen as derived from an estimate of uncertain consequences of possible actions and an estimate of uncertain future preferences for those consequences.' Nevertheless, organizational decision making has to do with a more complex situation, due to - among other things - the interaction of several actors. These actors - being groups or individuals - are not all equal. Some actors are more influential than others, or have more information to rely on. Voting and bargaining are introduced, sometimes treated as folkloristic, sometimes as intriguing mathematical processes. Optimal voting rules can be designed, bargaining games constructed.

For the benefit of a proper insight, the organizational multi-actor decision making can be divided in three classes of decision systems (see Hanken, 1981), (1) autocratic systems (in itself divided into hierarchical and polyhierarchical systems), (2) systems with collective decision making (divided into systems with voting, consensus systems and simple coalition systems) and (3) systems with individual decision making (divided into autonomous systems, bargaining systems and general coalition systems). One of the intriguing differences between individual and organizational decision making refers to the - mental - mapping of decision problems and the solution alternatives. In the case of individual decision making we can describe the mapping or modelling procedure by means of a personality theory, the theory of personal constructs (Kelly, 1963). In the case of organizational decision making the modelling is often guided by theory based rules, shared by those engaged in the modelling exercise. Several approaches are advocated, some systems analysis like, some along the lines of mathematical logic, and so on.

So far we neglected the process aspects of decision making in organizations. Several writers have been engaged in researching and describing decision processes in a normative or non-normative way (see Mintzberg et al., 1976; Davies & Morgan, 1982; Nutt, 1984). Enlighting is a decision process adopted from Davies and Morgan (see Van Dalen, 1987). This process encompasses four phases, together forming a sequence in time, albeit that several times feedback loops guarantee the adaptation to new circumstances. The process starts with a so-called *garbage can phase*, in which problems are competing with each other and with problem solutions, helped by participants with time and energy to commit themselves to these problems. This phase is specially dedicated to problem identification and model building. Several techniques are at hand and can be used. However, lots of process problems have to be solved first. E.g., problems with respect to the bargaining as practiced by the - potential - participants in the game of solving a decision problem. OR can have a beneficial function with respect to this phase in clarifying problem structures and the solvability of problems. OR can have a beneficial function with respect to the bargaining process too (see Brams, Lucas & Straffin, 1983).

The next phase in the schematic decision process is the *political phase* mainly consisting of negotiating activities. As a result of the activities in the former phase, the number of active participants is diminished. For mean-while, those persons who concluded that their interests are not so much at stake left the scene.

The input to this phase is the output of the garbage can phase: a clear picture of the nature of the decision problem and the goals to be reached. There is an agreement on the terms of reference and the remaining participants do share the picture of the decision problem. The second phase is characterized by informality, interaction of small groups of those directly concerned, bargaining and compromising and sometimes the active intervention of authoritative administrators (managers). The output of this phase mostly includes agreement on possible fields of solutions to the decision problem, with the supposition that these fields coincide with already existing policy lines. During this phase the number of active participants often is diminished further.

In the third phase, main activities are directed towards the construction of a feasible solution to the decision process. To all the persons concerned it is clear that a definite solution has to be found. Sometimes bargains have to be reaffirmated, as a result of the testing of solutions against criteria of acceptability, feasibility, maintainability, etc. As a consequence of the ever diminishing number of active participants, the solution to be chosen in the end has to be legitimated to the broader group of those influenced by the solution or by the way the solution is implemented. So we call this phase the *legitimation phase.* The output of this phase can be characterized by e.g. an agreed line of action, commitment of several groups and persons to the solution (the decision), accepted guidelines for execution of the decision, etc.

The last phase, called the *implementation phase*, ends with an operational decision or a policy-strategy, e.g. an optimal production schedule, that is maintained over a longer period. As a consequence of the process idea, the execution of the policy or the single decision will result in some kind of feed back, leading on its part to alterations in the optimizing program if necessary.

It is interesting to infer from this presentation of several aspects of decision making what contributions can be expected from OR. To start with the process just presented, we can conclude that the obvious contribution will be in the area of model building, and it may be as obvious that model building has a function in bargaining processes as they are incorporated in the whole decision 'building' process. The construction of alternative models, the probing of the models by simulation and sensitivity analysis, will support the participants engaged in bargaining and compromising on their way of arriving at a sound decision. A splendid example is the use of macro econometric models in the bargaining around the conception of a policy note at the level of national government, or around the constitution of a new government (which is a regular and repeated issue). Also within the area of strategy building in private firms the use of modeling is fruitful. The probing of scenario's with their own suppositions gives a valuable insight into the consequences of the actual realization of decisions, policies, strategies, etc. These applications of OR in the context of organizational decision making are deviating from the original line of development, which was mainly in designing new solution techniques. So we may conclude that organizational practice has lead the development of OR in two main directions. The first one is about solution techniques, like LP, dynamic programming, networks, etc. The second one is in modelling problem situations. Quite a lot of different approaches have been developed, sometimes as an analogy of e.g. modelling techniques in informatics, sometimes as composition of original OR techniques, matched with techniques from other disciplines, as with scenario analysis.

The main theme in the development and use of OR in theory and practice is the stress laid on decision making (or perhaps better: decision building) as a context for all activities in this area. With respect to the presented decision process we may emphasize that OR is used within the process as well as over the process. OR instruments can e.g. be used within the process as model building facility or as solution facility with respect to the decision problem at hand. But OR instruments can also be used as a facility over the whole decision making process (or parts of it) e.g. in modeling bargaining constellations and designing optimal bargaining strategies. The composition of (working) groups may be approached with the help of OR too.

An interesting and fairly new offspring of the vast tree of OR techniques, methods, procedures, etc. is the manipulation and effective preservation of knowledge. This line of development is well known as expert systems or artificial intelligence (see Van Dalen, 1989). Such a development is the splendid blend of OR and IT (Information Technology). By means of all the new devices in IT, varying from sophisticated mainframes and PC's to networks in a variety of manifestations, it is possible now to use OR applications in a multitude of situations. Centrally as well as decentrally knowledge can be exploited in favor of sound decision making. Nevertheless, the applications of expert systems in practice are relatively scarce. Mostly their scope does not go beyond the diagnosis of problem situations, without reaching for solutions. However, these applications are valuable, while for situations that have to be characterized by large amounts of data, it is possible now to support the first decision making task: diagnosing the problem.

3. OR AND STRATEGY

Strategy is a concept with a long, and perhaps glorious, history. Well known is the meaning in the context of the military, in use for eras. It is used in contrast with and in addition to concepts as tactics and operations. The concept is mostly connected with the flavour of broad viewing and high places, and positions like those of generals. In this section we will treat the concept of strategy as in use in the field of OR as well as in the fields of business and informatization. The strategic meaning of OR in organizational contexts will be dealt with too.

The meaning of strategy in OR is less prestigious as in the military. Strategy, being regarded as a synonym of policy, is interpreted as a decision rule, resulting in a decision at every point of time, given the existing state of the system under study and eventually given the history of the system during some period of time. Depending upon the characteristics of the system some specializations of this interpretation can be given. One of the common specializations is the assumption of stationarity, resulting in decisions based solely on the existing state. Another specialization is introduced by the difference between finite and infinite horizons. The class of techniques related to the design of (optimal) strategies is - as a rule - labelled 'dynamic programming', suggesting by this title that ongoing change is a key feature of the connected type of decision making.

So far the meaning of strategy in the area of OR. In business organizations - and in a growing rate in non-profit and governmental organizations - strategy is used in a variety of ways. The main difference is the use of the 'phenomenon' internally or externally. Internally strategy is mostly connected with the ways people in organizations try to change the organization, to set agendas for innovations, to facilitate the diffusion process of innovations (the introduction of OR being one of them) in the organizational context, to influence others in e.g. bargaining situations, etc. This meaning is clearly related to the notion of dynamism that belongs to the make-up of the OR-interpretation of strategy. One can hold the statement that this type of strategy is constructing decisions on the basis of - at least - the current state of the organization.

The external version of business strategies has to do with the entrepeneurial status of the organization, and is connected to the way the organization functions in its market-environment. A strategy is sometimes described as 'the fundamental characteristics of the match that an organization achieves among its skills and resources and the opportunities and threats in its external environment that enables it to achieve goals and objectives' (see Chrisman et al., 1988). As an elaboration of the strategy concept several classifications are formulated, among which the classification of Miles & Snow (1978) is well known:

- *Prospectors*, being organizations which almost continually search for market opportunities, and can be considered creators of change and uncertainty. These organizations are usually not completely efficient, but rather stress effectiveness.
- *Defenders*, being organizations which have narrow product/market domains, and whose primary attention focuses on improving the efficiency of their existing operations.
- *Analyzers*, an intermediate organizational type, operating in one relatively stable and another relatively changing product/market domain. In the stable domain they operate routinely and efficient, in the more turbulent domains top-managers watch their competitors closely for new ideas and then rapidly

adopt those ideas which appear to be most promising.
- *Reactors*, being organizations in which top-managers frequently perceive change occurring in their organizational environments, but are unable to respond effectively. They can lack a consistent strategy-structure relationship too.

In addition to the elucidation of the last strategic type it may be stipulated that in theory and practice there is a continuing debate about the order relation between strategy and structure of the organization, in the form of 'strategy follows structure' versus 'structure follows strategy'.

The mutual importance of the organizational strategies and OR is twofold. On the one hand the strategic make-up of organizations really do structure the feasible space for OR applications. Some strategic types ask for strictly efficiency reenforcing OR approaches, while other types preferably are supported by more loosely structured OR techniques. On the other hand it is the question whether and to what extent OR applications are supporting the strategic position of organizations. This question is quite analogous to the one that is often posed with respect to the introduction of new technologies in organizations. Especially nowadays the introduction and diffusion of information and communication technology is questioned. As a consequence of the close relationship between OR and information technology (as stipulated earlier) one can make distinctions with respect to the strategic impact of OR applications as McFarlan (1983) makes with respect to information technology applications. Two kinds of strategic impact are important:
- strategic impact of existing applications;
- strategic impact of future applications (applications to be developed).

The two types of impact function as dimensions of a double dichotomy - low and high impact - and generate four organization types, as follows:
- Low impact of existing as well as future applications: *support organization*, referring to the fact that the effective functioning of the organization is not fully dependent on existing applications, while the development of new applications is not essential for the effectivity of the organization too (examples can be found in chemical industries).
- Low impact of existing, high impact of future applications: *turnaround organization*, traditionally not very dependent on IT/OR applications, but with the development over time such applications are gaining importance. New production factories e.g. with their CAD/CAM practice and with the integration of production and administrative systems demonstrate the meaning of this type.
- High impact of existing, low impact of future applications: *factory type*, traditionally strongly relying on applications of OR/IT systems. Future activities in this area are directed towards maintenance of the existing applications. Lot of retail-organizations are examples of this type.
- High impact of existing, as wel as future applications: *strategic type*. The effective functioning of the organization is strongly dependent on the functioning of sophisticated information systems. Moreover the competitive strength of these organizations is based on the creative use of information and the development of new OR/IT applications. Banking industry and insurance

companies are proper examples of this type. These kinds of organizations are sometimes labelled as 'expert companies' (see Feigenbaum, McCorduck & Nii, 1988), because their success is mostly determined by the effective use of human or artificial expertise.

So far the meaning of strategies in OR context is explained, with special emphasis on the strategic impact of OR applications (mostly in combination with IT applications). The four types of organizations just mentioned, are important as types themselves, but are far more important when placed in the context of change of organizations from one type to another. These transitions indicate the opportunities of OR applications. The character of OR applications related to management tasks (as supportive to them) will be illustrated hereafter.

4. Varieties of OR contributions

OR applications are related to decision making in organizations. Especially to managerial decision making. So some insight in the process of managerial work will be helpful.

The main task of management in organizations can be characterized as coordination. Coordination between people, between people and machinery, between streams of information, etc. The coordination tasks will be analyzed by means of a model of the so called management cycle, encompassing four phases. The phases are based upon a two-dimensional decomposition of the set of possible coordination activities. One of the dimensions is the modelling dimension, referring to the circumstance that coordination can be founded on the reality of organizations itself, or in the model (the image) of reality. The other dimension introduces time as a coordination variable, by distinguishing between feed forward and feed back. These two dimensions together constitute a four phase management cycle as follows (see Van Dalen, 1987):

	model-based	reality-based
feed forward	planning/ strategy	execution/ implementation
feed back	learning/ adaptation	evaluation/ control

OR applications, as supportive means to managerial decision making must have somehow a relation to the four phases. Planning c.q. strategy applications are very common in the field of OR. Several techniques (e.g., scheduling, forecasting, optimal lot-size) are developed and in operation. Mostly these applications are really used as planning facilities. But alternative uses are not excluded. One such alternative is the use of a planning technique as a diagnostic instrument (see Bronnenberg, 1989). Without exaggeration it may be stated that most applications of OR belong to the first phase of the management cycle. The following phase is receiving quite a lot attention too. In the reality of executing planned activities OR applications nowadays are often practised in combination with IT applications. One can think of transport programs, supporting the actual transport, e.g. by interactive routing, dependent upon actual road conditions, changed destinations, etc. The programming of CAD/CAM machinery is another example. In the evaluation phase, a combination of feed back and real activities, few applications are developed (e.g. related to quality control). Nevertheless decision problems do exist with respect to this phase, and applications certainly will be designed.

The fourth phase is the most neglected phase: learning or adaptation. This phase is about the improvement of the organisation, its decision making and its management as a system. So OR applications in this phase should emphasize the change of the model building as prerequisite of planning. As mentioned earlier model building is not the strongest part of OR applications. Nevertheless energy should be devoted to this type of decision problem.

The coordination activities, as reflected in the management cycle have to be connected with processes in the organization. Not restricted to the sector of private enterprises we can discriminate between seven essential processes, necessary for the continuity of an organization. Without suggesting any preemptive order between the processes they can be characterized by the following labels (see Hanken & Van Dalen, 1987):

- acquisition of new clients, sometimes new markets, or sometimes a combination of both;
- research and development, necessary to design new products/services;
- production and logistics, sometimes called the primary processes;
- material resources, like production or facilitating machinery;
- human resources, including tasks belonging to the personnel and organization function;
- financial resources, comprising both the acquisition as well as the distribution of financial means;
- informational resources, comprising traditional bookkeeping as well as innovative diagnostic experiences.

Each of the seven processes can be described with the help of systems theory tools. The simplest version of such a description uses the input-throughput-output metaphor, supplemented by supporting processes if necessary. An analysis based on such a description - focused on control (or decision making) mechanisms - will give insight into the behaviour of each of the processes and into the possibilities to exercise influence on them. With respect to the control of the processes, the phases of the management cycle are important as a framework for internal coordination. Moreover such an analysis is well suited for scanning the possibilities for OR applications. As an example we can take the financial resources process. The input of the process will be different types of funds, acquired in several ways. OR can be helpful in modelling the acquisition problem, differentiating among distinguishable fund resources, taking into account - among others - the cost of acquiring funds. In the end an optimal fund raising scheme might be possible to construct.

The throughput of the financial resources process can be characterized as a distribution problem. The acquired funds have to be distributed among more or less profitable projects. The opportunities for OR applications are obvious. Portfolio e.g. is one of the promising approaches. The output of the financial resources process consists mainly of payments to outside creditors. Here too, one can identify several opportunities for OR applications. E.g. cash management is among them. In accordance to the line of analysis loosely presented here, all processes can be scanned globally and in detail. The opportunities and challenges of OR applications can be evaluated against criteria of feasibility cost, quality, etc.

So far the coordination within the seven essential processes. However, the seven processes have to be coordinated mutually too. This generates a more complex task. Coordination mechanisms must be created between processes. In the traditional work organization these types of coordination mechanisms are often effectuated by ad hoc interpersonal contact. But in the future - as is often stated - more emphasis will be laid upon coordination and accompanying communication by means of information technological devices. So one may expect that the need for formal systematic coordination will grow. This confronts us with a complex task of interlinking different processes, each with their own coordination policies. E.g. the human resources process has to be coordinated with the financial resources process, and with the material resources process (not to leave out of the discussion the production and the R&D process). Challenges for sophisticated OR applications in abundance!

5. Discussion

In this contribution OR as a science and as an art is dealt with from the viewpoint of management. It is assumed that OR is supportive to managerial decision making in a broad sense. On the one hand OR contributes to proper coordination decisions, being the main theme of managerial work. On the other hand, a bit underexposed here, OR contributes to the proper production process in organizations, even to the product. More and more products are designed in such a way that a substantial component of the product itself consists of information and of some kind of optimization. The so called high-tech products are spreading all around the market, and nearly nobody can withdraw himself from the impact of these products.

OR, this time seen as a product, has a variety of appearances, varying from just model building to sophisticated optimization, with mixtures of model building and local optimization in between. Several levels of problem formulation, e.g. used in a nested form, underline the wide applicability. This rich variety of appearances makes it possible to apply OR methods and techniques in a lot of decision situations. Nevertheless, there are boundaries restricting the use of OR. OR employs the language of mathematics as a vehicle to formulate decision problems. OR needs quantified criteria to solve such problems. More abstractly formulated, OR is applicable only in decision situations one can describe by normative knowledge. Several examples of OR applications have shown that this normative knowledge is necessary, and sometimes has to be created artificially (e.g. in sensitivity analysis, simulation or penalty constructions).

References

1. S.J. Brams, W.F. Lucas, P.D. Straffin (1983). Political and related models. New York.
2. B.J.J.A.M. Bronnenberg (1989). Beoordeling van nieuwe produkten: voorspellingsmethoden als diagnose-instrument, in: J. Chr. Van Dalen, P.G.M. Hufen. Organisatie-Diagnose: Theorie en Praktijk, Leiden.
3. J.J. Chrisman, C.W. Hofer, W.R. Boulton (1988). Toward a system for classifying business strategies. *Academy of Management Review 13,* 413-429.
4. R. Cyert, J. March (1962). A behavioral theory of the firm. Englewood Cliffs.
5. J. Chr. van Dalen (1987). Agendering in organisaties, Enschede.
6. J. Chr. van Dalen (1989). Knowledge organized: expert systems in local government, in: I.T.M. Snellen, W.B.H.J. van de Donk, J.-P. Baquiast. *Expert Systems in Public Administration,* Amsterdam.
7. J.L. Davies, A.W. Morgan (1982). The politics of institutional change, in: L. Wagner. Agenda for institutional change in higher education, Guilford.
8. E. Feigenbaum, P. McCorduck, H.P. Nii (1988). The rise of the expert company. New York.
9. M.S. Feldman, J.G. March (1981). Information in organizations as signal and symbol. *Administrative Science Quarterly 25,* 171-186.
10. A.F.G. Hanken (1981). Cybernetics and society. Tunbridge Wells, Kent.
11. G.A. Kelly (1963). The psychology of personal constructs. New York.
12. W.F. McFarlan, J.L. McKenny, P. Pyburn (1983). The information Archipelagio - A plotting course. *Harvard Business Review.*
13. R.E. Miles, C.C. Snow (1978). *Organizational Strategy, Structure and Process,* New York.
14. H. Mintzberg, D. Raisinghani, A. Theoret (1979). The structure of 'unstructured' decision processes. *Administrative Science Quarterly.*
15. P.C. Nutt (1984). Types of organization decision processes. *Administrative Science Quarterly 29,* 414-450.
16. T. Saaty (1981). How to structure and make choice in complex problems. *Human Systems Management.*
17. H.A. Simon (1960). The new science of management decision. New York.
18. H.A. Simon (1979). Rational decision making in business organizations. *American Economic Review.*
19. C.A.J. Vlek, J.A. Michon (1980). Beslissen, wat is dat voor een proces. *Gedrag, 8/6,* 335-361.

Assignment and shortest path problems

B. Dorhout
Faculty of Applied Mathematics
University of Twente
Enschede

The author's algorithm for solving the linear assignment problem by means of a number of shortest path calculations is reviewed, and two simple postoptimal extensions are given.

1. Introduction

In 1973 a course was given at the Mathematical Centre in Amsterdam on the subject of mathematical programming. On that occasion the author was scheduled to give a lecture in which the properties of algorithms for the linear assignment problem would be compared. Thus he read a number of papers on the subject, tried out a couple of algorithms for which ALGOL-procedures were published, and wrote programs for the most promising algorithms for which no ALGOL-programs could be found. It came out that an algorithm described by Tomizawa (1971) gave the best results. Tomizawa solved an $n \times n$ problem iteratively, starting with the solution of a 1×1 problem, and afterwards deriving from the optimal solution to a $k \times k$ problem the optimal solution to a $(k+1) \times (k+1)$ problem, for $k = 1,...,n-1$. This was done by solving shortest path problems. As it seemed possible to apply Tomizawa's ideas in a more efficient way, the author conducted a number of experiments that led to a revised algorithm which outperformed all other algorithms tested. The main differences with Tomizawa's algorithm are that $k \times n$ problems are solved, and that good start solutions are used. This approach to solving assignment problems appears to be very successful. The algorithm became well known after a FORTRAN-translation of the author's program was published in a book written by Burkard and Derigs (1980). It will be described in section 2, in a way which is slightly different from its description by Dorhout (1973b).

In section 3 we will look at the increase of the objective value if an assignment is made obligatory. In particular we will show how the calculation of such increases can be made for more than one element simultaneously after the optimal solution of the assignment problem is obtained by our method. Only some further shortest path calculations have to be made which are of the same kind as those made in our algorithm.

In section 4 we describe an algorithm for the one-to-all shortest path problem with arbitrary distances based on the same principles as those applied in the algorithm for the linear assignment problem.

2. Solving the assignment problem by shortest path augmentations

We consider a complete directed bipartite graph $G = (I,J,A)$ with node sets I and J of cardinality m and n, respectively. To all arcs $(i,j) \in A$ weights c_{ij} are attached. Without loss of generality we assume that $m \leq n$. The linear assignment problem is to find a maximum matching of minimum weight. This problem can be formulated as an integer linear programming problem, but the integrality property of transportation problems allows us to leave out integrality constraints. If we define

$$x_{ij} = \begin{cases} 1 \text{ if } j \in J \text{ is assigned (matched) to } i \in I, \\ 0 \text{ otherwise,} \end{cases}$$

then we obtain the optimal solution of the assignment problem by solving its LP-relaxation: minimize

$$\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \tag{2.1}$$

subject to

$$\sum_{j \in J} x_{ij} = 1, \qquad i \in I, \tag{2.2}$$

$$\sum_{i \in I} x_{ij} \leq 1, \qquad j \in J, \tag{2.3}$$

$$x_{ij} \geq 0, \qquad i \in I,\ j \in J. \tag{2.4}$$

In order to solve this problem, we choose an arbitrary subset $I_1 \subseteq I$ with $|I_1| = 1$. Problem (2.1),...,(2.4), with I replaced by I_1 will be called (P_1). It can be solved by inspection. In subsequent iterations I_k is extended with one new element of I to I_{k+1}, $k = 1,...,m-1$, with $I_m = I$. In these iterations the optimal solutions to the corresponding problems $(P_2),...,(P_m)$ are constructed. This is done by Dijkstra's method for the one-to-all shortest path problem with nonnegative distances. In each iteration the distances are changed in such a way that they never will become negative. This is possible if solutions of the dual problems are known.

The dual linear programming problem (D_m) of (2.1),...,(2.4) is: maximize

$$\sum_{i \in I} u_i - \sum_{j \in J} v_j \tag{2.5}$$

subject to

$$u_i - v_j \leq c_{ij}, \qquad i \in I,\ j \in J, \tag{2.6}$$

$$v_j \geq 0, \qquad j \in J. \tag{2.7}$$

The complementary slackness conditions for optimal solutions to (P_m) and (D_m) are:

$$x_{ij} > 0 \text{ implies } u_i - v_j = c_{ij}, \ i \in I, \ j \in J, \tag{2.8}$$

and

$$\sum_{i \in I} x_{ij} < 1 \text{ implies } v_j = 0, \ j \in J. \tag{2.9}$$

Without loss of generality we may assume that $I = \{1,...,m\}$, $I_1 = \{1\}$, $I_k = \{1,...,k\}$, $k = 2,...,m$. Let G_k denote the complete subgraph of G with node sets I_k and J, and let $X_k = \{(i,j) | x_{ij}^k = 1\}$ be the matching found in iteration k. If we call *matched* all nodes incident with X_k and *free* all other nodes, then after iteration k all nodes of I_k are matched. The subset of free nodes of J will be denoted by $F_k = \{j \in J | \Sigma_{i \in I_k} x_{ij}^k = 0\}$.

Problem (P_1) is solved easily: find t from $c_{1t} = \min_{j \in J} c_{1j}$. Then clearly optimal solutions to (P_1) and its dual problem (D_1) are:

$$x_{1j}^1 = \begin{cases} 1 \text{ if } j = t, \\ 0 \text{ otherwise,} \end{cases} \tag{2.10}$$

$$u_1^1 = c_{1t}, \tag{2.11}$$

$$v_j^1 = 0, \ j \in J. \tag{2.12}$$

Now assume that optimal solutions x^k to (P_k) and (u^k, v^k) to (D_k) are known. Using these, we find optimal solutions to (P_{k+1}) and (D_{k+1}) by changing assignments on an alternating path. Such a path starts at the free node $k+1$ and ends in a free node of F_k. It consists of an alternating sequence of forward arcs $(i,j) \notin X_k$ and backward arcs $(i,j) \in X_k$. After the path has been found we make assignments $x_{ij}^{k+1} = 1$ on its forward arcs and delete all assignments on its backward arcs, thus augmenting the number of assignments by 1. The following steps are made.

- Extend G_k to G_{k+1}, define $u_{k+1}^k = \min_{j \in J}(c_{k+1,j} + v_j^k)$ and give length $\tilde{c}_{ij} = c_{ij} - u_i^k + v_j^k$ to each arc (i,j) of G_{k+1}. Then $\tilde{c}_{ij} \geqslant 0$ by the definition of u_{k+1}^k and feasibility of the solution to (D_k). Also, by complementary slackness, $\tilde{c}_{ij} = 0$ for all backward arcs (i,j).
- Apply Dijkstra's method, starting at node $k+1$, until the nearest free node $j_{k+1} \in J$ is reached, and derive X_{k+1} by adding to X_k all forward arcs in the shortest path from $k+1$ to j_{k+1} and deleting from X_k all backward arcs.
- Using d_h as denotation for the tentative distances from node $k+1$ to nodes h, found by Dijkstra's method, we define $\bar{d}$ by $\bar{d} = d_{j_{k+1}}$, and d_h^* by

$$d_h^* = \min(d_h, \bar{d}). \tag{2.13}$$

Then the optimal solution to (D_{k+1}) is

$$\begin{aligned} u_i^{k+1} &= u_i^k + \bar{d} - d_i^*, \ i \in I_{k+1}, \\ v_j^{k+1} &= v_j^k + \bar{d} - d_j^*, \ j \in J. \end{aligned} \tag{2.14}$$

Optimality of X_{k+1} and (u^{k+1},v^{k+1}) can be proved as follows (cf. Tomizawa, 1971).

Compare the tentative distances d_h with the real shortest distances $\hat{d}_h$, which satisfy

$$\hat{d}_j \leqslant \hat{d}_i + \tilde{c}_{ij}, \quad i \in I_{k+1}, \; j \in J, \tag{2.15}$$

with an equality sign holding if (i,j) is on the shortest path from $k+1$ to j. After the execution of Dijkstra's method, node set $I_{k+1} \cup J$ is partitioned into a set T of *tentatively* labeled nodes and a set P of *permanently* labeled nodes. We assume that the labeling procedure was executed in such a way that $d_i = d_j$ for all $(i,j) \in X_k$. Then

$$\begin{aligned} d_h &= \hat{d}_h = d_h^* \leqslant \bar{d}, & h \in P,\\ d_h &\geqslant \hat{d}_h \geqslant d_h^* = \bar{d}, & h \in T. \end{aligned}$$

From (2.14) we obtain

$$c_{ij} - u_i^{k+1} + v_j^{k+1} = \tilde{c}_{ij} + d_i^* - d_j^*, \; i \in I_{k+1}, \; j \in J. \tag{2.17}$$

Therefore, it follows from

$$\begin{aligned} \tilde{c}_{ij} + d_i^* - d_j^* &= \tilde{c}_{ij} + \hat{d}_i - \hat{d}_j \geqslant 0, & i \in P, \; j \in P,\\ \tilde{c}_{ij} + d_i^* - d_j^* &\geqslant \tilde{c}_{ij} + \hat{d}_i - \hat{d}_j \geqslant 0, & i \in P, \; j \in T,\\ \tilde{c}_{ij} + d_i^* - d_j^* &= \tilde{c}_{ij} + \bar{d} - \hat{d}_j \geqslant \tilde{c}_{ij} \geqslant 0, & i \in T, \; j \in P,\\ \tilde{c}_{ij} + d_i^* - d_j^* &= \tilde{c}_{ij} + \bar{d} - \bar{d} = \tilde{c}_{ij} \geqslant 0, & i \in T, \; j \in T, \end{aligned} \tag{2.18}$$

that $(u,v) = (u^{k+1},v^{k+1})$ satisfies (2.6) with $I = I_{k+1}$.

From (2.12), (2.13) and (2.14) it follows immediately that (2.7) is satisfied.

(2.8) is satisfied for all i and j with $(i,j) \in X_k$, since for these indices $d_i = d_j$ and from (2.14) the left hand side of (2.6) is not changed. (2.8) is also satisfied for all i and j with $(i,j) \in X_{k+1} \setminus X_k$ because these (i,j) are on the shortest path from $k+1$ to j_{k+1} and for those

$$c_{ij} - u_i^{k+1} + v_j^{k+1} = \tilde{c}_{ij} + d_i^* - d_j^* = \tilde{c}_{ij} + \hat{d}_i - \hat{d}_j = 0. \tag{2.19}$$

Since $F_{k+1} \subset T$, it follows from (2.14) and (2.13) that $d_j^* = \bar{d}$ and $v_j^{k+1} = v_j^k = 0$ for all $j \in F_{k+1}$. So (2.9) is also satisfied, and X_{k+1} is an optimal solution to (P_{k+1}). □

Notice that the optimal value of the objective function is

$$\sum_{i \in I} u_i^m - \sum_{j \in J} v_j^m = \sum_{k \in I} u_k^k \tag{2.20}$$

if it is assumed that nodes $k \in I$ are numbered in the order in which they are matched. From now on we will drop this assumption.

The method just described can be improved considerably by the application of start procedures. A very simple algorithm is this:

```
- $X := \emptyset$; $F := J$.
- For each $i \in I$:
        determine $j_i$ such that $c_{ij_i} = \min_{j \in J} c_{ij}$;
        if $j_i \in F$ then: $F := F \setminus \{j_i\}$; $X := X \cup \{(i,j_i)\}$; $\bar{u}_i := c_{ij}$.
```

If, with $k = |X|$, the sets I_k, X_k, and F_k are defined by $I_k = \{i \in I \mid j_i \in F\}$, $X_k = X$, and $F_k = F$ respectively, then optimal solutions to (P_k) and (D_k), satisfying complementary slackness conditions are X_k, and $u_i^k = \bar{u}_i$, $i \in I_k$, $v_j^k = 0$, $j \in J$. Starting with these solutions the procedure described above can be continued.

If $m = n$, then (2.2) implies that in (2.3) the $\leqslant$ signs can be replaced by $=$ signs and consequently (2.7) can be dropped from the dual problem. In that case the start procedure can be improved to:

```
- $X := \emptyset$; $F := J$; $H := I$ (with $H \subseteq I$ denoting the subset of free nodes in
  $I$).
- For each $i \in I$:
        determine $j_i$ such that $c_{ij_i} = \min_{j \in J} c_{ij}$; $\bar{u}_i := c_{ij_i}$.
- Define $J_p = \{j \in J \mid$ for some $i \in I$: $c_{ij} - \bar{u}_i = 0\}$ and $J_e = J \setminus J_p$.
  For each $j \in J_e$:
        set $\bar{v}_j := -\min_{i \in I}(c_{ij} - \bar{u}_i)$; denote by $i_j$ the value of $i$ for which this
        minimum is achieved;
        if $i_j \in H$ then: $H := H \setminus \{i_j\}$; $F := F \setminus \{j\}$; $X := X \cup \{(i_j,j)\}$.
- For each $i \in H$:
        if $j_i \in F$ then: $H := H \setminus \{i\}$; $F := F \setminus \{j_i\}$; $X := X \cup \{(i,j_i)\}$.
```

If now $k = |X|$ and I_k, X_k, and F_k are defined by $I_k = I \setminus H$, $X_k = X$, and $F_k = F$ respectively, then optimal solutions to (P_k) and (D_k), satisfying complementary slackness conditions are X_k, and $u_i^k = \bar{u}_i$, $i \in I_k$, $v_j^k = \bar{v}_j$, $j \in J$ respectively.

If all minima in this algorithm are achieved for one unique index, then, given the dual solution $(u,v) = (\bar{u},\bar{v})$, the cardinality of primal starting solutions cannot be greater than k.

This can be seen as follows. Consider the bipartite graph $\tilde{G} = (I,J,\tilde{A})$, with $\tilde{A} = \{(i,j) \in A \mid c_{ij} - \bar{u}_i + \bar{v}_j = 0\}$. Each $(i,j) \in \tilde{A}$ is either one of the arcs (i,j_i) or one of the arcs (i_j,j). Now from the way of scanning it is seen that the arcs of the first category form a forest, consisting of trees of diameter 2, and that these trees after adding the arcs of the second category grow to trees with a diameter of 4 or less. So a maximal matching will be built up if subsequently all arcs incident with a point of degree 1 are included in the matching, provided that the other node of the arc is still free. This is done in the procedure. □

Nawijn and Dorhout (1988) proved that for $m = n$ the expected fraction of assignments obtained with this starting procedure is asymptotically equal to $2 - \exp(-1/e) - \exp(-\exp(-1/e)) \cong 0.8073$, if the cost coefficients are independent identically distributed random variables with continuous distribution function. Moreover, this fraction is monotonously decreasing in n. If the distribution function is not continuous, the fraction of assignments obtained will still be higher, as in general there is more freedom in the choice of assignments. Although this procedure is quite satisfactory with respect to the ratio of the number of assignments to the number of operations, needed to obtain it, experiments by Jonker and Volgenant (1987) demonstrate that it is worthwhile to construct a starting solution with a low value of the objective function rather than a large number of assignments, even though this takes more time.

As Dijkstra's method requires $O(|A|)$ operations, the complexity of the algorithm described above is $O(m|A|)$ in the worst case. In practice the efficiency of the algorithms for Dijkstra's method strongly depends on sorting procedures and data structures used. Especially for applications in sparse graphs these factors play an important role.

Finally we remark that the algorithm, as described in this section, has the property that searching for minima of (reduced) costs is allways done over the outgoing arcs from nodes in I. So one can work with representations of the graph in 'forward star' form, and in-core out-of-core versions of the algorithm can be implemented very efficient.

3. Postoptimal analysis

In several situations one is interested in the increase of the minimum value of the objective function caused by making an assignment obligatory. A well-known example constitutes the solution of the asymmetric travelling salesman problem by branch and bound methods, for which the assignment problem is used as a relaxation of the original problem. Then the optimal solution to the original problem is found, if the optimal solution $\bar{X}$ to the assignment problem results in a tour. Otherwise at least one arc in $\bar{X}$ has to be exchanged with an arc not in $\bar{X}$. An arc not in $\bar{X}$ will not be part of an optimal tour if forcing it in the solution of the assignment problem will give an objective value higher than the value z^* of a known feasible tour. An arc in $\bar{X}$ will be part of a minimal tour with certainty, if its exclusion causes a value higher than z^*. This is the case, for example, if for each of the arcs with the same tail, inclusion in the tour gives a value higher than z^*.

As remarked by Kreuzberger (1971) and Kindervater et al. (1985), it is not necessary, in the application mentioned above, to resolve an assignment problem completely for each arc that possibly is included or excluded in its solution. Kreuzberger described an algorithm for the computation of the increase of the objective function of an $n \times n$ problem, if assignments (i,j) are forced. He did this simultaneously for fixed i and all $j \in J$, starting from the optimal primal and dual solutions of the original problem. So in order to obtain these numbers for all $(i,j) \in A$, this algorithm has to be executed n times, once for each $i \in I$. Dorhout (1973a) demonstrated that the same numbers can be computed in n supplementary applications of Dijkstra's method after the solution of the assignment problem by the method described in section 2. Then in each of these applications the increase of the objective function is calculated simultaneously for all $i \in I$ and fixed $j \in J$. This can be seen as follows.

Assume that the optimal solutions to (P_n) and (D_n) are

$$\overline{X} = \{(i_1,j_1),...,(i_n,j_n)\}, \tag{3.1}$$

and $(\overline{u},\overline{v})$ respectively, and that one wishes to know the increase of the optimal value of the objective function, if (i_k,j_1) is made obligatory. Then one has to delete from G nodes i_k and j_1 and all arcs incident with these nodes. As a consequence the assignments (i_1,j_1) and (i_k,j_k) in the optimal solution for G are cancelled, and the remaining assignment problem (P') has to be reoptimized. This is accomplished by making one iteration of the algorithm described in section 2, in which the shortest path from node i_1 to node j_k is determined, with arc lengths

$$\overline{c}_{ij} = c_{ij} - \overline{u}_i + \overline{v}_j \geqslant 0, \ i \in I \setminus \{i_k\}, \ j \in J \setminus \{j_1\} \tag{3.2}$$

(with $\overline{c}_{ij} = 0$ for all $(i,j) \in \overline{X}$).

If d_r denotes the distance from i_1 to nodes $r \in I \cup J$, the increase of the minimal objective value is

$$\overline{c}_{i_k j_1} + d_{j_k}. \tag{3.3}$$

Often one is interested in the increases of the minimal objective value for *all* (i_k,j_1), with $i_k \in I \setminus \{i_1\}$, and fixed j_1. Then Dijkstra's method gives an alternating spanning tree T of shortest paths from node j_1 to all other nodes of G. This tree represents a basic solution to the assignment problem, and, with

$$\begin{aligned} \overline{u}'_i &= \overline{u}_i - d_i, \quad i \in I, \\ \overline{v}'_j &= \overline{v}_j - d_j, \quad j \in J, \end{aligned} \tag{3.4}$$

we have

$$\overline{c}'_{ij} = c_{ij} - \overline{u}'_i + \overline{v}'_j \geqslant 0, \qquad (i,j) \in A, \tag{3.5}$$

$$\overline{c}'_{ij} = 0, \qquad (i,j) \in T, \tag{3.6}$$

showing that complementary slackness properties hold for these solutions of the primal assignment problem and its dual. Now, an arbitrary arc (i_k,j_1), $i_k \neq i_1$, forms an alternating cycle with arcs of T. So, if the matching is altered by changing all assignments on this cycle, the objective function increases, by (3.6), with $\overline{c}'_{i_k j_1}$. This amount is the minimum increase of the objective function if (i_k,j_1) is forced in the solution, since the feasibility and complementary slackness relations with the $\overline{u}'_i$ and $\overline{v}'_j$ values from (3.4) are unchanged for the resulting $(n-1) \times (n-1)$ problem. Applying Dijkstra's method in this way for all $j \in J$ gives the minimum increase of the objective function for all unassigned arcs (i,j) in $O(n^3)$ operations.

4. Shortest paths with negative arc lengths allowed

Let $G = (N,A)$ be a directed graph with node set $N = \{1,...,n\}$ and arc set A. We consider the problem of determining the shortest paths from one arbitrary node to all other nodes, with given lengths c_{ij} for all arcs (i,j) in A. (Without loss of generality we may assume that G does not have multiple arcs or loops.) As the problem only makes sense if G does not contain negative directed cycles, we assume that there is no sequence of arcs $(i_1,i_2),...,(i_{k-1},i_k)$, (i_k,i_1) with $c_{i_1i_2}+...+c_{i_ki_1}<0$.

If $r = |\{i\in N\setminus\{1\}|\exists\, j\in N: (i,j)\in A,\ c_{ij}<0\}|$, the complexity of the algorithm is $O(r|A|)$.

Assume that G is connected and that one wants to know the shortest paths from node 1. As they form a spanning tree of G, we may define x_{ij} as the number of shortest paths that contain (i,j). The problem can be formulated as to minimize

$$\sum_{i=1}^{n}\sum_{j=2}^{n} c_{ij}x_{ij} \tag{4.1}$$

subject to

$$\sum_{k=2}^{n} x_{1k} = n-1, \tag{4.2}$$

$$-\sum_{i=1}^{n} x_{ij}+\sum_{k=2}^{n} x_{jk} = -1, \quad j = 2,...,n, \tag{4.3}$$

$$x_{ij} \geqslant 0, \quad i = 1,...,n,\ j = 2,...,n. \tag{4.4}$$

Here, for ease of notation, we have assumed that with combinations ij of indices only those are meant for which $(i,j)\in A$.

Now (cf. Wagner, 1975), after the introduction of nonnegative slack variables x_{jj}, with $c_{jj} = 0$, $j = 2,...,n$, this problem is transformed into the nondegenerate transportation problem: minimize (4.1) subject to

$$\sum_{j=2}^{n} x_{ij} = n-1, \quad i = 1,...,n, \tag{4.5}$$

$$\sum_{i=1}^{n} x_{ij} = n, \quad j = 2,...,n, \tag{4.6}$$

and (4.4), which now includes nonnegativity constraints for x_{jj}, $j = 2,...,n$. If G is not connected, this problem does not have a feasible solution. If G does not have a negative cycle, the optimal values of all x_{jj} are positive.

The dual of problem (4.1), (4.4), (4.5), (4.6) is: maximize

$$(n-1)\sum_{i=1}^{n} u_i - n\sum_{j=2}^{n} v_j \tag{4.7}$$

subject to

$$u_i - v_j \leqslant c_{ij,}\ i = 1,...,n,\ j = 2,...,n. \tag{4.8}$$

(4.1), (4.4), (4.5), (4.6) is solved by first solving the problem: minimize

$$\sum_{i=2}^{n}\sum_{j=2}^{n} c_{ij}x_{ij} \tag{4.9}$$

subject to

$$\sum_{j=2}^{n} x_{ij} = n-1, \quad i = 2,...,n, \tag{4.10}$$

$$\sum_{i=2}^{n} x_{ij} = n-1, \quad j = 2,...,n, \tag{4.11}$$

$$x_{ij} \geqslant 0, \qquad i = 2,...,n,\ j = 2,...,n. \tag{4.12}$$

First we replace the right-hand sides in (4.10) and (4.11) with 1's, and solve the resulting assignment problem by the algorithm described in section 2. It is started by matching (i,i) for all those i, for which there is no $c_{ij}<0$, $j = 1,...,n$. After that the optimal primal and dual solutions are obtained in at most r iterations. Multiplication of the optimal x-values by $n-1$ gives the optimal solution to (4.9),...,(4.12). The optimal values of the dual problems are equal. If they are $u_i = \bar{u}_i$, $v_i = \bar{v}_i$, $i = 2,...,n$, then after suppletion with $\bar{u}_1 = \min_{j=1,...,n}(c_{1j}+\bar{v}_j)$ we determine the shortest alternating tree from node 1 to all other nodes in the bipartite graph $G' = (I,J,B)$, with $I = \{1,...,n\}$, $J = \{2,...,n\}$, $B = A \cup \bigcup_{j \in J}\{(j,j)\}$, and arc lengths $c_{ij} = c_{ij}-\bar{u}_i+\bar{v}_j$, $(i,j)\in B$. If $d_1 = \bar{u}_1$ and d_j-d_1 are the distances from node 1 to j in G', for all $j \in I \cup J$, then $(\bar{\bar{u}}, \bar{\bar{v}})$, defined by

$$\begin{aligned} \bar{\bar{u}}_i &= u_i - \bar{d}_i, \quad i \in I, \\ \bar{\bar{v}}_j &= \bar{v}_j - d_j, \quad j \in J, \end{aligned} \tag{4.13}$$

is the optimal solution to (4.7), (4.8). Just like in section 2 is $c_{ij}-\bar{\bar{u}}_i+\bar{\bar{v}}_j = 0$ for all arcs (i,j) of the shortest path tree T. So the optimal primal solution is obtained by sending a flow $n-1$ from node 1 to each node $j \in J$ over the arcs of T. This is possible because all forward arcs of T have infinite capacity and all backward arcs have capacity $n-1$. So T is the optimal tree in G', and the arcs (i,j) in T with $i \neq j$ correspond with the optimal tree in G. Moreover,

$$\bar{\bar{u}}_j = \bar{\bar{v}}_j, \quad j = 2,...,n, \tag{4.14}$$

and these numbers are equal to the shortest distances from node 1 to nodes j in G.

References

R.E. Burkard and U. Derigs (1980), *Assignment and Matching Problems: Solution Methods with FORTRAN-Programs*, Springer, Berlin.

U. Derigs (1988), *Programming in Networks and Graphs*, Springer, Berlin.

E.W. Dijkstra (1959), *A note on two problems in connexion with graphs*, Numerische Mathematik 1, p. 269-271.

B. Dorhout (1973a), *Enige toewijzingsproblemen*, Communication at the Ninth Dutch Mathematical Congress, Leiden.

B. DORHOUT (1973b), *Het lineaire toewijzingsprobleem: vergelijking van algoritmen*, Report BN 21, Stichting Mathematisch Centrum, Amsterdam.

B. DORHOUT (1977), *Experiments with some algorithms for the linear assignment problem*, Report BW 39, Stichting Mathematisch Centrum, Amsterdam.

R. JONKER and A. VOLGENANT (1987), *A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems*, Computing 38, p. 325-340.

G. KINDERVATER, A. VOLGENANT, G. DE LEVE and V. VAN GIJLSWIJK (1985), *On dual solutions of the linear assignment problem*, European Journal of Operational Research 19, p. 76-81.

H. KREUZBERGER (1971), *Eine Methode zur Bestimmung mehrerer Lösungen für das Zuordnungsproblem*, Angewandte Informatik 9, p. 407-414.

W.M. NAWIJN and B. DORHOUT (1988), *On the expected number of assignments in reduced matrices for the linear assignment problem*, Operations Research Letters, to appear.

N. TOMIZAWA (1971), *On Some Techniques Useful for Solution of Transportation Network Problems*, Networks 1, p. 173-194.

H.M. WAGNER (1975), *Principles of Operations Research*, 2nd edition, Prentice-Hall, Englewood Cliffs.

Heuristics for the hierarchical network design problem

C. Duin, A. Volgenant
Institute of Actuarial Sciences and Econometrics
University of Amsterdam

The Hierarchical Network Design Problem is a minimum spanning tree problem involving two weight functions: the edges on the elementary path between two given nodes are to be weighted by a more expensive cost function. This problem is shown to be NP-hard.

Three heuristics are considered: the method introduced by Current, Revelle and Cohon in 1986, a revised version, and a newly developed heuristic based on branch chord exchanges. An implementation for the latter is given of square order complexity in the number of nodes. This heuristic appears to be best on 3000 symmetric test problems (up to 100 nodes and from sparse to full dense), yielding nearly always the optimal value. In asymmetric networks the revised heuristic is expected to be the best.

1. Introduction

Most models for network problems define only one cost function on the edges of the network. However, in real life problems often more than one cost function is involved. Current, Revelle and Cohon (1986) were the first to study a minimum spanning tree problem with two costs on each edge of a network. Their Hierarchical Network Design Problem can be formulated as follows.

Let $G = (N,s,t,E,c,d)$ be a (directed) double weighted graph with N the set of nodes; s and t 'special' nodes (the source and target); E the set of edges; and c respectively d, a primary and secondary weight function on E.

The (Directed) Hierarchical Network Design Problem denoted $(D)HNDP$, is the problem of finding a (directed) spanning tree T (rooted into t) of minimum hierarchical weight:

$$z(T) = \Sigma_{(i,j)\in P} c_{ij} + \Sigma_{(i,j)\in T\setminus P} d_{ij},$$

where P denotes the fundamental path from s to t in T is primary weighted.

Applications arise in situations where an overall connection is required but an expensive high quality linkage is only necessary between the two special nodes. For example, in the design of a new road infrastructure two major towns have to be connected by highways while others can be linked with unimproved roads.

Two special cases of the HNDP are well known and satisfactorily solved: when $c = d$ the problem is an ordinary Minimum Spanning Tree problem; when $d = 0$ it reduces to a Shortest Path problem. On the other hand $(D)HNDP$ is a special case of the Hierarchical Steiner problem that considers two or more special nodes, see Duin & Volgenant (1989).

As most applications approve this, we will restrict ourselves to the undirected *HNDP*, i.e., the two cost functions c, d are symmetric $n \times n$ matrices ($n = |N|$) with $c_{ij} \geqslant d_{ij}$ for all $i,j \in N$ ($c_{ij} = d_{ij} = \infty$ if $(i,j) \notin E$). For this special case Duin & Volgenant (1989) developed reduction tests that reduce the size of the problem graph by showing redundancy of edges and/or nodes and sharpened the linear programming lower bound of Current et al. The complexity of the problem (with $c \geqslant d$) was still open. Here we will prove that it is *NP*-hard. In section 3 two heuristics are considered for the *HNDP*, both of $O(n^2)$ complexity and with worst case ratio 2. In section 4 we develop a more sophisticated greedy branch chord exchange algorithm and a lower bounding procedure. Computational results in section 5 illustrate the performance of these three heuristics and show the superiority of the branch chord heuristic.

Let us first agree on the notation and terminology to be used:

- We often refer to a subgraph in (N,E) as a subset of N or a subset of E. Also subsets of N or E may be identified with the subgraphs they induce in (N,E).
- For any $n \times n$ matrix p and a subset of edges B the quantity **p(B)** denotes $\Sigma_{(i,j) \in B} p_{ij}$.
- The symbol $\mathbf{T}^*$ stands for an optimal solution, $\mathbf{P}^*$ is its primary weighted path connecting s to t and $\mathbf{z}^*$ denotes its optimal weight: $c(\mathbf{P}^*) + d(T^* \setminus P^*)$.
- For any path P from s to t, the tree T^*_P is an optimum solution under the extra condition $P^* = P$; let z^*_P denotes its objective value.
- With **S** we denote a minimum spanning tree with respect to the secondary weights d.
- For any two nodes i,j in a tree T spanning N, the subset of edges $\mathbf{T{<}i,j{>}}$ constitutes the elementary path in T connecting nodes i and j.
- For typesetting reasons suffixes may be bracketed, e.g., we write $c(i_0,j_0)$ instead of $c_{i_0 j_0}$.

2. A polynomial transformation of 3-satisfiability to HNDP

With the number of special nodes bounded the Steiner problem is polynomially solvable, while the Hierarchical Steiner problem is not, even not the *HNDP*. We will prove this by showing that the *HNDP* problem is *NP*-hard.

It suffices to transform an arbitrary instance of a known *NP*-complete problem to an instance of (the recognition version of) the undirected *HNDP* with $c \geqslant d$. The problem 3-Satisfiability (3-SAT) is suited.

Let $C = \{c_i | i = 1,2,...,m)$ be a collection of three-element sets of literals over a set of boolean variables U, say $c_i = \{x_i, y_i, z_i\}$. A literal x over U is either equal to a boolean variable u of U or to **u** the negation of such a variable. Now 3-SAT asks whether there exists a truth assignment $\alpha: U \rightarrow \{\text{true,false}\}$, that satisfies C, i.e., for each c_i in C at least one of the 3 literals x_i, y_i, z_i is true under α. The transformation we use to reduce the instance (U,C) of 3-SAT to an instance G of *HNDP* is of the constructive type.

For any literal x let $|x|$ denote the number of occurrences of x in the sets

$\{x_i,y_i,z_i\}$ for $i = 1,2,...,m$. The *HNDP* problem graph G is complete and undirected. It has as node set N a union of m disjoint subsets N_i, constructed from $\{x_i,y_i,z_i\}$ as follows:

$$N_i = \{vx_i,wx_i\} \cup N_i(x_i) \cup \{vy_i,wy_i\} \cup N_i(y_i) \cup \{vz_i,wz_i\} \cup N_i(z_i)$$

with $N_i(x)$ containing exactly $|\mathbf{x}|$ vertices for any literal x. So the number of nodes n in N is

$$6m + \Sigma_{i=1}^{m}(|\mathbf{x}_i| + |\mathbf{y}_i| + |\mathbf{z}_i|) = 6m + 2\Sigma_{r\in U}|r|\,|\mathbf{r}|.$$

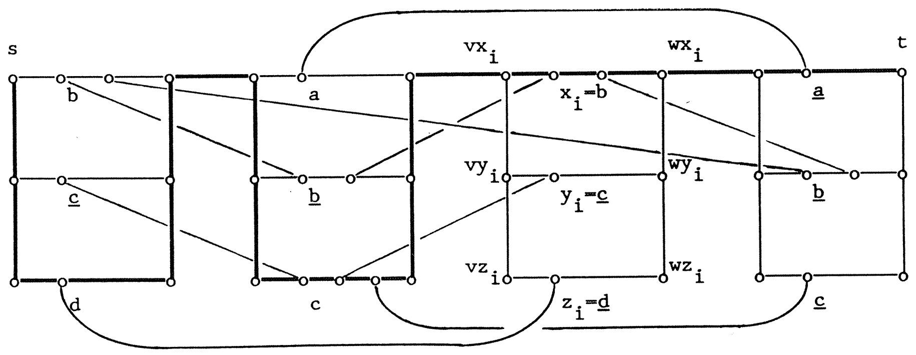
The primary weight function assumes only values 1 and ∞. The edges of primary weight 1 are the edges incident with vertices of N_i $(i = 1,2,...,m)$ that are continuously drawn in figure 1. All other edges have primary weight ∞. The secondary weight function assumes value 1 everywhere except on a set E_0 with $\Sigma_{r\in U}|r|\,|\mathbf{r}|$ edges in it, where it attains zero value. For every pair of literals $r \in c_i$, $s \in c_j$ with $r = \mathrm{s}$, precisely one edge exists in E_0. This edge connects a vertex of $N_i(r)$ with a vertex of $N_j(s)$; they can be chosen in such a way that no edges of E_0 share a common node. The construction of G can be done in time polynomial in $\max\{m, |U|\}$.

Truth assignments α on U that satisfy C, canonically correspond with optimal primary paths from $s = vx_1$ to $t = wx_m$ in G. For every $i = 1,2,...,m$ the path P_α^* visits the nodes of only one of the sets $N_i(x_i)$, $N_i(y_i)$ or $N_i(z_i)$, corresponding to a literal that is true under α (at least one is); see figure 1. Now the question whether C is satisfiable is equivalent with the question whether there exists an *HNDP* solution of weight $n - |E_0| - 1$. To understand this, note that for any successful truth assignment α the path P_α^* cannot contain both endnodes of an edge in E_0, so P_α^* can be extended to an optimal *HNDP* solution T with E_0 entirely in it of weight $z(T) = n - |E_0| - 1$. On the other hand if we have an *HNDP* solution T of weight $z(T) = n - |E_0| - 1$, then $E_0 \subset T$, so $P = T{<}s,t{>}$ cannot contain both sets $N_i(r)$ and $N_j(\mathbf{r})$ for any pair $r \in c_i, \mathbf{r} \in c_j$. Thus when defining

$$\begin{aligned} \alpha(r) \quad &= \text{true} \ \text{ if } r \in c_i \text{ and } N_i(r) \subset P \text{ for some } i = 1,2,...,m, \\ &= \text{false otherwise,} \end{aligned}$$

this solution corresponds to a correctly defined truth assignment α satisfying C.

The given transformation is an adaption of the transformation Gabow, Maheswari and Osterweil (1976) use to show that the search for a path in a digraph with forbidden pairs of nodes is *NP*-complete. With the adaption it follows that their problem is also *NP*-complete in an undirected graph with disjoint pairs of nodes forbidden.



$U = \{a,b,c,d\}$; $C = \{\{b,\mathbf{c},d\}, \{a,\mathbf{b},c\}, \{b,\mathbf{c},\mathbf{d},\}, \{\mathbf{a},\mathbf{b},\mathbf{c}\}\}$;
dashed: edges of E_0; continuous: edges with primary weight 1;
——: $P^\star_\alpha$ for α = (a,false), (b,true), $c,\mathit{true})$, $(d,\mathit{true})\}$;

FIGURE 1. Transforming an instance of 3-SAT into an instance of HNDP.

3. Heuristics for the *HNDP*

A special case of the *HNDP* was already mentioned in the introduction: when matrix d equals $\varnothing$ the problem is a Shortest Path Problem. The heuristic described by Current, Revelle and Cohon (1986) emphasizes this aspect of *HNDP*; the steps of the algorithm are:

Step 1. Determine a shortest path P from s to t with respect to the cost matrix c.

Step 2. Set the secondary arc costs of P on zero and extend it to a minimal spanning tree U on the transformed secondary cost matrix.

Step 3. With the arcs of P primary weighted and the arcs in $U \setminus P$ secondary weighted the tree U is a feasible solution.

Current et al. recommend to apply the above procedure for the second best (third best etc.) shortest path, selecting the solution with the minimum weight. Let z_{CK} be the minimum weight for the K best shortest paths. The time complexity to obtain z_{CK} is $O(Kn^2)$ in undirected graphs and $O(Kn^3)$ in directed graphs, which are the time complexities for finding the K-best shortest paths by respectively the algorithm of Yen (1971) and of Katho (1982). We will only consider $C1$ here.

Steps 2 and 3 are optimal in the sense that the best possible solution with primary path fixed at P is returned, i.e. $z_{C1} = z^\star_P$. We can write

$$z^\star_P = c(P) + d(S) - b_P,$$

for some $b_P \geqslant 0$, denoting the savings P induces in secondary connection costs.

In its attempt to approximate the optimal primary path, $C1$ merely minimizes the primary path costs $c(P)$ neglecting its secondary costs consequences. With other or more nodes on the primary path P, the savings b_P may increase more than the cost of the primary path, see figure 2.

To approximate this effect, we use a matrix p, defined as

$$p_{ij} = c_{ij} - \mu_j \qquad i,j \in N,$$

where

$$\begin{aligned} \mu_i &= \min\{d_{ki} \mid k \in N\} && i \in N \setminus \{s\}, \\ &= 0 && i = s. \end{aligned}$$

For any path P from 1 to n the difference

$$c(P) - p(P) = \Sigma_{i \in P} \mu_i$$

is a lower bound on the savings in step 2, where we have all the nodes of P in the secondary *MST* at zero cost.

It seems reasonable to revise step 1 of the heuristic of Current et al. by choosing the path P as the shortest with respect to matrix p instead of c. With this modification the heuristic is likely to produce better solutions. In order to decide which nodes to incorporate in the primary path instead of connecting them by secondary arcs, not only the primary cost of P is a criterion for its choice, but also the lower bound $\Sigma_{i \in P} \mu_i$ on the savings in secondary connection cost.

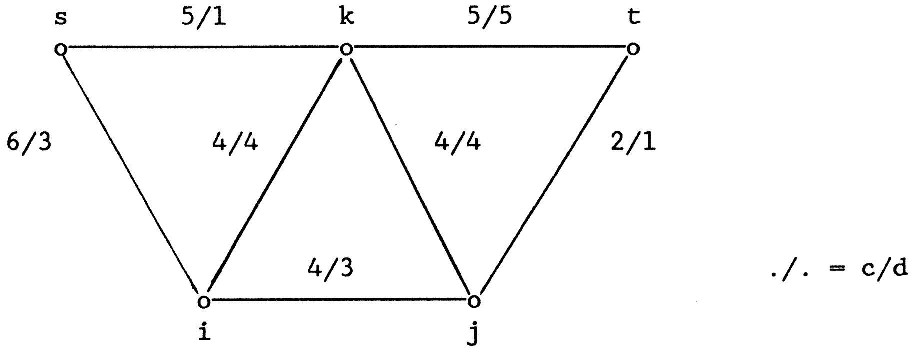


FIGURE 2. Going via k implies $z = 10+4$; going via i,j implies $z = 12+1$.

The reduction test Primary Edge Elimination as formulated in Duin & Volgenant (1989) can eliminate arcs of the primary path computed by the original heuristic, while it fails on the primary arcs chosen by the modified version. Moreover the new procedure produces a lower bound value on the cost of the hierarchical network, namely $\mu + \pi_{st}$, where π_{st} stands for the length of the shortest path from s to t with respect to $((p))$ and μ is defined as $\Sigma_{i \in N} \mu_i$. The argument is simple; for any feasible solution T we have

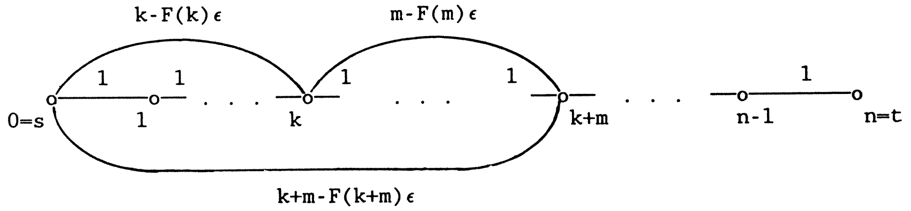
$$z(T) \geqslant \mu + c(P) - \Sigma_{i \in P} \mu_i \geqslant \mu + p(P) \geqslant \mu + \pi_{st}.$$

In a graph with nonnegative edge weights a shortest path between two specified nodes can be found in order n^2 time with the algorithm of Dijkstra. A minimum spanning tree can be found in the same time complexity order. It follows that the heuristics have time complexity $O(n^2)$.

To determine the worst case ratios, consider the network in figure 3. Here all secondary weights are equal to one. The primary weights are defined as

$$\begin{aligned} c_{ij} &= |i-j| - F(|i-j|)\epsilon && \text{if } |i-j|>1, \\ &= 1 && \text{if } |i-j| = 1; \end{aligned}$$

where $F(k)$ is the k-th Fibonacci number and ϵ is an infinitesimal number smaller than $1/nF(n)$. The optimal value here is $z^* = n$ with $T^*(=P^*=S)$ equal to the horizontal line $s-t$. As $F(k+m)>F(k)+F(m)$ for any two nodes the direct connection is the shortest primary connection.



(all secondary weights equal to 1; $F(i+1) = F(i)+F(i-1)$, $F(1) = F(2) = 1$)

FIGURE 3. Worst case for heuristic *CK*.

Current's heuristic chooses $P = \{(s,t)\}$ and has outcome $2n-1-F(n)\epsilon(>2n-1-\frac{1}{n})$, whereas the modified heuristic ($p_{i,i+1}$ being zero) chooses the optimal primary path. For growing value of n the ratio z_{C1}/z^* approximates 2. Moreover when $s-t$ paths are considered in order of increasing primary length, the optimal solution is found for the 2^{n-1}-th path and one can show that for any polynomial $K(n)$ the ratio z_{CK}/z^* approximates 2 for $n \to \infty$.

Suppose the problem graph of figure 3 is extended by edges (k,k') for $k = 1,2,...,n$ with primary and secondary weight zero. Then again $z^* = n$, only this time (the μ-values being zero) $C1$ as well as its revised version $C\mu$ choose nonoptimally with outcome $2n-1-F(n)$. So also z_μ/z^* approximates 2 for $n \to \infty$.

On the other hand we have that z_μ and z_{C1} are at most $\gamma + d(S) \leq 2z^*$, where γ is the length of the shortest primary path from s to t. Thus both heuristics have worst case ratio 2. In section 5 we will compare them on test problems.

4. Bounds by branch chord methods

For the undirected *HNDP* we know that all secondary edges in an optimal solution are part of S, see Duin & Volgenant (1989). The best solution T_P^* with weight z_p^* for a compulsory primary path P can be found as follows:

> Initially take S as the 'solution'. Then bring in with primary weight each edge of P, while deleting the longest of the **secondary weighted** edges from the cycle that would otherwise have been created.

In other words the solution T_P^* is formed by exchanging a set of branches B_P in S for the set P of chords; the savings in secondary connection costs b_P, defined as $c(P)+d(S)-z_P^*$, is equal to $d(B_P)$. This approach leads to a lower bound as well as an upper bound procedure.

The lower bound procedure uses the notion of bottleneck distance. For any nodes $i,j \in N$, we define the bottleneck distance b_{ij} as the maximum secondary weight on the elementary path $S<i,j>$ in the minimum spanning tree S. It is a straightforward check that, conform the definition given in Duin & Volgenant (1986), for all $i,j \in N$:

$$b_{ij} = \min[\max\{d_{vw} \mid (v,w) \in P_{ij}\} \mid P_{ij} \text{ is a path connecting } i \text{ and } j]. \qquad (3)$$

Lemma. *For any primary path P the value b_P is at most $\Sigma_{(i,j)\in P} b_{ij}$.*

Proof. Consider the above given process for transforming S into T_P^* with weight z_P^*. Assume P has m edges. For $r = 0,1,...,m$ P^r stands for the subset of the first r edges in P. Let T^r denote the solution at stage r with P^r as compulsory set of edges, e.g., $T^0 = S$, $T^m = T_P^*$. Then for $r = 0,1,...,m$ the tree T^r is the *MST* with respect to the modified secondary weight matrix d^r. By definition d^r is equal to d except on P^r: $d_{ij}^r = 0$ for $(i,j) \in P^r$.

When T^{r+1} is formed out of T^r, say by bringing in $(i,j) \in P^{r+1} \setminus P^r$, the secondary weight of some longest d^r-branch in $T^r<i,j>$ is returned and this weight is equal to the bottleneck length b_{ij}^r of (i,j) with respect to d^r. Because of equality (3) and $d^r \leqslant d$, we have $b_{ij}^r \leqslant b_{ij}$ and the desired inequality follows:

$$b_P = d(B_P) = \Sigma_{(i,j)\in P} b_{ij}^r \leqslant \Sigma_{(i,j)\in P} b_{ij}.$$

Proposition. *Let γ^b denote the length of the shortest path from s to t with respect to difference matrix $c^b = c - b$. Then $\gamma^b + d(S)$ is a lower bound for the HNDP with source s and target t.*

Proof. Applying the lemma to P^* we have

$$\begin{aligned} z^* &= c(P^*) - b_{P^*} + d(S) \geqslant c(P^*) - \Sigma_{(i,j)\in P^*} b_{ij} + d(S) \\ &= c^b(P^*) + d(S) \geqslant \gamma^b + d(S). \end{aligned}$$

The idea for a new approximation algorithm is to actually perform the branch chord exchanges when computing the heuristical primary path. Starting from s we reach out for t in a greedy way.

At any time during the algorithm we maintain for every node j: z_j,λ_j and e_j. The weight of a best solution found so far for the *HNDP* with target node j, is denoted by z_j. In this solution λ_j is the predecessor node of j on the primary path from s to j and e_j denotes the branch in B_P that is replaced by (λ_j,j). We proceed as in the shortest path algorithm of Dijkstra (1959). The set N is partitioned in a set of labeled nodes L and a set of unlabeled nodes $N \setminus L$. For each labeled node u we store the solution tree T^u with weight z_u. Initially only s is labeled, T^s is S and we terminate as soon as t gets labeled. In the scanning step the edges incident to a newly labeled node are considered as primary edges in an attempt to improve the z_j values for $j \in N \setminus L$. For any tree T, let T_{ij} be the longest secondary weighted edge on the elementary path $T_{<i,j>}$.

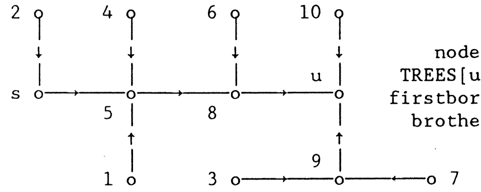
Now the complete algorithm is given below.

```
Step 1  (initialization)
            $L := \{s\};\; T^s := S;$
            for every $j \in N \setminus L$: $\lambda_j := s,\; e_j := T^s_{sj},\; z_j := d(S)+c_{sj}-d(e_j)$;
Step 2  (labeling a new node)
            $u := argmin\{z_j \mid j \in N \setminus L\};\; L := L \cup \{u\}$;
            $T^u := (T^{\lambda_u} \cup \{(\lambda_u,u)\}) \setminus \{e_u\}$;
            if $u = t$ then STOP:
Step 3  (scanning node u)
(a)  Give all edges of the primary weighted path in $T^u$ secondary weight zero;
(b)  for every $j \in N \setminus L$ do

            $z := z_u + c_{uj} - d(T^u_{uj})$,

            if $z < z_j$ then $z_j := z,\; e_j := T^u_{uj}$;

(c)  Restore the original secondary weights on the primary path in $T^u$;
     goto step 2.
```

In step 3(b) the solution T^u is transformed to a solution with value z of the *HNDP* with target j by extending the primary path $<s,...,\lambda(\lambda_u),\lambda_u,u>$ with edge (u,j). The additional cost is c_{uj} minus the longest of the secondary weighted edges on $T^u_{<u,j>}$; step 3(a) ensures that none of the primary weighted edges is considered here.

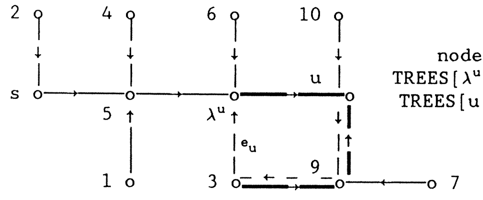
The algorithm ends with the approximate solution T^t of weight z_t. In a non-negatively weighted graph subpaths of shortest paths must also be shortest. The heuristic is based on the observation that for optimal primary paths this is usually also true. If $<s,i_1,i_2,...,i_m>$ is an optimal primary path for the HNDP problem with target i_m then the subpath $<s,i_1,i_2,...,i_{m-1}>$ is often an optimal primary path for the problem with target i_{m-1}. If this were true in general the algorithm would have been exact.

We will now show that with a suitable data structure the algorithm can be implemented in $O(n^2)$ time. The trees T^u are stored in an $n \times n$ array TREES of which row u represents T^u rooted into u, see figure 4.



nodes:	s	1	2	3	4	5	6	7	8	9	10	u
TREES[u]:	5	5	s	9	5	8	8	9	u	u	u	0
firstborn:	2	0	0	0	0	4	0	0	6	3	0	10
brother:	1	0	0	7	s	0	5	0	9	0	8	0

FIGURE 4. Data structures for tree T^u.



nodes:	s	1	2	3	4	5	6	7	λ^u	9	10	u
TREES[λ^u]:	5	5	s	λ^u	5	λ^u	λ^u	9	0	3	u	9
TREES[u]:	.	.	.	9	.	.	.	.	u	u	.	0

_ _ _: delete
——: insert

FIGURE 5. Forming T^u out of $T^\lambda u$.

```
scan( j: node for which step 3 is to be performed in current call;
    ej: edge assuming value $T^u_{uj}$ in current call;
    lej: real number assuming value $d(T^u_{uj})$)
begin
  if brother[j] > 0 then scan(brother[j],lej,ej);
  if $d(j,T^u[j]) > lej$ then
  begin $lej := d(j, T^u[j])$; $ej := (j,T^u[j])$ end;
  if $j \in N \setminus L$ then
  begin
    $z := z_u + c[u,j] - lej$;
    if $z < z[j]$ then
    begin $\lambda[j] := u$; $e[j] := ej$; $z[j] := z$ end;
  end;
  if firstborn[j]>0 then scan(firstborn[j],lej,ej)
end
```

FIGURE 6. Linear time procedure for step 3(b).

It suffices to show that each execution of step 2 and 3 requires $O(n)$ time.

This is easy for step 2. When making T^u, one first copies vector T^{λ_u}. Then this tree is rooted into u, see figure 5, by reversing the arrows along the path from u until and with the first node of e_u (which eliminates e_u from the tree) and by the assignment TREES[u][λ_u] := u the new edge (λ_u,u) is inserted. To perform step 3(b) efficiently one first generates in linear time two vectors called 'brother', and 'firstborn' for the tree T^u representing its offspring, see figure 4. After that the call scan(firstborn[u],0,0) of the recursive procedure given in figure 6 performs step 3(b) in $O(n)$ time.

5. Computational results on the *HNDP*

We give a computational comparison on problems with up to 100 vertices and three edge densities: sparse (with average degree 3), medium (with average degree 10), and complete. Nodes drawn as points in a square with uniformly random coordinates, were randomly connected by line segments to form a connected graph of the desired density.

In each graph (N,E), 50 *HNDP* instances were solved with 'Euclidean' weights assigned to the edges and the same 50 problems were solved with 'random' weights. In the Euclidean problem the secondary weight on an edge is taken as the Euclidean length (rounded up to integer value) of the associated line segment, while in the random problem it is a uniformly random number (rounded up integer) in the interval (0,100]. Both types have primary weights c_{ij} as the truncated value of $r_{ij} \cdot d_{ij}$ with $r_{ij} \in [1,2]$ uniformly random.

For each number of nodes and edges we generated 20 different graphs (N,E) and the results of $20^* 5 = 100$ *HNDP* instances are summarized in a line of table 1 for Euclidean weights and in table 2 for random weights. Every *HNDP* instance was solved by the heuristic of Current, Revelle and Cohon ($C1$), its adjusted version ($C\mu$) and the branch chord heuristic (BC). The results can be compared in three main columns. Information is given about the gaps between the upper bound considered and the best of the lower bounds (BL) of section 3 and 4 (nearly always the bottleneck bound). Each gap was measured as a percentage of the BL value; the table summarizes the number of gaps, the maximum gap and the average over the non-zero gaps.

As expected the number of gaps increases with the problem size, while the extent of the (relative) gap decreases; both tendencies are stronger for the Euclidean problems. In the (medium) dense case the Euclidean problems appear to be more difficult than the corresponding random problems. In random problems of larger size, the number of $C\mu \setminus BL$ and $BC \setminus BL$ gaps is maximal for medium dense problems. For complete random problems the latter number tends to become zero. This can be explained: secondary edge weights are uniform in [1,100], so in a dense graph the weights of all edges in the secondary MST will approximate value 1, i.e., $b_P = |P| - 1$. This is recognized by the BC heuristic, the bottleneck lower bound and the revised heuristic.

The results for the branch chord heuristic excel. For each of the 6 possible comparisons our program counted the number of times one heuristic performed worse than the other. Of the 3000 problems considered BC was beaten only once (by the revised heuristic). One may wonder which of the values BC or LB is (dominantly) non-optimal in case of a non zero BC/BL gap. For each problem size the instance attaining the maximum BC/BL gap was solved to

Table 1. Results on Euclidean graphs
(average of the non zero gaps and maximal gaps between lower and upper bound)

problem graph		heuristic $C1$			heuristic $C\mu$			heuristic BC		
nodes	degree	#	a	m%	#	a	m%	#	a	m%
20	3	29	2.1	6.9	12	1.7	3.6	3	1.0	1.6
	10	55	2.8	8.6	27	1.8	6.2	6	.9	1.1
	20	68	3.9	14.9	26	2.2	5.9	7	1.5	3.5
40	3	28	.9	3.2	16	.7	1.7	7	.4	1.4
	10	53	1.3	5.3	35	.9	3.2	14	.5	1.2
	40	80	3.7	11.2	38	1.4	5.2	6	.6	1.1
60	3	41	.9	2.9	27	.6	2.9	15	.2	.7
	10	61	1.1	4.4	35	.6	2.2	9	.2	1.2
	60	86	3.1	9.1	38	1.2	3.6	11	.3	.5
80	3	27	.5	2.4	23	.4	.9	7	.1	.2
	10	59	.9	2.9	47	.5	1.3	26	.3	.6
	80	82	2.7	8.0	59	.8	2.6	16	.3	1.1
100	3	35	.5	1.4	34	.4	.9	10	.1	.4
	10	56	.9	3.3	43	.4	1.1	20	.2	.4
	100	92	2.6	5.4	64	.8	3.0	14	.2	.3

optimality using techniques from Duin and Volgenant (1989). In all but two cases the BC heuristic was optimal.

We conclude that the BC heuristic is superior. However it cannot be applied on asymmetric problems, in contrast to the other two heuristics. The heuristic CK will probably perform better when applied for the $K(>1)$ best shortest paths. But then again the K-best path version of $C\mu$ having a similar implementation, is expected to perform better.

References

J.R. Current, C.S. Revelle and J.L. Cohon (1986), *The hierarchical network design problem*, European Journal of Operational Research 27, 57-66.

C.W. Duin and A. Volgenant (1989), *Reducing the hierarchical network design problem*, European Journal of Operational Research 39, 332-344.

E.W. Dijkstra (1959), *A note on two problems in connexion with graphs*, Numerische Mathematik 1, 269-271.

Table 2. Results on random graphs
(average of the non zero gaps and maximal gaps between lower and upper bound)

problem graph		heuristic $C1$			heuristic $C\mu$			heuristic BC		
nodes	degree	♯	a	m%	♯	a	m%	♯	a	m%
20	3	32	2.7	9.7	17	1.6	6.1	5	.8	2.1
	10	40	3.0	9.3	21	2.3	4.9	6	2.2	2.7
	20	68	3.8	9.6	21	2.8	9.1	5	1.3	2.5
40	3	31	1.3	5.5	27	.8	3.2	14	.6	1.1
	10	39	1.8	4.6	25	.5	3.9	3	.6	.9
	40	41	2.5	5.0	32	2.1	4.7	1	1.0	1.0
60	3	33	1.0	3.1	26	.6	1.5	11	.3	.9
	10	45	1.2	2.7	31	1.0	3.0	6	.4	.7
	60	40	2.1	6.2	41	1.8	3.7	8	1.0	1.3
80	3	39	.9	2.3	36	.6	1.4	15	.5	1.1
	10	56	1.4	3.9	39	.6	1.8	10	.2	.6
	80	30	1.9	4.2	28	1.8	4.7	1	1.2	1.2
100	3	35	.5	1.4	30	.4	1.4	15	.2	.4
	10	53	.9	2.6	44	.6	2.6	14	.3	.9
	100	28	2.1	8.8	22	1.6	2.9	0	-	-

H.N. Gabow, S.N. Maheswari and L. Osterweil (1976), *On two problems in the generation of program test paths*, IEEE Transactions on Software Engineering SE-2, 227-231.

M.R. Garey and D.S. Johnson (1979), *Computers and intractability, a guide to the theory of NP-completeness*, Freeman, San Francisco.

N. Katho, T. Ibaraki and H. Mine (1982), *An efficient algorithm for K shortest simple paths*, Networks 12, 411-427.

R.C. Prim (1957), *Shortest connection networks and some generalizations*, Bell Systems Technical Journal 36, 1389-1401.

J.Y. Yen (1971), *Finding the K shortest loopless paths in a network*, Management Science 17, 712-716.

Some approximation approaches in large-scale Markov decision problems: applications to one warehouse multiple retailer systems

A. Federgruen
Graduate School of Business
Columbia University, New York, NY 10027

1. Introduction and summary

Professor Gijs de Leve's treatise on generalized Markovian decision processes (1964) marked one of the founding contributions to the field of dynamic programming and Markov decision processes (MDP's). It, as well as his popular course on dynamic programming, inspired my own original research, see e.g., De Leve et al. (1977a, 1977b) and Federgruen (1978), as well as that of so many prominent scholars in the Netherlands. This paper is devoted to this special scholar and mentor at the occasion of the 25th anniversary of his chaired professorship at the University of Amsterdam.

Until recently, almost all research efforts in the dynamic programming area were focused on the development of efficient solution methods and the characterization of qualitative properties of *fully optimal* strategies. Most applications suffer unfortunately from the famous *curse of dimensionality,* i.e., the corresponding MDP's have state and action sets of large dimension and efficient algorithms for the determination of fully optimal strategies fail to exist. More importantly, the complexity of the structure of fully optimal strategies makes them unattractive and hard—if not impossible—to implement, even if such strategies could be computed efficiently.

It is therefore that in the last 5 to 10 years, the focus has shifted—and with considerable success—towards the identification of close-to-optimal, but not necessarily fully optimal, policies of relatively *simple* structure which are easy to compute and to implement, as well as accurate and easily computable *approximations* of the *total system-wide cost* for use in design and parametric studies.

Most approximation approaches start with an exact formulation of the planning problem as a dynamic program or Markov decision problem. As pointed out above, the large dimension of the associated state (and action) spaces precludes, in general, exact solution of these dynamic programs. The exact model is therefore replaced by an approximate one through the application of one or several manipulations of the problem like those used in large scale mathematical programming (relaxations, restrictions, projections, cost approximations, cf. GEOFFRION (1970)). These distinctions are important, because the properties of an approximation depend on the types and sequence of manipulations applied. If only relaxations of lower bound cost approximations are used, for example, the resulting approximation is a *lower bound* on the true optimal cost of the problem. This fact is very helpful in assessing optimality gaps for any heuristic strategy, since the cost of an appropriately constructed feasible strategy provides an upper bound on the optimal cost. (As is the case with most *mathematical programming* approximation methods such as Lagrangian relaxation, the heuristic strategy is usually based on the solution of the approximate model.) If the difference between upper and lower bounds is small, we can conclude *both* that the approximation is accurate *and* that the constructed policy is a good one.

Another approach is *first* to *restrict* the policy space to a more convenient and qualitatively appealing class. If determination of an optimal strategy within the chosen class is still intractable and the restriction is followed by one or more relaxations, the result is a lower bound, not on the original problem, but on the minimum cost among all policies within the class so that optimality gaps may be assessed with respect to the chosen class of strategies only.

One of the most basic multi-echelon distribution systems is that consisting of a central depot which supplies a collection of outlets $\{1, \ldots, J\}$ where *exogenous, random* demands for a *single* commodity must be filled. (The demand process may be correlated across outlets.) The deport places orders with an outside supplier which arrive there after a 'supplier leadtime' of L time units. The outlets are supplied by shipments from the depot which require a (second) shipment leadtime of l_j time units ($j = 1, \ldots, J$). Instantaneous perfect information about all inventory levels is assumed. Unfilled demand is backlogged. (To facilitate the exposition below, we assume that all leadtimes are deterministic. Random leadtimes, with arbitrary leadtime distributions, often an important factor in the design of an appropriate distribution strategy, may however be treated as well; we merely assume that consecutive orders or shipments to a given outlet do *not* cross in time, i.e., arrive in the sequence in which they were originated, and that leadtimes are independent of the demand process, see ZIPKIN (1986).)

Distribution systems of this type apply one of the following two basic management philosophies:

(a) Push systems, and
(b) Pull systems.

In Push systems, there is a central decision maker who possesses continuously or periodically updated information about inventories and sales at all relevant facilities. This information is used to decide

(i) when to place orders with an outside supplier,

(ii) in what quantities,
(iii) when to transfer stock from the upper to the lower echelon, and
(iv) how to allocate the released stock among the retailers.

In a Pull system, it is each outlet which places orders with the depot on the basis of its own *local* information (inventory level and shipments outstanding). The warehouse responds passively to these orders and fills them on a first-come-first-serve basis as long as sufficient stocks are available. The depot decides on all orders to the outside supplier, either on the basis of its own inventory position only, or on that of *all* facilities.

Analysis of a Push system model

Assume a Push system is adopted and that a system-wide replenishment strategy is sought to achieve an 'optimal' tradeoff of the following three cost components:

(i) variable shipment costs assumed to be proportional with the shipment volumes between the depot and the outlets;
(ii) inventory carrying and backlogging penalty costs, assumed to be proportional with the inventory levels and backlog sizes respectively;
(iii) costs for outside order, consisting of a fixed component and a variable one which is proportional with the order size (see FEDERGRUEN and ZIPKIN (1984a) for more general order cost structures).

In general, centralized ordering as represented here offers *two* distinct advantages. The first advantage is the possibility to exploit economies of scale in the order costs. Note also that one could choose to decide the future allocations at the same time as the order. Postponing the allocations permits one to observe the demands in the intervening l periods, and thus to make better-informed decisions. The phrase *'statistical economies of scale'* has been used (EPPEN and SCHRAGE (1981)) to describe this effect.

Another important distinction is that between systems where inventory is carried at the depot, and systems without central inventories. (The latter applies, for example, when the 'depot' does not represent a physical location at all, but rather a centralized ordering function; here decisions about shipment to the ultimate destinations can be postponed until L periods after the bulk order is placed. Even if it does correspond to a physical location, the depot acts as a transshipment center, not a stocking point.) ROSENFIELD and PENDROCK (1980) refer to such systems as 'uncoupled' and 'coupled' respectively.

For systems without central inventories and an infinite planning horizon, we refer to EPPEN and SCHRAGE (1981) and FEDERGRUEN and ZIPKIN (1984c) for a detailed discussion of approximation methods of the above described type. FEDERGRUEN and ZIPKIN (1984a) treat coupled systems with a finite planning horizon and possibly non-stationary parameters and demand distributions. Similar approximation analyses and results have been obtained for systems *with* central inventories and finite or infinite planning horizons, see FEDERGRUEN and ZIPKIN (1984b, 1989).

In Section 2, we discuss an alternative model for one-warehouse multiple retailer systems and give a detailed outline of successful relaxation and restriction approaches for this model.

2. An alternative model for coupled systems

An alternative model for two-echelon distribution systems *without* central stock, goes back to SILVER (1965). Here the assumption is that orders are received instantaneously by the depot, i.e., $L = 0$; alternatively, if L is positive, one assumes that any order to the outside supplier is to be allocated among the outlets, the very moment it is placed by the depot. This restriction eliminates the possibility of exploiting the *statistical economies of scale* mentioned in Section 1 (and hence the need to allocate incoming orders), by effectively eliminating the first leadtime component (L). Contrary to the model in Section 1, Silver's allows for *all* shipment and order cost functions to consist of *fixed* (as well as variable) components.

SILVER (1965) as well as all subsequent papers on this type of model, assumes that inventories are monitored *continuously* rather than periodically, but this distinction from the periodic review models in Section 1 is not essential. In fact, Federgruen and Zipkin have used continuous-time analogs of their models in some applications and NADDOR (1975) treats a periodic review analog of the Silver model. The earlier papers by VEINOTT (1965), BESSLER and VEINOTT (1966) and IGNALL and VEINOTT (1969) may all be viewed as periodic analogs of Silver's model, in the absence of *any* economies of scale i.e., linear costs throughout. (They may therefore also be viewed as special cases of the models in Section 1. On the other hand, these models allow for joint constraints in inventory positions.) These three papers show that myopic policies are optimal under various assumptions; and it is of interest that the (approximate) myopia results for the more general models mentioned in Section 1 are reminiscent of theirs. SOBEL (1977) extends the results in these three papers to demand processes which are correlated across outlets as well as time, following a general autoregressive moving-average structure.

Returning to Silver's model, demands are assumed to be generated by independent unit or compound Poisson processes. The independence of the demand processes represents an additional restriction. *This* multilocation model can easily be formulated as a semi-Markov decision problem; once again, in view of the dimensionality of the state space, exact solution methods are computationally intractible. Moreover, IGNALL (1969) showed that an optimal rule may fail to have a simple form (even in the case of *two* outlets) and would therefore be hard or impossible to implement even if it could be computed with reasonable resources.

The proposed *approximation* methods may be partitioned into those based on *relaxations* and those starting with a *restriction* of the policy space. (We use the term relaxation in the general sense popularized by GEOFFRION (1970), i.e., any approximation of a minimization model which results in a lower bound, e.g., expansions of the feasible set and/or replenishments of the objective function by lower bound objectives.) Since the vast majority of the literature on the Silver model is based on the restriction approach, and relaxation approaches have started to appear only recently, we treat the former first.

2.1. Restriction approaches

Most of the literature confines the policy space to so-called (s,c,S) or *can-order* policies: three parameters S_j, c_j, and s_j are specified for each location j with $s_j \leqslant c_j < S_j$. An order (to the outside supplier) is triggered by location j when its inventory position falls to or below the *reorder level* s_j; any location $i \neq j$ whose inventory position is at or below its *can-order level* c_i, is included in this order, and the inventory position of each location k included in the order is increased to the *order-up-to-level* S_k. (Silver's original model considered only the special cases where all can-order levels are set equal to either (i) the reorder levels c or (ii) the order-up-to levels S. The former case amounts to managing each item by itself ignoring opportunities to exploit economies of scale. The class of can-order policies was first introduced in BALINTFY (1964) in a random yield but deterministic demand model.)

For the special case of unit Poisson demands, SILVER (1974) provides an iterative method to compute a suboptimal policy in the above described class of (s,c,S) rules. THOMPSTONE and SILVER (1975) present a heuristic method for the special case of compound Poisson demands and *zero* leadtimes, by using a transformation of the compound Poisson distribution into an 'equivalent' unit Poisson distribution. Solution methods for the most general case are due to SILVER (1981) and FEDERGRUEN et al. (1984). See also PETERSON and SILVER (1979).

All of the above methods employ *decomposition* as an additional type of approximation, i.e., the multilocation model is decomposed into J single-location problems with iteratively adapted characteristics. (Decomposition techniques are common to many mathematical programming solution methods or evaluation and control methods for queueing networks.) Each single-location problem has *normal* replenishment opportunities at the location's demand epochs. The fixed cost of a *normal* order is given by the fixed depot order cost plus the fixed shipment cost to this specific location. In addition there are *special* replenishment opportunities where the fixed order costs are limited to the fixed shipment cost only; these opportunities arise at epochs generated by an independent Poisson process which is an approximation of the superposition of the order processes triggered by the other locations. (It is this representation of the superposition of these order processes, which represents the approximate element in the decomposition method.) For each location, the mean time between consecutive 'special replenishment opportunities' (and hence the parameter of the Poisson process representing these special epochs) is adapted iteratively. In FEDERGRUEN et al. (1984) an optimal (s,c,S) rule is found for each of the single location problems via a specialized and highly efficient policy-iteration method. (The latter generalizes the policy iteration methods in FEDERGRUEN and ZIPKIN (1984d, 1985) for finding optimal (s,S) policies in ordinary single location problems.) SILVER (1974), THOMPSTONE and SILVER (1975) and SILVER (1981) employ heuristic evaluation and search methods for the single location problems. The numerical study in FEDERGRUEN et al. (1984) suggests that the algorithm presented there, provides accurate (approximate) evaluations of the average cost of (s,c,S) policies as well as other performance measures of interest. On the other hand, in this restriction approach no bounds are obtained for the optimality gap of the generated (s,c,S) policies.

Analogous to the approaches mentioned in Section 1, one could restrict oneself to policies which determine when to place an order and of what size on the basis of the *aggregate* inventory position only. Indeed RENBERG and PLANCHE (1967) in a lesser known paper adopt a restriction approach of tis type: an order is placed when the aggregate inventory position falls to or below some system-wide reorder point R and all locations' inventory positions are increased to specific order-up-to levels $\{S_j\colon j=1,\ldots,J\}$. In the absence of fixed shipment costs and as shown for the models mentioned in Section 1, the aggregate inventory position represents an adequate proxy for the system state (the vector of the outlets' inventory positions) and indeed a perfect state description in the relaxed model ibid. The same is true when it is optimal to replenish all outlets with (roughly) equal frequencies. In other settings the restriction appears undersirable, even though it is quite commonly used practice: e.g., IBM's IMPACT system is based on this type of restriction approach, see KLEYNEN and RENS (1978) or VOLLMANN et al. (1984). See also PANTUMSINCHAI (1988) for a (not altogether conclusive) numerical comparison between $(W,S_1,\ldots,S_J)$ and (s,c,S) policies.

2.2. Relaxation approaches

Observe that in the Silver model, the need for centralized control arises exclusively because of the fixed cost that is incurred for orders to the outside supplier. Note also that if this *nonseparable* cost structure were replaced by a *separable lower bound* cost structure, the problem would decompose into J independent single location problems. As pointed out in the introduction of this section, such an approximation is referred to as a *relaxation*, since it results in a *lower bound* for the minimum achievable cost value.

A *separable lower bound* cost structure is obtained when allocating the fixed (depot) order cost K in fixed proportions $(\alpha_1,\alpha_2,\ldots,\alpha_J)$ with $\Sigma_{j=1}^J\alpha_j=1$ and $\alpha_j\geqslant 0$ $(j=1,\ldots,J)$. (Note that under the approximate cost structure, the fixed costs incurred under *any* replenishment strategy are less than under the exact cost structure.) In the resulting relaxed model it is clearly optimal to manage each location separately according to an (s,S) policy and the optimal (s,S) policy is easily found for each of the locations $j=1,\ldots,J$ with one of the algorithms discussed above. Thus, for $j=1,\ldots,J$, let

k_j = fixed shipment cost for location j,
$T_j(s,S)$ = expected time between consecutive order by location j under an (s,S) policy,
$C_j(s,S)$ = expected long run average holding and shortage cost for location j under the (s,S) policy.

We thus obtain the following solution of the relaxed problem:

$$(LB)_\alpha := \sum_{j=1}^{J}\psi_j(\alpha_j), \quad \text{where}$$

$$\psi_j(\alpha_j) := \min_{s,S}\{\frac{(\alpha_j K + k_j)}{T_j(s,S)} + C_j(s,S)\}.$$

Note that the functions $\psi_j(\cdot)$ are piecewise linear and *concave* and may be evaluated with the above mentioned standard algorithms for the determination of optimal (s,S) policies in single location models. For *each* allocation vector α, we thus obtain a lower bound (LB_α) for the minimum achievable cost value. The *best* such bound is clearly given by

$$(LB) := \max\ \{\sum_j \psi_j(\alpha_j) \mid \sum_j \alpha_j = 1;\ \alpha_j \geqslant 0\ (j=1, \ldots, J)\}.$$

Evaluation of (LB) thus reduces to maximizing a separable concave objective subject to a single constraint, an optimization problem for which many efficient solution methods exist, see e.g., ZIPKIN (1980) or FEDERGRUEN and GROENEVELT (1986). The above construction of the lower bound (LB) bears considerable similarity to the lower bounds obtained for mathematical programs via the popular technique of Lagrangian relaxation. (LB_α) has the additional advantage of being *separable* in α.

The above presentation is distilled from FEDERGRUEN and ZHENG (1989). The lower bound (LB) was however first presented in ATKINS and IYOGUN (1987, 1988). (The latter suggest employing the α-vector which maximizes the lower bound in the deterministic, constant demand rate analog of this model. Federgruen and Zheng show that the true value of (LB) may be efficiently computed.)

When computing (LB) we obtain the vector α^* maximizing (LB_α) and the corresponding J-tuple of (s,S) pairs. This vector of (s,S) policies represents a *feasible* system-wide replenishment strategy. For significant values of K it is however unlikely to be efficient since with probability 1 all orders generated under this strategy are made for *one* location only, i.e., the locations are managed independently and FEDERGRUEN et al. (1984) have shown that even the *best* independent control policy can be significantly more expensive than various heuristic coordinated strategies.

The following procedure generates a vastly superior replenishment strategy instead: For the allocation vector α^*, determine an optimal $(m_j, v_j^{\#})$ policy for each location $j=1, \ldots, J$, treating m as a *continuous* time-variable. (It is empirically known that in single location problems, optimal (m,v) policies come close to being globally optimal.) Next, round the vector of replenishment cycles $(m_j: j=1, \ldots, J)$ to a neighboring power-of-two $(m_j^*: j=1, \ldots, J)$, i.e., a vector in which all components are power-of-two multiples of a common base period, and implement the resulting $\{(m_j^*, v_j^{\#}): j=1, \ldots, J)$ policy. This rounding procedure is standard in constructing feasible strategies from lower bounds for many deterministic, constant demand rate models (e.g., ROUNDY (1985, 1986)) where it can be shown to result in very minor cost increases (2% or 6% at *worst*, depending upon the exact implementation of the rounding procedure). Note that the vector m^* consists of few *distinct* components. This together with the nestedness of the replenishment intervals in a power-of-two vector, induces a large degree of joint replenishments among the different locations.

ATKINS and IYOGUN (1988) use different heuristics for the determination of the vector of replenishment interval (m_j: $j=1,\ldots,J$): (i) $m_j=m$ for *all* locations, with m the *best* common interval value or (ii) $m_j=m$ for all locations j with $\alpha_j^*>0$, and m_j chosen as an (integer) multiple for all other locations j, i.e., with $\alpha_j^*=0$. Atkins and Iyogun show in a limited numerical study that their heuristic strategies outperform the (s,c,S) policies of Section 2.1 except for settings in which the fixed order cost K is small. In their sample, the cost value of the policies is on average 12% higher than the specific (suboptimal) lower bound computed in their method. Moreover it appears that the relaxation methods of ATKINS and IYOGUN (1988) and FEDERGRUEN and ZHENG (1989) are, in addition, computationally simpler than the methods of Section 2.1.

The relaxation approach has the additional advantage of being applicable to more general joint cost structures; indeed, as shown in FEDERGRUEN and ZHENG (1989), it is applicable to any joint cost structure which satisfies a general economies-of-scale (submodularity) property. The only change occurs in the specification of the lower bound (LB). Under a general submodular cost structure, the separable concave objective $\Sigma_j \Psi_j(\cdot)$ is to be maximized over a more general polyhedron, which is however still of a special (polymatroidal) structure allowing for highly efficient solution methods, see FEDERGRUEN and GROENEVELT (1986) and GROENEVELT (1985).

REFERENCES

D. ATKINS, P. IYOGUN (1987). A lower bound for a class of inventory/production problems. *Oper. Res. Letters 6,* 63-67.

D. ATKINS, P. IYOGUN (1988). Periodic versus can-order policies for coordinated multi-item inventory systems. *Management Science 34,* 791-795.

J.L. BALINTFY (1967). On a basic class of multi-item inventory problems. *Management Science 10,* 287-297.

A.A. BESSLER, A.F. VEINOTT (1966). Optimal policy for a dynamic multiechelon inventory model. *Naval Research Logistics Quarterly 13,* 355-389.

G. DE LEVE (1964). *Generalized Markovian Decision Processes,* Mathematical Centre Tracts 3 and 4, Mathematical Centrum, Amsterdam.

G. DE LEVE, A. FEDERGRUEN, H.C. TIJMS (1977a). A general Markov decision method I: model and method. *Advances of Applied Probability 9,* 295-316.

G. DE LEVE, A. FEDERGRUEN, H.C. TIJMS (1977b). A general Markov decision method II: applications. *Advances of Applied Probability 9,* 316-336.

G. EPPEN, L. SCHRAGE (1981). Centralized ordering policies in a multiwarehouse system with leadtimes and random demand. in *Multi-Level Production/Inventory Control Systems: Theory and Practice,* L. SCHWARZ (ed.), North-Holland, Amsterdam, 51-69.

A. FEDERGRUEN (1978). *Markovian Control Problems: Functional Equations and Algorithms.* Ph.D. dissertation, Mathematisch Centrum, Amsterdam.

A. FEDERGRUEN, A. GROENEVELT (1986). The greedy procedure of resource allocation problems: necessary and sufficient conditions for optimality. *Operations Research 34,* 909-919.

A. Federgruen, A. Groenevelt, H.C. Tijms (1984). Coordinated replenishments in a multi-item inventory system with compound Poisson demand and constant lead times. *Management Science 20,* 344-357.

A. Federgruen, Y.S Zheng (1989). Coordinated multi item inventory systems with stochastic demands and general joint cost structures (in preparation).

A. Federgruen, P. Zipkin (1984a). Approximation of dynamic, multilocation production and inventory problems. *Management Science 30,* 69-84.

A. Federgruen, P. Zipkin (1984b). Computational issues in an infinite-horizon, multi-echelon inventory model. *Operations Research 32,* 818-836.

A. Federgruen, P. Zipkin (1984c). Allocation policies and cost approximation for multilocation inventory systems. *Naval Research Logistics Quarterly 31,* 97-131.

A. Federgruen, P. Zipkin (1984d). An efficient algorithm for computing optimal (s,S) policies. *Operations Research 32,* 1268-1286.

A. Federgruen, P. Zipkin (1985). Computing optimal (s,S) policies in inventory models with continuous demands. *Advances of Applied Probability 17,* 424-443.

A. Federgruen, P. Zipkin (1989). Approximations for dynamic two-echelon inventory systems with centralized stocks (in preparation).

A. Geoffrion (1970). Elements of large-scale mathematical programming. *Management Science 16,* 652-691.

H. Groenevelt (1985). Two algorithm for maximizing a separable concave function over a polymatroid feasible region, Graduate School of Management, University of Rochester, Rochester.

E. Ignall (1969). Optimal continuous review policies for two product inventory systems with joint set-up costs. *Management Science 15,* 284-304.

E. Ignall, A. Veinott (1978). IMPACT. *Production and Inventory Management,* First Quarter.

E. Naddor (1975). Optimal and heuristic decisions in single and multi-item inventory systems. *Management Science 21,* 1234-1249.

R. Peterson, E. Silver (1979). *Decision Systems for Inventory Management and Production Planning,* Wiley, New York, 1979.

B. Renberg, R. Planche (1967). Un modèle pour la gestion simultanée des n articles d'un stock. *Revue d'Informatique et de Recherche Opérationnelle 1*(6), 47-59.

D. Rosenfield, M. Pendrock (1980). The effects of warehouse configuration design on inventory levels and holding costs. *Sloan Management Rev. 24,* 4, 21-33.

R. Roundy (1985). 98% effective-integer ratio lot-sizing for one warehouse multi-retailer systems. *Management Science 31,* 1416-1430.

R. Roundy (1986). 98% effective ratio lot-sizing rule for a multi-product, multi-stage production inventory system. *Mathematics of Operations Research 11,* 699-727.

E. Silver (1965). Some characteristics of a special joint-order inventory model.

Operations Research 13, 319-322.

E. SILVER (1974). A control system for coordinated inventory replenishment. *International Journal Production Research 12,* 647-671.

E. SILVER (1981). Establishing reorder points in the (S,c,s) coordinated control system under compound Poisson demand. *International Journal of Production Research 9,* 743-750.

M. SOBEL (1987). Dynamic affine logistics models, SUNY Stony Brook Working Paper, Stony Brook, NY.

R. THOMPSTONE, E. SILVER (1975). A coordinated inventory control system for compound Poisson demand and zero lead time. *International Journal of Production Research 13,* 581-602.

A. VEINOTT (1965). Optimal policy for a multiproduct dynamic nonstationary inventory problem. *Management Science 12,* 206-222.

T. VOLLMANN, W. BERRY, D. WHYBARK (1984). *Manufacturing Planning and Control Systems,* Dow Jones-Irwin, 549-560.

P. ZIPKIN (1980). Simple ranking methods for allocation of one resource. *Management science 26,* 34-43.

P. ZIPKIN (1986). Stochastic leadtimes in continuous-time inventory models. *Naval Research Logistics Quarterly 33,* 763-774.

On the transition probabilities of a Markov process with interventions

A. Hordijk
University of Leiden

Consider a Markov process. Any time the Markov process enters a given set U, an intervention brings it outside U. The transition probabilities for the process with these interventions are constructed. It is shown that they satisfy the Chapman-Kolmogorov equations.

1. Introduction

In December of 1967 I became research associate at the statistics department of the Mathematical Centre in Amsterdam. After I had studied a couple of introductory books on operations research, professor De Leve asked me to read and to correct the second volume of his Mathematical Centre Tracts on Generalized Markovian Decision Processes (reference [1]). In this tract the probabilistic background of the generalized Markovian decision process is treated. This decision process is a Markov process with extra instantaneous transitions, any time a closed set, say U, is entered. The original Markov process has as state space $(E, \mathfrak{B})$ a finite dimensional Borel space. Moreover, the sample paths are continuous from the right and have only a finite number of discontinuities in each bounded time interval. Consequently, entrance times in the set U are well defined. If the process enters U in state u then there will be an intervention of the decision maker which gives the process an instantaneous transition to a state in $U^c = E \setminus U$. The transition probabilities of this intervention are given by $q(u, B), u \in U, B = \mathfrak{B} \cap U^c$. The main problem in the tract was the question whether the process with interventions is again a strong Markov process. Since the analysis of the tract is very technical and inaccessible I spent many months in struggling my way through it. The analysis of De Leve is basically a sample path analysis. After reading the tract I started to approach the problem with constructing the transition probabilities and showing that they satisfy the Chapman-Kolmogorov equations. When the editors of De Leve's Liber Amicorum asked me for a contribution I thought of my twenty years old notes on this problem. Since extending the research is not possible for lack of time, I have restricted myself to rewrite them in english. Hopefully, the original telegram style in dutch has become a readable paper. Unfortunately, some assumptions about measurability still remain. It seems to me that it should be possible to get rid of them. Clearly, the analysis of this paper does not solve De Leve's original problem. Since the transition probabilities do satisfy the Chapman-Kolmogorov equations they correspond to a Markov process. However, to assure that they correspond to a strong Markov process, it seems that continuity properties on the transition probabilities are needed (cf. theorem 3.2 and 3.14 of [2]). If the original process is a standard process (see [2] p. 104) then an interesting question is, under which conditions on the instantaneous transition probabilities $q(u,B)$ are the transition probabilities of the process with interventions (see (4.1)) stochastically continuous functions. In case they are, they correspond to a standard Markov process and hence the strong Markov property holds (cf. [2] theorem 3.14).

We assume that the Markov process is a strongly measurable, complete, strong Markov process on the state space $(E,\mathfrak{B})$ with sample paths $x_t(\omega)$, $t \geqslant 0$, $\omega \in \Omega$. For $U^c = E \setminus U$ an open Borel set, let τ be the first exit time from U^c. Then the process with sample paths,

$$\tilde{x}_t(\omega) = x_{\min(t,\tau(\omega))}(\omega)$$

defines a strong Markov process (cf. [2] theorem 10.2). This process is called the stopped process. It is stopped at the first exit time of U^c or equivalently the first entrance time of U. We will denote the transition probabilities of the stopped process with $P_0(t, x, B)$, $t \geqslant 0$, $x \in E$ and $B \in \mathfrak{B}$. These transition probabilities are the building stones of our analysis. In the assumptions 3.1 and 3.2 we will assume existence and monotone dominance of the right hand side derivatives of $P_0(t, x, B)$ for $t=0$, $x \in U^c$ and $B \in \mathfrak{B} \cap U$. It would be interesting to analyze whether this assumption holds for the practical problems to which De Leve's Markovian decision analysis has been applied (see [4]).

2. Discrete time hitting transition probabilities

It is well known ([2]) that the transition probabilities $P_0(t, x, B)$, $t \in T$, $x \in E$, of the stopped process are monotone nondecreasing and right continuous in t for $B \in \mathfrak{B} \cap U$. The discrete time hitting transition probabilities are defined through the jumps in $P_0(t, x, B)$ as function of t. For all $t>0$, $x \in U^c$ and $B \in \mathfrak{B} \cap U$ we introduce the notation,

$$p_d(t, x, B) = P_0(t, x, B) - \lim_{h \downarrow 0} P_0(t-h, x, B).$$

It follows from the properties of the $P_0(t, x, B)$ transition probabilities that $p_d(t, x, B)$ is for fixed B a measurable function of the pair (t, x). Moreover, for any pair (t, x), $p_d(t, x, B)$ is a finite measure on $\mathfrak{B} \cap U$.

Since for all x and $t_1, \cdots, t_n$,

$$\sum_{i=1}^{n} p_d(t_i, x, U) \leqslant P_0(\max_i t_i, x, U) \leqslant 1,$$

we conclude that $p_d(t, x, U)$ has at most a countable number of t values for which it is positive. However these t values may depend on x.

We introduce $T(t, x)$ as follows; $\tau \in T(t, x)$ if $p_d(\tau, y, U) > 0$ on a set of y's with positive $P_0(t, x, \cdot)$-measure. We will show that $T(t, x)$ has at most a countable number of elements. Therefore we need the following relation.

LEMMA 2.1. *The following equality holds for all* $t \geq 0$, $\tau > 0$ *and* $x \in U^c$,

$$p_d(t+\tau, x, U) = \int_{U^c} P_0(t, x, dy) p_d(\tau, y, U).$$

PROOF.

$$\begin{aligned} p_d(t+\tau, x, U) &= \lim_{h \downarrow 0} (P_0(t+\tau, x, U) - P_0(t+\tau-h, x, U)) \\ &= \lim_{h \downarrow 0} (P_0(t, x, U) + \int_{U^c} P_0(t, x, dy) P_0(\tau, y, U) - P_0(t, x, U) \\ &\qquad - \int_{U^c} P_0(t, x, dy) P_0(\tau-h, y, U)) \\ &= \lim_{h \downarrow 0} \int_{U^c} P_0(t, x, dy)(P_0(\tau, y, U) - P_0(\tau-h, y, U)) \\ &= \int_{U^c} P_0(t, x, dy) p_d(\tau, y, U), \end{aligned}$$

where for the last equality the monotone convergence theorem is used. □

From this relation we conclude that

$$p_d(t+\tau, x, U) > 0 \text{ if and only if (iff) } \tau \in T(t, x).$$

Since $\{x\}$ has $P_0(0, x, \cdot)$-measure equal to one, it follows that

$$t + \tau \in T(0, x) \text{ iff } \tau \in T(t, x). \tag{2.1}$$

Since $T(0, x)$ is a finite or countable set the same is true for $T(t, x)$.

For any $B \in \mathfrak{B} \cap U$ the t-values of the jumps in $P_0(t, x, B)$ an contained in $T(0, x)$. The discrete time hitting transition probabilities are defined as the sum of all jumps, which is possible since there are at most a countable number of them. For any $x \in U^c$ and $B \in \mathfrak{B} \cap U$ we define,

$$P_d(t, x, B) = \sum_{\substack{0 < \tau \leq t \\ \tau \in T(0, x)}} p_d(\tau, x, B).$$

Remark that in this definition we take x, B inside resp. outside U. Hence they represent the discrete part of the probabilities of hitting the set U in subset B before or at time t.

We conclude this section with a relation for the discrete time hitting transition probabilities.

LEMMA 2.2. *The following equality is true for any* $x \in U^c$ *and* $B \in \mathfrak{B} \cap U$,

$$\int_{U^c} P_0(t, x, dy) P_d(s, y, B) = P_d(t+s, x, B) - P_d(t, x, B).$$

PROOF. From the definition of the discrete time hitting probabilities we have,

$$P_d(t+s,x,B) - P_d(t,x,B) = \sum_{\substack{t<\tau\leq t+s \\ \tau\in T(0,x)}} p_d(\tau,x,B).$$

With lemma 2.1 and relation (2.1) we find,

$$\int_{U^c} P_0(t,x,dy)P_d(s,y,B) = \int_{U^c} P_0(t,x,dy)(\sum_{\substack{0<\tau\leq s \\ \tau\in T(t,x)}} p_d(\tau,y,B))$$

$$= \sum_{\substack{0<\tau\leq s \\ \tau\in T(t,x)}} p_d(t+\tau,x,B) = \sum_{\substack{0<\tau\leq s \\ t+\tau\in T(0,x)}} p_d(\tau,x,B),$$

where in the second expression $p_d(\tau,y,B)$ is integrated with respect to $P_0(t,x,dy)$ and therefore we may restrict the summation to the τ's in $T(t,x)$. Combining the relations gives the assertion of the lemma. □

3. THE HITTING TRANSITION RATES

In this section we define the hitting transition rates. As far as we know their existence is not always guaranteed for the general state space $(E, \mathfrak{B})$. Therefore we make the following assumption.

ASSUMPTION 3.1. For any $x\in U^c$ and $B\in\mathfrak{B}\cap U$, $\lim_{h\downarrow 0} h^{-1}P_0(h,x,B)$ exists .

We call these limits the hitting transition rates at $t=0$. They will be denoted by $p_0(0,x,B)$. We also need the following properties of these hitting transition rates.

ASSUMPTION 3.2.

(i) $p_0(0,x,\cdot)$ is a measure on $\mathfrak{B}\cap U$,

(ii) for some finite and measurable function $g(x)$ and constant $h_0>0$ with $P_0(h,x,U)\leq hg(x)$ for all $0<h\leq h_0$, the integral $\int_{U^c}P_0(t,x,dy)g(y)$ is finite for all t and x.

Assumption 3.2 (ii) will be used to apply the monotone convergence theorem. From the existence of the time derivative at $t=0$ follows the existence of the time derivative at any time point. This consequence of the semigroup property is fairly standard. Let us state it in the following lemma.

LEMMA 3.1. *For any $x\in U^c$ and $B\in\mathfrak{B}\cap U$,*

$$\lim_{h\downarrow 0} h^{-1}(P_0(t+h,x,B)-P_0(t,x,B))$$

exists. With the notation $p_0(t,x,B)$ for this limit the following relation holds,

$$p_0(t,x,B) = \int_{U^c} P_0(t,x,dy)p_0(0,y,B) \tag{3.1}$$

PROOF. From

$$P_0(t+h, x, B) = \int_E P_0(t, x, dy) P_0(h, y, B)$$

it easily follows for $x \in U^c$ and $B \in \mathcal{B} \cap U$,

$$\begin{aligned} &h^{-1}(P_0(t+h, x, B) - P_0(t, x, B)) \\ &= \int_{U^c} P_0(t, x, dy)(h^{-1} P_0(h, y, B)). \end{aligned} \tag{3.2}$$

From the monotone convergence theorem and assumption 3.2 (ii) we conclude that the limit of the right-hand side does exist and that the relation (3.1) is satisfied. □

We will call $p_0(t, x, B)$, $t \leqslant 0$, $x \in U^c$, $B \in \mathcal{B} \cap U$ the hitting transition rates.

It is well known that $P_0(t, x, B)$ is a measurable function of the pair (t, x). From this it follows with standard arguments that $p_0(t, x, B)$ is also measurable in (t, x). It is also straightforward to show that for any t and $x \in U^c$ the function $p_0(t, x, \cdot)$ is a finite nonnegative measure on $\mathcal{B} \cap U$.

For later use we state in the following lemma a generalization of the relation (3.1).

LEMMA 3.2. *The following equality holds for all t, $s \geqslant 0$, $x \in U^c$ and $B \in \mathcal{B} \cap U$,*

$$p_0(t+s, x, B) = \int_{U^c} P_0(t, x, dy) p_0(s, y, B).$$

PROOF. It is easily seen that for $x \in U^c$ and $B \in \mathcal{B} \cap U$,

$$\begin{aligned} &h^{-1}(P_0(t+s+h, x, B) - P_0(t+s, x, B)) \\ &= \int_{U^c} P_0(t, x, dy)(h^{-1}(P_0(s+h, y, B) - P_0(s, y, B))). \end{aligned} \tag{3.3}$$

From relation (3.2) and assumption 3.2(ii) we find that the right-hand side is dominated by,

$$\int_{U^c} P_0(t, x, dy) \int_{U^c} P_0(s, y, dz) g(z) \leqslant \int_{U^c} P_0(t+s, x, dz) g(z) < \infty.$$

The assertion of the lemma follows from taking the limits for $h \downarrow 0$ in (3.3) and using the monotone convergence theorem. □

The discrete time hitting transition probabilities P_d, were defined as the sum over all jumps. We now define the continuous time hitting transition probabilities P_c, as the difference of P_0 and P_d, more precisely; for all $x \in U^c$ and $B \in \mathcal{B} \cap U$ let,

$$P_c(t, x, B) = P_0(t, x, B) - P_d(t, x, B).$$

It is easily seen that $P_c(t, x, B)$ as function of t is continuous and nondecreasing. Moreover, we showed that $p_0(t, x, B)$ is its righthand side derivative. Since $P_c(0, x, B) = P_0(0, x, B) = 0$ for $x \in U^c$ and $B \in \mathfrak{B} \cap U$, it follows from a well-known result in real analysis (see [3] p. 596) that,

$$P_c(t, x, B) = \int_0^t p_0(\tau, x, B) d\tau. \tag{3.4}$$

Hence the hitting transition rates are the time derivatives of the continuous time hitting transition probabilities.

4. The transition probabilities of the process with interventions

In this section we suppose that when the process enters the set U, say in state u, there will be an intervention which at the same instant brings the system to a state outside U. The new state is a random variable, its distribution is given by the jump transition probabilities $q(u, \cdot)$. As seen from the notation $q(u, \cdot)$ the distribution of the new state may depend on the entrance state u of the set U.

We assume that $q(u, B)$, $u \in U$ and $B \in \mathfrak{B} \cap U^c$ are transition probabilities. Hence $q(u, B)$ for $B \in \mathfrak{B} \cap U^c$ is measurable as function of u.

In [1] the process with interventions is studied via a sample path method. Here, we will approach it via the transition probabilities. We will give an explicit expression for the transition probabilities. Therefore, we need the regularity assumption that with probability one the process with interventions will enter the set U in any finite time interval only a finite number of times. Then in an inductive way we will define the transition probabilities given the process enters the set U exactly k, $k \in \mathbb{N}_0$ times.

The transition probabilities without entering U are given by,

$$P_0(t, x, B), \; t \geqslant 0, \; x \in U^c, \; B \in \mathfrak{B} \cap U^c.$$

Now suppose the transition probabilities with entering U precisely $(k-1)$ times or equivalently, with $(k-1)$ interventions, are given by $P_{k-1}(t, x, B)$. Then the transition probabilities with k interventions are equal to,

$$P_k(t, x, B) = \int_0^t d\tau \Big(\int_U p_0(\tau, x, du)\Big(\int_{U^c} q(u, dy) P_{k-1}(t-\tau, y, B)\Big)\Big)$$
$$+ \sum_{\substack{0<\tau\leqslant t \\ \tau \in T(0,x)}} \Big(\int_U p_d(\tau, x, du)\Big(\int_{U^c} q(u, dy) P_{k-1}(t-\tau, y, B)\Big)\Big).$$

For the existence of the integrals above we need that $P_{k-1}(t, x, B)$ is measurable in (t, x). For a process on a semi-compact state space (see [2]) this does not seem to be a severe restriction. However, we do not know general conditions implying it for all $k \geqslant 0$. Also in the proofs below we will interchange at various places the order of integration. Since all integrands are nonnegative there is no strong restriction of generality in assuming that this is correct. Our claim is that the transition probabilities of the process with interventions are given by,

$$P(t, x, B) = \sum_{k=0}^{\infty} P_k(t, x, B), \tag{4.1}$$

for $t \geqslant 0, x \in U^c$ and $B \in \mathfrak{B} \cap U^c$. To justify our claim we have to show that $P(t, x, B)$, $t \geqslant 0, x \in \mathfrak{B} \cap U^c$ is a semigroup of transition probabilities and moreover that they correspond to the process with interventions. The measurability properties follow from the assumption that they hold for P_k for all $k \geqslant 0$. With standard arguments it can be shown from the regularity assumption on the finiteness of the number of interventions in $[0, t]$ that $P(t, x, \cdot)$ is a probability measure on $\mathfrak{B} \cap U^c$. We will take it for granted that these transition probabilities do model the process with interventions. What remains to show is that the semigroup property holds. In the next section we show this by proving that the Chapman-Kolmogorov relation is satisfied.

In the remainder of this section we do some preparatory work.

LEMMA 4.1. *For all $t, s \geqslant 0, x \in U^c$ and $B \in \mathfrak{B} \cap U^c$,*

$$\int_{U^c} P_0(t, x, dy)(\int_0^s dt(\int_U p_0(\tau, y, du)(\int_{U^c} q(u, dz)P_k(s-\tau, z, B)))$$
$$= \int_t^{t+s} d\tau(\int_U p_0(\tau, x, du)(\int_{U^c} q(u, dz)P_k(t+s-\tau, z, B))).$$

PROOF. The proof follows straightforward from lemma 3.2 and an interchange of the order of integration. □

The analogous result with the discrete time hitting transition probabilities is:

LEMMA 4.2. *For all $t, s \geqslant 0, x \in U^c$ and $B \in \mathfrak{B} \cap U^c$,*

$$\int_{U^c} P_0(t, x, dy)(\sum_{\substack{0<\tau\leqslant s \\ \tau \in T(0,y)}} (\int_U p_d(\tau, y, du)(\int_{U^c} q(u, dz)P_k(s-\tau, z, B))))$$
$$= \sum_{\substack{t<\tau\leqslant t+s \\ \tau \in T(0,x)}} (\int_U p_d(\tau, x, du)(\int_{U^c} q(u, dz)P_k(t+s-\tau, z, B))).$$

PROOF. The proof is essentially the application of lemma 2.2. However, for clearness we do not refer to this lemma. In the first expression $p_d(\tau, y, \cdot)$ is integrated with respect to $P_0(t, x, dy)$. Therefore we may restrict to τ-values for which $p_d(\tau, y, U)$ is positive on a set of y's with positive $P_0(t, x, \cdot)$-measure. This set of τ's was defined as $T(t, x)$. Hence the first expression with interchanging the order of summation and integration and only the relevant part explicitly written out, is equal to

$$\sum_{\substack{0<\tau\leqslant s \\ \tau \in T(t,x)}} \int_{U^c} P_0(t, x, dy)p_d(\tau, y, du) \cdots .$$

With lemma 2.1 and relation (2.1) we find that this is equal to,

$$\sum_{\substack{t<\tau\leqslant t+s\\ \tau\in T(0,x)}} p_d(\tau, x, du)\cdots .$$

From which the assertion follows. □

5. The Chapman-Kolmogorov equations

The transition probabilities with exactly k interventions were denoted by $P_k(t, x, B)$, $t\geqslant 0$, $x\in U^c$, $B\in\mathcal{B}\cap U^c$. If there are n interventions in the interval $(0, t+s]$ and k in $(0,t]$ then there should be $(n-k)$ interventions $(t, t+s]$. Hence the equality of the following lemma is intuitively clear. It gives the main step to prove the Chapman-Kolmogorov equations.

Lemma 5.1. *For all $n\in\mathbb{N}_0$, $x\in U^c$ and $B\in\mathcal{B}\cap U^c$,*

$$\sum_{k+l=n}\int_{U^c} P_k(t, x, dy)P_l(s, y, B) = P_n(t+s, x, B).$$

Proof. The proof goes with induction on n. The Chapman-Kolmogorov equations for the $P_0(t, x, B)$ transition probabilities on $(U^c, \mathcal{B}\cap U^c)$ give the relation for $n=0$. Suppose the relation holds for n equal to m. To prove the relation for $m+1$ we split the sum $\Sigma_{k+l=m+1}\cdots$ in $\Sigma_{k+l=m+1, k\geqslant 1}$ and the term with $(k=0, l=m+1)$.

(i)
$$\sum_{\substack{k+l=m+1\\ k\geqslant 1}}\int_{U^c} P_k(t, x, dy)P_l(s, y, B)=$$

$$\sum_{\substack{k+l=m+1\\ k\geqslant 1}}\int_{U^c}[(\int_0^t d\tau + \sum_{\substack{0<\tau\leqslant t\\ \tau\in T(0,x)}})\int_U (p_0(\tau, x, du)+p_d(\tau, x, du))\cdot$$

$$(\int_{U^c} q(u, dz)P_{k-1}(t-\tau, z, dy))]P_l(s, y, B)$$

$$= (\int_0^t dt + \sum_{\substack{0<\tau\leqslant t\\ \tau\in T(0,x)}})\int_U (p_0(\tau, x, du)+p_d(\tau, x, du))\cdot$$

$$(\int_{U^c} q(u, dz)(\sum_{\substack{k+l=m+1\\ k\geqslant 1}}\int_{U^c} P_{k-1}(t-\tau, z, dy)P_l(s, y, B)))$$

$$= (\int_0^t d\tau + \sum_{\substack{0<\tau\leqslant t\\ \tau\in T(0,x)}})\int_U (p_0(\tau, x, du)+p_d(\tau, x, du))\cdot$$

$$(\int_{U^c} q(u, dz)P_m(t+s-\tau, z, B)),$$

where in the above shorthand notation the p_0, p_d has to be combined with the integral resp. the summation over τ; the last equality follows from the induction hypothesis.

(ii) The lemma's 4.1 and 4.2 imply that

$$\int_{U^c} P_0(t, x, dy)P_{m+1}(s, y, B) =$$

$$\Big(\int_t^{t+s} + \sum_{\substack{t<\tau\leq t+s \\ \tau\in T(0,x)}}\Big) \int_U (p_0(\tau, x, du) + p_d(\tau, x, du))\cdot$$

$$\Big(\int_{U^c} q(u, dz)P_m(t+s-\tau, z, B)\Big).$$

The final expressions of (i) and (ii) added together give the defining relation of $P_{m+1}(t+s, x, B)$ and the assertion is proved. □

The main result of this paper follows now easily. It shows that the transition probabilities of the process with interventions satisfy the Chapman-Kolmogorov equations.

THEOREM 5.1. *For all $t, s \geqslant 0$, $x \in U^c$ and $B \in \mathfrak{B} \cap U^c$,*

$$\int_{U^c} P(t, x, dy)P(s, y, B) = P(t+s, x, B).$$

PROOF. The proof follows by interchanging the order of integration and summation. Indeed,

$$\int_{U^c} P(t, x, dy)P(s, y, B)$$

$$= \int_{U^c} \Big(\sum_{k=0}^{\infty} P_k(t, x, dy)\Big)\Big(\sum_{l=0}^{\infty} P_l(s, y, B)\Big)$$

$$= \sum_{n=0}^{\infty} \sum_{k+l=n} \Big(\int_{U^c} P_k(t, x, dy)P_l(s, y, B)\Big)$$

$$= \sum_{n=0}^{\infty} P_n(t+s, x, B) = P(t+s, x, B),$$

where the equality before the last one is a consequence of lemma 5.1. □

REFERENCES

1. G. DE LEVE, *Generalized Markovian Decision Processes, Part II, Probabilistic Background,* 2nd Edition, Mathematical Centre Tracts, 4.
2. E.B. DYNKIN, *Markov Processes,* Volume 1, Springer-Verlag.
3. E.W. HOBSON, *The Theory of Functions of a Real Variable,* Cambridge University Press.
4. G. DE LEVE, A. FEDERGRUEN and H.C. TIJMS. *A general Markov decision method, I: model and method, II: applications,* Adv. Appl. Prob. 9 (1977), pp. 296-315, pp. 316-335.

Teaching linear assignment by Mack's algorithm

R. Jonker
Department of Mathematics and Systems Engineering
Koninklijke/Shell Laboratorium, Amsterdam

A. Volgenant
Institute of Actuarial Sciences and Econometrics
University of Amsterdam

In many introductory textbooks on Operations Research and Mathematical Programming the Linear Assignment Problem is solved by the Hungarian Method, probably because this is the oldest method. We highlight another method, the Bradford method of Mack; we think it to be more suited to explain a solution method for the Linear Assignment Problem. After a short review of assignment algorithms, we give an exposition of elements of these algorithms. Then we explain Mack's algorithm illustrated with an example and we will show it to be near - equivalent to the Hungarian Method. Finally we indicate an improvement on Mack's Method.

1. Introduction

In many introductory textbooks on Operations Research and Mathematical Programming, e.g., Daellenbach, George and McNickle (1983), and Taha (1987), the Linear Assignment Problem is introduced by the Hungarian Method, probably because this is the oldest method and maybe because not many textbooks treat any other method.

We highlight another method, the Bradford method of Mack (1969), which is theoretically equivalent to the Hungarian method; we think it to be more suited to explain a solution method for the Linear Assignment Problem. In 1969, Mack developed this method, which can be considered a forerunner of Tomizawa's (1971).

We start with a short review of assignment algorithms, followed by an exposition of elements of these algorithms. Then we explain Mack's algorithm illustrated with an example and we will show it to be near-equivalent to the Hungarian Method. Finally we indicate how Mack's method can be improved.

As is well known, the Linear Assignment Problem on the $n \times n$ cost matrix (c_{ij}) can be formulated as a linear program:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot x_{ij}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1,...,n),$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1,...,n),$$

$$x_{ij} \in \{0,1\} \quad (i,j = 1,...,n).$$

The zero-one constraints on the x_{ij} can be relaxed to nonnegativity restrictions, yielding the dual problem:

$$\max \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$$

subject to

$$c_{ij} - u_i - v_j \geqslant 0 \quad (i,j = 1,...,n).$$

From the dual variables u_i and v_j we may calculate the reduced costs $c_{ij} - u_i - v_j$ $(i,j = 1,...,n)$. So, the dual problem is to find a reduction of the cost matrix with maximum sum and non-negative reduced costs.

In this and the following sections, n is the number of assignments to make. Indices i and j refer to rows and columns respectively; x_i is the index of the column assigned to row i and y_j the index of the row assigned to column j, with $x_i = 0$ for an unassigned row i and $y_j = 0$ for an unassigned column j; the dual variable u_i corresponds to row i and v_j to column j. We denote the reduced costs $c_{ij} - u_i - v_j$ by $cred_{ij}$, and we may refer to the dual variables as 'prices'. An unassigned row or column is 'free'.

2. A review of algorithms

Methods to solve the Linear Assignment Problem can be classified (roughly) in three categories:

a. algorithms based on maximum flow,

b. algorithms based on shortest paths,

c. algorithms based on the simplex method.

Most algorithms based on *maximum flow* are primal-dual methods. Papadimitriou & Steiglitz (1982) give an introduction to these methods that is very well suited for use in the classroom. A primal-dual method for the Linear Assignment Problem performs the following steps:

step 1. find a feasible dual solution;

step 2. solve a restricted primal problem, that is, find a (partial) primal solution that has complementary slackness with the dual solution;

step 3. terminate if the solution of the restricted primal problem solves the Linear Assignment Problem; otherwise, solve a restricted dual problem to adjust the dual solution leading to a less restricted primal problem, and return to step 2.

Essentially, step 2 consists of solving a maximum flow problem on an auxiliary graph. The notion 'complementary slackness' known from the theory of linear programming will be described in section 3 for the use in the context of the Linear Assignment Problem.

Historically seen Kuhn's Hungarian algorithm (1955, 1956) was actually the 'father' of the general primal-dual algorithm. The original method had computational complexity $O(n^4)$, but later $O(n^3)$ versions were developed (Lawler (1976)). Bertsekas (1981) also presented a primal-dual algorithm. The method is of the Hungarian type, and the best version even uses the Hungarian algorithm itself.

The methods based on *shortest paths* are dual algorithms in the sense that dual feasibility exists and primal feasibility has to be reached. This is achieved by considering the Linear Assignment Problem as a minimum cost flow problem, which can be solved by steps that involve finding shortest (augmenting) paths on an auxiliary graph.

In this group two algorithms, both of time complexity $O(n^3)$, stand out: Tomizawa's from 1971 and Hung & Rom's from 1980. The latter method is the more ingenious, but the former approach the faster. The algorithm of Tomizawa is initialized with a partial primal solution and a corresponding feasible dual solution. The partial assignment is augmented into a complete solution by primal steps in each of which one shortest augmenting path is determined using Dijkstra's method (1959). Hung & Rom's initial solution is complete, but may be infeasible. They determine in each step a shortest path tree, which takes more effort, but may lead to finding more augmenting paths per iteration.

Jonker & Volgenant (1987) described a Linear Assignment Problem algorithm including a Pascal implementation, that appears to be faster than the best known methods from the literature.

We highlight in this note the so-called Bradford method of Mack (1969), especially for its intuitively appealing presentation. It resembles the method of Hung & Rom, but, as originally presented, has computational complexity $O(n^4)$. Adapting it to obtain complexity $O(n^3)$ results in an algorithm close to Tomizawa's.

The *linear programming* based algorithms in the third category are (very) specialized versions of the simplex method. The best published results are from Barr, Glover & Klingman (1977). A major difficulty with all of these methods is the phenomenon of zero pivot steps. This can be illustrated in the classroom by solving a Linear Assignment Problem example as a transportation problem, showing a lot of degeneracy: almost half of the basic variables are equal to zero. A drawback is also their relatively complex implementation, as compared to the other approaches. Computational experiments (Hung & Rom (1980)) show that they are outperformed by the best primal-dual and dual algorithms.

The $O(n^3)$ Signature algorithm presented by Balinski (1985) also belongs to this category. It considers feasible dual solutions corresponding to trees in the bipartite graph of row and column nodes. Since its first publication, some refinements have been published, but, up to now, no computational results have been presented.

3. Elements of linear assignment algorithms

Most primal-dual and dual Linear Assignment Problem algorithms are based on only a few standard operations:

- *initialization:* a feasible dual solution u_i $(i = 1,...,n)$ and v_j $(j = 1,...,n)$ is determined; the primal solution x_i $(i = 1,...,n)$, with corresponding y_j $(j = 1,...,n)$, is initialized so that complementary slackness holds, that is, $x_i = j$ only if $cred_{ij} = 0$;
- *finding an augmenting path:* a sequence of, alternately, row and column indices is determined, with the first an unassigned row, the last an unassigned column, and the intermediate columns and rows assigned in successive pairs;
- *augmentation:* augmentation of a partial solution can take place along an augmenting path by assigning all rows in the path to their succeeding column, which results in one more assignment;
- *adjustment of the dual solution:* prices are adjusted either to obtain at least one additional zero reduced cost coefficient, while maintaining complementary slackness, or to restore complementary slackness after augmentation of a partial assignment.

The concept of augmentation along alternating paths is the basis for every assignment algorithm. How to adjust the dual solution merits some thought. Each algorithm specifies its own rules for this operation. The purpose is to maintain complementary slackness, that is,

$$c_{ij} - u_i - v_j \geqslant 0 \qquad (i,j = 1,...,n), \tag{1}$$

$$c_{ik} - u_i - v_k = 0, \text{ if } x_i = k \quad (i = 1,...,n). \tag{2}$$

Substituting the u_i from (2) into (1) leads to

$$c_{ik} - v_k \leqslant c_{ij} - v_j \quad (j = 1,...,n).$$

This implies that for every assigned column k $(y_k = i)$ the v_k must be chosen so that

$$c_{ik} - v_k = \min\{c_i - v_j \,|\, j = 1,...,n\},$$

and for every assigned row i $(x_i = k)$ the u_i must be set at

$$u_i = c_{ik} - v_k.$$

So all assignments in a (partial) solution must correspond to row minima in the reduced costs matrix. After augmentation of a partial solution, this trivial observation will usually show the best way to adjust prices. It even forms the basis of the assignment algorithm of Mack (1969) in which algorithm only the values of the v-variables have to be recorded, making it easier to understand the solution procedure.

4. The algorithm of Mack

Mack's linear assignment method is easy to understand, and easy to use. We will show it to be near-equivalent to both the algorithm of Tomizawa and the Hungarian method. So, in a way, Mack's algorithm provides the best statement for an Hungarian-type assignment algorithm. The method is based on two trivial observations for the Linear Assignment Problem:

1. the cost matrix can be reduced without influencing the optimal solution;
2. an optimal solution is found if for a certain reduced cost matrix the row minima occur in different columns.

As long as 2 is not fulfilled, Mack's algorithm adjusts the reduced costs in such a way that the row minima are spread over more columns.

$$\begin{array}{cccc} 3^* & 7 & 6 & 6 \\ 1^* & 6 & 8 & 8 \\ 3 & 0^* & 8 & 1 \\ 0^* & 7 & 9 & 9 \end{array}$$

(a) start

$$\begin{array}{cccc} 6^* & 7 & 6 & 6 \\ 4^* & 6 & 8 & 8 \\ 6 & 0^* & 8 & 1 \\ 3^* & 7 & 9 & 9 \end{array}$$

(b)

$$\begin{array}{cccc} 6 & 7 & 6^* & 6 \\ 4^* & 6 & 8 & 8 \\ 6 & 0^* & 8 & 1 \\ 3^* & 7 & 9 & 9 \end{array}$$

(c)

$$\begin{array}{cccc} 8 & 7 & 6^* & 6 \\ 6^* & 6 & 8 & 8 \\ 8 & 0^* & 8 & 1 \\ 5^* & 7 & 9 & 9 \end{array}$$

(d)

$$\begin{array}{cccc} 9 & 8 & 6^* & 6 \\ 7^* & 7 & 8 & 8 \\ 9 & 1^* & 8 & 1 \\ 6^* & 8 & 9 & 9 \end{array}$$

(e)

$$\begin{array}{cccc} 9 & 8 & 6^* & 6 \\ 7 & 7^* & 8 & 8 \\ 9 & 1 & 8 & 1^* \\ 6^* & 8 & 9 & 9 \end{array}$$

(f) optimal

Figure 1. An assignment problem solved by Mack's algorithm (bases are starred).

We illustrate the method on a simple example, which has the cost matrix given in Figure 1(a). The column where for a certain row *i* the minimum current reduced costs occur is called the ***base*** of row *i*, denoted base_i. (The bases are starred in Figure 1.) The method terminates if every column contains one base.

First consider column 1, which contains more than one base. By increasing its entries we create an alternative position for one of the bases in this column. If column 1 is increased by 3, an alternative position is found for base_1, underlined in matrix (b). This position being free, we switch base_1 from column 1 to column 3 in matrix (c), using in fact an augmenting path of length two. In the next iteration we consider again column 1, still containing two bases. After increasing column 1 by the amount 2, we find a new possibility for base_2 in matrix (d). The corresponding column 2 is already occupied by a base, so from now on we must increase columns 1 and 2 simultaneously. An increase of 1 for both columns leads to the cost matrix of Figure 1(e). In this matrix a path exists, alternating between bases and alternative bases, along which the current set of bases can be spread over one more column. The solution in Figure 1(f) is optimal. Note that after every step, in (c) and in (f), the bases are spread over one more column.

This is a more formal statement of the *assignment algorithm of Mack.*

step 1. *Initialization:* Determine the bases in the cost matrix.
step 2. *Termination,* if every column contains one base.
step 3. Select a column j that contains more than one base.
Set COL := $\{j\}$ and ROW := $\{i \mid \text{base}_i \in \text{COL}\}$.
step 4. For $i \in$ ROW set $m_i = \min\{cred_{ik} - cred_{i,\text{base}_i} \mid k \in \text{COL}\}$.
Determine $\delta = \min\{m_i \mid i \in \text{ROW}\}$.
Let rr be a row and kk a column for which δ is assumed.
step 5. *Adjust the dual solution:* Increase the reduced costs of all columns in COL by the amount δ.
If column kk contains no base, go to step 6. Otherwise go to step 7.
step 6. *Augmentation:* A path has been found consisting of, alternately, bases and alternative bases, starting in column kk:

- alter the current set of bases along the alternating path,
- go to step 2.

step 7. Column kk is base for some row(s):

- mark column kk as alternative base for row rr,
- COL := COL $\cup\{kk\}$; ROW := ROW $\cup\{i \mid \text{base}_i = kk\}$,
- go to step 4.

Bunday and Garside (1987) have published a computer program for Mack's method.

5. Improving Mack's algorithm

When considering the complexity of Mack's algorithm, step 4 turns out to be inefficient. Its formulation above requires $O(n^2)$ operations to update the row minima over the columns not in COL. An alternative formulation is:

step 4. For $k \in$ COL set $mm_k = \min\{cred_{ik} - cred_{i,\text{base}_i} \mid i \in \text{ROW}\}$.
Determine $\delta = \min\{mm_k \mid k \in \text{COL}\}$.
Let rr be a row and kk a column for which δ is assumed.

When δsum is the sum of the δ-values by which columns in COL were increased in previous applications of step 5 during the current iteration, we note that

$$mm_k + \delta\text{sum} = d_k \quad (k = 1,\ldots,n).$$

This implies that updating the row minima is equivalent to determining a shortest path from the original column j (from step 2) to any unoccupied column in an auxiliary network. As in Tomizawa, Dijkstra's method can be used as we consider the (non-negative) reduced costs. So in this formulation, the column minima over the rows in ROW can be updated in $O(n)$ operations, and the computational complexity of the method becomes $O(n^3)$.

Using the second formulation of step 4 yields a method that is almost equivalent to that of Tomizawa (1971). Note, however, that Mack's method was published in 1969. Tomizawa's method is obtained as follows:

- rows are assigned only if the minimum reduced costs occur in an unassigned

column, that is, every column is assigned at most once;
- the set ROW is initialized (in step 3) with any unassigned row and COL with the empty set.

The improvement of this section may be omitted when treating Mack's method in the classroom. For further improvements, e.g., about finding an initial solution (step 1), we refer to Jonker & Volgenant (1987).

6. Conclusion

We have highlighted Mack's Linear Assignment Problem algorithm to promote it as a method to be discussed in textbooks on Operations Research and Mathematical Programming. We think the given explanation as well as the cohesion with other Linear Assignment Problem algorithms to contain enough arguments for the recommendation to use the method for teaching students about the Linear Assignment Problem.

References

M.L. Balinski (1985), *Signature methods for the assignment problem*, Operations Research 33, 527-536.

R. Barr, F. Glover and D. Klingman (1977), *The alternating path basis algorithm for assignment problems*, Mathematical Programming 13, 1-13.

D.P. Bertsekas (1981), *A new algorithm for the linear assignment problem*, Mathematical Programming 21, 152-171.

B.D. Bunday, G.R. Garside (1987), *Linear programming in Pascal*, Edward Arnold, London.

H.G. Daellenbach, J.A. George and D.C. McNickle (1983), *Introduction to Operations Research Techniques*, second edition, Allyn and Bacon, Boston.

E.W. Dijkstra (1959), *A note on two problems in connexion with graphs*, Numerische Mathematik 1, 269-271.

M.S. Hung and W.O. Rom (1980), *Solving the assignment problem by relaxation*, Operations Research 28, 969-982.

R. Jonker and A. Volgenant (1987), *A shortest augmenting path algorithm for dense and sparse linear assignment problems*, Computing 38, 325-340.

H.W. Kuhn (1955), *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly 2, 83-97.

H.W. Kuhn (1956), *Variants of the Hungarian method for assignment problems*, Naval Research Logistics Quarterly 2, 253-258.

E.L. Lawler (1976), *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart & Winston, New York.

C. Mack (1969), *The Bradford method for the assignment problem*, New Journal of Statistics and Operational Research 1, 17-29.

C.H. Papadimitriou and K. Steiglitz (1982), *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, New Jersey.

H.A. Taha (1987), *Operations Research: An Introduction*, fourth edition, Macmillan, New York.

N. Tomizawa (1971), *On some techniques useful for the solution of transportation problems*, Networks 1, 173-194.

Parallel local search for the time-constrained traveling salesman problem

G.A.P. Kindervater
Erasmus University, Rotterdam

J.K. Lenstra
Eindhoven University of Technology
Centre for Mathematics and Computer Science (CWI), Amsterdam

M.W.P. Savelsbergh
Eindhoven University of Technology
Georgia Institute of Technology, Atlanta

In the time-constrained TSP, each city has to be visited within a given time interval. Such 'time windows' often occur in practice. When practical vehicle routing problems are solved in an interactive setting, one needs algorithms for the time-constrained TSP that combine a low running time with a high solution quality. Local search seems a natural approach. It is not obvious, however, how local search for the TSP has to be implemented so as to handle time windows efficiently. This is particularly true when parallel computer architectures are available. We consider these questions.

1. Introduction

On May 2, 1969, Professor Gijs de Leve showed his newly-appointed assistant around in the Mathematical Centre, then located in an old school building. 'Here is our library,' he said. 'And this is how you do research. You just pick up a journal, and - well, there isn't any Markov programming here, but this may interest you.' The journal was a recent issue of *Operations Research*, and the paper was Bellmore and Nemhauser's survey of the traveling salesman problem [Bellmore & Nemhauser, 1968].

This was neither the first nor the last time that De Leve put someone on the track of the traveling salesman. As a result, the TSP has always occupied a central position in the research in combinatorial optimization at the University of Amsterdam and at the Mathematical Centre. This has led to a long list of publications, which probably starts with the survey by Tijdeman [1968]. It includes De Leve's own elegant improvement of the assignment bound [Jonker, De Leve, Van der Velde & Volgenant, 1980] as well as the impressive computational work of Jonker [1986] and Volgenant [1987]. The latest additions focus on the availability of new computer architectures for interactive and parallel computing and their consequences for the TSP.

In this contribution, we review some of this recent work. We give a nontechnical summary in Section 2. Sections 3-7 provide more detail; most of this material is adapted from Martin Savelsbergh's dissertation on interactive vehicle routing [Savelsbergh, 1988] and Gerard Kindervater's dissertation on parallel combinatorial computing [Kindervater, 1989].

2. Nontechnical summary

2.1. Theory versus practice

The theory of operations research is concerned with the investigation of a broad class of mathematical models that are somehow inspired by practical decision situations, and with the design and analysis of algorithms for their solution. The practice of operations research is an even broader and considerably less scientific occupation. A huge and ever growing pile of literature is devoted to the tension between theory and practice and to the inadequacy of the mathematical models and methods in giving real-world solutions to real-world problems.

For the benefit of the reader, we summarize this literature in one paragraph. The first observation is that decision problems tend to be both *soft* and *hard*. At the practical side, the decision situation is usually ill-defined and the quality of a decision, as expressed by notions like feasibility and optimality, is an imprecise concept. Feasibility requirements may be loose rather than strict, and tradeoffs between optimality criteria are often not explicitly known but carried implicitly in the value judgement of the decision maker. At the mathematical side, any reasonable abstraction of the decision situation is likely to be computationally intractable in a well-defined sense. The second observation is that no solution can be better than the model to which it provides an answer. While the construction of models that are both realistic and tractable is a delicate affair, the implementation of solution procedures and their results in practice is far more difficult. For applications of operations research, the modeling stage is a minor obstacle in comparison to the implementation stage.

One way out of these complications, which has been much advocated, is to create a so-called *gap* between theory and practice and to try to fill it up with the literature on operations research. Another approach, which yields more mutual benefits, goes under the name of *man-machine interaction*. The idea is that man and machine each have their given and complementary capabilities. Human problem solving is empirical by nature, based on generalization, insight, and experience. Automated problem solving is normative and proceeds by the efficient application of general rules in specific situations. An *interactive planning system* combines the strengths of both approaches. Roughly speaking, the planner is in charge of the global problem aspects and takes care of all kinds of ad hoc constraints, and the computer performs the routine work, such as the manipulation and representation of data and the solution of detailed subproblems. (The reader should note the contrast with *artificial intelligence*, which is concerned with the automation of tasks that are better done by human beings.) We refer to Anthonisse, Lenstra & Savelsbergh [1989] for a further elaboration on the functional and technical characteristics of these types of systems.

2.2. Local search

The emergence of interactive planning systems has reinforced the need for algorithms that can handle problems of a realistic size and give solutions of a reasonable quality in a reasonable amount of time. One often employs some form of local search of the solution space. Although theoretical results on the performance of local search algorithms are scarce and mostly negative, it is generally acknowledged that their empirical behavior is excellent. In addition to being *effective* and *efficient*, local search is also *robust* and *easy to program*. That is, a local search method for a certain model is usually readily adapted to handle minor variations of the model, and developing a computer code requires much less effort than in the case of highly structured optimization algorithms or tailored approximation techniques.

Local search owes this flexibility and simplicity to the fact that it proceeds on the basis of relatively little information about the problem under consideration. One only has to specify an *initial feasible solution* and fast subroutines that, given a feasible solution, compute its *cost* (i.e., the value of the objective function) and its *neighborhood* (i.e., a set of feasible solutions that are in some sense close to it). Given a starting solution, its neighborhood is searched for a solution of lower cost. If such a solution exists, it becomes the new starting point and the search continues. Otherwise, a local optimum relative to the neighborhood definition has been found.

This heuristic solution approach enjoys an increasing popularity. Many variants have recently been proposed, such as simulated annealing, tabu search, neural nets, and genetic algorithms. It is not our purpose to discuss this class of so-called *homeopathic algorithms* [Van Hee, 1989]. Rather, we will consider a plain and simple local search method for the TSP and examine its implementation when time constraints are added to the model.

2.3. Local search for the TSP

Like so many other approaches in combinatorial optimization, local search was first seriously investigated in the context of the TSP. Lin [1965] calls a traveling salesman tour *k-optimal* when it cannot be improved by replacing j of its edges by j other edges, for any $j \leqslant k$. It is not known whether, for any fixed value of $k \geqslant 2$, a k-optimal tour can be *generated* in polynomial time. However, it is trivial to observe that the k-optimality of a given tour through n cities can be *verified* in $O(n^k)$ time: there are $\binom{n}{k}$ ways to delete k edges; for each of these, there is a constant number of candidate improvements (where the constant depends on k); and each of these candidates can be evaluated in constant time. For example, if $k=2$, two edges are replaced by two other edges, and only four cost coefficients have to be checked in order to compute the length of the new tour.

Now suppose that each city has its own time window during which it must be visited, and again consider the case $k=2$. If two edges are replaced by two other edges, then a certain segment of the tour will be traversed in the opposite direction. In addition to the test for improvement, there is now also a test for feasibility with respect to the time windows. This takes time proportional to the length of the reversed segment. In general, a straightforward implementation of the algorithm requires linear rather than constant time for evaluating a single k-exchange and thereby $\Theta(n^{k+1})$ time for verifying the k-optimality of a tour. We will present a way to avoid this additional factor of n and to verify k-optimality for the time-constrained TSP in $O(n^k)$ time.

2.4. Serial and parallel computing

So far, we have implicitly assumed that our algorithms were to be executed on a traditional computer, which performs at most one computation at a time. An algorithm for a given problem is *likable* if the number of computations involved is bounded by a polynomial function of the problem size, and the algorithm is *more likable* if the degree of the polynomial is lower. Thus, we do not now if there is a likable algorithm for generating a k-optimal tour. However, such an algorithm does exist for verifying k-optimality of a given tour, and we like our $O(n^k)$ approach better that the obvious $O(n^{k+1})$ implementation.

Now suppose that we have a computer that can perform a number of operations in parallel. Such a computer has a greater processing power than a serial one. This is especially important in the context of man-machine interaction, where the user expects fast answers in real time.

More specifically, assume that we have an unbounded number of processors that operate in parallel and communicate with each other in constant time. Consider, as an example, the simple problem of finding the maximum of n numbers $a_1, a_2, \ldots, a_n$. At the first stage, one processor takes the maximum of a_1 and a_2, another processor takes the maximum of a_3 and a_4, and so on. At the second stage, about $n/2$ numbers are left, and again pairwise maxima are taken. So it continues. After $\lceil \log n \rceil$ stages, we have the overall maximum. (All logarithms in this paper are to the base 2.) It follows that the problem is solvable in logarithmic time on a linear number of processors and that, in order to achieve this, each processor needs to know only a small fraction of the entire problem instance. Indeed, if a problem of size n is solved in $\log n$ time, no single processor is able to read all of the problem data. It appears that, when we can compute in parallel, we can find algorithms that are *more than likable*.

2.5. Parallel local search for the TSP

We have explained that the maximum of n numbers can be found by $n/2$ processors in $\log n$ time. Similarly, the k-optimality of a tour through n cities can be verified by $O(n^k)$ processors in $O(\log n)$ time: each processor evaluates a single k-exchange in constant time, and the best of these is selected in logarithmic time. In both cases, it is not hard to reduce the number of processors involved by a factor of $\log n$. Hence, for the TSP, $O(n^k/\log n)$ processors do in time $O(\log n)$ what a single processor can do in time $O(n^k)$. We thus achieve a *perfect speedup*.

When time constraints are added, complications occur. Evaluating a single k-exchange seems to be a serial process, but it is not too hard to design a parallel implementation that requires logarithmic time and a linear number of processors. This leads to an algorithm for verifying k-optimality in $O(\log n)$ time using $O(n^{k+1}/\log n)$ processors. Further improvements are possible, and we can save a factor of n in the number of processors, again achieving a perfect speedup.

2.6. *Yet another summary*

Section 3 gives a brief and informal introduction into the relevant concepts of complexity theory. Sections 4 and 5 discuss serial and parallel local search for the unconstrained TSP, respectively; this material is relatively straightforward. Section 6 presents our implementation of serial local search for the time-constrained TSP, and Section 7 deals with the parallel case.

3. Serialism, parallelism, and complexity

Complexity theory deals with the classification of problems based on the *running time* and the *work space* required by algorithms for their solution. When considering parallel algorithms, we also have to take the *number ofprocessors* into account. Complexity theory concentrates on *decision* problems (i.e., problems that produce a 'yes' or 'no' answer), but this is not a severe restriction, since most other problems can be reformulated in terms of a limited series of decision problems. An optimization problem, for example, can be solved by posing questions about the existence of a feasible solution with at most or at least a given value.

In this section, we discuss some aspects of complexity theory that are of importance to combinatorial optimization. We do not intend to go into much detail, and refer to Garey & Johnson [1979] and Cook [1981] for more complete expositions.

Sequential computers are reasonably represented by models of computation such as the Turing machine and the random access machine (RAM). Given these models, we can define several complexity classes. The class $\mathcal{P}$ contains the problems that are solvable in *polynomial time*, i.e., the running time is bounded by a polynomial in the problem size. The problems in $\mathcal{P}$ are often called *well solved* or *easy*. $\mathcal{P}$SPACE contains the problems that are solvable in *polynomial space*, i.e., in work space that is bounded by a polynomial in the problem size. A very well studied class included in $\mathcal{P}$SPACE is $\mathcal{NP}$, the class of problems for which a feasible solution can be recognized as such in polynomial time. It is obvious that $\mathcal{P} \subseteq \mathcal{NP} \subseteq \mathcal{P}$SPACE, and it is conjectured that both these inclusions are proper.

Another class contained in $\mathcal{P}$SPACE, which has not attracted much attention in the context of serial computations, is POLYLOGSPACE. It consists of the problems that are solvable in *polylog space*, i.e., in work space that is polynomially bounded in the logarithm of the problem size. Many problems in $\mathcal{P}$ belong to POLYLOGSPACE, but it is generally believed that $\mathcal{P} \not\subseteq$ POLYLOGSPACE. We do know, however, that POLYLOGSPACE $\neq$ $\mathcal{P}$SPACE.

The classes $\mathcal{P}$SPACE and $\mathcal{NP}$ have their *complete* members. The $\mathcal{P}$SPACE-complete problems are generalizations of all other problems in $\mathcal{P}$SPACE in terms of transformations that require polynomial time. More precisely: a problem is $\mathcal{P}$SPACE-*complete under polynomial-time transformations* if it belongs to $\mathcal{P}$SPACE and if any other problem in $\mathcal{P}$SPACE is reducible to it by a transformation that requires polynomial time. It follows that, if any $\mathcal{P}$SPACE-complete can be shown to belong to $\mathcal{P}$, then $\mathcal{P}$SPACE = $\mathcal{P}$. Since this equality is not believed to be true, a polynomial-time algorithm for a $\mathcal{P}$SPACE-complete problem is very unlikely to exist. For the class $\mathcal{NP}$ and its complete members, the same properties hold.

$\mathcal{P}$ also has its complete problems. The $\mathcal{P}$-complete problems generalize all other problems in $\mathcal{P}$ in terms of transformations that require logarithmic work space. Formally: a problem is *log space complete for* $\mathcal{P}$ or, better, *$\mathcal{P}$-complete under log-space transformations*, if it belongs to $\mathcal{P}$ and if any other problem in $\mathcal{P}$ is reducible to it by a transformation using logarithmic work space. If any $\mathcal{P}$-complete problem would belong to POLYLOGSPACE, then $\mathcal{P} \subseteq$ POLYLOGSPACE. As this inclusion is believed to be false, an algorithm for a $\mathcal{P}$-complete problem that uses only polylogarithmic work space cannot be expected to exist.

Serial and parallel computations are related by a hypothesis known as the *parallel computation thesis* [Chandra, Kozen & Stockmeyer, 1981; Goldschlager, 1982]: *time bounded parallel machines are polynomially related to space bounded sequential machines*. That is, for any function T of the problem size n, the class of problems solvable by a machine with unbounded parallelism in *time* $T(n)^{O(1)}$ (i.e., polynomial in $T(n)$) is equal to the class of problems solvable by a sequential machine in *space* $T(n)^{O(1)}$. This thesis is a *theorem* for many 'reasonable' parallel machine models and 'well-behaved' time bounds; see Van Emde Boas [1985] for a survey.

A frequently used model of parallel computation is the parallel random access machine, or PRAM. The PRAM is a machine with an unbounded number of processors and a shared memory. The processors perform their operations in a synchronized fashion. Simultaneous reads from the same memory location are allowed, but simultaneous writes into the same memory location are prohibited. The computation starts with one processor activated; at any step, an active processor can do a standard operation or activate another processor; and the computation stops when the initial processor halts.

Current technology prohibits the realization of a shared memory and, hence, of a machine with PRAM-like properties. However, the PRAM model is of theoretical interest. It helps us in investigating the intrinsic parallelism in problems and algorithms. For example, Fortune & Wyllie [1978] showed that the class of problems solvable in $T(n)^{O(1)}$ time by a PRAM is equal to the class of problems solvable in $T(n)^{O(1)}$ work space by a Turing machine, if $T(n) \geqslant \log n$.

As a consequence, the class of problems solvable by a PRAM in polynomial time is equal to $\mathcal{P}$SPACE. Since the PRAM is able to solve the apparently difficult problems in $\mathcal{P}$SPACE (such as the $\mathcal{P}$SPACE-complete and $\mathcal{NP}$-complete ones) in polynomial time, it is obviously an extremely powerful model. The theorem by Fortune & Wyllie also implies that the problems in POLYLOGSPACE are exactly the ones solvable by a PRAM in *polylog parallel time*, i.e., in time that is polynomially bounded in the logarithm of the problem size. This leads to a distinction within the class $\mathcal{P}$.

The problems in $\mathcal{P}$ belonging to POLYLOGSPACE are solvable in polylog parallel time. They can be considered to be among the *easiest* problems in $\mathcal{P}$, in the sense that the influence of problem size on solution time has been limited to a minimum. (It should be noted here that a further reduction to sublogarithmic solution time is generally impossible. One reason for this is that a PRAM needs $O(\log n)$ time to activate n processors. A similar reason is that in any realistic model of parallelism a constant upper bound on the maximum number of connections of any processor to other processors leads to a logarithmic lower bound on the communication time between processors. That is, a fixed degree implies at least a logarithmic diameter of the processor network.)

On the other hand, the $\mathcal{P}$-complete problems are unlikely to admit solution in polylog parallel time. If any such problem would be solvable in polylog parallel time, it would belong to POLYLOGSPACE, and it would follow that $\mathcal{P} \subseteq$ POLYLOGSPACE. Hence, their solution in polylog parallel time is not expected. Any solution method for these *hardest* problems in $\mathcal{P}$ is likely to require superlogarithmic time and is therefore, loosely speaking, probably 'inherently sequential' in nature. This does not imply, of course, that parallelism cannot yield substantial speedups.

We can, therefore, distinguish within $\mathcal{P}$ between the 'very easy' problems, which are solvable in polylog parallel time, and the 'not so easy' ones, for which such a speedup due to parallelism is unlikely.

The picture of the PRAM model as sketched above is in need of some qualification. The model is theoretically very useful, but its unbounded parallelism is hardly realistic. The reader will have no difficulty in verifying that a PRAM is able to activate a superpolynomial number of processors in subpolynomial time. If a polynomial time bound is considered reasonable, then certainly a polynomial bound on the number of processors should be imposed. It is a trivial observation, however, that the class of problems solvable if both bounds are respected is simply equal to $\mathcal{P}$. Within this more reasonable model, $\mathcal{NP}$-complete and $\mathcal{P}$SPACE-complete problems remain as hard as they were without parallelism.

Discussions along these lines have led to the consideration of *simultaneous resource bounds* and to the definition of new complexity classes. For example, *Nick (Pippenger)'s Class* $\mathcal{NC}$ contains all problems solvable in polylog parallel time on a polynomial number of processors, and *Steve (Cook)'s Class* $\mathcal{SC}$ contains all problems solvable in polynomial sequential time and polylog space. Some sort of extended parallel computation thesis might suggest that $\mathcal{NC} = \mathcal{SC}$. This is a major unresolved issue in complexity theory, and outside the scope of this paper. We refer to Johnson [1983] for further details and more references.

4. Local search for the TSP

In the traveling salesman problem, one is given a complete undirected graph G with vertex set $\{1, \ldots, n\}$ and a travel time d_{ij} for each edge $\{i,j\}$, and one wishes to find a Hamiltonian cycle (i.e., a cycle passing through each vertex exactly once) of minimum total duration. We assume that the travel times satisfy the triangle inequality, i.e., $d_{ij} + d_{jk} \geqslant d_{ik}$ for each triple (i,j,k). The TSP is a well-known $\mathcal{NP}$-hard problem, for which many optimization and approximation algorithms have been proposed; cf. Lawler, Lenstra, Rinnooy Kan & Shmoys [1985].

We consider the following local search algorithm for the TSP. Construct an initial Hamiltonian cycle by taking an arbitrary permutation of the vertices or by applying a specific heuristic method such as the *nearest neighbor* rule or the *double minimum spanning tree* algorithm. Then try to improve the tour by replacing a set of k of its edges by another set of k edges, and iterate until no further improvement is possible. Such replacements are called *k-exchanges*, and a tour that cannot be improved by a k-exchange is said to be *k-optimal*. We will consider the case

$k = 2$ in detail. For $k > 2$, the analysis is conceptually similar but technically more involved.

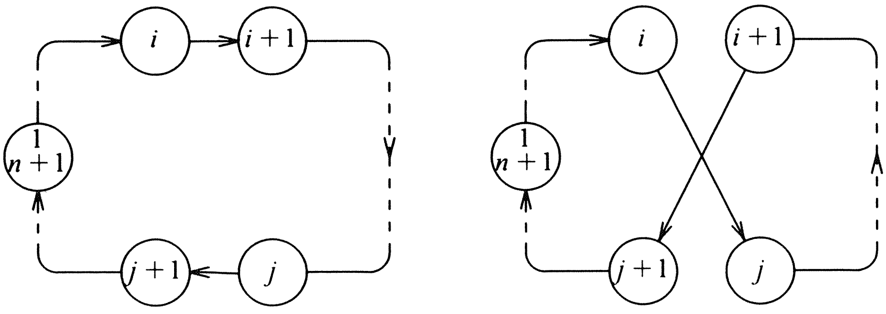


FIGURE 1. A 2-exchange.

For notational convenience, we consider the tour $(1,2,\ldots,n,n+1)$, where the origin 1 and the destination $n+1$ denote the same vertex. A 2-exchange replaces two edges $\{i,i+1\}$ and $\{j,j+1\}$ of the tour by two other edges $\{i,j\}$ and $\{i+1,j+1\}$, thereby reversing the path from $i+1$ to j; see Figure 1. It is an open question if there exists a polynomial-time algorithm that obtains a 2-optimal tour by a sequence of 2-exchanges [Johnson, Papadimitriou & Yannakakis, 1988]. We therefore restrict ourselves to deciding whether a given tour is 2-optimal.

Because the travel times between the vertices do not depend on the direction, a 2-exchange results in a local improvement if and only if

$$d_{ij} + d_{i+1,j+1} < d_{i,i+1} + d_{j,j+1}.$$

Testing a single 2-exchange for improvement involves only a constant amount of information and hence requires constant time. It follows that verifying 2-optimality takes $O(n^2)$ time. No algorithm that proceeds by enumerating all possible improvements can run faster, as there are $\binom{n}{2}$ 2-exchanges.

5. PARALLEL LOCAL SEARCH FOR THE TSP

Before discussing the verification of 2-optimality on the PRAM model, we will first consider an elementary problem and describe a basic technique in parallel computing for its solution.

The problem is to find the *partial sums* of a given sequence of n numbers. For the sake of simplicity, let $n = 2^m$ and suppose that the n numbers are given by $a_n, a_{n+1}, \ldots, a_{2n-1}$. We wish to find the partial sums $a_n + \cdots + a_{n+j}$ for $j = 0, \ldots, n-1$. The following procedure is due to Dekel & Sahni [1983]:

```
for $l \leftarrow m-1$ downto 0 do
    par $[2^l \leqslant j \leqslant 2^{l+1}-1]$ $a_j \leftarrow a_{2j} + a_{2j+1}$;
$b_1 \leftarrow a_1$;
for $l \leftarrow 1$ to $m$ do
    par $[2^l \leqslant j \leqslant 2^{l+1}-1]$ $b_j \leftarrow$ if $j$ odd then $b_{(j-1)/2}$ else $b_{j/2} - a_{j+1}$.
```

Here, a statement of the form '**par** $[\alpha \leqslant j \leqslant \omega]\, s_j$' denotes that the statements s_j are executed in parallel for all values of j in the indicated range.

The computation is illustrated in Figure 2. In the first phase, represented by solid arrows, the sum of the a_j's is calculated. Note that the a-value corresponding to a non-leaf node is set equal to the sum of all a-values corresponding to the leaves descending from that node. In the second phase, represented by dotted arrows, each parent node sends a b-value (starting with $b_1 = a_1$) to its children: the right child receives the same value, the left one receives that value minus the a-value of the right child. The b-value of a certain node is therefore equal to the sum of all a-values of the nodes of the same generation, except those with a higher index. This implies, in particular, that at the end we have $b_{n+j} = a_n + \cdots + a_{n+j}$ for $j = 0, \ldots, n-1$.

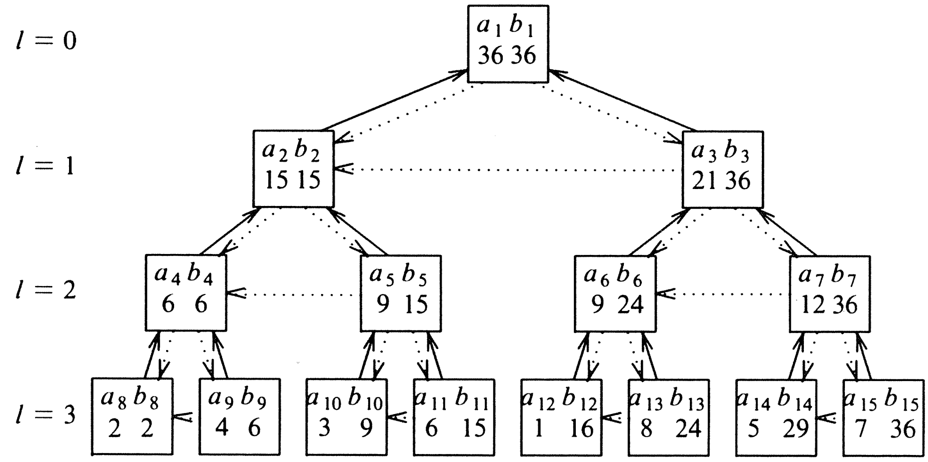


FIGURE 2. Partial sums: an instance with $n = 8$.

The algorithm requires $O(\log n)$ time and n processors. This can be improved to $O(\log n)$ time and $O(n/\log n)$ processors by a simple device. First, the set of n numbers is partitioned into $n/\log n$ groups of size $\log n$ each, and $n/\log n$ processors determine the sum of each group in the traditional serial way in $\log n$ time. After this aggregation process, the above algorithm computes the partial sums over the groups; this requires $O(n/\log n)$ processors and $O(\log n)$ time. Finally, a disaggregation process is applied with the same processor and time requirements.

In the form given above, the algorithm does not work for operations such as maximization. The partial sums algorithm uses subtraction, which has no equivalent in the case of maximization. We therefore present a version of the partial sums algorithm which is not quite so elegant as the original one, but which has the desired property since it makes use of addition only. It also runs in $O(\log n)$ time using $O(n/\log n)$ processors:

```
for $l \leftarrow m-1$ downto 0 do
    par $[2^l \leqslant j \leqslant 2^{l+1}-1]$ $a_j \leftarrow a_{2j}+a_{2j+1}$;
for $l \leftarrow 0$ to $m$ do
    par $[2^l \leqslant j \leqslant 2^{l+1}-1]$
        $b_j \leftarrow$ if $j=2^l$ then $a_j$ else if $j$ odd then $b_{(j-1)/2}$ else $b_{(j-2)/2}+a_j$.
```

We now return to the verification of 2-optimality. The following procedure decides whether or not the tour $(1,2,\ldots,n,n+1)$ is 2-optimal:

```
par $[1 \leqslant i < j \leqslant n]$ $\delta_{ij} \leftarrow d_{ij}+d_{i+1,j+1}-d_{i,i+1}-d_{j,j+1}$;
$\delta_{\min} \leftarrow \min\{\delta_{ij} \mid 1 \leqslant i < j \leqslant n\}$;
if $\delta_{\min} \geqslant 0$
then $(1,2,\ldots,n,n+1)$ is a 2-optimal tour
else let $i^*$ and $j^*$ be such that $\delta_{i^*j^*} = \delta_{\min}$,
    $(1,\ldots,i^*,j^*,j^*-1,\ldots,i^*+1,j^*+1,\ldots,n+1)$ is a shorter tour.
```

By adapting the first phase of the partial sums algorithm such that it computes the minimum of a set of numbers and also delivers an index for which the minimum is attained, the above procedure can be implemented to require $O(\log n)$ time and $O(n^2/\log n)$ processors. The total computational effort is $O(\log n \cdot n^2/\log n) = O(n^2)$, as it is in the serial case. This is called a *full processor utilization* or a *perfect speedup*.

Although the serial and parallel implementations seem similar, there is a basic distinction. When the tour under consideration is not 2-optimal, the serial algorithm will detect this after a number of steps that is somewhere in between 1 and $\binom{n}{2}$. In the parallel algorithm, confirmation and negation of 2-optimality always take the same amount of time.

6. Local search for the time-constrained TSP

In the TSP with time windows, each vertex i has a time window on the departure time, denoted by $[s_i,t_i]$. The time window is opened at time s_i and closed at time t_i. If the salesman arrives at i before s_i, he has to wait; if he arrives after t_i, he is late and his tour is infeasible.

Due to the presence of time windows, there are feasible and infeasible tours, and this complexifies the problem. To start with, the problem of determining the existence of a feasible tour is $\mathcal{NP}$-complete in the strong sense. This follows from the observation that the unconstrained TSP has a tour of duration no more than B if and only if there is a feasible tour for the constrained TSP in which each vertex has a time window $[0,B]$.

Secondly, when applying local search, we have to test all candidate improvements for feasibility. A k-exchange influences the arrival times at all vertices visited after the first change in the tour. This may lead to changes in the departure times and even to infeasibility. In a straightforward implementation, we need $O(n)$ time to handle a single k-exchange, which results in a time complexity of $O(n^{k+1})$ for the verification of k-optimality. We will show how to reduce this time bound by an order n, thereby obtaining the same time complexity as in the unconstrained case.

The basic idea is the use of a specific *search strategy* in combination with a set of *global variables* such that testing the feasibility of a single exchange and maintaining the set of global variables require no more than constant time. We consider the case $k = 2$ in detail.

As before, we consider the tour $(1,2,\ldots,n,n+1)$. We assume that this tour is feasible. A 2-exchange involves the replacement of the edges $\{i,i+1\}$ and $\{j,j+1\}$ by the edges $\{i,j\}$ and $\{i+1,j+1\}$. Such an exchange is both feasible and profitable if and only if the following three conditions are satisfied:

(1) the reversed path $(j,\ldots,i+1)$ is feasible, i.e., the new departure time at vertex k is not larger than t_k, for $k=i+1,\ldots,j$;

(2) the new departure time at vertex $j+1$ is smaller than it was before the exchange;

(3) a part of the gain at vertex $j+1$ can be carried through to the destination, i.e., the original departure time at vertex k is strictly larger than s_k, for $k=j+1,\ldots,n$.

Condition (3) needs further consideration. If it is violated, the exchange will not affect the duration of the tour. However, it will reduce the duration of the path from 1 to $k-1$, for the smallest k for which violation occurs. In the sequel, we will drop condition (3), for two reasons. First, introducing some slack may be beneficial for the rest of the procedure, even though the slack cannot be carried through to the end of the tour. In addition, taking condition (3) into account would make the presentation needlessly complicated.

We propose a *search strategy* that examines the 2-exchanges in lexicographic order. We choose i successively equal to $1,2,\ldots,n-2$; this will be referred to as the outer loop. For a fixed value of i, we choose j successively equal to $i+2,i+3,\ldots,n$; this will be called the inner loop. In the inner loop, the previously reversed path $(j-1,\ldots,i+1)$ is repeatedly expanded with the edge $\{j,j-1\}$; cf. Figure 3.

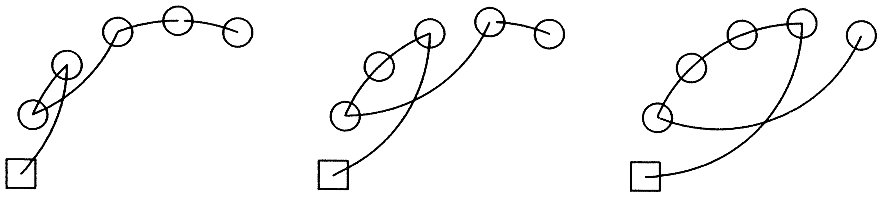


FIGURE 3. The search strategy for 2-exchanges.

In the following, we assume that i is fixed and consider the inner loop. The departure time at vertex k in the tour $(1,2,\ldots,n,n+1)$ will be denoted by D_k, for $k=1,\ldots,n+1$. The waiting and departure times at vertex k after reversal of the path $(i+1,\ldots,j)$ will be denoted by W_k^j and D_k^j, respectively, for $k>i$.

We define three *global variables*, which will be maintained throughout the inner loop. We suppose that the reversed path $(j-1,\ldots,i+1)$ has been considered. First, T is equal to the total travel time along this path:

$$T = \textstyle\sum_{k=i+1}^{j-2} d_{k,k+1}.$$

Secondly, W is equal to the total waiting time along the path after departing from vertex $j-1$:

$$W = \textstyle\sum_{k=i+1}^{j-2} W_k^{j-1}.$$

Thirdly, S is equal to the maximum forward shift in time of the departure time at vertex $j-1$ that would cause no time window violation along the path:

$$S = \min_{i+1 \leqslant k \leqslant j-1}[t_k - (D_{j-1}^{j-1} + \textstyle\sum_{l=k}^{j-2} d_{l,l+1})].$$

Expanding the reversed path $(j-1, \ldots, i+1)$ with the edge $\{j,j-1\}$ may change the arrival time at vertex $j-1$ and thereby all departure times along the path $(j-1, \ldots, i+1)$. We define a *local variable* Δ to denote the difference between the new arrival time and the old departure time at vertex $j-1$:

$$\Delta = D_j^j + d_{j,j-1} - D_{j-1}^{j-1}.$$

Δ can be computed in constant time, using $D_j^j = \max\{s_j, D_i + d_{ij}\}$ and $D_{j-1}^{j-1} = \max\{s_{j-1}, D_i + d_{i,j-1}\}$.

In order to prove that we can verify 2-optimality of the tour $(1,2, \ldots, n, n+1)$ in $O(n^2)$ time, we have to establish two facts: it is possible to update the values of the global variables in constant time, and the new values allow us to handle a single 2-exchange in constant time.

As to updating the global variables, we note that the definition of Δ covers two cases. In the case that $\Delta < 0$, the triangle inequality implies that the old arrival at $j-1$ cannot have been later than the new arrival. It follows that the old arrival and departure times did not coincide, so that the old departure occurred at the opening of the time window. But then we have that $-\Delta = W_{j-1}^j$, the new waiting time at $j-1$. In the case that $\Delta \geqslant 0$, we obviously have $\Delta = D_{j-1}^j - D_{j-1}^{j-1}$, the forward shift of the departure time at $j-1$. We conclude that the new values of the global variables are obtained by

$$T \leftarrow T + d_{j,j+1},$$

$$W \leftarrow \max\{W - \Delta, 0\},$$

$$S \leftarrow \min\{t_j - D_j^j, S - \Delta\}.$$

These updates require constant time.

As to handling a single 2-exchange, the conditions (1), requiring feasibility, and (2), stipulating profitability at vertex $j+1$, can be written as

$$(1)\ D_k^j \leqslant t_k \ \text{ for } k = i+1, \ldots, j,$$

$$(2)\ D_{j+1}^j < D_{j+1}.$$

The inequalities (1) are obviously equivalent to $S \geqslant 0$; see Savelsbergh [1988] for a formal proof. For inequality (2), we observe that the new departure time at $j+1$ satisfies

$$D_{j+1}^j = \max\{s_{j+1}, D_j^j + T + W + d_{i+1,j+1}\}.$$

We conclude that conditions (1) and (2) can be tested in constant time.

7. Parallel local search for the time-constrained TSP

We will now present a parallel algorithm for verifying 2-optimality of a time-constrained TSP tour. It requires $O(\log n)$ time and $O(n^2/\log n)$ processors, and thereby has the same resource requirements as in the unconstrained case.

Again, we consider the tour $(1,2,\ldots,n,n+1)$, which is assumed to be feasible. We start by computing all partial path lengths along the tour. This enables us to construct the tours that can be obtained by a 2-exchange. Our algorithm has five phases.

(1) We first compute all partial sums T_{ij} of travel times along the tour:

par $[1 \leqslant i \leqslant j \leqslant n+1]\ T_{ij} \leftarrow \sum_{k=i}^{j-1} d_{k,k+1}$.

By application of the partial sums algorithm from Section 5, this phase requires $O(\log n)$ time and $O(n^2/\log n)$ processors.

(2) We now investigate the effect of the time windows on the paths along the tour. For each pair of vertices $\{i,j\}$ with $i<j$, we define E_{ij} as the earliest possible departure time at vertex j when traveling along the tour from i to j, and E_{ji} as the earliest possible departure time at vertex i when traveling from j to i in the reverse direction along the tour. Note that $E_{1,n+1}$ is the arrival time at vertex 1. Further, let L_{ij} denote the latest possible departure time at vertex i such that the path from i to j remains feasible, and let L_{ji} denote the latest possible departure time at vertex j such that the path from j to i remains feasible. We then have:

par $[1 \leqslant i \leqslant j \leqslant n+1]\ E_{ij} \leftarrow \max_{i \leqslant k \leqslant j}(s_k + T_{kj})$;
par $[1 \leqslant i \leqslant j \leqslant n+1]\ E_{ji} \leftarrow \max_{i \leqslant k \leqslant j}(s_k + T_{ik})$;
par $[1 \leqslant i \leqslant j \leqslant n+1]\ L_{ij} \leftarrow \min_{i \leqslant k \leqslant j}(t_k - T_{ik})$;
par $[1 \leqslant i \leqslant j \leqslant n+1]\ L_{ji} \leftarrow \min_{i \leqslant k \leqslant j}(t_k - T_{kj})$.

Using the partial sums algorithm from Section 5 with addition replaced by maximization or minimization, we have the same time and processor requirements as in phase (1).

(3) Given the earliest and latest possible departure times relative to paths along the tour, we compute the earliest departure time $D_{ij}(k)$ at any vertex k and the earliest arrival time A_{ij} at the origin after the replacement of the edges $\{i,i+1\}$ and $\{j,j+1\}$ by the edges $\{i,j\}$ and $\{i+1,j+1\}$:

par $[1 \leqslant i < j \leqslant n]\ D_{ij}(j) \leftarrow \max\{E_{1i} + d_{ij}, s_j\}$;
par $[1 \leqslant i < j \leqslant n]\ D_{ij}(i+1) \leftarrow \max\{D_{ij}(j) + T_{i+1,j}, E_{j,i+1}\}$;
par $[1 \leqslant i < j \leqslant n]\ D_{ij}(j+1) \leftarrow \max\{D_{ij}(i+1) + d_{i+1,j+1}, s_{j+1}\}$;
par $[1 \leqslant i < j \leqslant n]\ A_{ij} \leftarrow \max\{D_{ij}(j+1) + T_{j+1,n+1}, E_{j+1,n+1}\}$.

For this phase we need $O(1)$ time and $O(n^2)$ processors, or $O(\log n)$ time and $O(n^2/\log n)$ processors.

(4) We then test for the feasibility of the tours obtained by 2-exchanges, using boolean variables F_{ij}:

par $[1 \leqslant i < j \leqslant n]\ F_{ij} \leftarrow (D_{ij}(j) \leqslant L_{j,i+1})\ \&\ (D_{ij}(j+1) \leqslant L_{j+1,n+1})$.

The first condition tests for feasibility at the vertices $i+1,\ldots,j$ and the second one at the vertices $j+1, \ldots, n+1$. As in the previous phase, we need $O(1)$ time and $O(n^2)$ processors, or $O(\log n)$ time and $O(n^2/\log n)$ processors.

(5) Finally, we decide whether or not the given tour is 2-optimal in the same way as in the case without time windows:

```
$A_{\min} \leftarrow \min\{A_{ij} \mid F_{ij}, 1 \leqslant i < j \leqslant n\}$;
if $E_{1,n+1} \leqslant A_{\min}$
then $(1,2, \ldots, n, n+1)$ is a 2-optimal tour
else let $i^*$ and $j^*$ be such that $F_{i^*j^*}$ & $A_{i^*j^*} = A_{\min}$,
     $(1,\ldots,i^*,j^*,j^*-1,\ldots,i^*+1,j^*+1,\ldots,n+1)$ is a better feasible tour.
```

For this last phase, the same time and processor bounds as before suffice. So, we end up with an algorithm that runs in $O(\log n)$ time using $O(n^2/\log n)$ processors, which is the same as in the case without time windows.

For each fixed $k>2$, we can derive a logarithmic-time algorithm along similar lines. One has to take into account that, given k edges, several k-exchanges are possible. Further, the influence of a k-exchange on a tour is more complex. However, it is not hard to see that the running time remains $O(\log n)$ using $O(n^k/\log n)$ processors, which is optimal with respect to the number $\Theta(n^k)$ of k-exchanges.

References

J.M. Anthonisse, J.K. Lenstra, M.W.P. Savelsbergh (1989). Behind the screen: DSS from an OR point of view. *Decision Support Syst. 4*, 413-419.

M. Bellmore, G.L. Nemhauser (1968). The traveling salesman problem: a survey. *Oper. Res. 16*, 538-558.

A.K. Chandra, D.C. Kozen, L.J. Stockmeyer (1981). Alternation. *J. Assoc. Comput. Mach. 28*, 114-133.

S.A. Cook (1981). Towards a complexity theory of synchronous parallel computation. *Enseign. Math. (2) 27*, 99-124.

E. Dekel, S. Sahni (1983). Binary trees and parallel scheduling algorithms. *IEEE Trans. Comput. C-32*, 307-315.

S. Fortune, J. Wyllie (1978). Parallelism in random access machines. *Proc. 10th Annual ACM Symp. Theory of Computing*, 114-118.

M.R. Garey, D.S. Johnson (1979). *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, San Francisco.

L.M. Goldschlager (1982). A universal connection pattern for parallel computers. *J. Assoc. Comput. Mach. 29*, 1073-1086.

D.S. Johnson (1983). The NP-completeness column: an ongoing guide; seventh edition. *J. Algorithms 4*, 189-203.

D.S. Johnson, C.H. Papadimitriou, M. Yannakakis (1988). How easy is local search? *J. Comput. Syst. Sci. 37*, 79-100.

R. Jonker (1986). *Traveling Salesman and Assignment Algorithms: Design and Implementation*, Ph.D. thesis, University of Amsterdam.

R. Jonker, G. de Leve, J.A. van der Velde, A. Volgenant (1980). Rounding [Bounding] symmetric traveling salesman problems with an asymmetric assignment problem. *Oper. Res. 28*, 623-627.

G.A.P. KINDERVATER (1989). *Exercises in Parallel Combinatorial Computing*, Ph.D. thesis, Centre for Mathematics and Computer Science, Amsterdam.

E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN, D.B. SHMOYS (eds.) (1985). *The Traveling Salesman Problem: a Guided Tour of Combinatorial Optimization*, Wiley, Chichester.

S. LIN (1965). Computer solutions of the traveling salesman problem. *Bell System Tech. J. 44*, 2245-2269.

M.W.P. SAVELSBERGH (1988). *Computer Aided Routing*, Ph.D. thesis, Centre for Mathematics and Computer Science, Amsterdam.

R. TIJDEMAN (1968). *Het Handelsreizigersprobleem, een Literatuuronderzoek*, Report S385, Mathematical Centre, Amsterdam.

P. VAN EMDE BOAS (1985). The second machine class: models of parallelism. J. VAN LEEUWEN, J.K. LENSTRA (eds.). *Parallel Computers and Computations*, CWI Syllabus 9, Centre for Mathematics and Computer Science, Amsterdam, 133-161.

K.M. VAN HEE (1989). Private communication.

A. VOLGENANT (1987). *Contributions to the Solution of the Traveling Salesman Problem and Related Problems*, Ph.D. thesis, University of Amsterdam.

Constructing valid inequalities for combinatorial optimization problems

A.W.J. Kolen
Quantitative Economics
Limburg University, Maastricht

Two construction methods for valid inequalities for combinatorial optimization problems will be presented. The first method is based on a description of the problem by a (mixed) integer programming formulation using auxiliary variables in addition to the original variables and constructs the projection of the linear programming relaxation of this formulation on the space of the original variables. In the second method we will derive an explicit description of a valid inequality simply by assuming properties of the coefficients of a facet inducing valid inequality for the convex hull of feasible solutions. Both construction methods will be demonstrated on the economic lot-sizing problem.

1. Introduction

Cutting plane algorithms have been used very successfully for solving combinatorial optimization problems. In a cutting plane algorithm we start with an initial (mixed) integer programming formulation of the combinatorial optimization problem. If the optimal solution of the linear programming (LP) relaxation is a feasible solution to the combinatorial problem, then we have found an optimal solution. Otherwise we try to find valid inequalities for the combinatorial problem (i.e., inequalities which are satisfied by all feasible solutions) which are violated by the optimal LP-solution. These inequalities are added to the LP-formulation which is then reoptimized. This continues until either we find a feasible solution to the combinatorial problem or we can't find any more violated inequalities among the known valid inequalities. In the latter case we frequently end with a very good bound on the optimal solution value. Together with a good feasible solution this can be used to fathom nodes in a branch-and-bound procedure. So in order to apply a cutting plane algorithm we first of all need valid inequalities and secondly we need to be able to solve the separation problem, i.e., given a solution we must be able to find valid inequalities which are violated by the given solution. In this paper we discuss two methods for constructing valid inequalities which can sometimes be used to find classes of valid inequalities for combinatorial optimization problems.

The first method which we call the *projection method* is based on a description of the problem as a (mixed) integer programming formulation using auxiliary variables in addition to the original variables. The LP-relaxation of the constraint set of this formulation is then projected on the space of the original variables. For more details we refer to Section 2.

The second method which we call the *polyhedral method* is based on properties of inequalities needed in a minimal description of a polyhedron. We will see that by assuming certain properties about the coefficients of these inequalities we will be able to obtain an explicit description of these inequalities. The theoretical foundation for the polyhedral method as well as a more detailed description will be given in Section 3.

Both construction methods will be demonstrated on the *economic lot-sizing problem*. In the economic lot-sizing problem we have to decide when and how much to produce in order to meet the known positive demands for a given product over a finite discrete planning horizon so as to minimize total cost consisting of production and inventory costs. The inventory costs are linear in the number of items in stock at the end of each time period. The production costs decompose into two parts; a fixed set-up cost is incurred whenever we set-up production in addition to a cost linear in the number of items produced. The economic lot-sizing problem is the problem of minimizing

$$\sum_{i=1}^{n}(p_ix_i+f_iy_i+h_is_i)$$

subject to

$$x_i+s_{i-1}-s_i = d_i, \quad i = 1,2,...,n \tag{1.1}$$

$$s_0 = s_n = 0 \tag{1.2}$$

$$x_i \leqslant d_{in}y_i, \quad i = 1,2,...,n \tag{1.3}$$

$$x_i,s_i \geqslant 0, \quad i = 1,2,...,n \tag{1.4}$$

$$y_i \in \{0,1\}, \quad i = 1,2,...,n, \tag{1.5}$$

where n is the number of time periods; f_i is the set-up cost and p_i is the production cost per item in period i; h_i is the inventory cost per item in stock at the end of period i; d_i is the positive demand in period i, $i = 1,2,...,n$. With d_{pk} we denote the total demand in periods $p,p+1,...,k$, i.e., $d_{pk} = \Sigma_{i=p}^{k}d_i$. The variable y_i indicates whether we set up production in period i ($y_i=1$) or not ($y_i=0$), the variable x_i is the amount produced in period i and s_i is the inventory level at the end of period i, $i = 1,2,...,n$. Since backlogging is not allowed we will always produce for future periods. Because the inventory at the end of period n is zero we will never produce more than d_{in} in period i, $i = 1,2,...,n$. Constraints (1.3) guarantee that a set-up cost is incurred whenever we produce in a period. Note that we may decide to set up production without starting production. Constraints (1.1) are balancing constraints which tell us that the total amount entering period i (production in period i and inventory at the end of period $i-1$) equals the amount leaving period i (demand in period i and inventory at the end of period i). We can eliminate the inventory variables from the formulation, namely $s_i = \Sigma_{t=1}^{i}x_t-d_{1i}$. If we define $c_i = p_i+\Sigma_{t=i}^{n}h_t$, $i = 1,2,...,n$, then we obtain the following formulation in what we call the original variables of the economic lot-sizing problem:

minimize

$$\sum_{i=1}^{n}(c_ix_i+f_iy_i)$$

subject to

$$\sum_{i=1}^{n}x_i = d_{in} \tag{1.6}$$

$$\sum_{i=1}^{t}x_i \geqslant d_{1t}, \quad t = 1,2,...,n-1 \tag{1.7}$$

$$x_i \leqslant d_{in}y_i, \quad i = 1,2,...,n \tag{1.8}$$

$$x_i \geqslant 0, \quad i = 1,2,...,n \tag{1.9}$$

$$y_i \in \{0,1\}, \quad i = 1,2,...,n \tag{1.10}$$

It can be shown that optimal solutions to this problem can be characterized by $I,J \subseteq \{1,2,...,n\}$, $1 \in I \subseteq J$, where I is the set of production periods and J is the set of set-up periods. The production in a period equals the total demand from this period until the next production period (or period n in case of the last production period). This means that production is only started when inventory is zero (see Van Hoesel, Kolen & Wagelmans (1989)).

2. The projection method

Let $S \subseteq \mathbf{R}_+^n$ be a finite set for which we want to find a linear description (i.e., a description in terms of the solution set of a number of linear inequalities) of the convex hull of S, where the convex hull is denoted by conv(S).

Assume the existence of a polyhedron $P \subseteq \mathbf{R}_+^{n+m}$ such that for every $x \in S$ there exists a $z \in \mathbf{R}_+^m$ such that $(x,z) \in P$. If we define the projection, $proj_x(P)$, of the polyhedron P on the space of the x-variables by

$$proj_x(P) = \{x \in \mathbf{R}_+^n \mid (x,z) \in P \text{ for some } z \geqslant 0\},$$

then another way of stating our assumption is

$$conv(S) \subseteq proj_x(P). \tag{2.1}$$

If on the other hand we can prove that for every $c \in \mathbf{R}^n$ the optimization problem

$$\text{minimize } cx \tag{2.2}$$
$$\text{subject to } (x,z) \in P,$$

has an optimal solution (x^*,z^*) with $x^* \in S$, then since (2.2) is equivalent to the optimization problem

$$\text{minimize } cx \tag{2.3}$$
$$\text{subject to } x \in proj_x(P),$$

we have proved that every extreme point of $\text{proj}_x(P)$ belongs to S and hence

$$\text{proj}_x(P) \subseteq \text{conv}(S). \tag{2.4}$$

It follows from (2.1) and (2.4) that in this case

$$\text{conv}(S) = \text{proj}_x(P). \tag{2.5}$$

Each inequality in a linear description of $\text{proj}_x(P)$ will provide a valid inequality for conv(S) if (2.1) holds. If in addition (2.5) holds, then the linear descriptions of conv(S) and $\text{proj}_x(P)$ are identical.

To obtain a linear description of $\text{proj}_x(P)$, we use the result that the cone $\{u \mid B^T u \geqslant 0, u \geqslant 0\}$ defined by some integer matrix B has a finite set of generators $\{u^j \mid j = 1,2,...,J\}$, i.e., every element in the cone can be written as a nonnegative combination of these generators. The following theorem shows how to obtain a linear description of $\text{proj}_x(P)$.

Theorem 1. *If* $P = \{(x,z) \in \mathbf{R}_+^{n+m} \mid Ax + Bz \leqslant d\}$, *then* $\text{proj}_x(P) = \{x \in \mathbf{R}_+^n \mid (u^j)^T Ax \leqslant (u^j)^T d,\ j = 1,2,...,J\}$, *where* $\{u^j \mid j = 1,2,...,J\}$ *is a finite set of generators of* $\{u \mid B^T u \geqslant 0,\ u \geqslant 0\}$.

Proof. According to Farkas' Lemma statements (i) and (ii) are equivalent:
(i) there exists $z \geqslant 0$ such that $Bz \leqslant d - Ax$,
(ii) $B^T u \geqslant 0$ implies $(d - Ax)^T u \geqslant 0$, for all $u \geqslant 0$.
Using the set of generators $\{u^j \mid j = 1,2,...,J\}$ statement (ii) is equivalent to:
(iii) $(d - Ax)^T u^j \geqslant 0$ for all $j = 1,2,...,J$.
Since (i) is equivalent to $x \in \text{proj}_x(P)$ the result follows. □

Corollary 1. *If* $P = \{(x,z) \in \mathbf{R}_+^{n+m} \mid Ax + Bz \leqslant d, z \geqslant 0, x \in Q\}$, *where* Q *is an arbitrary subset of* $\mathbf{R}^n$, *then* $\text{proj}_x(P) = \{x \in \mathbf{R}_+^n \mid (u^j)^T Ax \leqslant (u^j)^T d,\ j = 1,2,...,J,\ x \in Q\}$.

Proof. Similar to the proof of Theorem 1. □

Even if we cannot completely characterize a set of generators of the cone $\{u \mid B^T u \geqslant 0, u \geqslant 0\}$, every element u of the cone will give us a valid inequality $(d - Ax)^T u \geqslant 0$ for $\text{proj}_x(P)$. One of the first problems on which the projection technique was used is the perfect matchable subgraph problem on a bipartite graph (Balas & Pulleyblank (1983)).

We will describe two situations which frequently arise and in which the projection method can be applied, namely disaggregation and dynamic programming.

2.1. Disaggregation

When using disaggregation we are looking at the problem at a more detailed level than is necessary to describe it. The original variables are decomposed into a sum of new auxiliary variables. At the cost of a larger model with more variables disaggregation allows us to obtain a better formulation in the sense that the LP-relaxation yields a better bound for the optimal value than the LP-relaxation of the original formulation.

In the economic lot-sizing problem we know that the production in period i will be used to satisfy demand in future periods. Therefore a natural disaggregation is obtained by introducing auxiliary variables q_{it} indicating the production in period i used to satisfy demand in period $t, t \geqslant i$. We have $x_i = \Sigma_{t=i}^{n} q_{it}$. The new formulation of the economic lot-sizing problem is to minimize

$$\sum_{i=1}^{n} (c_i x_i + f_i y_i)$$

subject to

$$\sum_{i=1}^{t} q_{it} = d_t, \qquad t = 1,2,...,n \tag{2.6}$$

$$\sum_{t=i}^{n} q_{it} = x_i, \qquad i = 1,2,...,n \tag{2.7}$$

$$q_{it} \leqslant d_t y_i, \qquad i = 1,2,...,n, \; t = i,...,n \tag{2.8}$$

$$0 \leqslant y_i \leqslant 1, \; x_i \geqslant 0, \; q_{it} \geqslant 0, \; i = 1,2,...,n, \; t = i,...,n \tag{2.9}$$

$$y_i \in \{0,1\}, \qquad i = 1,2,...,n. \tag{2.10}$$

Constraints (2.6) say that the demand in period t must be produced in periods $1,2,...,t$, i.e., backlogging is not allowd. Constraints (2.8) say that we can only produce in period i to satisfy the demand in period t whenever a set-up has occurred in period i, and that the quantity produced q_{it} will never exceed the total demand d_t. It is easy to show that if (x,y,q) is a solution of (2.6)-(2.10), then (x,y) is a solution of (1.6)-(1.10). Conversely, given (x,y) satisfying (1.6)-(1.10) it is easy to find a vector q such that (x,y,q) satisfies (2.6)-(2.10). (For example, q can be constructed by always assuming that demand in a period is satisfied from the oldest production period for which still some production is left over.)

It is well known that the LP-relaxation of the above problem given by (2.6)-(2.9) always has an integer optimal solution. For a constructive proof see Wagelmans, Kolen & Van Hoesel (1989). Therefore if S is the set of solutions to the economic lot-sizing problem characterized by the subsets $I,J \subseteq \{1,2,...,n\}$ as described at the end of Section 1, then

$$\text{conv}(S) = \text{proj}_x(P_1),$$

where P_1 is the polyhedron defined by (2.6)-(2.9).

The projection of P_1 still is very difficult to describe. However by introducing additional auxiliary variables we obtain a polyhedron P_2 for which the projection is easy to describe and since $\text{proj}_{x,y}(P_1) = \text{proj}_{x,y}(P_2)$ will serve to find a linear description of conv(S). The polyhedron P_2 is given by $P_2 = \{(x,y,q,s) | (x,y,q,s)$ satisfy (2.7), (2.8), (2.9), (2.11) and $s_i \geqslant 0$, $i = 1,2,...,n-1\}$, where (2.11) is given by

$$\sum_{i=1}^{t} q_{it} + s_{t-1} - s_t = d_t, \quad t = 1,2,...,n, \tag{2.11}$$

with $s_0 = s_n = 0$. The quantity q_{it} can be interpreted as the amount produced in period i for periods $t, t+1,..,n$, s_t is the inventory at the end of period t. We will first prove that the projections of P_1 and P_2 are identical.

THEOREM 2. $\text{proj}_{x,y}(P_1) = \text{proj}_{x,y}(P_2)$.

PROOF. Clearly $\text{proj}_{x,y}(P_1) \subseteq \text{proj}_{x,y}(P_2)$; take $s = 0$. The proof that $\text{proj}_{x,y}(P_2) \subseteq \text{proj}_{x,y}(P_1)$ follows an idea of Pochet (1987). Given $(x,y) \in \text{proj}_{x,y}(P_2)$ let q and s be such that $(x,y,q,s) \in P_2$ and $\Sigma_{i=1}^{n-1} s_i$ is minimal. We shall prove that $\Sigma_{i=1}^{n-1} s_i = 0$, i.e., $(x,y,q) \in P_1$ and hence $(x,y) \in proj_{x,y}(P_1)$.

Assume $\Sigma_{i=1}^{n-1} s_i > 0$. Then there exist $t_1, t_2, 1 \leqslant t_1 < t_2 \leqslant n$ such that $s_{t_1-1} = 0$, $s_{t_2} = 0$ and $s_p > 0$, $p = t_1,...,t_2-1$.

Claim 1. If $q_{i,t_1} > 0$, then $q_{i,t_2} = d_{t_2} y_i$.

PROOF. If $q_{i,t_1} > 0$ and $q_{i,t_2} < d_{t_2} y_i$, then we decrease q_{i,t_1} and s_p, $p = t_1,...,t_2-1$ with ϵ and increase q_{i,t_2} with ϵ, where $\epsilon > 0$ is sufficiently small. Then $(x,y,\hat{q},\hat{s}) \in P_2$, where $(\hat{q},\hat{s})$ is the new vector defined above. Since $\Sigma_{i=1}^{n-1} \hat{s}_i < \Sigma_{i=1}^{n-1} s_i$ contradicts the minimality of $\Sigma_{i=1}^{n-1} s_i$ the claim follows. □

Claim 2. $\Sigma_{i=1}^{t_1} (q_{i,t_1}/d_{t_1}) \leqslant \Sigma_{i=1}^{t_2} (q_{i,t_2}/d_{t_2})$.

PROOF. Since $q_{i,t_1} \leqslant d_{t_1} y_i$ it follows from Claim 1 that whenever $q_{i,t_1} > 0$ then $(q_{i,t_1}/d_{t_1}) \leqslant y_i = (q_{i,t_2}/d_{t_2})$. □

Now $s_{t_1-1} = 0$, $s_{t_1} > 0$ and $\Sigma_{i=1}^{t_1} q_{i,t_1} + s_{t_1-1} - s_{t_1} = d_{t_1}$ imply $\Sigma_{i=1}^{t_1} q_{i,t_1} > d_{t_1}$, i.e.,

$$\sum_{i=1}^{t_1} (q_{i,t_1}/d_{t_1}) > 1. \tag{2.12}$$

Furthermore $s_{t_2} = 0$, $s_{t_2-1} > 0$ and $\Sigma_{i=1}^{t_2} q_{i,t_2} + s_{t_2-1} - s_{t_2} = d_{t_2}$ imply $\Sigma_{i=1}^{t_2} q_{i,t_2} < d_{t_2}$, i.e.,

$$\sum_{i=1}^{t_2} (q_{i,t_2}/d_{t_2}) < 1. \tag{2.13}$$

From (2.12) and (2.13) it follows that $\Sigma_{i=1}^{t_2} (q_{i,t_2}/d_{t_2}) < \Sigma_{i=1}^{t_1} (q_{i,t_1}/d_{t_1})$ contradicting Claim 2. We conclude that $\Sigma_{i=1}^{n-1} s_i = 0$. □

Let β_i be the dual variable associated with constraint i of (2.7), π_{it} the variable associated with constraint i,t of (2.8), and α_t the variable associated with constraint t of (2.11). Then the cone corresponding to the projection of P_2 is described by

$$\alpha_t+\beta_i+\pi_{it}\geqslant 0, \qquad i = 1,2,\dots,n,\ t = i,\dots,n \tag{2.14}$$

$$-\alpha_t+\alpha_{t+1}\geqslant 0, \qquad t = 1,\dots,n-1 \tag{2.15}$$

$$\pi_{it}\geqslant 0, \qquad i = 1,2,\dots,n,\ t = i,\dots,n. \tag{2.16}$$

We will prove in Theorem 3 that a finite set of generators for this cone is given by

$$(\alpha_t = 1,\ t = 1,\dots,n,\ \beta_i = -1,\ i = 1,\dots,n;\ \text{all other values zero}) \tag{2.17}$$

$$(\pi_{it} = 1;\ \text{all other values zero}),\ i = 1,\dots,n,\ t = i,\dots,n \tag{2.18}$$

$$(\beta_i = 1;\ \text{all other values zero}),\ i = 1,\dots,n \tag{2.19}$$

$$(\beta_i = -1;\ \pi_{it} = 1,\ t = i,\dots,n;\ \text{all other values zero}),\ i = 1,\dots,n \tag{2.20}$$

$$(\alpha_t = -1,\ t = 1,\dots,l,\ \beta_i = 1,\ i\in S,\ \pi_{it} = 1,\ i\in\{1,\dots,l\}\setminus S,\ t = i,\dots,l;$$
$$\text{all other values zero}),\ l = 1,\dots,n,\ S\subseteq\{1,\dots,l\}. \tag{2.21}$$

As an element (α,β,π) of the cone leads to the inequality $\Sigma_{t=1}^{n}\alpha_t d_t + \Sigma_{i=1}^{n}\beta_i x_i + \Sigma_{i=1}^{n}\Sigma_{t=i}^{n}\pi_{it}d_t y_i\geqslant 0$ the generators (2.17)-(2.21) lead to the linear inequalities (2.22)-(2.26) which together with $0\leqslant y_i\leqslant 1$, $x_i\geqslant 0$, $i = 1,2,\dots,n$ (see Corollary 1) form a linear description of the convex hull of feasible solutions for the economic lot-sizing problem.

$$\sum_{i=1}^{n} x_i\leqslant d_{in} \tag{2.22}$$

$$y_i\geqslant 0 \tag{2.23}$$

$$x_i\geqslant 0 \tag{2.24}$$

$$x_i\leqslant d_{in}y_i \tag{2.25}$$

$$\sum_{i\in S}x_i + \sum_{i\in\{1,\dots,l\}\setminus S} d_{il}y_i\geqslant d_{1l},\ l = 1,\dots,n,\ S\subseteq\{1,\dots,l\} \tag{2.26}$$

Constraints (2.26) are the so called (S,l)-inequalities defined by Barány, Van Roy & Wolsey (1984), who were also the first to prove that the linear description is complete. Although the description is complete it is not a minimal description. We leave it to the reader to prove that for example constraints (2.25) are redundant.

THEOREM 3. *The solutions defined by (2.17)-(2.21) form a set of generators for the cone defined by (2.14)-(2.16).*

PROOF. We will prove that any point in the cone can be written as a nonnegative combination of the points defined by (2.17)-(2.21). Let (α,β,π) be a point in the cone.

Since the point defined by (2.17) satisfies all inequalities with equality we can subtract a positive multiple of it from (α,β,π) and still have a point of the cone. Therefore we may assume that $\alpha\leqslant 0$.

If $\beta_i<0$ for some i, then it follows from $\alpha_t+\beta_i+\pi_{it}\geqslant 0$ that $\pi_{it}\geqslant-\beta_i$ (remember that $\alpha_t\leqslant 0$) for all t, $t\geqslant i$. Hence we can subtract the $-\beta_i$ multiple of point i of (2.20) from (α,β,π) and still have a point of the cone. Therefore we may assume that $\beta\geqslant 0$.

Assume $\alpha\neq 0$. Let l be the largest index t for which $\alpha_t<0$. Then from $\alpha_t\leqslant\alpha_{t+1}$ it follows that $\alpha_1\leqslant\cdots\leqslant\alpha_l<0$ and $\alpha_{l+1}=\cdots=\alpha_n=0$. Define $S=\{i\in\{1,\dots,l\}\mid\beta_i>0\}$ and $\epsilon=\min\{\min_{i\in S}\{\beta_i\},-\alpha_l\}$. We claim that if we subtract the ϵ multiple of the solution defined by l and S of (2.21) from (α,β,π), then we still have a point of the cone. In order to prove this claim we need to show that the new point $(\hat{\alpha},\hat{\beta},\hat{\pi})$ satisfies the cone inequalities. As to (2.14):

If $i\in S, t\leqslant l$, then $\hat{\alpha}_t+\hat{\beta}_i+\hat{\pi}_{it}=(\alpha_t+\epsilon)+(\beta_i-\epsilon)+\pi_{it)}\geqslant 0$.
If $i\notin S, t\leqslant l$, then $\hat{\alpha}_t+\hat{\beta}_i+\hat{\pi}_{it}=(\alpha_t+\epsilon)+\beta_i+(\pi_{it}-\epsilon)\geqslant 0$.
If $i\in S, t>l$, then $\hat{\alpha}_t+\hat{\beta}_i+\hat{\pi}_{it}=\alpha_t+(\beta_i-\epsilon)+\pi_{it}\geqslant 0$ since $\alpha_t=0$, $\pi_{it}\geqslant 0$ and $\beta_i\geqslant\epsilon$.
If $i\notin S, t>l$, then $\hat{\alpha}_t+\hat{\beta}_i+\hat{\pi}_{it}=\alpha_t+\beta_i+\pi_{it}\geqslant 0$.

As to (2.15), it is clear that $\hat{\alpha}_t\leqslant\hat{\alpha}_{t+1}, t=1,\dots,n-1$ since $\hat{\alpha}_l=\alpha_l+\epsilon\leqslant 0=\hat{\alpha}_{l+1}$. As to (2.16):

If $i\notin S$, $t\leqslant l$, then $\pi_{it}\geqslant\epsilon$ since $\alpha_t+\beta_i+\pi_{it}=\alpha_t+\pi_{it}\geqslant 0$ and $-\alpha_t\geqslant\epsilon$. Hence $\hat{\pi}_{it}=\pi_{it}-\epsilon\geqslant 0$; in all other cases $\hat{\pi}_{it}=\pi_{it}$.

We conclude that $(\hat{\alpha},\hat{\beta},\hat{\pi})$ satisfies the cone inequalities. Note that $\alpha_t=0\Rightarrow\hat{\alpha}_t=0$ and $\beta_i=0\Rightarrow\hat{\beta}_i=0$. Furthermore there is at least one index such that $\alpha_p<0$ and $\hat{\alpha}_p=0$, or $\beta_p>0$ and $\hat{\beta}_p=0$. By repeating the argument we arrive at a solution (α,β,π) for which $\alpha=0$.

It is trivial to see that a solution (α,β,π) for which $\alpha=0$, $\beta\geqslant 0$, $\pi\geqslant 0$ can be written as a nonnegative combination of points in (2.18) and (2.19). □

2.2. Dynamic programming

Let $S\subseteq\mathbf{R}^n_+$, a finite set, be the set of feasible solutions to a problem for which a dynamic programming formulation exists. When we consider the state space of the dynamic programming formulation, then we can construct a digraph $G=(\mathcal{V},\mathcal{A})$ with the property that there is a one-to-one correspondence between a feasible solution and a path between two given vertices $s,t\in\mathcal{V}$. The vertices of the graph correspond to states and arcs to feasible state transitions. Let us assume that there exists an $n\times|\mathcal{A}|$ integer matrix B such that x is a feasible solution if and only if there exists a path from s to t with (0,1) incidence vector z such that $x=Bz$. Define the polyhedron P by $P=\{(x,z)\in\mathbf{R}^{n+|\mathcal{A}|}_+\mid Az=f,Bz=x\}$, where A is the vertex-arc incidence matrix of G and f is a vector with zero components except the components corresponding to s and t which are 1 and -1 respectively.

THEOREM 4. $\mathrm{conv}(S) = \mathrm{Proj}_x(P)$.

PROOF. Let $x \in \mathrm{conv}(S)$. Then x can be written as a convex combination of elements in S. Take z to be the identical convex combination of incidence vectors of paths corresponding to these elements in S. Then $Az = f$, $Bz = x$ and hence $x \in \mathrm{Proj}_x(P)$.

Let $x \in \mathrm{Proj}_x(P)$ and $z \geqslant 0$ such that $Az = f$, $Bz = x$. Since every $z \geqslant 0$ with $Az = f$ can be written as a convex combination of incidence vectors of paths from s to t we can write x as the identical convex combination of elements in S corresponding to these incidence vectors. □

Let C be a finite set of generators of the cone $\{(u,v) | A^T u + B^T v \geqslant 0\}$. Then applying Theorem 1 we find that

$$\mathrm{conv}(S) = \{x \in \mathbf{R}^n_+ \,|\, f^T u + x^T v \geqslant 0,\ (u,v) \in C\}. \tag{2.27}$$

Let us now apply this to the economic lot-sizing problem. As mentioned in Section 1 an element of S is characterized by two subsets $I,J \subseteq \{1,2,...,n\}$ with $1 \in I \subseteq J$. Here I corresponds to the production periods and J corresponds to set-up periods. If i and j are two consecutive production periods, then $x_i = d_{i,j-1}$.

We shall now construct the graph $G = (\mathcal{V},\mathcal{A})$ such that there is a one-to-one correspondence between solutions of the economic lot-sizing problem, and paths between two given vertices s and t of $\mathcal{V}$. The vertex set $\mathcal{V}$ is defined by

$$\mathcal{V} = \{(i,j) | i,j = 1,2,...,n+1,\ i \leqslant j\}.$$

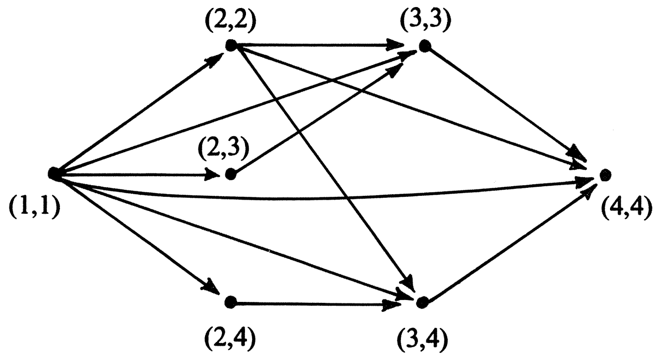
The arc set $\mathcal{A}$ is defined by

$$\mathcal{A} = \{((i,i),(k,j)) | i,k,j = 1,2,...,n+1,\ i < k \leqslant j\} \cup$$
$$\{((k,j),(m,j)) | k,m,j = 1,2,...,n+1,\ k < m \leqslant j\}.$$

The two vertices $s,t \in \mathcal{V}$ for which the paths are considered are given by $s = (1,1)$ and $t = (n+1,n+1)$.

Let us now convince ourselves that there is a one-to-one correspondence between a path from $(1,1)$ to $(n+1,n+1)$ and a feasible solution of the economic lot-sizing problem characterized by subsets $I,J \subseteq \{1,2,...,n\}$. If the path uses the arc between (i,i) and (k,j), then the interpretation is that we start production in period i and $x_i = d_{i,j-1}$. If $k < j$, then there is a set-up in period k without production. Note that vertex (k,j) with $k < j$ is only connected by an arc to vertices (m,j), $k < m \leqslant j$. If $m < j$, then there is another set-up in period m without production. Eventually the path starting with the arc $((i,i),(k,j))$ will arrive in vertex (j,j). From there the above argument can be repeated starting with (j,j). We have shown that a path from $(1,1)$ to $(n+1,n+1)$ can be interpreted as a feasible solution for the economic lot-sizing problem. It will be clear from the above argument that to every feasible solution there also corresponds a path from $(1,1)$ to $(n+1,n+1)$.

The graph for $n = 3$ is given below.



The path (1,1), (2,3), (3,3), (4,4) corresponds with

$$x_1 = d_{12}, x_3 = d_3, y_1 = y_2 = y_3 = 1.$$

The solution $x_1 = d_{13}, y_1 = y_3 = 1$ corresponds to the path

(1,1),(3,4),(4,4).

From the above argument it is also clear how to express x_i and y_i in terms of the incidence vectors of paths from (1,1) to $(n+1,n+1)$. Let $z(a)$ be one if arc $a \in \mathcal{Q}$ is used in the path, and zero otherwise. We start production in period i only if vertex (i,i) is on the path. The amount we produce is determined by the next vertex on the path. If this is vertex (k,j), then $x_i = d_{i,j-1}$. A path visits (i,i) if and only if one of the outgoing arcs of (i,i) is used. Furthermore since (i,i) is visited at most once we have the relation

$$x_i = \sum_{(k,j):i<k\leqslant j} d_{i,j-1} z((i,i),(k,j)),\ i = 1,2,...,n. \tag{2.28}$$

Similarly we set up production in period i if vertex (i,i) or a vertex (i,j), $i<j$ is on the path. Also note that at most one vertex with first component i can be on a path from (1,1) to $(n+1,n+1)$. A vertex is on the path if and only if one of the outgoing arcs is on the path. Hence we have the relation

$$y_i = \sum_{(k,j):i<k\leqslant j} z((i,i),(k,j)) + \tag{2.29}$$
$$\sum_{(k,j):i<k\leqslant j} z((i,j),(k,j)),\ i = 1,2,...,n.$$

In order to describe the cone related to this problem let the variable λ_{ij} correspond to the constraint of $Az = f$ related to vertex (i,j), the variable α_i correspond to constraint i of (2.28), and the variable β_i correspond to constraint i of (2.29). In the cone (see definition of C in (2.27)) we have a constraint for each arc in the graph. The constraint are given by

$$\lambda_{ii} - \lambda_{kj} + d_{i,j-1}\alpha_i + \beta_i \geqslant 0, \text{ for all } i,k,j,\ i<k\leqslant j, \tag{2.30}$$

$$\lambda_{ij} - \lambda_{kj} + \beta_i \geq 0, \qquad \text{for all } i,k,j,\ i<k\leq j, \tag{2.31}$$

where (2.30) corresponds to the arc $((i,i),(k,j))$ and (2.31) to the arc $((i,j),(k,j))$.

According to (2.27) every element (λ,α,β) of the cone will generate the inequality

$$\sum_{i=1}^{n} \alpha_i x_i + \sum_{i=1}^{n} \beta_i y_i \geq \lambda_{n+1,n+1} - \lambda_{11} \tag{2.32}$$

for the economic lot-sizing problem.

Although a finite set of generators for this cone has been conjectured no proved finite set of generators is known. It would be interesting to find a proof because the dynamic programming technique can also be applied to extensions of the economic lot-sizing problem such as the problem with start-up cost (Wolsey (1988)). For this problem a complete linear description and dual algorithm is described by Van Hoesel, Kolen, Wagelmans and Wolsey (1989). We finish this section by giving the element of the cone leading to an (S,l)-inequality:

$$\alpha_i = 1, \quad i \in S$$

$$\beta_i = d_{il}, \quad i \in \{1,...,l\} \setminus S$$

$$\lambda_{kj} = -d_{jl}, \quad 1 \leq k \leq j \leq l$$

all other values are zero.

3. The polyhedral method

In order to understand the polyhedral method we first need some basic results about polyhedra. For more details and proofs we refer to Nemhauser & Wolsey (1988) or Schrijver (1986).

The dimension of a polyhedron P is k if the maximum number of affinely independent points in P is $k+1$. Let $ax \leq \alpha_0$ be a valid inequality for P, i.e., $ax \leq \alpha_0$ for all $x \in P$. Then $F = \{x \in P \mid ax = \alpha_0\}$ is called a face of P induced by $ax \leq \alpha_0$. F is a proper face if $F \neq \phi$ and $F \neq P$. If $dim(F) = dim(P) - 1$, then the face F is called a facet. As we will see facets play an important role in a minimal description of the polyhedron, i.e., minimal with respect to the number of linear inequalities. Given a description of a polyhedron P as the solution set of a number of linear inequalities an inequality is an *implicit equation* if it is satisfied with equality for all points in P. Theorem 4 describes how to find a minimal description of a polyhedron P.

Theorem 4. *Let $P \subseteq \mathbf{R}^n$ be a polyhedron defined by the equations $d_i x = c_i, i = 1,2,...,k$ and the inequalities $a_i x \leq b_i, i = 1,2,...,p$ where none of the inequalities is an implicit equation. Then this is a minimal description of P if and only if*

(1) *The dimension of P is equal to $n-k$.*

(2) *Every inequality $a_i x \leq b_i$ induces a distinct facet, $i = 1,2,...,p$.* □

It follows from Theorem 4 that in order to find a linear description of a polyhedron P we need to find the implicit equations and for each facet one inequality inducing this facet. It is well known that the inequality inducing a facet is unique up to multiplication by a positive scalar and addition of any linear combination of the implicit equations.

The polyhedral method for obtaining valid inequalities for our combinatorial optimization problem proceeds in the following way. We assume that the set of implicit equations is known. For most combinatorial optimization problems the set of implicit equations is indeed trivial to find. Next we assume that $ax \leqslant \alpha_0$ is a facet inducing inequality. We are going to partition the set of all facet inducing inequalities into classes, where each class is characterized by some properties of the vector a. An example would be the class of facet inducing inequalities $ax \leqslant \alpha_0$ with $a_i < 0$ for some i. The properties of a given class are chosen in such a way that we are able to prove that $ax = \alpha_0$ implies $bx = \beta_0$, where $bx \leqslant \beta_0$ is an explicitly known valid inequality, not an implicit equation, for P. Therefore we have proved that $F = \{x \in P \mid ax = \alpha_0\} \subseteq F_1 = \{x \in P \mid bx = \beta_0\} \neq P$. Since F is a facet, i.e., a proper face of maximal dimension, it follows that $F = F_1$. Hence we have been able to find an explicitly known valid inequality inducing the facet. By applying this to all classes of the partition we find a linear description of P, which however is not necessarily minimal. The reason is that some of the properties assumed in the facet inducing inequality $ax \leqslant \alpha_0$ need not occur. However by considering all possible properties we obtain a complete linear description of P.

Let us now demonstrate the polyhedral method on the economic lot-sizing problem. The polyhedron P is the convex hull of feasible solutions where each feasible solution is characterized by subsets $I,J \subseteq \{1,2,...,n\}, 1 \in I \subseteq J$ as described in Section 1. The implicit equations of the economic lot-sizing problem (Barany, Van Roy & Wolsey (1984)) are given by $\Sigma_{i=1}^{n} x_i = d_{1n}$ and $y_1 = 1$. Let $ax + by \leqslant c$ be a facet inducing inequality which is different from the inequalities we already know for the economic lot-sizing problem, i.e., $x_i \geqslant 0$, $y_i \geqslant 0$, $y_i \leqslant 1$, $x_i \leqslant d_{in} y_i$ and $\Sigma_{t=1}^{i} x_t \geqslant d_{1i}$, $i = 1,2,...,n$. By adding a suitable linear combination of the implicit equations we can assume without loss of generality that $a \leqslant 0$, $\max_i\{a_i\} = 0$ and $b_1 = 0$. Define $F = \{(x,y) \in P \mid ax + by = c\}$.

In order to prove that $x \in F$ implies $bx = \beta_0$ for some valid inequality $bx \leqslant \beta_0$ for P it is necessary and sufficient to prove this for every extreme point of P belonging to F. This follows from the fact that every element in F can be written as a convex combination of extreme points of P belonging to F. The proof always follows the same line. We assume that there exists an extreme point $(x,y) \in F$ for which $bx < \beta_0$. Next we show that in this case we can find an $(\hat{x},\hat{y}) \in P$ such that $a\hat{x} + b\hat{y} > c$, contradicting the validity of $ax + by \leqslant c$ for all $(x,y) \in P$. Hence for every extreme point $(x,y) \in F$ we have $bx = \beta_0$ thereby proving the desired result.

We will now present the partitioning of all facet inducing inequalities, different from the given inequalities, into classes such that an explicit description of these inequalities can be found. We leave it to the reader to verify that the solution $(\hat{x},\hat{y})$ constructed belongs to P and satisfies $a\hat{x}+b\hat{y}>c$.

(i) Consider the class of facet inducing inequalities $ax+by\leqslant c$ with $b_j>0$ for some j.
Claim: $(x,y)\in F$ implies $y_j = 1$.
Proof: Assume there exists an extreme point $(x,y)\in F$ with $y_j = 0$. Then $(\hat{x},\hat{y})$ can be obtained from (x,y) by increasing y_j to one. □

We conclude that this class does not lead to a new inequality. For the remaining classes we may assume that $b_j\leqslant 0$ for all j.

(ii) Consider the class of facet inducing inequalities $ax+by\leqslant c$ with $a = 0$ and $b_j<0$ for some j.
Claim: $(x,y)\in F$ implies $y_j = 0$.
Proof: Assume there exists an extreme point $(x,y)\in F$ with $y_j = 1$. Then $(\hat{x},\hat{y})$ can be obtained from (x,y) by increasing x_1 with x_j and decreasing x_j and y_j to zero. □

Again this does not lead to a new inequality.

(iii) Consider the class of facet inducing inequalities $ax+by\leqslant c$ with $b = 0$ and $a\leqslant 0, a\neq 0$. Define $S = \{i\,|\,a_i<0\}$.
Claim: $S = \{1,2,..,l\}$ for some $l, 1\leqslant l<n$.
Proof: Since $\max_i\{a_i\} = 0$ it follows that $S\neq\{1,2,...,n\}$.
Assume that there exist indices $i,j,i<j$ such that $a_i = 0$ and $a_j<0$, i.e., assume that S is not of the desired form. Because we assumed that $F\neq\{(x,y)\in P\,|\,x_j = 0\}$ there exists an extreme point $(x,y)\in F$ with $x_j>0$. Then (x',y') obtained from (x,y) by increasing x_i with x_j, increasing y_i to one if necessary and decreasing x_j to zero, is a feasible point in P for which $ax'+by'>c$. As this contradicts the validity of the inequality $ax+by\leqslant c$ for P the claim follows. □
Claim: $(x,y)\in F$ implies $\Sigma_{i=1}^{l}x_i = d_{1l}$.
Proof: Assume there exists an extreme point $(x,y)\in F$ with $\Sigma_{i=1}^{l}x_i>d_{1l}$ and let p, $p\leqslant l$ be the largest index for which $x_p>0$. Then $(\hat{x},\hat{y})$ can be obtained from (x,y) by decreasing x_p with $\epsilon>0$, increasing x_{l+1} with ϵ and increasing y_{l+1} to one if necessary, where $\epsilon\leqslant\min\{x_p,\Sigma_{i=1}^{l}x_i-d_{1l}\}$. □

So far no new inequality has been obtained. This will change when we consider the last class of facet inducing inequalities.

(iv) Consider the class of facet inducing inequalities $ax+by\leqslant c$ with $a\leqslant 0$, $a\neq 0$, $b\leqslant 0$, $b\neq 0$. Define $S = \{i\,|\,a_i<0\}$ and $T = \{j\,|\,b_j<0\}$.
Claim: $S\cup T = \{1,2,...,l\}$ for some l, $1\leqslant l\leqslant n$.
Proof: Assume $a_i = b_i = 0$ and $a_j<0$, $i<j$. Because we assumed that $F\neq\{(x,y)\in P\,|\,x_j = 0\}$ there exists an extreme point $(x,y)\in F$ with $x_j>0$. Then (x',y') obtained from (x,y) by increasing x_i by x_j, increasing y_i to one if necessary, and decreasing x_j to zero, is a feasible point in P for which $ax'+by'>c$. As this contradicts the validity of the inequality $ax+by\leqslant c$ for P we conclude that our assumption does not hold.
Assume $a_i = b_i = 0$ and $b_j<0,i<j$. Because we assumed that $F\neq\{(x,y)\in P\,|\,y_j = 0\}$ there exists an extreme point $(x,y)\in F$ with

$y_j = 1$. Then (x',y') obtained from (x,y) by increasing x_i by x_j, increasing y_i to one if necessary and decreasing x_j and y_j to zero, is a feasible point in P for which $ax'+by'>c$. As this contradicts the validity of the inequality $ax+by \leqslant c$ for P the claim follows.

Claim: For all $(x,y)\in F$ there is at most one $j\in T\setminus S$ with $y_j = 1$.

Proof: Assume there exists an extreme point $(x,y)\in F$ with $y_j = y_k = 1, j,k\in T\setminus S$, $j<k$. Then (x',y') obtained from (x,y) by increasing x_j with x_k and decreasing x_k and y_k to zero, is a feasible point in P for which $ax'+by'>c$. As this contradicts the validity of the inequality $ax+by\leqslant c$ for P the claim follows. □

Claim: $(x,y)\in F$ implies $\Sigma_{i\in S}x_i+\Sigma_{i\in\{1,2,...,l\}\setminus S}d_{il}y_i = d_{1l}$.

Proof: Assume there exists an extreme point $(x,y)\in F$ with $\Sigma_{i\in S}x_i+\Sigma_{i\in\{1,2,...,l\}\setminus S}d_{il}y_i>d_{1l}$. Note that $\{1,2,...,l\}\setminus S = T\setminus S$. We distinguish two cases.

(i) There is no $j\in T\setminus S$ with $y_j = 1$.
Then $\Sigma_{i\in S}x_i>d_{1l}$. Therefore $l<n$. Let $p\in S$ be the largest index for which $x_p>0$. Then $(\hat{x},\hat{y})$ can be obtained from (x,y) by decreasing x_p with $\epsilon>0$, increasing x_{l+1} by ϵ and increase y_{l+1} to one if necessary, where $\epsilon\leqslant\min\{x_p,\Sigma_{i\in S}x_i-d_{1l}\}$.

(ii) There is exactly one index $j\in T\setminus S$ with $y_j = 1$.
Then $\Sigma_{i\in S}x_i+d_{jl}>d_{1l}$, i.e., $\Sigma_{i\in S}x_i>d_{i,j-1}$. Let $p\in S$ be the largest index for which $x_p>0$. If $p<j$, then $(\hat{x},\hat{y})$ can be obtained from (x,y) by decreasing x_p with $\epsilon>0$ and increasing x_j with ϵ, where $\epsilon\leqslant\Sigma_{i\in S}x_i-d_{1,j-1}$. If $p>j$, then $(\hat{x},\hat{y})$ can be obtained from (x,y) by decreasing x_p to zero and increasing x_j with x_p.

This completes the proof of the polyhedral method applied to the economic lot-sizing problem. □

Until now the polyhedral method has mostly been used in those cases where a linear description was conjectured to be complete. The polyhedral method has less frequently been used to find yet unknown valid inequalities. Polyhedral method proofs of the completenes of a linear description have for example been given for the matroid intersection problem (Pulleyblank (1983)) and the matching problem (Lovász (1979)).

4. Conclusion

We have formulated the projection method and the polyhedral method for constructing valid inequalities for combinatorial optimization problems and demonstrated the usefulness of these methods by applying them to the economic lot-sizing problem.

References

E. Balas & W.R. Pulleyblank (1983), *The Perfect Matchable Subgraph Polytope of a Bipartite Graph*, Networks 3, 495-516.

I. Barány, T.J. Van Roy & L.A. Wolsey (1984), *Uncapacitated Lot-sizing: The Convex Hull of Solutions*, Mathematical Programming Study 22, 32-43.

L. Lovász (1979), *Graph Theory and Integer Programming*, Annals of Discrete Mathematics 4, 141-158.

G.L. Nemhauser & L.A. Wolsey (1988), *Integer and Combinatorial Optimization*, Wiley & Sons.

Y. Pochet (1987), *Lot-sizing Problems: Reformulations and Cutting Plane Algorithms*, Ph.D. Thesis, CORE, Louvain-la-Neuve.

W.R. Pulleyblank (1983), *Polyhedral Combinatorics*, Mathematical Programming: The State of the Art, Springer Verlag, 312-345.

A. Schrijver (1986), *Theory of Linear and Integer Programming*, Wiley & Sons.

S. Van Hoesel, A. Kolen & A. Wagelmans (1989), *A Dual Algorithm for the Economic Lot-sizing Problem*, Report Rijksuniversiteit Limburg.

S. Van Hoesel, A. Kolen, A. Wagelmans & L. Wolsey (1989), *A Complete Linear Description and Dual Algorithm for the Economic Lot-sizing Problem with Start-up Cost*, Report, Rijksuniveriteit Limburg.

A. Wagelmans, A. Kolen & S. van Hoesel (1989), *A Parametric Analysis of the Economic Lot-sizing Problem*, Report, Rijksuniversiteit Limburg.

L.A. Wolsey (1988), *Uncapacitated Lot-sizing Problems with Start-up Costs*, CORE discussion paper 8801, Louvain-La-Neuve.

On the differentiability of the set of efficient (μ,σ^2) combinations in the Markowitz portfolio selection method

J. Kriens
Department of Econometrics
Tilburg University

In this paper differentiability properties of the set of efficient (μ,σ^2) combinations are discussed. After a review of statements made in the literature, two conditions for nondifferentiable points are derived and illustrated with some numerical examples.

I am grateful to H. Geerts and A. Volgenant for their critical remarks on an earlier version of this paper and to L.W.G. Strijbosch for his assistance in doing the computations.

CONTENTS

1. GENERAL

Markowitz studied the following investment problem, cf. H.M. Markowitz (1956), (1959). An investor wants to invest an amount b in the securities $1,...,n$. If he invests an amount x_j in security j, then

$$\sum_{j=1}^{n} x_j = b. \tag{1.1}$$

There may be more linear constraints; suppose

$$\mathcal{A}X = B \tag{1.2}$$

and

$$X \geqslant \mathbb{O} \tag{1.3}$$

should be satisfied with $\mathcal{Q}$ an $(m \times n)$-matrix, B an m-vector and $X' = (x_1,...,x_n)$.

The yearly revenue on one dollar invested in security j is a random variable $\mathbf{r}_j$ with $\mathcal{E}\ \mathbf{r}_j = \mu_j$; the covariance matrix of the $\mathbf{r}_j$ equals $\mathcal{C}$. Denote the yearly revenue of a portfolio $X = (x_1,...,x_n)'$ by $\mathbf{r}(X)$, the expected value of $\mathbf{r}(X)$ by $\mu(X)$, its variance by $\sigma^2(X)$ and let $M' = (\mu_1,...,\mu_n)$. Then

$$\mu(X) = M'X \tag{1.4}$$

and

$$\sigma^2(X) = X'\mathcal{C}X. \tag{1.5}$$

In order to find 'good' solutions of the problem, a risk averse investor may put a restriction on $\mu(X)$ and then minimize $\sigma^2(X)$, or put a restriction on $\sigma^2(X)$ and next maximize $\mu(X)$. Markowitz studies the problem from a more general point of view and introduces the notion of *efficient portfolio.* A feasible portfolio $X = \bar{X}$ is efficient if:

(a) no feasible portfolio has a revenue with larger or equal expected value and smaller variance, and

(b) no feasible portfolio has a revenue with smaller or equal variance and larger expected value;

cf. H.M. Markowitz (1959), p. 310. In the (μ,σ^2)-plane this means that if a portfolio $X = \bar{X}$ is efficient, there do not exist feasible portfolios with corresponding $(\mu(X),\sigma^2(X))$ points in the closed rectangle $\leqslant\sigma^2(\bar{X})$ and $\geqslant\mu(\bar{X})$, cf. fig. 1.1.

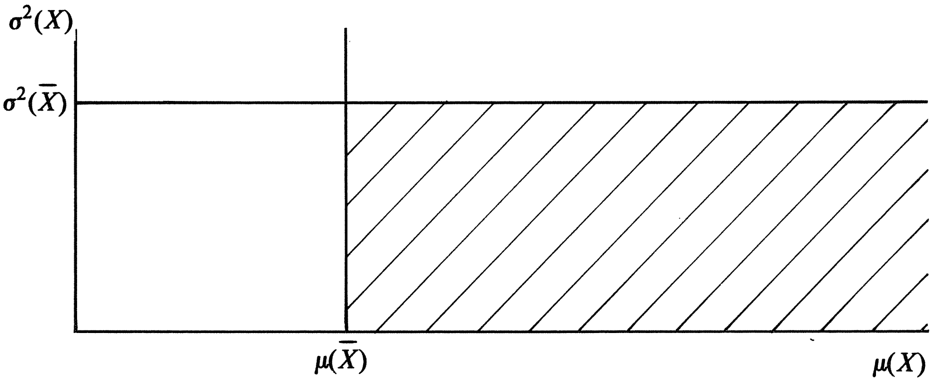


FIGURE 1.1. No feasible portfolio with $(\mu(X),\sigma^2(X))$ in the shaded area.

According to Markowitz all efficient portfolios can be derived by solving

$$\min_X\{X'\mathcal{C}X - \lambda M'X \mid \mathcal{Q}X = B \wedge X \geqslant \mathbb{O}\} \tag{1.6}$$

for all $\lambda \geqslant 0$, cf. H.M. Markowitz (1959), p. 315-316. A precise and more general statement of the theorem underlying the algorithm is given by J. Kriens en

J.Th. Van Lieshout (1988). In our case their theorem reduces to:

THEOREM. *A feasible portfolio $X = \bar{X}$ is efficient if and only if*
(a) *there exists a $\bar{\lambda}>0$ such that*

$$\min_{X}\{X'\mathcal{C}X-\bar{\lambda}M'X\,|\,\mathcal{A}X = B\wedge X\geqslant\Theta\} = \bar{X}'\mathcal{C}\bar{X}-\bar{\lambda}M'\bar{X}, \tag{1.7}$$

or (b)

$$\max_{X}[M'X\,|\,X'\mathcal{C}X = \min_{Y}\{Y'\mathcal{C}Y\,|\,\mathcal{A}Y = B\wedge Y\geqslant 0\}] = M'\bar{X}, \tag{1.8}$$

or (c)

$$\min_{X}[X'\mathcal{C}X\,|\,M'X = \max_{Y}\{M'Y\,|\,\mathcal{A}Y = B\wedge Y\geqslant 0\}] = \bar{X}'\mathcal{C}\bar{X}. \tag{1.9}$$

Note that strictly speaking condition (c) can be omitted because $M'X$ is a linear function of X.

Usually one starts with setting $\lambda = 0$ in (1.7), thus with determining the minimum value possible of the variance. Next λ is raised to get new efficient portfolios. For specific values of λ there is a change in the basis; suppose these values are $\bar{\lambda}_1, \ldots, \bar{\lambda}_k$ and that the corresponding efficient solutions are $\bar{X}_1,\ldots,\bar{X}_k$. We form the (sub)sequence $\bar{X}_{j_1},\ldots,\bar{X}_{j_l}$ $(l\leqslant k)$ from $\bar{X}_1,\ldots,\bar{X}_k$ for which the $(\bar{\mu},\bar{\sigma}^2)$ combinations are different. This (sub)sequence is *the set of corner portfolios*.

The set of all $(\mu(\bar{X}),\sigma^2(\bar{X}))$ points in the (μ,σ^2)-plane corresponding to efficient portfolios $\bar{X}$ is *the set of efficient* (μ,σ^2) *combinations* of the problem. Between the (μ,σ^2) points of two adjacent corner portfolios it is part of a strictly convex parabola, cf. J. Kriens and J.Th. van Lieshout (1988), p. 185.

The question discussed in this paper concerns the differentiability properties of this set in the (μ,σ^2) points corresponding to corner portfolios. Section 2 reviews some statements made in the literature, section 3 summarizes the expressions given by J. Kriens and J.Th. van Lieshout (1988) for the values of the basic variables in a basic feasible solution and section 4 presents an example of nondifferentiability. Next necessary and sufficient conditions are derived for getting points of nondifferentiability, which conditions are verified for some numerical examples in section 6.

2. DRIVING THROUGH THE LITERATURE ON DIFFERENTIABILITY PROPERTIES

Markowitz himself is not very clear in his statements on differentiability properties of the set of efficient (μ,σ^2) combinations. In his book he writes, cf. H.M. Markowitz (1959), p. 153: *The set of points representing efficient portfolios turns a corner, forms a sharp kink, as our passenger transfers from one critical line to another. There is typically no such kink, however, in the curve describing the relation between E and V for efficient portfolios. .... The relationship between V and E transfers from one parabola to the other without discontinuity or kink* (E is in our notation μ and V is σ^2).

And then two paragraphs further down: *It is, however, possible for the curve relating efficient V to efficient E to have a kink. ..... Whenever a kink occurs, it must be of this nature* *rather than of this nature* *.*

Markowitz does not give a numerical example with a point in which the set of efficient (μ,σ^2) points is not differentiable.

After the book by Markowitz many articles and books appeared with statements on the differentiability properties of the set of efficient (μ,σ^2) combinations. It is not planned to revue them all but just to mention a few of the 'highlights' in the literature. Keep in mind: the function in question is not necessarily differentiable everywhere, cf. the example in section 4.

An amusing mixture of mathematical and economic arguments is given by E.F. Fama and M.H. Miller (1972), p. 243. In a footnote they remark: *We should note, for the mathematically more sophisticated, that the efficient set curve need not be differentiable everywhere, so that, strictly speaking, the representation of equilibrium in terms of a 'tangency' could be incorrect. It can be shown, however, that the maximum number of points at which the efficient set curve is not differentiable cannot be greater than the number N of available assets. With infinitely divisible assets, the number of efficient portfolios is infinite; that is, the efficient set curve is continuous. Thus these nondifferentiable points do not greatly detract from our conclusions; in mathematical terms, they constitute a set of measure* 0.

As stated at the end of section 1, between two corner portfolios the set is part of a convex parabola (as already shown by Markowitz); from the algorithm based on (1.6) it follows directly that the number of corner portfolios is finite, so Fama and Miller's conclusion is trivial and not very informative.

G.P. Szegö (1980) devotes chapter 12 to the investment problem with only the constraints (1.1) and (1.3). He introduces the notion *region of admissable portfolios $\mathfrak{R}^n$ in the (μ,σ^2) plane*, defined parametrically by the equations (1.4) and (1.5) subject to (1.1) and (1.3). The boundary $\overline{\mathfrak{B}}^n$ of this region is defined by the minimal values of (1.5) subject to (1.1), (1.3) and (1.4) and therefore coincides with the set of efficient (μ,σ^2) points. His conclusion about the differentiability of this set runs (cf. p. 135): *In all circumstances, however, it follows that 'The boundary $\overline{\mathfrak{B}}^n$ of the region of admissable portfolios with nonnegativity constraints on the allocation vector ... is represented on the plane (ν,π) by a continuously differentiable curve composed of a sequence of arcs of parabolas each of which belongs to the boundary of the region of admissable portfolios of a subset of the set of n investments'*. (The plane (ν,π) is our (μ,σ^2) plane.)

The 'proof' is based on Szegö's analysis of the properties of $\overline{\mathfrak{B}}^n$. He also developes an algorithm to identify $\overline{\mathfrak{B}}^n$. The argument is rather lengthy and will not be repeated here. Moreover his conclusion on p. 135 that *their common points ... are true tangency points* is not generally correct as is shown by the example in section 4.

The last author to be quoted is J. Vörös. He states: *It can easily be seen that parabolas describing efficient return-variance connection at intervals $[c_{i-1},c_i]$ and $[c_i,c_{i+1}]$ respectively have the same values at c_i and do not intersect each other.*

Otherwise the solution would not be optimal at interval $[c_{i-1},c_i]$. *Thus we can state the following theorem. The function* $Z^+(C)$ *is continuously differentiable and convex*, cf. J. Vörös (1986), p. 298 (c is in our notation μ and $Z^+(c)$ is $\sigma^2(\mu)$).

To be sure he modifies this statement in a subsequent contribution, cf. J. Vörös (1987), p. 305. The theorem now runs: *The efficient frontier* $Z_+^2(c)$ *is continuously differentiable except in points where* $a_i = a_j$ *for all* $i,j \in M$ (a_i is μ_i in our notation, $Z_+^2(c)$ is again $\sigma^2(\mu)$ and M is the set of x_j-variables being in the basis). Because the condition $a_i = a_j$ for all $i,j \in M$ only makes sense if M contains at least 2 elements, as Vörös assumes indeed, the restriction in the theorem relates to efficient (μ,σ^2) points with 2 or more x_j variables in the basis. The proof does not take into account cases in which M contains only one element, and then the set may be indifferentiable as the example in section 4 shows. So, this mere point already implies that the formulation as well as the proof of the theorem is not correct.

Vörös develops the same algorithm for identifying the efficient (μ,σ^2) points as Szegö did, but both do not prove that all efficient points are actually found in this way. Note that this algorithm is different from the algorithm based on (1.6). As a matter of fact the solution presented for the second problem in J. Vörös (1986) is incorrect; the efficient point with minimum variance is the point (8.3×10^3;10.53×10^6) and not the point shown in Vörös' figure 2.

3. $\mathcal{E}$XPLICIT EXPRESSIONS FOR EFFICIENT PORTFOLIOS

Starting from the Kuhn-Tucker conditions for the optimal solution of (1.6), Kriens and Van Lieshout (1988) derive an expression for the values of the basic variables which, if $\mathcal{C}$ is positive definite, holds for every efficient portfolio. With constraints

$$\mathcal{A}X \leqslant B \tag{3.1}$$

rather than (1.2), the Kuhn-Tucker conditions run

$$-2\mathcal{C}\bar{X}-\mathcal{A}'\bar{U}+\bar{V} = -\bar{\lambda}M, \tag{3.2}$$

$$\mathcal{A}\bar{X}+\bar{Y} = B, \tag{3.3}$$

$$\bar{V}'\bar{X} = 0,\ \bar{U}'\bar{Y} = 0,\ \bar{X},\bar{Y},\bar{U},\bar{V} \geqslant \mathcal{O}; \tag{3.4}$$

$\bar{Y}$ contains the values of the slack variables, $\bar{U}$ and $\bar{V}$ the values of the vectors of Lagrange multipliers.

Omitting bars to get variables X,Y,U and V, the equations (3.2) and (3.3) can be summarized as

X'	Y'	U'	V'	
$-2\mathcal{C}$	$\mathcal{O}$	$-\mathcal{A}'$	$\mathcal{I}$	$-\bar{\lambda}M$
$\mathcal{A}$	$\mathcal{I}$	$\mathcal{O}$	$\mathcal{O}$	B

(3.5)

If

$$Z'_b = (X'_b, Y'_b, U'_b, V'_b) \tag{3.6}$$

denotes the set of basic variables for a given efficient portfolio, (3.5) can be partitioned into

$$\begin{array}{|ccccccccc|}
\hline
X'_b & X'_{nb} & Y'_b & Y'_{nb} & U'_b & U'_{nb} & V'_b & V'_{nb} & \\
\hline
-2\mathcal{C}_{b_1} & -2\mathcal{C}_{nb_1} & \mathcal{O} & \mathcal{O} & -\mathcal{A}'_{b_1} & \mathcal{A}'_{b_2} & \mathcal{O} & \mathcal{I} & -\bar{\lambda}M_{b_1} \\
-2\mathcal{C}_{b_2} & -2\mathcal{C}_{nb_2} & \mathcal{O} & \mathcal{O} & -\mathcal{A}'_{nb_1} & \mathcal{A}'_{nb_2} & \mathcal{I} & \mathcal{O} & -\bar{\lambda}M_{b_2} \\
\mathcal{A}_{b_1} & \mathcal{A}_{nb_1} & \mathcal{O} & \mathcal{I} & \mathcal{O} & \mathcal{O} & \mathcal{O} & \mathcal{O} & B_{b_1} \\
\mathcal{A}_{b_2} & \mathcal{A}_{nb_2} & \mathcal{I} & \mathcal{O} & \mathcal{O} & \mathcal{O} & \mathcal{O} & \mathcal{O} & B_{b_2} \\
\hline
\end{array} \tag{3.7}$$

The matrix $-2\mathcal{C}$ is partitioned into the square matrices $-2\mathcal{C}_{b_1}$ and $-2\mathcal{C}_{nb_2}$ corresponding to basic and non-basic variables x_j and into $-2\mathcal{C}_{b_2}$ and $-2\mathcal{C}_{nb_1}$ with $\mathcal{C}_{b_2} = \mathcal{C}'_{nb_1} \cdot \mathcal{A}_{b_1}, \mathcal{A}_{nb_1}$ and B_{b_1} represent the active constraints, $\mathcal{A}_{b_2}$, $\mathcal{A}_{nb_2}$ and B_{b_2} the non-active constraints. Therefore there are identity matrices in the fourth place of the Y'_b column and in the third place of the Y'_{nb} column. The matrix of coefficients of basic variables is

$$\mathfrak{B} = \begin{bmatrix} -2\mathcal{C}_{b_1} & \mathcal{O} & -\mathcal{A}'_{b_1} & \mathcal{O} \\ -2\mathcal{C}_{b_2} & \mathcal{O} & -\mathcal{A}'_{nb_1} & \mathcal{I} \\ \mathcal{A}_{b_1} & \mathcal{O} & \mathcal{O} & \mathcal{O} \\ \mathcal{A}_{b_2} & \mathcal{I} & \mathcal{O} & \mathcal{O} \end{bmatrix} \tag{3.8}$$

To facilitate computations Kriens and Van Lieshout reshuffle (3.8) into

$$\mathfrak{B}_v = \begin{bmatrix} -2\mathcal{C}_{b_1} & -\mathcal{A}'_{b_1} & \mathcal{O} & \mathcal{O} \\ \mathcal{A}_{b_1} & \mathcal{O} & \mathcal{O} & \mathcal{O} \\ -2\mathcal{C}_{b_2} & -\mathcal{A}'_{nb_1} & \mathcal{I} & \mathcal{O} \\ \mathcal{A}_{b_2} & \mathcal{O} & \mathcal{O} & \mathcal{I} \end{bmatrix} \tag{3.9}$$

The values of the basic variables are

$$\bar{Z}_{bv} = \mathfrak{B}_v^{-1} \begin{bmatrix} \mathcal{O} \\ B_{b_1} \\ \mathcal{O} \\ B_{b_2} \end{bmatrix} - \bar{\lambda}\mathfrak{B}_v^{-1} \begin{bmatrix} M_{b_1} \\ \mathcal{O} \\ M_{b_2} \\ \mathcal{O} \end{bmatrix} \tag{3.10}$$

with $\bar{Z}'_{bv} = (\bar{X}'_b, \bar{U}'_b, \bar{V}'_b, \bar{Y}'_b)$. Explicit expressions for the values of the basic variables are found by computing $\mathfrak{B}_v^{-1}$:

$$\mathcal{B}_v^{-1} = \left[\begin{array}{c|c} \begin{bmatrix}-2\mathcal{C}_{b_1} & -\mathcal{A}'_{b_1}\\ \mathcal{A}_{b_1} & \mathcal{O}\end{bmatrix}^{-1} & \mathcal{O} \\ \hline -\begin{bmatrix}-2\mathcal{C}_{b_2} & -\mathcal{A}'_{nb_1}\\ \mathcal{A}_{b_2} & \mathcal{O}\end{bmatrix}\begin{bmatrix}-2\mathcal{C}_{b_1} & -\mathcal{A}'_{b_1}\\ \mathcal{A}_{b_1} & \mathcal{O}\end{bmatrix}^{-1} & \begin{bmatrix}\mathcal{I} & \mathcal{O}\\ \mathcal{O} & \mathcal{I}\end{bmatrix}\end{array}\right] \tag{3.11}$$

with

$$\begin{bmatrix}-2\mathcal{C}_{b_1} & -\mathcal{A}'_{b_1}\\ \mathcal{A}_{b_1} & \mathcal{O}\end{bmatrix}^{-1} =$$

$$\left[\begin{array}{c|c} -\frac{1}{2}\mathcal{C}_{b_1}^{-1} + \frac{1}{2}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1} & \mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1} \\ \hline -(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1} & -2(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\end{array}\right]. \tag{3.12}$$

Substituting (3.12) into (3.11) and the result into (3.10), they find

$$\overline{X}_b = A + D\overline{\lambda} \tag{3.13}$$

with

$$A = \mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}B_{b_1} \tag{3.14}$$

and

$$D = \frac{1}{2}[\mathcal{C}_{b_1}^{-1} - \mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}]M_{b_1}. \tag{3.15}$$

The corresponding values $\mu(\overline{X}_b)$ and $\sigma^2(\overline{X}_b)$ are

$$\mu(\overline{X}_b) = M'_{b_1}A + M'_{b_1}D\overline{\lambda} \tag{3.16}$$

$$\sigma^2(\overline{X}_b) = A'\mathcal{C}_{b_1}A + 2A'\mathcal{C}_{b_1}D\overline{\lambda} + D'\mathcal{C}_{b_1}D\overline{\lambda}^2. \tag{3.17}$$

For the proofs, see appendix A of their contribution.

4. LOOKING AT AN EXAMPLE OF NONDIFFERENTIABILITY

The following example has a point of nondifferentiability; it originates with Markowitz and was handed to me by Vörös. The data are

$$M = \begin{bmatrix}1\\3\\5\end{bmatrix}, \mathcal{C} = \begin{bmatrix}3 & 3 & -1\\ 3 & 11 & 23\\ -1 & 23 & 75\end{bmatrix}, \ \mathcal{A} = (1\ 1\ 1),\ B = (1). \tag{4.1}$$

For this problem conditions (3.2),...,(3.4) reduce to

$$\begin{array}{rrrrrrrll}
-6x_1 & -6x_2 & +2x_3 & -u_1 & +v_1 & & & = -\lambda & \\
-6x_1 & -22x_2 & -46x_3 & -u_1 & & +v_2 & & = -3\lambda & (4.2) \\
2x_1 & -46x_2 & -150x_3 & -u_1 & & & +v_3 & = 5\lambda & \\
x_1 & +x_2 & +x_3 & +y_{j-1} & & & & = 1 & (4.3)
\end{array}$$

$$\sum_{j=1}^{3} v_j x_j = 0,\ u_1 y_1 = 0,\ X,Y,U,V \geqslant \Theta; \qquad (4.4)$$

the bars denoting optimal values are omitted.

In order to perform the portfolio solution analysis a user written subroutine has been linked to the linear optimization package LINDO. In that subroutine special features of LINDO like the parametric analysis option have been used.

TABLE 4.1. Basic solutions of the example.

$\bar\lambda$	$\bar x_1$	$\bar x_2$	$\bar x_3$	$\bar\mu$	$\bar\sigma^2$
0	0.950	0	0.050	1.20	2.80
3	0.875	0	0.125	1.50	3.25
4	0.500	0.500	0	2.00	5.00
8	0	1.000	0	3.00	11.00
12	0	1.000	0	3.00	11.00
52	0	0	1.000	5.00	75.00

With formulae (3.16) and (3.17) the relationships between $\mu(\bar X_b),\bar\lambda$ and $\sigma^2(\bar X_b),\bar\lambda$ can be derived. It is found that for the corner portfolio $\bar X' = (0\ 1\ 0)$ with $(\bar\mu,\bar\sigma^2) = (3,11)$ the left hand side derivative of the efficient (μ,σ^2) set equals 8 in (3,11) whereas the right hand side derivative equals 12. So the set of efficient (μ,σ^2) points is not differentiable in the point (3,11). In the computations this property is revealed by the production of two successive bases with different values of $\bar\lambda$ but the same optimal $\bar X$-vector. The results are also in agreement with

$$\left[\frac{d\sigma^2}{d\mu}\right]_{(\bar\mu,\bar\sigma^2)} = \bar\lambda \qquad (4.5)$$

if the set is differentiable, $\lim\limits_{\mu\uparrow 3}\frac{d\sigma^2}{d\mu} = 8$ and $\lim\limits_{\mu\downarrow 3}\frac{d\sigma^2}{d\mu} = 12$.

However, the algorithm does not show any computational problems, this as opposed to a conjecture by Vörös concerning his own algorithm: *This counterexample shows that the procedure suggested by Szegö and of the author may not be valid so generally as it is stated ...*, cf. J. Vörös (1987), p. 305.

5. EVIDENT NECESSARY AND SUFFICIENT CONDITIONS FOR NONDIFFERENTIABILITY

Inspection of the example in section 4 makes clear that a point of nondifferentiability in the set of efficient (μ,σ^2) points comes into being if for a range of $\bar{\lambda}$ values the vector $\bar{X}_b$ remains the same. From (3.13) it follows that this is the case if and only if D equals $\mathcal{O}$. Define $\mu_{\min} := \min_i \mu_i$ and $\mu_{\max} := \max_i \mu_i$; then for an efficient (μ,σ^2) point with $\mu \in (\mu_{\min}, \mu_{\max})$ a necessary and sufficient condition for nondifferentiability runs $D = \mathcal{O}$. The next two theorems exploit this property for the problem with only the restrictions (1.1) and (1.3).

THEOREM 5.1. *If in the investment problem subject to* (1.1) *and* (1.3) $\mathcal{C}$ *is positive definite and a corner portfolio with* $\mu \in (\mu_{\min}, \mu_{\max})$ *contains only one x-variable* $>$ 0, *then the set of efficient* (μ,σ^2) *points is nondifferentiable in that point.*

PROOF. Suppose $\bar{x}_i > 0$, then $\bar{x}_i = b$, $\mathcal{C}_{b_1} = (c_{ii})$, $\mathcal{A}_{b_1} = (1)$, $M_{b_1} = (\mu_i)$. From (3.15) it follows

$$D = -\frac{1}{2}\mathcal{C}_{b_1}^{-1}[\mathcal{I}-\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}]M_{b_1}. \tag{5.1}$$

Substitution of the values of $\mathcal{A}_{b_1}$ and $\mathcal{C}_{b_1}^{-1}$ shows

$$\mathcal{I}-\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1} = \mathcal{O}, \tag{5.2}$$

so $D = \mathcal{O}$ and $\bar{X}_b = A$, cf. (3.13). □

Now suppose a corner portfolio contains as basic variables $X'_b = (x_1,\ldots,x_k)$ $(k>1)$. Define

$$\mathcal{M} = (m_{ij}) := \mathcal{C}_{b_1}^{-1} \tag{5.3}$$

$$f := \sum_{i=1}^{k}\sum_{j=1}^{k} m_{ij} \tag{5.4}$$

$$d := \sum_{i=1}^{k}(\sum_{j=1}^{k} m_{ij}\mu_j). \tag{5.5}$$

THEOREM 5.2. *If in the investment problem subject to* (1.1) *and* (1.3) $\mathcal{C}$ *is positive definite and a corner portfolio with* $\mu \in (\mu_{\min}, \mu_{\max})$ *contains* $k(>1)$ *variables* $>$ 0, *then the set of efficient* (μ,σ^2) *points is nondifferentiable in that point if and only if all corresponding* μ*-values are equal to* d/f.

PROOF. Let

$$X_b = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_k \end{bmatrix}, \mathcal{C}_{b_1} = \begin{bmatrix} c_{11}\ldots c_{1k} \\ \cdot \\ \cdot \\ c_{k1}\ldots c_{kk} \end{bmatrix}, \mathcal{A}'_{b_1} = \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ 1 \end{bmatrix}, M_{b_1} = \begin{bmatrix} \mu_1 \\ \cdot \\ \cdot \\ \mu_k \end{bmatrix},$$

then

$$(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1} = \frac{1}{f} \tag{5.6}$$

and $D = \mathcal{O}$ can be reduced to

$$f\begin{bmatrix}\mu_1\\ \cdot\\ \cdot\\ \cdot\\ \mu_k\end{bmatrix} - \begin{bmatrix}d\\ \cdot\\ \cdot\\ \cdot\\ d\end{bmatrix} = \begin{bmatrix}0\\ \cdot\\ \cdot\\ \cdot\\ 0\end{bmatrix}, \tag{5.7}$$

or

$$\forall_{i\in\{1,\ldots,k\}}\ \mu_i = \frac{d}{f}. \tag{5.8}$$

So $D = \mathcal{O}$ if and only if (5.8) holds. □

REMARK. As a consequence of these theorems, D may be a zero vector and therefore the statement by Kriens and Van Lieshout (1988) that $M'_{b_1}\cdot D$ is always $\neq 0$ (p. 187) cannot be generally correct. In their 'proof', see appendix B of the article, the matrix $\mathfrak{B}_v^*$ does not necessarily have an inverse as is illustrated by the example in section 4: for the efficient portfolio (0 1 0) their matrix $\mathfrak{B}_v^*$ equals

$$\mathfrak{B}_v^* = \left[\begin{array}{cccc|c} -22 & -1 & 0 & 0 & 3\\ 1 & 0 & 0 & 0 & 0\\ -6 & -1 & 1 & 0 & 1\\ -46 & -1 & 0 & 1 & 5\\ - & - & - & - & -\\ 3 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{5.9}$$

6. VERIFICATION OF THE CONDITIONS IN SOME EXAMPLES

In this section forementioned formulae and conditions are illustrated with the help of some examples.

EXAMPLE 6.1. For the data, see section 4. In the case of corner portfolio $\bar{X}' = (0\ 1\ 0)$ there is only one x-variable > 0 and the set of efficient (μ,σ^2) points is indeed nondifferentiable in the corresponding point $(\overline{\mu},\overline{\sigma}^2) = (3,11)$. Substitution of the data in (3.15) leads to $D = \mathcal{O}$.

The behaviour of the dual variables is also clear. If (3.12) is substituted in (3.11) and the result into (3.10), we get

$$\bar{U}_b = -2(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}B_{b_1} + \bar{\lambda}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}M_{b_1}. \tag{6.1}$$

For $\bar{\lambda} = 0$, x_1 and x_3 are basic variables and then

$$\bar{u}_1 = 2(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}B_{b_1} = -5.6;$$

if we look at the corner portfolio $\bar{X}' = (0\ 1\ 0)$, then only x_2 is basic variable and from (6.1) it follows

$$\bar{u}_1 = -44+3\bar{\lambda},$$

so if $\bar{\lambda}$ rises from 8 to 12, the value of $\bar{u}_1$ rises from -20 to -8.

In the same way the values of $\bar{V}_b$ can be derived from the third 'row' in (3.10). Therefore we need the elements in the third 'row' of (3.11). The first two elements in this 'row' of $\mathfrak{B}_v^{-1}$ are

$$\mathcal{C}_{b_2}\{-\mathcal{C}_{b_1}^{-1}+\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\} \qquad (6.2)$$
$$-\mathcal{A}'_{nb_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}$$

and

$$2\mathcal{C}_{b_2}\{\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\}+\mathcal{A}'_{nb_1}\{-2(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\}; \qquad (6.3)$$

the third element equals $\mathcal{I}$ and the fourth $\mathcal{O}$. So

$$\bar{V}_b = [2\mathcal{C}_{b_2}\{\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\}+\mathcal{A}'_{nb_1}\{-2(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\}]\cdot B_{b_1}$$
$$-\lambda[\{\mathcal{C}_{b_2}\{-\mathcal{C}_{b_1}^{-1}+\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\}- \qquad (6.4)$$
$$\mathcal{A}'_{nb_1}(\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{b_1})^{-1}\cdot\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\}M_{b_1}+\mathcal{I}\cdot M_{b_2}].$$

If conditions (1.2) only consist of $\Sigma_{j=1}^{n}x_j = 1$, then, using (5.4) and (5.6), (6.4) can be simplified to

$$\bar{V}_b = \frac{2}{f}(\mathcal{C}_{b_2}\mathcal{C}_{b_1}^{-1}\mathcal{A}'_{nb_1}-\mathcal{A}'_{nb_1}) \qquad (6.5)$$
$$-\bar{\lambda}[\{\mathcal{C}_{b_2}\mathcal{C}_{b_1}\{-\mathcal{I}+\frac{1}{f}\mathcal{A}'_{b_1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\}-\frac{1}{f}\mathcal{A}'_{nb_1}\mathcal{A}_{b_1}\mathcal{C}_{b_1}^{-1}\}\cdot M_{b_1}+\mathcal{I}M_{b_2}].$$

In the case of the efficient portfolio (0 1 0) in the example, (6.5) reduces to

$$\bar{V}_b = \begin{bmatrix}\bar{v}_1\\ \bar{v}_3\end{bmatrix} = \begin{bmatrix}-16+2\bar{\lambda}\\ 24-2\bar{\lambda}\end{bmatrix}. \qquad (6.6)$$

If $\bar{\lambda}$ is raised, for $\bar{\lambda} = 8$, x_1 leaves the basis and v_1 comes in and for $\bar{\lambda} = 12$, $\bar{v}_3 = 0$, so for $\bar{\lambda}>12, v_3$ leaves the basis and x_3 comes in; cf. also table 4.1.

EXAMPLE 6.2. Assume

$$M = \begin{bmatrix}1\\5\\5\\10\end{bmatrix}, \mathcal{C} = \begin{bmatrix}1 & 1.4 & 1.5 & 16\\ 1.4 & 4 & 0 & 32\\ 1.5 & 0 & 8 & 6\\ 16 & 32 & 6 & 400\end{bmatrix}, \mathcal{A} = (1\ 1\ 1\ 1), B = (1). \qquad (6.7)$$

Starting from the conditions (3.2),...,(3.4) the LINDO optimization routine generates the basic solutions presented in table 6.1.

The set of efficient (μ,σ^2) points is not differentiable in the point $(\bar{\mu},\bar{\sigma}^2) =$ (5.00,2.67) corresponding to the efficient portfolio $\bar{X}' =$ (0 0.67 0.33 0). According to theorem 5.2 this behaviour was to be expected. The set of

TABLE 6.1. Basic solutions of example 6.2.

$\bar{\lambda}$	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_4$	$\bar{\mu}$	$\bar{\sigma}^2$
0	1.00	0	0	0	1.00	1.00
0.200	1.00	0	0	0	1.00	1.00
0.227	0.98	0.02	0	0	1.10	1.02
0.617	0	0.67	0.33	0	5.00	2.67
8.267	0	0.67	0.33	0	5.00	2.67
36.471	0	0	0.76	0.24	6.18	28.98
157.600	0	0	0	1.00	10.00	400.00

corresponding $\bar{\lambda}$ values equals $[0.617 \leqslant \bar{\lambda} \leqslant 8.267]$. Substitution of

$$M_{b_1} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \mathcal{C}_{b_1} = \begin{bmatrix} 4 & 0 \\ 0 & 8 \end{bmatrix}, \mathcal{A}_{b_1} = (1\ 1), M_{b_1} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \tag{6.8}$$

in (3.15) results in $D = \Theta$.

The values of the dual variables can be derived by substituting (6.8) into (6.1) and (6.5) respectively. In the latter case we find

$$\bar{V}_b = \begin{bmatrix} \bar{v}_1 \\ \bar{v}_4 \end{bmatrix} = \begin{bmatrix} -2.467+4\bar{\lambda} \\ 41.333-5\bar{\lambda} \end{bmatrix}; \tag{6.9}$$

so for $\bar{\lambda} = 0.617$, $\bar{v}_1$ is > 0 and enters the basis, whereas for $\bar{\lambda} = 8.267$, $\bar{v}_4$ becomes < 0 and leaves the basis.

The last example is due to H. Geerts; in this case the theorems of section 5 do not apply because besides condition (1.1) there is one more constraint.

EXAMPLE 6.3. Let

$$M = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \mathcal{C} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}, \mathcal{A} = \begin{bmatrix} 1 & 1 \\ \frac{8}{7} & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ \frac{15}{14} \end{bmatrix}. \tag{6.10}$$

Using the conditions (3.2),...,(3.4) the basic solutions presented in table 6.2 are found.

The set of efficient (μ,σ^2) points is nondifferentiable in $(\bar{\mu},\bar{\sigma}^2) = (1.500, 1.250)$, the corresponding values of $\bar{\lambda}$ are $[3.000 \leqslant \bar{\lambda} \leqslant 3.333]$. The value of D equals Θ because as all reciprocals exist the expression between square brackets in (5.1) is the zero matrix. There are no basic variables v_j whereas the expression for $\bar{U}_b$ follows from (6.1) and runs

$$\bar{U}_b = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} = \begin{bmatrix} 20 - 6\bar{\lambda} \\ -21 + 7\bar{\lambda} \end{bmatrix}. \tag{6.11}$$

TABLE 6.2. Basic solutions of example 6.3.

$\bar{\lambda}$	$\bar{x}_1$	$\bar{x}_2$	$\bar{\mu}$	$\bar{\sigma}^2$
0	0	0	0	0
1.333	0.333	0.667	1.333	0.889
3.000	0.500	0.500	1.500	1.250
3.333	0.500	0.500	1.500	1.250
8.750	0.938	0	1.875	3.516

7. ℰVIDENTIAL MATTER

E.F. FAMA and M.H. MILLER (1972), *The Theory of Finance*, Holt, Rinehart and Winston, New York.

J. KRIENS and J.TH. VAN LIESHOUT (1988), *Notes on the Markowitz portfolio selection method*, Statistica Neerlandica 42, 181-191.

H.M. MARKOWITZ (1956), *The Optimization of a Quadratic Function subject to Linear Constraints*, Naval Research Logistics Quarterly 3, 111-133.

H.M. MARKOWITZ (1959), *Portfolio Selection*, John Wiley and Sons, New York.

G.P. SZEGÖ (1980), *Portfolio Theory with Application to Bank Asset Management*, Academic Press, New York.

J. VÖRÖS (1986), *Portfolio analysis - An analytic derivation of the efficient portfolio frontier*, European Journal of Operational Research 23, 294-300.

J. VÖRÖS (1987), *The explicit derivation of the efficient portfolio frontier in the case of degeneracy and general singularity*, European Journal of Operational Research 32, 302-310.

Probabilistic analysis of algorithms

A.H.G. Rinnooy Kan
Econometric Institute
Erasmus University, Rotterdam

L. Stougie
Institute of Actuarial Sciences and Econometrics
University of Amsterdam

Probabilistic analysis of algorithms for combinatorial optimization problems has only recently become an active research area. The nature of mathematical techniques used in the analysis and of the results obtained is illustrated by a selection of examples from the literature. The selection is made in such a way that various problem classes in combinatorial optimization are represented. At several points, challenging open problems are mentioned as an encouragement for further research in this area.

1. Introduction

Suppose that two thieves meet on a regular basis to divide the proceeds of their joint effort. Each stolen object has a specific value and has to be assigned to one of the two. For obvious reasons, they are interested in a quick and fair partitioning scheme.

In spite of its apparent simplicity, the above combinatorial problem is not easy to solve if we insist on an optimal solution, i.e. one in which the difference between the values assigned to each thief is as small as possible. According to the theory of computational complexity this problem belongs to the class of *NP-complete* problems. This implies that any optimization method for its solution can be expected to perform very poorly on some occasions: more formally, its *worst case running* time is likely to grow *exponentially* with *problem size*.

Hence, in choosing a solution scheme, the thieves will be forced into a trade-off between two features of algorithmic quality: the *computational effort* (the smaller running time, the better) on one hand and the *computational result* (the smaller deviation from optimality, the better) on the other hand. As mentioned above it is very unlikely we can obtain an algorithm that simultaneously guarantees a fast (polynomially bounded) running time and a zero deviation from the optimal solution value.

One possible way out of this dilemma is to insist no longer on an *absolute* guarantee. For practical purposes an algorithm that, with respect to both effort and result, does well in the majority of cases or even on average might be perfectly acceptable.

Probability theory provides the natural setting for such an analysis of algorithms. This analysis starts from a specification of a *probability distribution* over the *class of all problem instances*. The running time and the solution value of a particular algorithm are then *random variables*, whose behaviour can be studied and evaluated. This approach can therefore be viewed as the analytical counterpart to the familiar *experimental* analysis in which an algorithm is tried out on a set of supposedly representative test problems and evaluated statistically. Here, we obtain the rigor of mathematical analysis. However we do so at a price; at the current state of the art, only relatively straightforward solution methods have admitted a detailed probabilistic analysis. More intricate algorithms usually give rise to probabilities that are conditioned on the outcome of certain algorithmic steps, which complicates a direct analysis dramatically.

Although the probabilistic analysis of algorithms has only recently become an active research area, it has already generated an impressive number of publications. A concise survey of this area would require the prior introduction of many techniques from probability theory so as to encompass the diversity of ideas and approaches that one finds in the literature. A recent annotated bibliography [26] provides an up to date survey of the available publications. In this paper we will illustrate the nature of the analysis and of the results by a selection of examples from the literature.

In Section 2 we consider the problem of the two thieves, which is known as the PARTITION problem in which one seeks to minimize the size of largest share. This problem is an example of a *number problem*, a problem whose input consists primarily of numbers. In Section 3 we review some representative results that are known for *geometric* problems in *Euclidean* space. In Section 4, we examine combinatorial optimization problems defined on *graphs and networks*. Finally, concluding remarks and directions for future research are given in Section 5.

We conclude this introduction by a short digression on modes of *stochastic convergence*, an essential concept if we want to analyse the behaviour of random variables such as the error of an approximation algorithm with increasing problem size.

Almost sure convergence of a sequence of random variables $y_1, y_2, \dots$ to a constant c means that

$$Pr\{\lim_{n \to \infty} y_n = c\} = 1.$$

It implies the weaker *convergence in probability*, which stands for

$$\lim_{n \to \infty} Pr\{|y_n - c| > \epsilon\} = 0$$

for every $\epsilon > 0$. The reverse implication holds if the latter convergence is fast enough to have for every $\epsilon > 0$

$$\sum_{n=1}^{\infty} Pr\{|y_n - c| > \epsilon\} < \infty.$$

Similarly, convergence of y_n to c *in expectation*, i.e.,

$$\lim_{n\to\infty} |Ey_n - c| = 0,$$

also implies convergence in probability, with the reverse implication holding under additional integrability conditions on y_n.

2. The partition problem

Perhaps the simplest way to solve the Partition problem of the two thieves is to allow each thief to choose a particular item in turn until they have all been assigned. If the j-th item has value a_j ($j=1,2,...,n$), then this amounts to ordering the items according to decreasing a_j values $a^{(n)} \geqslant a^{(n-1)} \geqslant ... \geqslant a^{(1)}$, one thieve receives $a^{(n)} + a^{(n-2)} + ...$, the other $a^{(n-1)} + a^{(n-3)} + ...$.

This is clearly a fast approximation algorithm ($O(n\log n)$) for sorting the items) that may, however, produce a very inequitable result: in the worst case, the first thieve may receive up to 50 percent more than the optimal partition would grant him. (Take $a_1 = 2$, $a_{2i} = a_{2i+1} = 2^{-i} (i \geqslant 1)$.) How about the average case behaviour? To answer that question we specify a probability distribution over all problem instances. Let us assume that the a_j are *independently identically distributed* random variables that have a distribution with finite first moment.

Under this assumption the optimal solution value z_n^{OPT} of the partition problem (i.e. the smallest possible size of the larger share) is a random variable that turns out to be almost surely (a.s.) *asymptotic to* the expected value of the lower bound $\Sigma_{j=i}^n a_j/2$:

$$\frac{z_n^{OPT}}{E\sum_{j=i}^{n} a_j/2} = \frac{z_n^{OPT}}{\frac{n}{2}Ea_1} \to 1 \text{ (a.s.)}. \tag{2.1}$$

This result provides a first example of *asymptotic probabilistic value analysis*: for n large enough the optimal solution value can be guessed with increasing (relative) accuracy. What about the size of the larger share z_n^H produced by the approximation scheme proposed? We have that

$$z_n^H = \frac{1}{2}\sum_{j=1}^{n} a_j + \frac{1}{2}[(a^{(n)} - a^{(n-1)}) + (a^{(n-2)} - a^{(n-3)}) + ...],$$

$$\leqslant \frac{1}{2}\sum_{j=1}^{n} a_j + \frac{1}{2}a^{(n)}.$$

Under the assumption of the finite first moment for the distribution of the a_j, we have that the strong law of large numbers implies that

$$\frac{\sum_{j=1}^{n} a_j}{nEa_1} \to 1 \text{ (a.s.)}$$

and that $a^{(n)} = o(n)$ (a.s.), so that

$$\frac{z_n^H}{\frac{n}{2}Ea_1} \to 1 \text{ (a.s.)}. \tag{2.2}$$

(2.1) and (2.2) together imply that

$$\frac{z_n^H}{z_n^{OPT}} \to 1 \text{ (a.s.)}.$$

The approximation algorithm is called *asymptotically optimal*: its *relative error* $(z_n^H - z_n^{OPT})/z_n^{OPT}$ a.s. goes to 0. Hence a probabilistic analysis leads to a much more optimistic conclusion than a worst case one.

What about the *absolute error* $z_n^H - z_n^{OPT}$ or the absolute difference between the two shares? Neither of these two quantities goes to 0 for the above heuristic, so there is room for improvement. A slightly more sophisticated scheme would be to allow each thief in turn to select items until the value of his share exceeds the value of the current share of his colleague. From a worst case point of view this is a much more reasonable approach: the larger share never exceeds its smallest possible size by more than 16.6... percent [16]. (For a worst case example take $a_1 = a_2 = 3$, $a_3 = a_4 = a_5 = 2$.) In a probabilistic sense the difference is even more impressive. Of course, the relative error again goes to 0. However, if we assume that, next to finite first moment, the distribution of the a_j is strictly increasing over an interval [0,b] with $b > 0$, then also the absolute difference between the two shares d_n^H (which is clearly an upper bound on the absolute error) satisfies

$$d_n^H \to 0 \text{ (a.s.)}$$

[11]. To prove this result one observes that

$$\begin{aligned} d_n^H &\leqslant \max\nolimits_{1 \leqslant k \leqslant n}\{d_{n-1}^H - a^{(1)}, a^{(1)}\} \\ &\leqslant a^{([\delta n])} + \max\{a^{(n)} - \sum_{j=i}^{[\delta n]} a^{(j)}, 0\} \end{aligned}$$

for any $\delta > 0$. The first term converges to δ a.s. and can therefore be made arbitrarily small; for any fixed δ, the second term converges a.s. to 0 since $\Sigma_{j=i}^{[\delta n]} a_j = O(n)$, whereas $a^{(n)} = o(n)$ as mentioned before.

The two results presented so far demonstrate the importance of the theory of *order statistics* for the analysis of approximation algorithms that involve the sorting of numbers; *priority rules* generally fall into this class.

One weakness of the type of results presented above is its asymptotic nature, i.e., its validity only for *sufficiently large* values of n. Ideally one would like to have a precise indication of what values of n are sufficiently large.

At the very least, one could be grateful for information on the rate at which d_n^H converges to 0. It can be shown [12] that

$$\limsup_{n \to \infty} \frac{d_n^H}{\log\log n / n} = O(1) \text{ (a.s.)}.$$

However, it is also known that d_n^{OPT}, the smallest possible difference, satisfies

$$\limsup_{n \to \infty} \frac{d_n^{OPT}}{n^2 2^{-n}} = O(1) \text{ (a.s.)}$$

[23]. Hence, the exponential effort that may be required for the computation of the optimal partition is at least rewarded by an exponential decrease to 0 of the difference between the two shares. Can this also be achieved a.s. in polynomial time?

The answer to this question is unknown, but an improvement on the rate of convergence of the previous heuristic is obtained by a *differencing method.* Essentially, two items are assigned at a time and their difference is compensated for. More precisely, in the first iteration $a^{(n)}$ will be assigned to one thief, and $a^{(n-1)}$ to the other. The items will then be replaced by a single item with value $a^{(n)} - a^{(n-1)}$ and the process is repeated on the set of $n-1$ items until only one item remains; its value represents the difference between the two shares. A simple backtracking procedure establishes the partition in terms of the original items.

In the worst case this method is not better then the previous one. The probabilistic analysis of its performance is difficult: each step in the algorithm conditions the probability distributions of events encountered in the succeeding steps in a complicated fashion.

In [22] this obstacle is overcome by changing the algorithm so that (with high probability) the value produced will not be affected but its modified behaviour can be analyzed rigorously. Through this approach it was established that

$$\limsup_{n\to\infty} \frac{\mathrm{d}_n^H}{n^{-\log n}} = O(1) \text{ (a.s.).}$$

Thus, in $O(n\log n)$ time this method guarantees a rate of convergence that is superpolynomial, yet subexponential. It is tempting to conjecture that this is best possible for a polynomial time heuristic.

We have dealt with this simple example since it exhibits many of the ingredients typically encountered in a probabilistic analysis:

- *a combinatorial problem* that may be difficult to solve (the PARTITION problem is NP-complete);
- *a probability distribution* over all problem instances to generate problem data as realizations of independent and identically distributed (i.i.d.) random variables;
- *a probabilistic value analysis* that yields an asymptotic characterization of the optimal solution value as a simple function of the problem data ($\mathrm{z}_n^{OPT}/(nE\mathbf{a}_1/2)\to 1$ (a.s.));
- *a probabilistic error analysis* of a fast heuristic to prove that its relative or absolute error tends to 0 with increasing problem size in some stochastic sence:
- *a rate of convergence analysis* that yields some indication of how large problem sizes might have to be in order to demonstrate asymptotic behaviour in practice. Moreover, it allows for further differentation among the heuristics.

Similar analyses have been carried out for many other combinatorial problems. An example is provided by the MULTIKNAPSACK problem:

$$\max\{\sum_{j=1}^{n} c_j x_j | \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i \ (i=1,...,m), x_j \in \{0,1\} \ (j=1,...,n)\}.$$

Let us assume that the $(m+1)$-dimensional vectors $(\mathrm{c}_j, \mathrm{a}_{1j},..., \mathrm{a}_{mj})$ are i.i.d. with bounded support and that $b_i = n\beta_i$, with β_i fixed. As above we are interested in the optimal solution value as a function of $\beta = (\beta_1,...,\beta_m)$ and in heuristics whose error vanishes asymptotically with high probability.

The analysis of this problem in [38] and [15] is of interest in that it exploits the close relationship (in a probabilistic sence) between a difficult nonconvex combinatorial optimization problem and its convex LP relaxation, obtained by replacing the constraints $x_j \in \{0,1\}$ by $0 \leqslant x_j \leqslant 1$ $(j=1,...,n)$. It is easy to verify that the absolute difference between the solution values of these two problems is bounded by $(\max_{j=1,...,n} \mathrm{c}_j)m$, so that the relative error that is made by focussing on the LP relaxation goes to 0. But the LP relaxation (or rather, its dual) is much easier to analyze: its value is given by

$$\min_\lambda \mathrm{L}_n(\lambda),$$

with $\lambda = (\lambda_1,...,\lambda_m)$ and

$$\mathrm{L}_n(\lambda) = \sum_{i=1}^{m} \lambda_i b_i + \max\{\sum_j (\mathrm{c}_j - \sum_{i=1}^{m} \lambda_i \mathrm{a}_{ij}) x_j | 0 \leqslant x_j \leqslant 1\}. \tag{2.3}$$

The maximization problem in (2.3) is solved by setting

$$x_j = \mathrm{x}_j(\lambda) = \begin{cases} 1 & \text{if } \mathrm{c}_j - \sum_{i=1}^{m} \lambda_i \mathrm{a}_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

The strong law of large numbers implies that $\mathrm{L}_n(\lambda)/n$ is asymptotically equal to its expectation

$$L(\lambda) = \sum_{i=1}^{m} \lambda_i \beta_i + E\mathrm{c}_1 \mathrm{x}_1(\lambda) - \sum_{i=1}^{m} \lambda_i E\mathrm{a}_{i1} \mathrm{x}_1(\lambda),$$

for every value of $\lambda > 0$. To establish the asymptotic characterization of the optimal solution value the above convergence must be shown to hold uniformly over the class of functions $\mathrm{L}_n(\lambda)$ induced by all possible (interesting) values of λ.

In [15] such a uniform strong law of large numbers is obtained by using elements from empirical process theory that essentially relate uniform convergence to combinatorial properties of the class of functions studied. It then follows almost immediately that the optimal solution value z_n^{OPT} satisfies

$$\frac{\mathrm{z}_n^{OPT}}{n} \rightarrow \min_\lambda L(\lambda) \text{ (a.s.).}$$

The minimum of $L(\lambda)$ can actually be computed in closed form in some simple cases (e.g. for $m=1$).

A particularly nice feature of the above use of empirical process theory is that it yields, next to a uniform law of large numbers under slightly stronger conditions on the probability distribution, a *uniform law of the iterated logarithm*, which establishes a rate of convergence:

$$(\frac{n}{\log\log n})^{1/2}|\frac{z_n^{OPT}}{n} - \min_\lambda L(\lambda)| = O(1) \text{ (a.s.)},$$

[15].

The type of argument used suggests that it might be applicable to other combinatorial optimization problems with a minsum-objective and a continuous relaxation whose optimal solution value is relatively close to the integer one. And indeed, the approach has been successful in analyzing the SETCOVER problem [41], the k-MEDIAN location problem [45] and the UPWARD MATCHING problem [43].

A heuristic to solve the MULTIKNAPSACK problem, which is suggested by the above analysis , is the *generalized greedy heuristic* in which x_j's are set equal to 1 in order of nonincreasing ratio's $c_j/\Sigma_{i=1}^m\lambda_i a_{ij}$. If the λ_i's are chosen to be equal to the values minimizing $L(\lambda)$, then the relative error of this greedy method goes to 0 a.s. A heuristic whose absolute error vanishes asymptotically is not known for the MULTIKNAPSACK problem, however, and further analysis of the model reveals puzzling differences between the minimization and maximization version of MULTIKNAPSACK that still have to be resolved. Nonetheless, the probabilistic analysis of this model yields surprisingly high returns.

For another famous number problem, the BIN PACKING problem, a different type of probabilistic analysis has been carried out. The problem is to find the minimum number of unit size bins required to pack a collection of items with sizes $a_1,...,a_n$. Under the assumption that the items sizes are i.i.d. uniform on [0,1], probabilistic analysis from the perspective illustrated above has yielded many beautiful results (see e.g. [8], [29]). In [44] the question is considered which probability distribution on [0,1] from which the item sizes are drawn independently allow *perfect packing* of the bins. More formally, let z_n^{OPT} denote the optimal number of bins required to pack items of size $a_1,...,a_n$, which are i.i.d. random variables with distribution μ on [0,1]. Since z_n^{OPT} is subadditive, in the sense that

$$z_{k+l}^{OPT}(a_1,...,a_k,a_{k+1},...,a_{k+l}) \leqslant z_k^{OPT}(a_1,...,a_k) + z_l^{OPT}(a_{k+1},...,a_{k+l}),$$

it is known (cf. [28]) that there exists a constant $c(\mu)$, depending on the distribution μ, such that

$$\frac{z_n^{OPT}}{n} \rightarrow c(\mu) \text{ (a.s.)}.$$

It is clear that $z_n^{OPT} \geqslant \Sigma_{j=1}^n a_j$, and that $\Sigma_{j=1}^n a_j/n \rightarrow Ea_1$ (a.s.) by the strong law of large numbers. Thus $c(\mu) \geqslant Ea_1$. Now, answering the question which distributions μ allow perfect packing, amounts to characterizing the class of distributions μ for which $c(\mu) = Ea_1$.

Necessary and sufficient conditions on μ have been derived in [44]. We present here the implications of this general result for the intervals of uniform distributions, thereby answering a question posed for the first time in [21]. Consider the uniform distributions $\mu_{a,b}$ over intervals $[a,b]$ with $0 \leqslant a < b \leqslant 1$. Let us assume first that $0 \leqslant a < b \leqslant 1/2$. Then $\mu_{a,b}$ allows perfect packing of items whose size are drawn independently from it if and only if either $a+b=2/p$ for some integer $p \geqslant 3$, or if there exists an integer $p \geqslant 3$ such that $a \leqslant 1/(p+1) < 1/p \leqslant b$ and $b-a \geqslant 2/(p(p+1))$. If $0 \leqslant a < b \leqslant 1$ and $b > 1/2$ then $\mu_{a,b}$ allow perfect packing if and only if $a+b \leqslant 1$ and $\mu_{a,1-b}$ does (see [46]).

So far, all examples of number problems focused on the solution value and not on the running time of an algorithm as the random variable to be analyzed. However, one of the first results in this field of research involved the latter type of analysis. The problem in question is a number problem, and indeed probably the most famous number problem, LINEAR PROGRAMMING. This problem is well known to be solvable in polynomial time. Probabilistic analysis played a vital role in understanding the excellent practical performance (as compared to the exponential worst case performance) of the *simplex method*. The venerable history of the average case performance analysis of the simplex method illustrates the importance of an appropriate *probabilistic model*: ultimately, the concept of a random polytope being generated by m fixed hyperplanes in $\mathbf{R}^n$ and m coin flips to determine the direction of the corresponding inequalities reduced the computation of the average number of pivot steps to a combinatorial counting question. Within this model, various simplex variants admit of a quadratic upper bound on the expected number of iterations (including those in Phase 1), which takes us very close to the behaviour observed in practice ([18,1,2,49]).

In the next section we turn to problems with a geometric flavour, whose probabilistic analysis involves random sets of points in Euclidean space.

3. GEOMETRICAL PROBLEMS

In this section we consider problems, whose input includes n points in some finite dimensional metric space. We restrict ourselves here to problems in the Euclidean plane. The most famous problem of this type is surely the TRAVELING SALESMAN problem, which is to find the shortest tour connecting n points. It has a respectable history, of which its probabilistic analysis forms one of the most recent chapters (see [33]).

To facilitate the exposition we assume that the points are uniformly distributed over a fixed region, e.g. the unit square. Under such an assumption it is not difficult to arrive at an intuitive probabilistic value analysis of the TRAVELING SALESMAN problem. For large n, an optimal tour through the 2×2 square with $4n$ points will be approximately 4 times as long as an optimal tour in a unit square with n points. Scaling back the 2×2 square to a unit one, we conclude that the optimal tour length z_n^{OPT} is likely to grow proportionally to $\sqrt{n}$. Indeed, a heuristic from [7] shows that its value is bounded deterministically from above by $\sqrt{2n}$. And an argument based on expected nearest neighbour distances easily shows that there exists a positive constant c such that z_n^{OPT} is a.s. bounded by $c\sqrt{n}$ from below.

The actual convergence argument is much more difficult and was first provided in [5]: as expected

$$z_n^{OPT}/\sqrt{n}\rightarrow\beta \text{ (a.s.)}, \tag{3.1}$$

where β is a constant that has been estimated empirically to be equal to 0.765.

This result has been rederived by other researchers. A recent proof in [47] simplifies a proof in [48] considerably. It involves a technique useful in a broader context. Martingale inequalities are used as a basic ingredient. First, the asymptotics of the expectation Ez_n^{OPT} are determined. In particular, it is shown that

$$Ez_n^{OPT}=\phi(n)\sim\beta\sqrt{n}.$$

The argument for this result relies on subadditivity and self-similarity properties of the functional z_n^{OPT}, and embody the intuitive insight mentioned above. The unit square is divided into m^2 equal size subsquares. The expected length of the shortest tour in each of the subsquares is approximately $\frac{1}{m}\phi(\frac{n}{m^2})$. Sewing together all these shortest subtours can be done at an incremental cost of at most $3m$, so that $\phi(n)$ is bounded by $m\phi(\frac{n}{m^2})+3m$. Letting n and m go to infinity in a suitable way it can be proved that

$$\liminf_{n\rightarrow\infty}\phi(n)/\sqrt{n} = \limsup_{n\rightarrow\infty}\phi(n)/\sqrt{n}$$

and that this limit exists [48].

Once convergence has been established, the next step is to prove that

$$|z_n^{OPT}-Ez_n^{OPT}|/\sqrt{n}\rightarrow 0 \text{ (a.s.)}.$$

Let Σ_i denote the σ-field generated by the first i points $\{X_1,...,X_i\}$ from an infinite sequence, $i=1,2,...,n$ and let z_n^i denote the length of the shortest tour through the points $\{X_1,...,X_{i-1},X_{i+1},...,X_n\}$. Two nice properties of z_n^i are that $E(z_n^i|\Sigma_i)=E(z_n^i|\Sigma_{i-1})$ and $E(z_n^i|\Sigma_{i-1})=E(z_n^{OPT}|\Sigma_{i-1})$. Thus we have

$$\begin{aligned}d_i &= E(z_n^{OPT}|\Sigma_i)-E(z_n^{OPT}|\Sigma_{i-1})\\ &= E(z_n^{OPT}-z_n^i|\Sigma_i)-E(z_n^{OPT}-z_n^i|\Sigma_{i-1})\\ &= E(z_n^{OPT}-z_n^i|\Sigma_i).\end{aligned}$$

The sequence $\{d_i\}_{i=1}^n$ is called a martingale difference sequence, and we can write $z_n^{OPT}-Ez_n^{OPT}=\Sigma_{i=1}^n d_i$. For any such a sequence the following inequality holds:

$$Pr\{|\Sigma_{i=1}^n d_i|>t\}\leq 2\exp(-t^2/(2\sum_{i=1}^{n}\|d_i\|_\infty^2)), \tag{3.2}$$

for each $t>0$, where $\|d_i\|_\infty$ is the supnorm of d_i. We bound $\|d_i\|_\infty$ by $2\min_{j\neq i}d(X_i,X_j)$ by considering the insertion of X_i in a shortest tour through $\{X_1,...,X_{i-1},X_{i+1},...,X_n\}$. An appropriate analysis of the bound, conditional on the σ-fields, yields

$$\|d_i\|_\infty \leqslant 2C(n-i-1)^{-1/2}$$

for some constant C. This together with inequality (3.2), with $t=\epsilon\sqrt{n}$, yields

$$Pr\{|z_n^{OPT}-Ez_n^{OPT}|>\epsilon\sqrt{n}\}\leqslant 2\exp(-\epsilon^2 n/(C\ln n))$$

for any $\epsilon>0$. The Borel-Cantelli lemma concludes the argument, since obviously

$$\sum_{n=1}^{\infty} Pr\{|z_n^{OPT}-Ez_n^{OPT}|>\epsilon\sqrt{n}\}<\infty.$$

In [48] the result (3.1) is generalized to problems of arbitrary dimension, whose objective function can be viewed as a functional that is subadditive and has Euclidean properties (provided that a few other technical conditions are satisfied). The TRAVELING SALESMAN problem is one example, the MATCHING PROBLEM is another one.

Given the result of this probabilistic value analysis it now becomes attractive to search for heuristics whose absolute error is $o(\sqrt{n})$; their relative error then goes to 0 almost surely. As in the case of other Euclidean problems, *partitioning algorithms* do precisely that. Generally, in these heuristics, the region is appropriately partitioned into subregions, subproblems defined by the points in each subregion are analyzed separately, and a feasible solution to the problem as a whole is composed out of the separate results.

For the TRAVELING SALESMAN problem in the square, one partitioning approach is to execute an alternating sequence of horizontal and vertical *cuts* through the point with current median vertical and horizontal coordinate respectively, until the resulting rectangles contain no more than $\sqrt{\log n}$ points. Each of these subproblems is solved to optimality by some enumerative technique (say, dynamic programming) which takes $O(n^\epsilon)$ time per rectangle, and hence $O(n^{1+\epsilon})$ time overall, for some $\epsilon>0$. The resulting tours define a connected graph with even degree at each point; the *Euler walk* that visits each edge of this graph can be converted into a simple cycle of no greater length by eliminating all multiple visits. The difference between the length of this tour and the optimal one can be shown to be of the same order as the total perimeter of the subrectangles generated, which is easily seen to be $O(\sqrt{n})$ in this case. Thus, the relative error of the heuristic goes to 0 a.s. ([24,27]).

Not much is known about the rate of convergence to optimality of this heuristic, nor is any heuristic known whose absolute error goes to 0 asymptotically.

We close this section by discussing an interesting class of problems that cannot quite be handled by Steele's techniques. LOCATION problems are problems in which k depots have to be located so as to minimize the average distance between points of a fixed set and their closest depot (the k-MEDIAN problem) or the maximum of these distances (the k-CENTER problem). The probabilistic value analysis for both problems, leads, surprisingly, to similar results: provided that $k=O(n/\log n)$ both optimal solution values are asymptotically proportional to $1/\sqrt{k}$, albeit for different constants of proportionality ([20,51]).

The analysis leading to these results is of special interest, since it relies on the similarity between the original (*discrete*) problem for large n and the *continuous* problem in which customer demand is not concentrated in a finite number of points but spread uniformly and continuously over the entire region. For the k-MEDIAN problem, a rate of convergence has been established using empirical process theory (cf. Section 2) as a technical tool ([45]).

A simple partitioning heuristic in which a depot is located in each of k nonempty subsquares of size $1/\sqrt{k}$ by $1/\sqrt{k}$ already provides an $O(1/\sqrt{k})$ upper bound on the optimal solution values of both problems. An asymptotically optimal heuristic, however, is only obtained by partitioning the region into regular *hexagons* (the *honeycomb* heuristic), with the constant of proportionality being determined by the optimal solution value of the continuous problem with $k=1$ over one such a hexagon. This heuristic actually solves the continuous problem to optimality, and a detailed error analysis shows that, for n sufficiently large, its relative error in the discrete case becomes vanishingly small ([19,51]).

4. GRAPHS AND NETWORKS

We now turn to the rich area of combinatorial optimization problems defined on graphs and networks. One of the reasons for the wide variety of probabilistic results for this class of problems is the existence of a substantial theory dealing with *random graphs*. There are two definitions of this concept: $G_{n,p}$ is defined to be the (undirected) graph on n vertices for which each of the $n(n-1)/2$ edges occurs independently with equal probability p; G_n^N is defined as the graph on n vertices with N edges where each of the $\binom{n(n-1)/2}{N}$ undirected graphs occurs with equal probability. We refer to [39] for a survey of the theory. Especially for *structural graph optimization* problems, in which we are interested in graph properties that depend only on the node-edge incidence structure, random graphs provide a natural distribution over the set of all problem instances of size n.

Continuing in the spirit of the previous two sections we again refer to [26] for a list of references in the area and review only a few typical probabilistic analyses of heuristics for NP-complete problems. In doing so, we (reluctantly) exclude many beautiful results on problems of CONNECTIVITY and MATCHING that can be solved in worst case polynomial time.

A typical example of a difficult structural problem is the CLIQUE problem of finding a complete subgraph of G that has maximal size $\omega(G)$. To carry out a probabilistic value analysis of $\omega(G_{n,p})$ for fixed p, we observe that the expected number of cliques of size k in such a graph is equal to $\binom{n}{k}p^{k(k-1)/2}$. We would expect the maximal clique size k to occur when the expected number is approximately equal to 1, i.e. when (from Stirling's approximation of $k!$)

$$\frac{1}{\sqrt{2\pi k}}(\frac{nep^{(k-1)/2}}{k})^k \approx 1.$$

The left-hand side decreases very rapidly as k increases and passes through the value 1 when

$$\frac{nep^{(k-1)/2}}{k} = 1,$$

i.e., when

$$k = 2\log_{1/p}n + 2\log_{1/p}e - 2\log_{1/p}k + 1,$$

so that, approximately, $k \approx k(n,p)$ with

$$k(n,p) = 2\log_{1/p}n - 2\log_{1/p}\log_{1/p}n + 2\log_{1/p}(e/2) + 1.$$

This estimate turns out to be very sharp indeed. In [36] it is proved that for every $\epsilon > 0$,

$$\lim_{n \to \infty} Pr\{[k(n,p) - \epsilon] \leq z_n^{OPT} = \omega(G_{n,p}) \leq [k(n,p) + \epsilon]\} = 1,$$

so that, for large enough n, the size of the largest clique can be predicted to be one of two consecutive integers with high reliability.

This precise probabilistic value analysis again encourages the search for a fast heuristic whose absolute error compares favorably to $2\log_{1/p}n$. Consider, for instance, the sequential greedy algorithm, which considers the vertices of G in arbitrary order and adds a vertex to the current clique if it is adjacent to all its members. For an analysis of the performance of this method , one observes that the expected number of trials to increase the clique size from j to $j+1$ is $1/p^j$, so that we might guess the ultimate clique size z^H to satisfy

$$\sum_{j=0}^{z^H-1} 1/p^j = \frac{1 - 1/p^{z^H}}{1 - 1/p} = n,$$

i.e.,

$$Z^H \approx \log_{1/p}n.$$

A more precise analysis shows that, indeed, this greedy approach a.s. yields a clique of size $(1/2 - \epsilon)z_n^{OPT}$ [17]. Thus, the relative error does not go to 0, but is almost surely close to 50 percent. (There is no known polynomial time heuristic with any constant worst case bound on the relative error.)

The above result has immediate implications for the problem to find the INDEPENDENT SET of G of maximal size; it coincides with the maximal size clique in the complement of G. Again the sequential greedy approach, which picks up each successive vertex that is not adjacent to any member of the current independent set, produces an independent set whose size is a.s. close to 50 percent of the optimal value. The COLORING problem, which is to partition the vertices of G into the smallest possible number $\chi(G)$ of independent sets, is much harder to analyze: the asymptotic optimal solution value $z_n^{OPT} = \chi(G_{n,p})$ is known for $p = 1/2$ ([31], though this (Russian) announcement has not been verified). The heuristic method, which greedily finds an independent set as above, deletes it and repeats on the remaining graph, does poorly ([37]) but well enough to get within a factor $2+\epsilon$ a.s. ([17]).

Another class of structural graph problems for which probabilistic analysis has been successful is the HAMILTONIAN CIRCUIT problem of searching for a simple cycle containing all vertices. In [30] the conditions under which such a cycle exists have been established: in particular it was proved that for random graphs G_n^N there is a threshold value for the number of edges $N=\frac{1}{2}n\log n+\frac{1}{2}n\log\log n+cn$ above which a Hamiltonian circuit exists in G_n^N with high probability and below which it does not exist with high probability.

A heuristic that is successful in finding a Hamiltonian circuit with probability, if one exists, within polynomial time is presented in [3]. The general idea of this so-called *extension-rotation* algorithm is as follows. Given a path of vertices $\{v_0,...,v_k\}$ one of the neighbours of v_k, say w is selected at random and the edge (v_k, w) is added to the path. If w is not in the path, it is adjoined to it, so that we obtain a path that contains one edge more than the previous one (extension), and the method is applied to w. If $w\in\{v_0,...,v_{k-1}\}$, say $w=v_l$, then the edge (v_l,v_{l+1}) is removed from the path and the method is applied to v_{l+1} (rotation). If N exceeds the threshold presented before by a sufficient amount (e.g. $N=cn\log n$ for large enough c) this method will be successful with high probability. We notice that the above heuristic is different from those considered before in that here the steps are partially dictated by a random mechanism. We will return to these so called *randomized algorithms* in the following section.

We now turn briefly to number problems on weighted graphs, i.e., graphs with weights on the edges, an area which mixes features addressed in Section 2 with the theory of random graphs. Here most results refer to problems that admit a worst case polynomially bounded algorithm like the LINEAR ASSIGNMENT problem ([50,10]), the MINIMUM SPANNING TREE problem ([13]) and the SHORTEST PATH problem ([40]).

An NP-complete problem that belongs to this class is again the TRAVELING SALESMAN problem this time defined on an undirected graph. In [14] a random weighted graph model is presented together with an algorithm that solves problems generated from this model exactly with high probability. The problems are defined on a complete graph with n vertices, and for each edge a weight is drawn independently uniform from the integers in the interval $[0,n/\sqrt{\log n}]$. By choosing the weights in this way, the graphs will have (with high probability) enough edges with weight 0 to make the algorithm work optimally but on the other hand not enough to have a Hamiltonian cycle of length 0. The algorithm works in two steps. The idea of the first step is to select the set v_0 of vertices that have a low degree of incidence to edges with weight 0, and then to construct a partitioning of the graph into vertex disjoint paths of smallest possible total length such that each of the vertices in v_0 is an inner point of one of these paths. That this is a lower bound on z_n^{OPT} is shown simply by deleting from an optimal tour all those edges that join two vertices not in v_0. An optimal set of paths can be constructed in polynomial time if the graph generated satisfies certain conditions. But these conditions are satisfied by the class of random graphs defined with high probability. Thus, the endpoints of the paths are vertices, each of which is incident to enough edges with weight 0 to allow the transformation of the paths obtained into a Hamiltonian tour in the second step, using only edges of length 0. This phase is performed by a deterministic extension-rotation algorithm; it will (with high probability) produce such a Hamiltonian circuit in polynomial time. Since for this phase only edges of weight 0 are used, the value produced in the first phase, which was a lower bound on z_n^{OPT}, must be optimal.

For the Asymmetric traveling salesman problem defined on a complete directed graph a heuristic that patches the subcycles appearing in the linear assignment relaxation together, achieves a relative error going to 0 in expectation (see [25]).

Perhaps the most peculiar result has been obtained for a generalization of the above problem, the Quadratic assignment problem;

$$\max\{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} c_{ij}d_{kl}x_{ij}x_{kl} |$$

$$\sum_{i=1}^{n} x_{ij} = 1\,(j=1,\ldots,n),\ \sum_{j=1}^{n} x_{ij}(i=1,\ldots,n), x_{ij} \in \{0,1\}\}.$$

In [6] and in [9] it is shown that for this problem with c_{ij} and d_{kl} i.i.d., the ratio of the best and the worst possible solution value tends to 1 in probability. It shows an unexpected side benefit of probabilistic analysis, in that it clearly indicates how not to generate test problems for an empirical analysis of heuristic solution methods!

5. Concluding remarks

Our first conclusion has to be that many topics of interest in this area have hardly been covered. A good example, announced in the previous section, is the work on *randomized algorithms*. These algorithms contain random steps; i.e. steps whose result depends partially on a random mechanism. Therefore, the solution value and the running time become random variables, even on fixed problem instances. An example of such an algorithm, presented before (cf. Section 4) is the extension-rotation algorithm for finding a Hamiltonian circuit in a graph.

Historically, Primality testing was the first successful algorithmic application of randomization. In [42], a number is submitted to k tests and declared to be prime if it passes all of them, with the probability of it being composite none the less being equal to 2^{-k}. Such an algorithm is called a *Monte Carlo method*, in contrast to a *Las Vegas method* in which the algorithm never produces an incorrect answer, but may, with small probability, produce no answer at all. The method for Graph isomorphism in [4] is of the latter type. These two examples are of special interest in that they concern two problems whose computational complexity (polynomially solvable or NP-complete) is still unknown.

The formal study of randomized algorithms is far from complete, and in particular the real power of randomization techniques remains a mysterious issue; for instance, it is not clear to what extent (if any) the class of problems that can be solved in randomized polynomial time (i.e. fast with high reliability) strictly includes the class of problems that can be solved in worst case polynomial time. A recent annotated bibliography ([35]) provides a useful survey of the area.

It should be clear by now that the area of probabilistic analysis still harbors many interesting research challenges. The purpose of the preceding sections has, again, not been to provide an exhaustive review, but to provide some typical examples that convey the flavour of this area. They have ranged from the very complete insight we have into various solution methods for the PARTITIONING problem to the less satisfactory state of the art for the CLIQUE and the COLORING problems. Clearly a lot of problems and a lot of algorithms await investigation. It is not hard to formulate open questions for probabilistic analysis; so far, however, it has turned out to be quite hard to come up with satisfactory answers for any but the simplest heuristics.

A particularly fascinating possibility is the development of a complexity theory that would lead to a class of problems for which solution to optimality in polynomial expected time is as unlikely as the equality of P and NP. A first step in that direction can be found in [34], where a TILING problem is introduced, together with a probability distribution over its problem instances, such that any other problem with a (mildly restricted type of) probability distribution is reducible to the TILING problem.

To establish completeness for other problems in this class is a major challenge of considerable interest. After all, the reasonable average behaviour of enumerative methods (and the remarkable success of a nonenumerative method based on computations in an integer lattice ([32])) to solve some NP-complete problems, and the apparent impossibility to find such algorithms for other NP-complete problems still defy theoretical explanation!

REFERENCES

1. I. ADLER, R.M. KARP, R. SHAMIR (1983). *A simplex variant solving an $m \times d$ linear program in $O(\min(m^2,d^2))$ expected number of pivot steps.* Report UCB CSD 83/158, Computer Sience Division, University of California, Berkeley.
2. I. ADLER, M. MEGGIDO (1983). *A simplex algorithm whose average number of steps is bounded between two quadratic functions of the smaller dimension.* Technical report, University of California, Berkeley.
3. D. ANGLUIN, L.G. VALIANT (1979). Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Science* 19, 155-193.
4. L. BABAI (1979). *Monte Carlo algorithms in graph isomorphism testing.* Rapport de Recherches DMS 79-10, Départment de Mathématique et de Statistique, Université de Montréal.

5. J. Beardwood, J.H. Halton, J.M. Hammersley (1959). The shortest path through many points. *Proceedings of the Cambridge Philosophical Society* 55, 299-327.
6. R.E. Burkhard, U. Fincke (1982). Probabilistic asymptotic properties of the quadratic assignment problem. *Zeitschrift für Operations Research* 27, 73-81.
7. L. Few (1955). The shortest path and the shortest road through *n* points. *Mathematika* 2, 141-144.
8. G.N. Frederickson (1980). Probabilistic analysis of simple one- and two-dimensional bin packing algorithms. *Information Processing Letters* 11, 156-161.
9. J.B.G. Frenk, M. van Houweninge, A.G.H. Rinnooy Kan (1985). Asymptotic properties of the quadratic assignment problem. *Mathematics of Operations Research* 10, 100-116.
10. J.B.G. Frenk, A.H.G. Rinnooy Kan (1984). *Order statistics and the linear assignment problem.* Report, Econometric Institute, Erasmus University, Rotterdam.
11. J.B.G. Frenk, A.H.G. Rinnooy Kan. (1987). The asymptotic optimality of the LPT heuristic. *Mathematics of Operations Research 12,* 241-254.
12. J.B.G. Frenk, A.H.G. Rinnooy Kan. On the rate of convergence to optimality of the LPT rule. *Discrete Applied Mathematics,* to appear
13. A.M. Frieze (1982). *On the value of a random minimum spanning tree problem.* Technical report, Department of Computer Science and Statistics, Queen Mary College, University of London.
14. A.M. Frieze (1987). On the exact solution of random traveling salesman problems with medium size integer coefficients. *SIAM Journal on Computing 16,* 1052-1072.
15. S. van de Geer, L. Stougie (1988). *On the rate of convergence of the multiknapsack value function.* Report, CWI, Amsterdam.
16. R.L. Graham (1969). Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics* 17, 263-269.
17. G.Grimmett, C.J.H. McDiarmid (1975). On colouring random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society* 77, 313-324.
18. M. Haimovich (1983). *The simplex method is very good! On the expected number of pivot steps and related properties of random linear programs.* Report, Columbia University, New York.
19. M. Haimovich (1984). *PhD. Thesis,* M.I.T., MA.
20. D. Hochbaum, J.M. Steele (1981). Steinhaus's geometric location problem for random samples in the plane. *Advances in Applied Probability* 14, 56-67.
21. N. Karmarkar (1982). Probabilistic analysis of some bin-packing problems. *Proceedings 23rd Annual Symposium on the Foundations of Computer Science,* 107-111.
22. N. Karmarkar, R.M. Karp. The diffencing method of set partitioning. *Mathematics of Operations Research,* to appear.

23. N. Karmarkar, R.M. Karp, G.S. Lueker, A. Odlyzko (1984). *The probability distribution of the optimal value of a partitioning problem.* Report, Bell Laboratories, Murray Hill, NJ.
24. R.M. Karp (1977). Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane. *Mathematics of Operations Research* 2, 209-224.
25. R.M. Karp (1979). A patching algorithm for the nonsymmetric traveling salesman problem. *SIAM Journal on Computing* 8, 561-573.
26. R.M. Karp, J.K. Lenstra, C.J.H. McDiarmid, A.G.H. Rinnooy Kan (1985). Probabilistic analysis. In M. O'hEigeartaigh, J.K. Lenstra, A.H.G. Rinnooy Kan (eds.), *Combinatorial optimization: Annotated bibliographies.* Wiley, Chichester, 52-88.
27. R.M. Karp, J.M. Steele (1985). Probabilistic analysis of heuristics. In E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (eds.), *The traveling salesman problem.* Wiley, Chichester, 181-206.
28. J.F.C. Kingman (1976). *Subadditive processes.* Lecture notes in Mathematics 539, Springer-Verlag, Berlin, 168-222.
29. W. Knödel (1981). A bin packing algorithm with complexity $O(n\log n)$ and performance 1 in the stochastic limit. In J. Gruska, M. Chytil (eds.), *Mathematical Foundations of Computer Science,* Lecture Notes in Computer Science 118, Springer, Berlin, 369-378.
30. J. Komlós, E. Szemerédi (1983). Limit distribution for the existence of Hamilonian cycles in random graphs. *Discrete Mathematics* 43, 55-63.
31. A.D. Korsunov (1980). The chromatic number of n-vertex graphs. *Metody Diskret. Analiz.* 35, p 14-44 (in Russian).
32. J.C. Lagarias, A.M. Odlyzko (1983). Solving low-density subset sum problems. *Proceedings of the 24th Annual IEEE Symposium on the Foundations of Computer Science,* 1-10.
33. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (eds.), (1985). *The traveling salesman problem.* Wiley, Chichester.
34. L.A. Levin (1984). Problems, complete in 'average' instance. *Proceedings of the 16th Annual ACM Symposium on the Theory of Computing,* 465.
35. F. Maffioli, M.G. Speranza, C. Vercellis (1985). Randomized algorithms. In M. O'hEigeartaigh, J.K. Lenstra, A.H.G. Rinnooy Kan (eds.), *Combinatorial optimization: Annotated bibliographies.* Wiley, Chichester, 89-105.
36. D.W. Matula (1976). *The largest clique size in a random graph.* Technical report CS7608, Department of Computer Science, Southern Methodist University, Dallas, TX.
37. C.J.H. McDairmid (1979). Colouring random graphs badly. In R.J. Wilson (ed.), *Graph theory and combinatorics.* Pitman Research Notes in Mathematics 34, Pitman, London, 76-86.
38. M. Meanti, A.H.G. Rinnooy Kan, L. Stougie, C. Vercellis. Probabilistic analysis of the multiknapsack value function. To appear in *Mathematical Programming.*
39. E.M. Palmer (1985). *Graphical evolution.* Wiley, New York.

40. Y. PERL (1977). *Average analysis of simple path algorithms.* Technical report UIUCDCS-R-77-905, Department of Computer Science, University of Illinois, Urbana-Champaign.
41. N. PIERSMA (1988). *Empirical process theory and combinatorial optimization: a survey.* Thesis, University of Amsterdam.
42. M.O. RABIN (1980). Probabilistic algorithms for testing primality. *Journal of Number Theory* 12, 128-138.
43. W.T. RHEE (1986). *On the optimal value of upward matching.* Report, Ohio State University, Columbus.
44. W.T. RHEE (1988). Optimal bin packing with items of random sizes. *Mathematics of Operations Research* 13, 140-151.
45. W.T. RHEE, M. TALAGRAND (1986). *A concentration inequality for the k-median problem.* Report, Ohio State University, Columbus.
46. W.T. RHEE, M. TALAGRAND (1986). *Some distributions that allow perfect packing.* Report, Ohio State University, Columbus.
47. W.T. RHEE, M. TALAGRAND (1987). Martingale inequalities and NP-complete problems. *Mathematics of Operations Research* 12, 177-181.
48. J.M. STEELE (1981). Subadditive Euclidean functionals and nonlinear growth in geometric probability. *Annals of Probability* 9, 365-376.
49. M.J. TODD, (1983) *Polynomial expected behaviour of a pivoting algorithm for linear complementarity and linear progamming problems.* Technical report 595, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY.
50. D.W. WALKUP (1979). On the expected value of a random assignment problem. *SIAM Journal on Computing* 8, 440-442.
51. E. ZEMEL (1984). Probabilistic analysis of geometric location problems. *Annals of Operations Research* 1, 215-238.

On the strong connectivity problem

A. Schrijver
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

We show that the strong connectivity problem is solvable in polynomial time in case each value α in the distance matrix with $0<\alpha<\infty$ is contained in a submatrix of form $\begin{bmatrix} 0 & 0 \\ \alpha & 0 \end{bmatrix}$ (up to permuting rows or columns), thus extending a result of Lucchesi.

The *strong connectivity augmentation problem* is:

(1) *given*: a directed graph $G = (V,A)$, a length function $l:V\times V\rightarrow\mathbb{Z}_+$ and an integer B,
find: a set $A'\subseteq V\times V$ so that the graph $(V,A\cup A')$ is strongly connected and so that $\Sigma_{a\in A'}l(a)<B$.

(cf. Garey and Johnson [3]). This problem is easily seen to be NP-complete, since the problem of finding a Hamiltonian cycle in a directed graph (V,A'') is reducible to (1): just take $A := \varnothing$, $l:V\times V\rightarrow\mathbb{Z}_+$ defined by:

(2) $$\begin{aligned} l(u,v) &:= 0 \text{ if } (u,v)\in A'', \\ &:= 1 \text{ if } (u,v)\notin A'', \end{aligned}$$

and $B := 1$ (cf. Eswaran and Tarjan [1]).

In fact the *traveling salesman problem*:

(3) *given*: a length function $l':V\times V\rightarrow\mathbb{Z}_+$ and an integer B',
find: a Hamiltonian cycle of length less than B'

is a direct special case of (1) (take $A := \varnothing$, $l(u,v) := l'(u,v)+B'$ and $B := B'|V|+B'$).

Another application of the strong connectivity augmentation problem is the planar feedback arc set problem (see below).

The strong connectivity augmentation problem is trivially equivalent to the *strong connectivity problem*:

(4) *given*: a length function $l:V\times V\rightarrow\mathbb{Z}_+\cup\{\infty\}$ and an integer B,
find: a subset $A'\subseteq V\times V$ so that (V,A') is strongly connected and so that $\Sigma_{a\in A'}l(a)<B$.

Indeed, (4) is just the case $A = \varnothing$ in (1). Conversely, (1) can be reduced to (4) by resetting $l(a) := 0$ whenever $a\in A$. Allowing $l(a) = \infty$ in (4) is irrelevant: we could replace any ∞ by the value B.

We may assume in (4) without loss of generality that for all $i,j,k \in V$:

(5) (i) $l(i,i) = 0$
(ii) if $l(i,j) = 0$ and $l(j,k) = 0$ then $l(i,k) = 0$.

It was shown by Lucchesi [5] (cf. Frank [2] and Lucchesi and Younger [6]) that the strong connectivity problem (4) is solvable in polynomial time if the following condition on the length function holds:

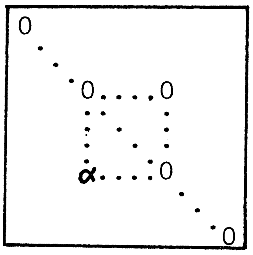
(6) for each $i,j \in V$: if $0 < l(i,j) < \infty$ then $l(j,i) = 0$.

Equivalently, the strong connectivity augmentation problem is solvable in polynomial time if:

(7) for each $i,j \in V$: if $0 < l(i,j) < B$ then $(j,i) \in A$.

So problem (1) is solvable in polynomial time if in the distance table we have that for each value α with $0 < \alpha < \infty$, the symmetric value is equal to 0:

(8)



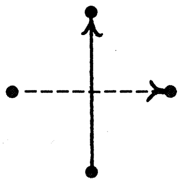
Lucchesi showed that this implies a polynomial-time algorithm for the following *feedback arc set problem*, in case G is planar:

(9) *given*: a directed graph $G = (V,A)$, a length function $l: A \rightarrow \mathbb{Z}_+$ and an integer B,
find: a subset $A' \subseteq A$ so that (V,A') is acyclic and so that $\Sigma_{a \in A'} l(a) > B$.

In general, this problem is NP-complete (Karp [4]).

To see that (9) is solvable in polynomial time if G is planar, we consider the planar dual graph $G^* = (F,A^*)$ of G, directed in such a way that each arc of G crosses its dual arc in G^* 'from left to right':

(10)



(where the uninterrupted arrow is an arc of G, and the interrupted arc is the dual arc in G^*). Define for each pair $(f,g) \in F \times F$:

(11) $$\left.\begin{array}{l} l^*(f,g) := 0 \\ l^*(g,f) := l(a) \end{array}\right\} \quad \left\{\begin{array}{l}\text{if } (f,g) = a^* \in A^*, \text{ where} \\ a^* \text{ is the dual arc of } a \in A,\end{array}\right.$$
$$l^*(f,g) := \infty \qquad \text{for all other pairs } (f,g).$$

Let $B^* := \Sigma_{a \in A} l(a) - B$. Then for each subset A' of A one has:

(12) (V,A') is acyclic $\Leftrightarrow$ $(F, A^* \cup [(A \setminus A')^*]^{-1})$ is strongly connected

(here C^{-1} denotes the set of inverse arcs of C). Moreover,

(13) $$\sum_{a \in A'} l(a) > B \Leftrightarrow \sum_{(g,f) \in [(A \setminus A')^*]^{-1}} l^*(g,f) < B^*.$$

This reduces the planar feedback arc set problem to the strong connectivity problem satisfying (6). Hence it is solvable in polynomial time.

Lucchesi's algorithm can also be used in a branch and bound method to solve the general strong connectivity problem. Typically, during the branching process, a node of the tree is labeled by a set R of 'required' arcs and a set F of 'forbidden' arcs. That is, the node only considers those subsets A' of $V \times V$ for which $R \subseteq A' \subseteq (V \times V) \setminus F$ and for which (V,A') is strongly connected. So the bound corresponding to the node should be a lower bound on the minimum length of these subsets A'.

In order to find such a bound, we can assume that R is reflexive (i.e., $(i,i) \in R$ for all i) and transitive (i.e., if (i,j) and (j,k) belong to R, then $(i,k) \in R$). Moreover, we can reset

(14) $$\begin{array}{ll} l(a) := 0 & \text{if } a \in R, \\ l(a) := \infty & \text{if } a \in F. \end{array}$$

If after this resetting, Lucchesi's condition:

(15) for all $i,j \in V$: if $0 < l(i,j) < \infty$ then $l(j,i) = 0$

is satisfied, Lucchesi's algorithm gives us the exact minimum value (instead of a lower bound) in polynomial time. This suggests that in our branching strategy, we should strive for a situation where (15) holds. That is, for choices of R and F satisfying:

(16) for all $i,j \in V$:$(i,j) \in R$, or $(j,i) \in R$, or both $(i,j) \in F$ and $(j,i) \in F$.

We show that the strong connectivity problem can also be solved in polynomial time if we weaken Lucchesi's condition (15) to:

(17) for all $i,j \in V$: if $0 < l(i,j) < \infty$ then $\exists i', j' \in V$ with
$l(i,i') = l(j',i') = l(j',j) = 0$.

This is indeed weaker than Lucchesi's condition, since if (15) holds we can take $i' = i$ and $j' = j$ in (17).

Condition (17) means that in the distance table we have that any value with $0 < \alpha < \infty$ is part of a 2×2-matrix $\begin{bmatrix} 0 & 0 \\ \alpha & 0 \end{bmatrix}$ as in:

$$\boxed{\begin{matrix}0 & 0\\ \alpha & 0\end{matrix}} \tag{18}$$

So the difference with Lucchesi's condition is that the diagonal elements of $\begin{bmatrix}0 & 0\\ \alpha & 0\end{bmatrix}$ need not be diagonal elements of the distance matrix.

THEOREM. *The strong connectivity problem is solvable in polynomial time if* (17) *is satisfied.*

PROOF. Let l satisfy (18). We may assume furthermore that $l(i,i) = 0$ for all $i \in V$, and if $l(i,j) = l(j,k) = 0$ then $l(i,k) = 0$ for all $i,j,k \in V$.

Suppose now that $0 < l(i,j) < \infty$ for some $i,j \in V$ while $l(j,i) \neq 0$. By (17) there exist $i',j' \in V$ so that $l(j',i) = l(j',i') = l(j,i') = 0$. We introduce two new points, i'' and j'' say. Let $\bar{V} := V \cup \{i'',j''\}$, and

$$\begin{aligned}
&\bar{l}(a,b) := l(a,b) \quad \text{if } a,b \in V, (a,b) \neq (i,j),\\
&\bar{l}(i,j) := \infty\\
&\bar{l}(i,i'') := \bar{l}(i'',i') := \bar{l}(j',j'') := \bar{l}(j',j'') := \bar{l}(j'',j) := 0,\\
&\bar{l}(a,b) := \infty \quad \text{for all other } a,b \in \bar{V}.
\end{aligned} \tag{19}$$

We show that the strong connectivity problem for $\bar{V},\bar{l}$ is equivalent to that for V,l. First, let A be a minimum length subset of $V \times V$ with (V,A) strongly connected. Let:

$$\begin{aligned}
&\bar{A} := A \cup \{(i,i''),(i'',i'),(j'',i''),(j',j''),(j'',j)\} && \text{if } (i,j) \notin A,\\
&\bar{A} := (A \setminus \{(i,j)\}) \cup \{(i,i''),(i'',i'),(j'',i''),(j',j''),(j'',j),(i'',j'')\} && \text{if } (i,j) \in A.
\end{aligned} \tag{20}$$

Clearly,

$$\sum_{a \in A} l(a) = \sum_{a \in \bar{A}} \bar{l}(a). \tag{21}$$

Moreover, $(V,\bar{A})$ is strongly connected. This follows directly from (20) if $(i,j) \notin A$. If $(i,j) \in A$, then $(i,i''),(i'',j''),(j'',j)$ form a path in $\bar{A}$ from i to j. Hence also in this case, $(\bar{V},\bar{A})$ is strongly connected.

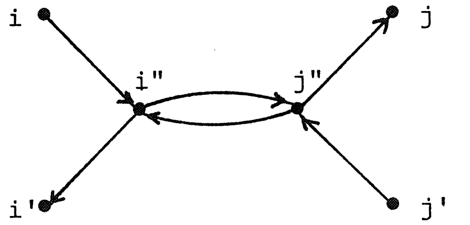
Conversely, let $\bar{A}$ be a minimum length subset of $\bar{V} \times \bar{V}$ with $(\bar{V},\bar{A})$ strongly connected. Without loss of generality, if $l(a,b) = 0$ then $(a,b) \in \bar{A}$. Let:

$$\begin{aligned}
&A := \bar{A} \cap (V \times V) && \text{if } (i'',j'') \notin \bar{A},\\
&A := (\bar{A} \cap (V \times V)) \cup \{(i,j)\} && \text{if } (i'',j'') \in \bar{A}.
\end{aligned} \tag{22}$$

Again (21) holds. Moreover (V,A) is strongly connected. To see this, take

$a,b \in V$. We show that A contains a path from a to b. Since $(\bar{V},\bar{A})$ is strongly connected, $\bar{A}$ contains a path P from a to b. Assume that P passes i'' and j'' as few as possible. If P does not traverse i'' nor j'', it is also a path in A. So supposes P traverses i'' or j''. Consider all arcs incident to i'' or j'' with finite length:

(23)



Since $(i,i'),(j',i'),(j',j) \in A$, and since $(i,j) \in A$ if $(i'',j'') \in \bar{A}$, it follows that P does not intersect $\{i'',j''\}$.

So replacing V,l by $\bar{V},\bar{l}$ gives an equivalent problem, and decreases the number of pairs (i,j) with $0 < l(i,j) < \infty$ and $l(j,i) \neq 0$. Therefore, after at most $|V|^2$ such replacements, we attain an equivalent strong connectivity problem satisfying Lucchesi's condition. This is solvable in polynomial time by Lucchesi's algorithm. □

This theorem suggests that in a branch and bound process, our branching strategy should strive for a situation where the following holds:

(24) for all $i,j \in V$: $(i,j) \in F$, or $(i,i'),(j',i'),(j',j) \in R$ for some $i',j' \in V$.

(The second alternative includes the case $(i,j) \in R$, by taking $i' = j$, $j' = i$.)

References

1. K.P. Eswaran and R.E. Tarjan, *Augmentation problems*, SIAM Journal on Computing 5 (1976) 653-665.
2. A. Frank, *How to make a digraph strongly connected*, Combinatorica 1 (1981) 145-153.
3. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.
4. R.M. Karp, *Reducibility among combinatorial problems*, in: Complexity of Computer Computations (R.E. Miller and J.W. Thatcher, eds.), Plenum Press, New York, 1972, pp. 85-103.
5. C.L. Lucchesi, *A minimax equality for directed graphs*, Ph.D. Thesis, University of Waterloo, Waterloo, Ontario, 1976.
6. C.L. Lucchesi and D.H. Younger, *A minimax theorem for directed graphs*, Journal of the London Mathematical Society (2) 17 (1978) 369-374.

Good probabilistic ideas are often simple

H.C. Tijms
Instituut voor Econometrie
Vrije Universiteit, Amsterdam

The applied fields of inventory and queueing lie at the heart of stochastic operations research. This paper deals with two typical problems from these fields and shows how they can be solved by using simple probabilistic methods.

1. Introduction

Operations Research is a rather young discipline. From the very beginning stochastic inventory and queueing problems have played a prominent role in the development of Operations Research. It is no coincidence that the development of Operations Research took place in the track of a greater acceptance of probabilistic and statistical methods for solving problems from daily life.

Nowadays widely used tools in stochastic operations research are renewal-reward processes and Markov decision processes. The roots of these tools go back to the early sixties. In that time De Leve made already an extensive use of these probabilistic methods in his research and teaching. His approach was always very intuitive and usually led to surprisingly elegant solutions. This fascinated me as a student who was used to formal thinking in probability theory. Chance plays a great role in one's life. My choice for stochastic operations research would not have been made without the motivating education I had in applied probability. A main lesson I learned is that good probabilistic ideas are often simple. In the remainder of this paper I will try to support this claim.

2. The periodic review (R,S) inventory system with restricted order size

2.1. Model

A widely used inventory control system is the periodic review system where at each review a replenishment order is placed for the cumulative demand since the previous review. This control rule assumes that there is no limitation on the size of the replenishment order. In practice this assumption is not always satisfied. In this section we will consider the case of a restricted order size. The inventory control model is as follows. The demands for a single product in the successive periods $t = 1,2,...$ are independent random variables having a common probability density $f(x)$ with mean μ and standard deviation σ. Demand in excess of the stock on hand is backordered until stock becomes available by the delivery of a replenishment order. The inventory position is reviewed every R periods, where R is a fixed positive integer. At each review the inventory position is ordered up to the level S provided that the order size does not exceed Q; otherwise, an amount of Q is ordered. Here Q is a given number, where it is assumed that

$$Q > R\mu.$$

This prevents the inventory position drifting to minus infinity. The goal is to compute the order-up-to-level S so that the following service level constraint is satisfied:

the fraction of demand satisfied directly from stock on hand $\geqslant \alpha$,

where α is a prespecified value (e.g., $\alpha = 0.95$). It will be shown in the next subsection how a computationally tractable method can be obtained using simple and basic probabilistic tools.

2.2. *Analysis*

For ease the analysis will assume that the lead time of any replenishment order is negligible. The outline of the analysis is as follows:

1. For a given control rule (R,S), it will be shown that the process describing the inventory position just after a review is probabilistically equivalent to the waiting-time process in the single-server $D/G/1$ queueing model with deterministic arrivals. The waiting-time distribution for the $D/G/1$ queueing model can be explicitly given for the class of Coxian-2 service time distributions.
2. The smallest order-up-to-level S_α achieving the service level α when the demand distribution is general will be approximated by the corresponding order-up-to-level for the case of a Coxian-2 demand distribution having the same first two or first three moments as the original demand distribution. This approximation requires that the review-time demand has a squared coefficient of variation of at least ½. Otherwise, the order-up-to-level S_α is approximated by linear extrapolation of the corresponding levels for two special Coxian-2 distributions (exponential and Erlang-2) with the same means as the original demand distribution. Here the extrapolation is with respect to the squared coefficient of variation of the review-time demand.

Let us first show that the process describing the inventory just after a review is equivalent to the waiting-time process in a $D/G/1$ queue. Fix an (R,S) control rule and define the random variable

$\Delta_i =$ the difference between the order-up-to-level S and the inventory position just after the i-th review.

Letting the random variable ξ_k be defined by

$$\xi_k = \text{the total demand between the } k\text{-th and } (k+1)\text{-st review,}$$

it follows that the inventory position just prior to the i-th review equals $S-\Delta_{i-1}-\xi_{i-1}$ and that an amount of $\min(Q,\Delta_{i-1}+\xi_{i-1})$ is ordered at the i-th review. It is now easily seen that

$$\Delta_i = \max(0,\Delta_{i-1}+\xi_{i-1}-Q) \quad \text{for } i = 1,2,...,$$

where $\Delta_0 = 0$ (assuming that the initial stock equals S). The same recurrence relation arises for the single-server $D/G/1$ queue in which the deterministic interarrival times of the customers are equal to Q and the service time of the k-th customer is distributed as ξ_k. Assuming service in order of arrival, let

$$W_i = \text{the waiting time of the } i\text{-th customer (excluding service time).}$$

It easily follows that $W_i = W_{i-1}+\xi_{i-1}-Q$ if $W_{i-1}+\xi_{i-1}-Q>0$ and $W_i = 0$ otherwise. This yields the famous Lindley equation,

$$W_i = \max(0, W_{i-1}+\xi_{i-1}-Q) \quad \text{for } i = 1,2,...,$$

where $W_0 = 0$. Consequently, the probability distribution of the 'inventory deficit' Δ_i is the same as that of the waiting-time of the i-th customer in the above $D/G/1$ queue. This observation goes back to De Kok [1] who also devised an interesting approximative method to solve the Lindley equation for the general $GI/G/1$ queue. His method can also be used to obtain a useful approximation for the service level of an (R,S) policy when the demand has a general distribution. However, an alternative approach can be given using the special class of Coxian-2 distributions and the idea of extrapolation with respect to the squared coefficient of variation of the review time demand. A key observation is that the limiting distribution function

$$W(x) = \lim_{i\to\infty} P\{W_i \leqslant x\}, \quad x \geqslant 0,$$

allows for a simple explicit expression when the service times have a Coxian-2 distribution. Before we proceed, let us describe the practically useful class of Coxian-2 distributions.

Coxian-2 distribution

A positive random variable S is said to have a Coxian-2 (C_2) distribution when S can be represented as

$$S = \begin{cases} U_1 & \text{with probability } 1-b, \\ U_1+U_2 & \text{with probability } b, \end{cases}$$

for some branching probability $0 \leqslant b \leqslant 1$, where U_1 are independently distributed exponential random variables with respective means $1/\mu_1$ and $1/\mu_2$. Any C_2 distributed random variable S can be shown to have a squared coefficient of variation of at least ½, where the squared coefficient of variation c_s^2 is defined by

$$c_s^2 = \frac{\sigma^2(S)}{E^2(S)}.$$

It is often convenient to fit a C_2-distribution to a positive random variable by matching its first two or first three moments. Let X be a positive random variable with $c_x^2 \geqslant ½$ and denote by $m_i = E(X^i)$ the i-th moment of X. If a three-moment fit to X by a C_2-distribution exists, the three parameters μ_1, μ_2 and b of this unique fit are given by

$$\mu_{1,2} = \frac{1}{2}[\alpha_1 \pm \sqrt{\alpha_1^2 - 4\alpha_2}\,],\ b = \frac{\mu_2}{\mu_1}(\mu_1 m_1 - 1),$$

where $\alpha_1 = (1 + ½ m_2 \alpha_2)/m_1$ and $\alpha_2 = (6m_1^2 - 3m_2)/(3m_2^2/2 - m_1 m_3)$. An infinite number of C_2-distributions can be fitted to X by matching only the first two moments. An appealing and very useful two-moment fit is the one with parameters

$$\mu_{1,2} = \frac{2}{m_1}\left[1 \pm \left\{\frac{c_x^2 - \frac{1}{2}}{c_x^2 + 1}\right\}^{\frac{1}{2}}\right],\ b = \frac{\mu_2}{\mu_1}(\mu_1 m_1 - 1).$$

This particular C_2-distribution has the same first three moments as a gamma distribution, cf. Tijms [4].

Let us now return to the stationary waiting-time distribution function $W(x)$ for the $D/G/1$ queue. In case the service times ξ_i have a C_2-distribution with parameters μ_1, μ_2, and b, then

$$W(x) = 1 - a_1 e^{-\eta_1 x} - a_2 e^{-\eta_2 x},\quad x \geqslant 0,$$

where η_1 and η_2 with $0 < \eta_1 < \min(\mu_1, \mu_2) \leqslant \eta_2$ are two real zeros of the equation

$$x^2 - (\mu_1 + \mu_2)x + \mu_1\mu_2 - \{\mu_1\mu_2 - (1-b)\mu_1 x\}e^{-xQ} = 0,$$

and the constants a_1 and a_2 are given by

$$a_1 = \frac{-\eta_1^2\eta_2 + \eta_1\eta_2(\mu_1 + \mu_2) - \eta_2\mu_1\mu_2}{\mu_1\mu_2(\eta_1 - \eta_2)},$$

$$a_2 = \frac{\eta_1\eta_2^2 - \eta_1\eta_2(\mu_1 + \mu_2) + \eta_1\mu_1\mu_2}{\mu_1\mu_2(\eta_1 - \eta_2)}.$$

An elementary proof of this result can be found in Van Ommeren and Nobel [6].

Let us next apply the above results to the periodic review inventory model with the (R,S) control rule. For any $v \leqslant S$, let

$$\Pi(v) = \lim_{i\to\infty} P\{\text{the inventory position just after the } i\text{-th review is} \leqslant v\}.$$

Assume for the moment that the review time demands ξ_i have a C_2-distribution with parameters μ_1, μ_2, and b. It is no restriction to assume that $\mu_1 > \mu_2$ (otherwise, redefine μ_1, μ_2, and b as $\mu_1 := \mu_2$, $\mu_2 := \mu_1$, and $b := 1 - (1-b)\mu_1/\mu_2$; then, the same density arises). The probability density of the ξ_i's is easily calculated as

$$f_R(x) = \begin{cases} p\mu_1 e^{-\mu_1 x} + (1-p)\mu_2 e^{-\mu_2 x} & \text{when } \mu_1 > \mu_2, \\ p\mu_1 e^{-\mu_1 x} + (1-p)\mu_1^2 x e^{-\mu_1 x} & \text{when } \mu_1 = \mu_2, \end{cases}$$

where $p = 1-b\mu_1/(\mu_1-\mu_2)$ when $\mu_1 \neq \mu_2$ and $p = 1-b$ when $\mu_1 = \mu_2$.

Since the inventory position after the i-th review equals $S-\Delta_i$ and the limiting distribution of the Markov process $\{\Delta_i\}$ is given by the function $W(x)$ above, it follows that

$$\Pi(v) = \begin{cases} a_1 e^{-\eta_1(S-v)} + a_2 e^{-\eta_2(S-v)} & \text{for } v < S, \\ 1 & \text{for } v = S, \end{cases}$$

where the constants η_1, η_2, a_1, and a_2 are specified above. Note that the probability distribution function $\Pi(v)$ has a mass of $1-a_1-a_2 = \eta_1\eta_2/(\mu_1\mu_2)$ at the point S. The distribution has a density $\pi(v)$ for $v<S$. We can now give a formula for the long-run fraction of demand not satisfied directly from stock on hand. Since it is assumed that the lead time of any replenishment order is zero, the net stock (= onhand stock minus backlog) is the same as the inventory position (= net stock plus stock on order). Thus, using Markov chain theory, it holds true with probability 1 that

the long-run fraction of demand not satisfied directly from stock on hand

$$= \frac{1}{R\mu}\{R\mu I(0) + \int_0^S \pi(v)dv \int_v^\infty (x-v)f_R(x)dx + (1-a_1-a_2)\int_S^\infty (x-S)f_R(x)dx\}.$$

To avoid technicalities, let us next assume that the parameters of the C_2 density of the review time demand satisfy $\mu_1 \neq \mu_2$ as will be usually the case. Then, using the short-hand notation $\alpha(S)$ for the long-run fraction of demand not satisfied directly from stock on hand under the (R,S) policy, we find the analytical expression

$$\alpha(S) = \frac{1}{R\mu}\{R\mu\sum_{i=1}^{2} a_i e^{-\eta_i S} + (1-a_1-a_2)\sum_{i=1}^{2}(p_i/\mu_i)e^{-\mu_i S}$$
$$+ \sum_{i=1}^{2}\sum_{j=1}^{2} p_i a_j \eta_j (e^{-\mu_i S} - e^{\eta_j S})/(\mu_i(\eta_j-\mu_i))\},$$

where $p_1 = p$ and $p_2 = 1-p$. For any specified service level α, Newton-Raphson or bisection can be used to find the order-up-to-level $S = S_\alpha$ for which $\alpha(S) = 1-\alpha$.

For the case of Coxian-2 distributed review time demand, we have a tractable method for the computation of the order-up-to-level achieving a prespecified service level. What to do for the case of a generally distributed review-time demand? To answer this question, we distinguish between two cases with respect to the squared coefficient of variation, c_ξ^2, of the review-time demand.

(a) $c_\xi^2 \geqslant ½$. Then we suggest to fit a Coxian-2 density to the review time demand ξ using a three-moment fit whenever possible or using otherwise a two-moment fit. This approximation step is justified by the empirical finding that the service level of an order-up-to-level policy is rather insensitive to more than the first two moments of the review time demand ξ provided that c_ξ^2 is not too large, the review-time demand density has a 'reasonable' shape (e.g. a unimodal density) and $R\mu/Q$ is not almost 1.

(b) $c_\xi^2 < ½$. Then we suggest to use the following approximation procedure. For a prespecified service level α, compute the order-up-to-levels S_α for the case of an exponentially distributed review-time demand and for the case of an Erlang-2 distributed review-time demand, where in both cases the mean of the review-time demand is equal to the mean $\mu_\xi = R\mu$ of the original distribution. This gives the respective order-up-to-levels $S_\alpha(1)$ and $S_\alpha(½)$. Using the linear interpolation formula

$$f = f_0 \times (x - x_1)/(x_0 - x_1) + f_1 \times (x - x_0)/(x_1 - x_0),$$

the order-up-to-level S_α for the actual review-time demand distribution is next approximated by

$$S_\alpha \approx 2(1 - c_\xi^2)S_\alpha(½) + 2(c_\xi^2 - ½)S_\alpha(1),$$

where the extrapolation is with respect to the squared coefficient of variation, c_ξ^2, of the review-time demand. It is an empirical finding that the linear interpolation approach cannot be applied directly to the service level of a given policy, but works satisfactorily for the critical points S_α provided that c_ξ^2 is not too small (say, $c_\xi^2 \geqslant ⅓$). This interpolation approach using 'percentiles' rather than 'probabilities' is generally useful, cf. Tijms [4].

Numerical illustration

Let us assume that the demand per period has a gamma distribution whose mean μ and squared coefficient of variation c^2 satisfy

$$\mu = 50, \quad c^2 \in \{1/2, 1, 1.5, 2, 3, 4\}.$$

Further, take the length of the review interval equal to

$$R = 2.$$

The maximal order size Q and the service level α are varied as $Q = 125$, 200 and $\alpha = 0.95$, 0.99. The mean and the squared coefficient of variation of the review-time demand are given by $\mu_\xi = 2\mu$ and $c_\xi^2 = c^2/2$. In the table we give for the various parameter combinations the approximate order-up-to-level S_α which has been calculated by the procedure sketched above. For the cases with $c_\xi^2 \geqslant ½$ a Coxian-2 density is fitted to the review-time demand by matching the first three moments. Also, we give in the table the actual service level of the S_α policy under the situation of gamma distributed demand. The actual service level and its 95% confidence interval are determined by computer simulation. The numerical results confirm that the approximative approach performs quite well, provided that c_ξ^2 is not too small.

TABLE 2.1. Numerical results.

	$Q=125$			
c_ξ^2	$S_{0.95}$	act.service	$S_{0.99}$	act.service
1/4	234	.944(±.002)	342	.988(±.002)
1/2	425	.952(±.002)	642	.990(±.002)
3/4	615	.950(±.005)	941	.991(±.002)
1	807	.952(±.005)	1240	.989(±.003)
1.5	1194	.950(±.007)	1842	.990(±.005)
2	1583	.950(±.009)	2444	.989(±.004)

	$Q=200$			
c_ξ^2	$S_{0.95}$	act.service	$S_{0.99}$	act.service
1/4	153	.940(±.001)	209	.986(±.001)
1/2	227	.950(±.001)	333	.991(±.001)
3/4	300	.950(±.002)	455	.990(±.001)
1	376	.948(±.002)	576	.990(±.001)
1.5	534	.950(±.002)	828	.989(±.001)
2	695	.950(±.003)	1081	.991(±.001)

3. Overflow probabilities in buffers with service interruptions

3.1. Model

Buffer overflow in communication and production systems is an important problem, particularly when those systems may be subject to random breakdowns. This section shows how simple probabilistic tools can be used to dimension the buffer size so that a very small overflow probability is achieved.

Let us consider a communication channel at which batches of packets arrive according to a Poisson process with rate λ. The batch size has a general discrete distribution

$$P(\text{batch size is } j) = \beta_j \quad \text{for } j = 1,2,...,$$

where $\beta = \Sigma\, j\beta_j$ denotes the average batch size. The packets are temporarily stored in a finite buffer to await transmission. Overflow occurs for those packets from an arriving batch which are in excess of the remaining buffer capacity. The packet transmission is synchronous, that is, one packet is taken out of the buffer for transmission at discrete clock times $t = 1,2,...,$ provided that the transmission channel is available. The packets have all a fixed length and the transmission time of each packet is one time slot. The channel is subject to random service interruptions. The on-times of the channel are assumed to have a geometric distribution

$$P(\text{on-time is } j) = (1-p)p^{j-1} \quad \text{for } j = 1,2,...,$$

while the off-time of the channel have a general discrete distribution

$$P(\text{off-time is } j) = q_j \text{ for } j = 1,...,M$$

for some finite $M \geqslant 1$. The on-times and off-times form a sequence of independent random variables. In other words, the process of on- and off-times is modeled as an alternating renewal process in which the breakdowns occur according to a Bernoulli process. A special case of this model is the simple model of random independent interruptions, where the channel fails in each time slot with a same probability. This simplest model with service interruptions was first studied in Heines [2] and was generalized in Tijms and Van Ommeren [5] and Woodside and Ho [7]. The analysis in these referencess will be refined and extended in this section. The goal is to find a computationally tractable method for the calculation of the buffer size so that the overflow probability of an arbitrary packet is less than a prespecified value α. In typical applications α is very small (e.g. $\alpha = 10^{-9}$). To achieve a very small overflow probability, it must be assumed that the offered load to the channel is not excessively high. The precise assumption will be given in subsection 3.3.

3.2. Analysis via an $M^x/G/1/K$ queueing model

Assume for the moment that the buffer size is fixed and that the buffer has room for K packets including any packet in service. In this subsection it will be shown how the communication system with random service interruptions can be analyzed via an $M^x/G/1/K$ queueing model with exceptional first services. This translation step will be crucial in our analysis. In the $M^x/G/1/K$ queueing model batches of customers arrive according to a Poisson process with rate λ. The batch-size has the discrete distribution $\{\beta_j\}$. There is a single server and a finite waiting room with capacity K including any customer in service. An arriving batch whose size exceeds the remaining capacity in the buffer is partially lost due to overflow. The service times of the customers are independent random variables. The service of each customer is distributed as the generic random variable S, except for the first customer in each batch that finds upon arrival the system empty. The service times of those first customers are distributed as the generic random variable S_{exc} (exceptional first service). In addition, there is a warming-up time W before the server can start actual service after an idle period. Hence we are in fact considering a variant of the standard $M^x/G/1/K$ queueing model. In the next subsection we show how to compute the overflow probability for the versatile model with exceptional first services.

In the remaining part of this section we will translate the communication model with service interruptions to the $M^x/G/1/K$ queueing model with exceptional first services. To do so, we need to specify the 'normal' service S, the 'exceptional' service S_{exc}, and the warming-up time W. The arrival rate λ, the batch-size distribution $\{\beta_j\}$, and the buffer capacity K are identical in both models. The translation step is easily understood by the following definition of the service time of a packet in the communication model:

> the service time of a packet = the number of time slots from the discrete clock time at which the packet comes in for its turn for

> transmission until the moment at which the transmission of the packet is successfully completed.

Further, we define for the communication model,

> the warming-up time = the time elapsed between the arrival of a batch finding the system empty and the beginning of the next time slot.

Let us first specify the probability distribution of the warming-up time W. This requires the calculation of the conditional probability $P\{T_1 > t \mid T_1 \leqslant 1\}$, where T_1 is the first arrival epoch in a Poisson arrival process. Thus we find

$$P\{W \leqslant x\} = \frac{e^{-\lambda(1-x)} - e^{-\lambda}}{1-e^{-\lambda}} \text{ for } 0 \leqslant x \leqslant 1.$$

We have to distinguish between two types of services. First, the service time of a packet whose turn comes directly after the service completion of a preceding packet. Second, the service time of a packet which is served as first one from a batch finding upon arrival no other packets in the system. For the first type of service, the channel was necessarily on during the actual execution of the preceding service. Thus, at the beginning of the new time slot either the on-time continues for a next slot with probability p or an off-time starts with probability $1-p$. Since the probability distribution of the off-time is given by $\{q_j, 1 \leqslant j \leqslant M\}$, it follows that the 'normal' service time S in the communication model has the probability distribution

$$P\{S=k\} = \begin{cases} p & \text{for } k=1, \\ (1-p)q_{k-1} & \text{for } 2 \leqslant k \leqslant M+1. \end{cases}$$

For the second type of service, the situation is more complicated. Then we need the distribution of the state of the system at the moment of the first arrival since the end of the last time slot at which a transmission was completed and the system was left empty. Let τ_0 be the time slot just following the latter time slot. At the beginning of time slot τ_0 the system is either in state (0,0) with probability p or in state $(k, 0)$ with probability $(1-p)q_k$ for $k = 1,...,M$. Here state (0,0) means that the channel is on and no arrival occurred in the preceding slot, whereas state $(k, 0)$ means that the channel is off with a remaining off-time of k time slots while no arrival occurred in the preceding slot. In addition, the system is said to be in state (0,1) if the channel is on and one or more arrivals occurred in the preceding time slot, and the system is said to be in state $(k, 1)$ if the channel is off with a remaining off-time of k slots while one or more arrivals occurred in the preceding slot. Some reflection shows that a Markov chain can be used to describe the behavior of the system from the beginning of the particular time slot τ_0 until the beginning of the time slot τ_1 at which one of the (absorbing) kstates (0,1) or $(k, 1)$ for $k = 1,...,M-1$ is reached for the first time. At the beginning of time slot τ_1 the system is no longer empty and an 'exceptional' service S_{exc} of a new packet is ready to start. Thus,

$$P\{S_{exc}=k\} = \phi_{k-1} \text{ for } k = 1,...,M,$$

where the (absorption) probability ϕ_j for $j = 0,...,M-1$ is defined as

$$\phi_j = \text{ the probability that the system is in state } (j,\, 1) \text{ at the beginning of time slot } \tau_1.$$

To calculate the ϕ_j's we define for $j = 0,...,M-1$ the absorption probabilities

$$f_{s,(j,\,1)} = \text{ the probability that the Markov chain will be absorbed in state } (j,\, 1) \text{ starting from state } s,$$

where $s \in S_0 = \{(0,0),(1,0),...,(M,0)\}$. Then

$$\phi_j = pf_{(0,0)(j,\,1)} + (1-p)\sum_{i=1}^{M} q_i f_{(i,\,0)(j,\,1)} \text{ for } j = 0,1,...,M-1.$$

Using standard arguments from Markov chain theory, it is easily verified that for any fixed j the absorption probabilities $f_{s,(j,\,1)}$ for $s \in S_0$ can be calculated as the unique solution to a system of $M+1$ linear equations. For $j = 0$, the linear equations are

$$f_{(0,0)(0,1)} = (1-e^{-\lambda})p + e^{-\lambda}pf_{(0,0)(0,1)} + e^{-\lambda}(1-p)\sum_{i=1}^{M} q_i f_{(0,i)(0,1)},$$

$$f_{(i,\,0)(0,1)} = (1-e^{-\lambda})U_{i-1} + e^{-\lambda}f_{(i-1,0)(0,1)} \text{ for } i = 1,...,M,$$

where U_m is an abbreviation for

$$U_m = \begin{cases} 1 \text{ for } m=0, \\ 0 \text{ for } m \neq 0. \end{cases}$$

For any fixed $j = 1,...,M-1$, the linear equations are

$$f_{(0,0)(j,\,1)} = (1-e^{-\lambda})(1-p)q_j + e^{-\lambda}pf_{(0,0)(j,\,1)} + e^{-\lambda}(1-p)\sum_{i=1}^{M} q_i f_{(i,\,0)(j,\,1)},$$

$$f_{(i,\,0)(j,\,1)} = (1-e^{-\lambda})U_{i-j-1} + e^{-\lambda}f_{(i-1,0)(j,\,1)} \text{ for } i = 1,...,M.$$

For any fixed j, the corresponding system of $M+1$ linear equations is very easy to solve. Each system is upper-diagonal and can be simply solved by backwards substitution. The details are easily worked out.

REMARK 3.1. Though the calculations of the absorption probabilities for the general case are not difficult, they become extremely simple for the geometric case. If the off-times have the geometric distribution

$$P\{\text{off-time} = j\} = (1-r)r^{j-1} \text{ for } j = 1,2,...,$$

we can restrict to a Markov chain with four states (i,j) with $i,j = 0,1$. The component $i = 0$ (1) means that the channel is on (off), and the component $j = 0$ (1) means that no (one or more) arrivals occurred in the preceding time slot. The states (0,1) and (1,1) are absorbing. Now we only have to solve two systems of two linear equations each. The absorption probabilities $f_{(0,0)(0,1)}$ and $f_{(1,0)(0,1)}$ are the unique solution to

$$f_{(0,0)(0,1)} = (1-e^{-\lambda})p + e^{-\lambda}pf_{(0,0)(0,1)} + e^{-\lambda}(1-p)f_{(1,0)(0,1)},$$
$$f_{(1,0)(0,1)} = (1-e^{-\lambda})(1-r) + e^{-\lambda}rf_{(1,0)(0,1)} + e^{-\lambda}(1-r)f_{(0,0)(0,1)}.$$

Similarly, the other two absorption probabilities follow from

$$f_{(0,0)(1,1)} = (1-e^{-\lambda})(1-p) + e^{-\lambda}pf_{(0,0)(1,1)} + e^{-\lambda}(1-p)f_{(1,0)(1,1)},$$
$$f_{(1,0)(1,1)} = (1-e^{-\lambda})r + e^{-\lambda}rf_{(1,0)(1,1)} + e^{-\lambda}(1-r)f_{(0,0)(1,1)}.$$

The two systems of two linear equations each can be explicitly solved. The probability distribution of the 'exceptional' service time S_{exc} is now calculated as

$$P\{S_{exc}=1\} = pf_{(0,0)(0,1)} + (1-p)f_{(1,0)(0,1)},$$
$$P\{S_{exc}=j\} = [pf_{(0,0)(1,1)} + (1-p)f_{(1,0)(1,1)}](1-r)r^{j-2} \text{ for } j \geq 2.$$

3.3. *The overflow probability*

The overflow probability will be calculated using the method of regenerative processes. This intuitive and powerful probabilistic approach got its popularity in the OR community after the appearance of the book of Ross [3] in 1970. However, in the sixties, De Leve made already extensive use of the same approach under the name of 'herhalingsprogrammering'.

Numerous stochastic processes arising, for example, in queueing and inventory systems have the property that they regenerate themselves at certain points in time. Then the behavior of the process after each regeneration epoch is a probabilistic replica of the behavior starting at time zero and is independent of the behavior before the regeneration epoch. It will be intuitively clear that the long-run behavior of a regenerative stochastic process can be studied in terms of its behavior during a single regeneration cycle.

For the communication model, let us say that a cycle starts each time an arriving batch finds the system idle. For the model with a buffer capacity of K packets, define

$$N(K) = \text{the number of packets served during one cycle.}$$

In particular, $N(\infty)$ denotes the number of packets served during one cycle for the model with $K = \infty$ (i.e., the infinite capacity model). To ensure that $N(\infty)$ has a proper probability distribution, we need the assumption

$$\rho = \lambda\beta E(S) < 1,$$

where $E(S) = 1 + (1-p)\Sigma_k kq_k$ denotes the mean of a 'normal' service. Without this assumption we cannot guarantee a very small overflow probability. Also, for the communication model with capacity K, define

$$\pi_{loss}(K) = \text{the long-run fraction of packets that overflow.}$$

We now derive the following lemmas.

LEMMA 3.1. *For any $K \geqslant 1$,*

$$\pi_{loss}(K) = \frac{\beta-\rho+\lambda\beta\{E(W)+E(S_{exc})\}-(1-\rho)EN(K)}{\beta-\rho+\lambda\beta\{E(W)+E(S_{exc}\}+\rho EN(K)}.$$

PROOF. The proof is based on simple probabilistic arguments. First,

the long-run average input of accepted packets
= the long-run average output of accepted packets.

Since the average arrival rate of packets is $\lambda\beta$, we have

the long-run average input of accepted packets $= \lambda\beta(1-\pi_{loss}(K))$.

By the theory of regenerative processes,

the long-run average output of accepted packets

$$= \frac{E[\text{number of packets served during one cycle}]}{E[\text{length of one cycle}]}.$$

The numerator of this ratio is by definition equal to $EN(K)$. Further,

$$E[\text{length of one cycle}] = E(W)+E(S_{exc})+\{EN(K)-1\}E(S)+\frac{1}{\lambda},$$

using Wald's equation to justify the third term in the right-hand side of this equation. Combining the above relations, the desired result follows. □

LEMMA 3.2. *For any $K \geqslant 1$,*

$$EN(K) = \sum_{j=0}^{K-1} q_j(\infty)\times EN(\infty),$$

where for the infinite capacity model $q_j(\infty)$ is defined as the long-run fraction of service completion epochs at which j packets are left behind in the system.

PROOF. The assumption of a Poissonian arrival process of batches is crucial in the proof. Using the memoryless property of the Poisson process and the fact that the packets are served one at a time, it can be seen that for any fixed $0 \leqslant j \leqslant K-1$ the probability distribution of the number of service completion epochs at which j packets are left behind in one cycle for the model with finite capacity K is identical to the corresponding probability distribution for the infinite capacity model. In the latter model, the ratio of the expected number of service completions at which j packets are left behind in one cycle and the expected number of service completions in one cycle is equal to $q_j(\infty)$, by the theory of regenerative processes. Hence $EN(K)/EN(\infty)$ equals $\Sigma_{j=0}^{K-1} q_j(\infty)$, yielding the desired result. □

As a consequence of the Lemmas 3.1 and 3.2, we have expressed the loss probability $\pi_{loss}(K)$ for the finite capacity model in terms of quantities for the infinite capacity model. The latter model is well-studied. We first note:

LEMMA 3.3.

$$EN(\infty) = \frac{\beta - \rho + \lambda\beta\{E(W) + E(S_{exc})\}}{1-\rho}.$$

This result is also an immediate corollary of Lemma 3.1 and the fact that $\pi_{loss}(\infty) = 0$. The probability distribution $\{q_j(\infty)\}$ is the equilibrium distribution of an embedded Markov chain $\{X_n\}$, where

X_n = the number of packets left behind at the n-th service completion epoch in the infinite capacity model.

Using standard arguments from Markov chain theory, we have

$$q_n(\infty) = \sum_{k=1}^{n+1} q_k(\infty) a_{n+1-k} + q_0(\infty) \sum_{k=1}^{n+1} \beta_k a^*_{n+1-k}, \; n = 0,1,...,$$

where

$a_k(a^*_k)$ = the probability that a total of k packets arrive during the normal service time S (during the sum of the warming-up time W and the exceptional service time S_{exc}).

Note that

$$q_0(\infty) = \frac{1}{EN(\infty)},$$

where an explicit expression for $EN(\infty)$ is given above. Next we apply the basic technique of generating functions. It is a matter of routine algebra to verify that

$$\sum_{n=0}^{\infty} q_n(\infty) z^n = q_0(\infty) \frac{\{A(z) - \beta(z) A^*(z)\}}{A(z) - z},$$

where $\beta(z)$,$A(z)$ and $A^*(z)$ are the generating functions of the probability distributions $\{\beta_j\}$,$\{a_j\}$ and $\{a^*_j\}$. These generating functions can be calculated from

$$\beta(z) = \sum_{j=1}^{\infty} \beta_j z^j, \; A(z) = \sum_{t=1}^{\infty} e^{\lambda t\{\beta(z)-1\}} P\{S=t\},$$

$$A^*(z) = \frac{(e^{\lambda\beta(z)} - 1) e^{-\lambda}}{\beta(z)(1-e^{-\lambda})} \times \sum_{t=1}^{\infty} e^{\lambda t\{\beta(z)-1\}} P\{S_{exc}=t\}.$$

Note that $A^*(z)$ is the product of the generating functions of the probability distributions of the number of arrivals during W and the number of arrivals during S_{exc}. Using the specific structure of the distributions of S and S_{exc}, the expressions for $A(z)$ and $A^*(z)$ can be further simplified.

A practically useful asymptotic expansion for the $q_j(\infty)$'s can be obtained from the generating function under the following assumption:

ASSUMPTION. *The convergence radius R of the power series $\beta(z) = \Sigma_{j=1}^{\infty}\beta_j z^j$ is larger than* 1.

Loosely put, this assumption requires that the batch-size distribution has no extremely long tail. For example, the assumption is satisfied with $R = \infty$ when $\{\beta_j\}$ has finite support. The generating functions above were originally defined only for $|z| \leqslant 1$. However, under the assumption, they can be analytically extended beyond the unit circle. From complex function analysis it is known that the smallest zero of $A(z) - z$ in the domain beyond the unit circle determines the asymptotic expansion of $q_j(\infty)$ for j large. A minor modification of the proof of Theorem 1 in Tijms and Van Ommeren [5] yields the important result:

THEOREM 3.4. *For large K,*

$$\sum_{j=K}^{\infty} q_j(\infty) \sim q_0(\infty) \frac{[\beta(z_0)A^*(z_0) - A(z_0)]}{\{A'(z_0) - 1\}(z_0 - 1)} z_0^{-K},$$

where $z_0 \in (1,R)$ is the unique number satisfying

$$\sum_{t=1}^{\infty} e^{\lambda t\{\beta(z_0)-1\}} P\{S = t\} = z_0.$$

We are now in a position to state our main result. Therefore, we first note that by the Lemmas 3.1-3.3,

$$\pi_{loss}(K) = \frac{(1-\rho)[1 - \sum_{j=0}^{K-1} q_j(\infty)]}{1 - \rho + \rho \sum_{j=0}^{K-1} q_j(\infty)}.$$

Hence the following final result is obtained from Theorem 3.4.

THEOREM 3.5. *For α small enough, the minimal buffer size K satisfying $\pi_{loss}(K) \leqslant \alpha$ can be approximately calculated from*

$$K(\alpha) \approx \frac{1}{ln(z_0)} \ln\{\frac{\gamma(1-\rho+\rho\alpha)}{\alpha}\},$$

where the constant γ is given by

$$\gamma = \frac{(1-\rho)[\beta(z_0)A^*(z_0) - A(z_0)]}{[\beta - \rho + \lambda\beta\{E(W) + E(S_{exc})\}][A'(z_0) - 1](z_0 - 1)}.$$

References

1. A.G. de Kok (1989). *A moment-iteration method for approximating the waiting-time characteristics of the GI/G/1 queue*. Probability in the Engineering and Informational Sciences 3, 273-288.
2. T.S. Heines (1983). *On the behavior of buffers with random service interruptions*. IEEE Transactions on Communication 28, 573-576.
3. S.M. Ross (1970). *Applied probability models with optimization applications*. Holden-Day, Inc., San Francisco.
4. H.C. Tijms (1986). *Stochastic modeling and analysis: A computational approach*, Wiley, Chichester.
5. H.C. Tijms, J.C.W. van Ommeren (1989). *Asymptotic analysis for buffer behavior in communication systems*, Probability in the Engineering and Informational Sciences 3, 1-12.
6. J.C.W. van Ommeren, R.D. Nobel (1989). *On the waiting time distribution in a GI/G/1 queue with a Coxion-2 service time distribution*. Statistica Neerlandica 43, 85-90.
7. C.M. Woodside, E.D.S. Ho (1987). *Engineering calculation of overflow probabilities in buffers with Markov interrupted service*. IEEE Transactions on Communication 35, 1272-1277.

Partial contraction in Christofides' lower bound for the traveling salesman problem

A. Volgenant
Institute of Actuarial Science & Econometrics, University of Amsterdam

Christofides' lower bound for the Traveling Salesman Problem is based on solving repeatedly a Linear Assignment Problem. In each step of the iterative method the subtours of the Linear Assignment solution are contracted to nodes. We consider the idea of partial contraction that contracts subtours one at a time. Together with already known improvements such as omitting cities and use of special dual solutions, it appears that for Euclidean problems of size 100 the bound can be substantially improved.

Keywords: Traveling Salesman Problem, lower bound.
Subject Classification: (AMS 1980) Primary: 94C15; Secondary: 05C35
Acknowledgement: M. Smit assisted with the computational results.

1. Introduction

Christofides (1972) introduced for the Traveling Salesman Problem (TSP) a lower bound that is computed by repeatedly solving the Linear Assignment Problem (LAP). The algorithm can be described in six steps.

Step 1 Initialization:
C becomes the start matrix with distances d_{ij}, $i,j=1,2,...,n$, with n the number of cities;
$L:=0$.

Step 2 Compute a lower bound:
Solve the assignment problem (LAP), with value $Z(C)$;
$L:=L+Z(C)$;
if the solution is a tour, then go to Step 6.

Step 3 Reduction:
Reduce the distance matrix with the dual solution (u,v) $c_{ij}:=c_{ij}-u_i-v_j$.

Step 4 Contraction:
Replace the subtours by nodes with the distance between two subtours given by the minimum of the distances between the nodes of these subtours.

```
Step 5   Compression:
         Check whether the new matrix (the relative cost matrix) obeys
         the triangle inequality;
         if not, replace the matrix by the shortest path matrix;
         go to Step 2.
Step 6   End:
         The TSP has a lower bound with value L.
```

The correctness of the bound follows from the observation that on the contracted problem a 'semi' TSP has to be solved: the new 'cities' have to be visited at least once. When such a problem obeys the triangle inequality, it is equivalent to a standard TSP. Step 5 takes care of this property.

For this lower bound a number of improvements are known. Kindervater, Volgenant, De Leve and Van Gijlswijk (1985) recommend to compute in the Reduction step a dual solution that is a weighted average of 2n dual solutions in order to suppress values for the duals that would give rise to small distances in the contracted matrix. This improvement is exploited in the computational results in the following sections.

Volgenant and Jonker (1985) suggested partial contraction in the Contraction step: instead of replacing all subtours of the LAP-solution, contraction can be done for an arbitrary choice from the set of subtours, e.g., one subtour. The correctness is based on the correlation between the partial contraction of a subtour and a constraint of the original TSP:

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geqslant 1,$$

with S the set of cities in the considered subtour possibly after one or more previous contractions.

The compression step can be done simpler after a partial contraction. Assuming one subtour has been contracted with the largest index, say k, the related partial compression can be given as

$$\tilde{c}_{ij} = \min\{c_{ij}, \min\{c_{ik} + c_{kj} \mid i,j < k\}\}.$$

The direct purpose of partial contraction is to decrease the number of subtours. By the presence of many subtours the lower bound can be poor. The question is now how many and which subtours to contract partially. It may be best to contract only 2-subtours, as then the number of subtours is reduced best.

Christofides' algorithm in its original version asks $O(n^3)$ computations; applying partial steps does not change this order. The previous and the present cost matrix differ only a little, so that a simple adjustment of $O(n^2)$ produces a new optimal LAP solution. Also partial contraction of one subtour and the related compression can be done in the same order, just as the solution of the LAP in step 2 of the next iteration. All the computational results are given without computer times, as we did not try to implement efficiently the considered approaches.

We have restricted the computational results to the Euclidean type of problem, as this type is known to be the most difficult one. The (ten) test problems are of size 100 with the coordinates drawn uniformly on the interval [1, 1000]. The presented approaches excluding the geometric based ones, however, can also be applied to other types, such as random table problems. Some of the approaches in the last sections are only valid for symmetric problems.

One may argue that an assignment lower bound for the TSP is not the best there is. For many variants of the TSP, however, the bound is very suited, as shown by Volgenant and Jonker (1987) for the generalized TSP, defined as the problem in which the salesperson can visit every node at most once and some penalty cost is incurred for every unvisited node.

In the following sections we will first consider partial contraction of the smallest subtour, then a less successful variation. In section 4 we will consider partial contraction in a later iteration of Christofides' algorithm, and finally we will treat the comparison and combination with omitting cities.

2. Partial contraction of the smallest subtour

We suggest first a simple rule for partial contraction: choose the subtour with the smallest length. This choice is intuitively appealing, as the contraction of such a subtour can involve only small reductions of distances in the compression step.

We have applied this rule to the ten test problems, both for contraction of the subtour with the smallest, as well as for contraction of the subtour with the largest length, in order to show the different influence on the value of the lower bound.

After the modified first iteration, the algorithm has been applied without modification, i.e., in the next iterations the distance matrix has been completely contracted and completely compressed. The computational results are given in Table 1.

Clearly, one time partial contraction of the subtour with minimal length gives nearly always an improvement. Contraction of the largest subtour however, yields on the average a worse lower bound. The question now remains how often partial contraction can be repeated with an increasing lower bound.

After partial contraction (and especially after complete contraction) it is a disadvantage that the original distance matrix has been replaced with a relative cost matrix, as this disturbs in general the triangle inequality and makes compression necessary. After partial contraction the new matrix however is not very different from the original one. So we can do one or more times partial contraction without damaging seriously the quality of the lower bound.

Encouraged by the improvement of the bound by one time partial contraction we experimented with two times contraction of the smallest subtour. On the average the results are better; see Table 2 (in order to comfort the comparison some columns of Table 1 have been repeated). It seems worth the trouble to apply partial contraction a few times before using complete contraction. As a heuristic rule partial contraction has been applied until $k+1$ times

Number	Christofides	Contraction on subtour		Optimal
problem	lower bound	largest	smallest	value
1	95.0	93.9	95.1*	7935
2	92.2	91.9	92.4*	7893
3	92.8	93.0	93.7*	8006
4	93.1	93.1	93.3*	7820
5	93.4	93.5	93.6*	7469
6	93.9*	93.8	93.7	7217
7	95.2	95.1	95.4*	7980
8	94.9	94.8	95.7*	7436
9	92.3	92.5	92.6*	7686
10	91.9	91.7	91.9*	7372
Average	93.5	93.3	93.7*	7681

TABLE 1. Relative lower bounds related to the optimal values (= 100%) for partial contraction on the largest and the smallest subtour (best values marked with a star).

Number	Christofides	Partial contraction of the smallest subtour			
problem	lower bound	1 time	2 times	*k* times	*k*
1	95.0	95.1	95.4	95.9*	5
2	92.2	92.4	92.5	93.0*	3
3	92.8	93.7	94.8	95.0*	3
4	93.1	93.3	93.4*	93.4*	2
5	93.4	93.6*	93.3	93.6*	1
6	93.9*	93.7	93.6	93.9*	0
7	95.2	95.4	95.6*	95.6*	2
8	94.9	95.7*	95.6	95.7*	1
9	92.3	92.6	92.7*	92.7*	2
10	91.9*	91.9	91.9	91.9*	0
Average	93.5	93.7	93.9	94.1*	1.9

TABLE 2. Comparison of one and several times partial contraction of the smallest subtour. The best values have been marked with a star.

gives no improvement over k times partial contraction, i.e., the value of L in step 2 no longer increases. The results have been reported in Table 2 together with the number of times that partial contraction has been applied. The results indicate that contraction of the smallest subtour several times is not always the best.

It is also interesting to see the results of partial contraction of two times the largest subtour. This repeated contraction however gives worse results than one time contraction of the largest subtour. It is clearly useless to continue with partial contraction of the largest subtour.

After evaluation of the lengths of the contracted subtours, the conclusion must be that merely the length does not explain the quality of the obtained lower bounds. Intuitively one can say that if a small subtour is in the center of the set of cities (nodes) in a Euclidean problem, the contraction of this subtour can exercise a larger negative influence on the lower bound than the contraction of a larger subtour lying at the edge of the set.

3. Partial contraction of the subtour with the largest D-value

Volgenant and Jonker (1985) suggested to contract only those 2-subtours S, say with $S = \{1,2\}$, with non negative value of

$$D = \min\{c_{i1} + c_{2j} - c_{ij} | i,j = 3,...,n; i \neq j\}.$$

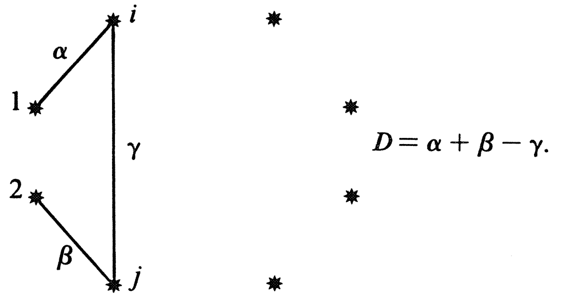


FIGURE 1. Illustration of the D-value for a subtour $S = \{1,2\}$.

In this way compression becomes superfluous and cannot cause loss of quality of the lower bound.

A nonnegative D-value is a strong condition (see Figure 1): for large problems (e.g. with 100 cities, as in the test problems), there exist often nodes i and j for which the quantity between accolades is negative. We think that the smaller the value of n, say after a number of iterations of Christofides' algorithm, the larger the chance to find a nonnegative D-value.

An alternative choice for the D-value is the sum of the decrements in the compression step

$$\sum_{i \neq j} \{c_{i1} + c_{2j} - c_{ij} | c_{i1} + c_{2j} - c_{ij} < 0\}.$$

We did not (yet) consider this alternative.

The comparison of partial contraction of the largest and the smallest subtour in Table 1 is in favor of the latter. This may be caused by the value of D, on the average -8 for the smallest and -81 for the largest subtour. Now the question is whether the D-value gives a usable condition for the contraction of a subtour.

In Table 3 one time partial contraction of the smallest subtour is compared with partial contraction of the subtour with the largest D-value. We see, that the contraction of the 2-subtour with the largest D-value is not better than partial contraction of the smallest subtour. Apparently the D-value alone is not a good measure for the successful contraction of a subtour.

The length of the subtour has also to be considered. The average length is 18 for the smallest subtour and 122 for the subtours with the largest D-value. The average D-value of the smallest respectively largest subtours is -8 and 55. One may think that partial contraction is successful on small subtours with large, nonnegative D-values. Examples as problem 4 in Table 3 show however, that the contraction of such a subtour can also yield a bound worse that the standard bound of Christofides. In this problem the largest D-value is 28 for a subtour with length 42.

It remains difficult to under stand why it is better to contract partially the smallest subtour than the subtour with the largest D-value. We did not find a condition related to the D-value for the successful partial contraction of a subtour.

Number problem	Christofides lower bound	Partial contraction smallest subtour	Partial contraction subtour largest D-value
1	95.0	95.1*	94.7
2	92.2	92.4	92.7*
3	92.8	93.7*	93.1
4	93.1	93.3*	92.6
5	93.4	93.6*	93.2
6	93.9*	93.7	93.7
7	95.2	95.4*	95.3
8	94.9	95.7*	94.9
9	92.3	92.6*	92.3
10	91.9	91.9	92.3*
Average	93.5	93.7*	93.5

TABLE 3. Comparison of one time partial contraction smallest subtour with partial contraction subtour with largest D-value. The best values have been marked with a star.

4. PARTIAL CONTRACTION IN A LATER ITERATION

During the subsequent iterations of Christofides' algorithm the size of the problem decreases, with an increasing chance on nonnegative D-values, so it may be useful to do partial contraction after a number of standard iterations in Christofides' algorithm.

To assess this idea, partial contraction of the smallest subtour has been done first, followed by respectively two and three times complete contraction and then partial contraction of the subtour with the largest D-value. The modified approach is completed by the standard steps of the algorithm of Christofides. After the second partial contraction the problem is much smaller. For the test problems the size of 100 is reduced on the average to 19 after the third and to 8 after the fourth iteration.

It can also be useful to contract the smallest subtour in iteration 4 and 5 instead of the subtour with the largest D-value as this approach was also successful in the first iteration (see Table 3). The results can be best compared with partial contraction of the smallest subtour in the first two iterations, because then also two subtours are contracted. The computational results are in Table 4 together with some columns repeated from Table 2.

To illustrate this, the second column of Table 4 gives the result of two times partial contraction of the smallest subtour in the first two iterations; the third and fourth column give the results of contraction of the smallest subtour in the first iteration and contraction of the subtour with the largest D-value in iteration 4 and 5. The last two columns give the results of partially contracting two times the smallest subtour, in the first iteration and thereafter in iteration 4 or 5.

It is clear that partial contraction in a later iteration is less favorable than at the start. Furthermore it is clear that we can contract better the subtour with the smallest length than the one with the largest D-value. So the length of the subtour appears to be a better measure for the success of partial contraction than the D-value.

Number	Smallest subtour	D-value in	D-value in	Smallest subtour in	
problem	in iteration 2	iteration 4	iteration 5	iteration 4	iteration 5
1	95.4*	95.2	95.1	95.3	95.2
2	92.5*	92.1	92.2	92.2	92.2
3	94.8*	93.7	93.7	93.8	93.6
4	93.4*	93.3	93.3	93.3	93.3
5	93.3	93.2	93.6*	93.6*	93.6*
6	93.6	93.4	93.7	94.0*	93.7
7	95.6*	95.4	95.3	95.4	95.3
8	95.6	95.6	95.7*	95.7*	95.7*
9	92.7*	92.6	92.6	92.7*	92.6
10	91.9	92.6*	91.7	91.9	91.9
Average	93.9*	93.7	93.7	93.8	93.7

TABLE 4. Results for partial contraction of the smallest subtour in iteration 1 followed by several contraction strategies: smallest subtour in iteration 2, the subtour with the largest D-value in iterations 4 respectively 5, and smallest subtour in these iterations (best values marked with a star).

5. A comparison with omitting cities

Partial contraction resembles the principle of omitting cities, as exploited by Volgenant, Van der Sluis and Jonker (1987). In some cases, however, partial contraction gives better results. The example (see Figure 2) is the same as gives by Volgenant et al. (1987) to illustrate that omitting cities can improve the LAP lower bound.

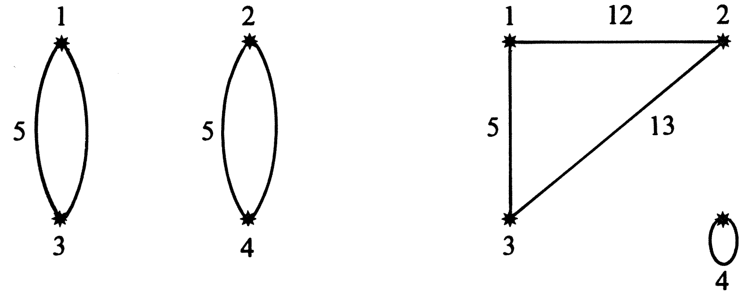


FIGURE 2. Omitting city 4 increases the LAP lower bound from 20 to 30.

Partial contraction does something else. The left part of Figure 2 is given in a distance matrix in the left part of Table 5. In the example city 4 was omitted and the subtour $S = \{3,4\}$ contracted. This subtour has length 10. After reduction we obtain the distance matrix in the middle part of Table 5. The dual variables have the values $u_3 = u_4 = v_3 = v_4 = 2.5$. After compression we obtain the distance matrix in the right part of Table 5. The solution now has the (optimal) value 10 + 5 + 9.5 + 9.5 = 34. Although for other choices of the dual values a less good result can be obtained it is useful to compare both methods for larger problems.

The principle of omitting cities has been applied within an other method, the Many Routes Approach, because this approach (without omitting cities) produced better results that the original method of Christofides. Therefore we outline first this approach in the next section.

	1	2	3	4
1	*	$\underline{5}$	12	13
2	$\underline{5}$	*	13	12
3	12	13	*	$\underline{5}$
4	13	12	$\underline{5}$	*

	1	2	3	4
1	*	5	9.5	10.5
2	5	*	10.5	9.5
3	9.5	10.5	*	0
4	10.5	9.5	0	*

	1	2	3
1	*	$\underline{5}$	9.5
2	5	*	$\underline{9.5}$
3	$\underline{9.5}$	9.5	*

TABLE 5. Partial contraction of subtour (3,4) increases the LAP lower bound from 20 to 34, the optimal value (underlined values indicate the LAP solution).

6. THE MANY ROUTES APPROACH AND PARTIAL CONTRACTION

The Many Routes Problem is an extension of the Two Routes Problem (2-RP), introduced by Jonker, De Leve, Van der Velde and Volgenant (1980). The 2-RP is the problem to find two paths of minimal total length that connect two given cities, both paths starting in the first and ending in the second city. All other cities must lie on exactly one of these paths. The problem is equivalent to the TSP in case of symmetry.

Van Leeuwen and Volgenant (1983) generalized the problem to the Many-Routes Problem in the context of Vehicle Scheduling. Stewart (1985) applied it to lower bounds for the Euclidean TSP. Volgenant, Van der Sluis and Jonker (1987) improved these bounds and combined the MRP with the algorithm of Christofides. How many routes are considered in this approach depends on the problem instance; it varies from 4 to 10 for the used test problems.

In the previous section an example has been given illustrating that in some cases partial contraction is better than omitting cities. Therefore partial contraction has been applied in the m-RP lower bound, instead of omitting cities.

The smallest subtour has been contracted as this, applied on Christofides' method, gave better results that other rules, for example the contraction of the subtour with the largest D-value. Further the subtour that is closest to the edge has been partially contracted. The results are in Table 6 just as that for one and two times partial contraction of the smallest subtour in combination with the Many Routes Approach (MRA).

Number	MRA and	MRA and partial contraction		
problem	omitting	1 x smallest	2 x smallest	1 x edge
1	97.1	97.7	97.8	97.9*
2	95.4	95.8	95.9*	95.7
3	95.5	95.8*	95.7	95.8*
4	95.4*	93.8	91.4	91.4
5	96.7	96.6	96.8*	96.4
6	94.9*	94.1	94.6	94.2
7	95.7	96.0*	95.9	94.9
8	96.6*	95.4	95.3	95.3
9	95.5*	94.0	94.3	93.9
10	96.0*	95.2	95.4	95.0
Average	95.8*	95.4	95.3	95.1

TABLE 6. Comparison of omitting cities and partial contraction of one and two times smallest subtour and one time subtour closest to the edge of the set of cities (best values marked with a star).

We see that in general partial contraction looses from omitting cities, although for some individual problems better results are obtained. As two times partial contraction of the smallest subtour gives worse bounds than one time, there is no reason to continue with partial contraction of the smallest subtour. It is disappointing that partial contraction after the many routes approach, of the subtour closest to the edge is worse than the other results. So it seems useless to repeat partial contraction of the subtour closest to the edge.

Although partial contraction instead of omitting cities is not more profitable, it may be useful to combine these approaches, e.g., first omitting cities and then partial contraction. Just as in the first sections the smallest subtour has been first contracted improving the original method. Therefore partial contraction has been repeated a second time, yielding nearly the same results as for one time. Analogously to Section 2, partial contraction of the smallest subtour has been repeated until $k+1$ times gives no improvement over k times. The results are in Table 7.

We see that problem 7 is somewhat special in the sense that one time partial contraction of the smallest subtour gives no improvement, while two times gives a better lower bound than the original method. So the smallest subtour is not always the best choice for the subtour to be contracted.

Finally, after omitting cities, partial contraction has been done of the subtour that is closest to the edge of the set of cities. The related results are better than the one of the original method with omitting cities, but on the average not better than partial contraction of the smallest subtour. The result was on the average 96.1% of the optimal solution. Again problem 7 was special as the smallest subtour is also closest to the edge, while contraction of this subtour gave no improvement. Apparently also other factors influence the results. It remains open to find them.

Number	MRA and	MRA, omitting and partial contraction smallest subtour			
problem	omitting cities	1 time	2 times	k times	k
1	97.1	98.0	98.1^*	98.1^*	2
2	95.4	95.8	95.9	96.1^*	3
3	95.5	96.0^*	96.0	96.0^*	1
4	95.4	95.7^*	95.7	95.7^*	1
5	96.7	96.9^*	96.0	96.9^*	1
6	94.9	95.3	95.4	95.6^*	3
7	95.7^*	95.7	95.8^*	95.7	0
8	96.6^*	96.6	96.5	96.6^*	0
9	95.5	95.9^*	95.6	95.9^*	1
10	96.9	96.3^*	96.1	96.3^*	1
Average	95.8	96.2	96.2	96.3^*	1.3

TABLE 7. Partial contraction of the smallest subtour after MRA and omitting cities compared to MRA and omitting cities (best values marked with a star).

7. Conclusions

The application of partial contraction of the smallest subtour during Christofides' method gives an improvement of the lower bound, both for the original method as well as for the Many Routes Approach with omitting cities. In both cases the gap for the Euclidean test problems is decreased with about 10% compared to the best known improved Christofides' lower bound.

The contraction of the smallest subtour is not always the best to do. The contraction of other subtours, such as the one with the largest D-value or the subtour that is closest to the edge of the set of cities, are on the average less successful. For some problems better results can be obtained.

Our aim to show empirically the usefulness of partial contraction has been reached. Further research may give better rules for the choice of the subtours to be contracted.

References

N. Christofides (1972). Bounds for the Traveling Salesman Problem. *Operations Research 20*, 1044-1056.

R. Jonker, G. de Leve, J.A. van der Velde and A. Volgenant (1980). Bounding Symmetric Traveling Salesman Problems with an Asymmetric Assignment Problem. *Operations Research 28*, 623-627.

G. Kindervater, A. Volgenant, G. de Leve and V. van Gijlswijk (1985). On Dual Solutions of the Linear Assignment Problem. *European Journal of Operational Research 19*, 76-81.

P.H. van Leeuwen and A. Volgenant (1983). Solving Symmetric Vehicle Routing Problems Asymmetrically. *European Journal of Operational Research 12*, 388-393.

W.R. Stewart (1985). An Improved Assignment Lower Bound for the Euclidian Traveling Salesman Problem. *Operations Research Letters 4*, 55-60.

A. Volgenant and R. Jonker (1985). Improving Christofides' Lower Bound for the Traveling Salesman Problem. *Optimization 16*, 691-704.

A. Volgenant, H.J. van der Sluis and R. Jonker (1987). Better Assignment Lower Bounds for the Euclidean Traveling Salesman Problem. *Optimization 18*, 393-403.

A. Volgenant and R. Jonker (1987). On Some Generalizations of the Traveling Salesman Problem. *Journal of the Operational Research Society 38*, 1073-1079.

Stochastic games and the total reward criterion

O.J. Vrieze
Department of Mathematics
University of Limburg
Maastricht

We discuss the state of the theory of two-person zero-sum stochastic games with the total reward criterion.

1. Introduction

The theory of stochastic games started with the fundamental paper of Shapley (1953). These games are stochastic models of competitive behaviour in a dynamic setting. They include as special cases the static noncooperative games the repeated games with complete information and the Markovian decision processes. Strikingly, the theory of dynamic noncooperative games and the theory of Markovian decision processes evolved for many years along separate lines. The commonly used techniques in stochastic games in the early approaches stem from the theory of functions and from fixed point theorems. Only during the 1970's the interrelationship between these two fields was fully recognized. From that time on many new results in the theory of stochastic games emerged by combining techniques from these two research areas.

Three dutch research centres served as pioneers in this respect. First, there was the game theory group around Stef Tijs at the Mathematics Department of the Catholic University at Nijmegen, which was mainly focussed on static noncooperative games like matrix games and bimatrix games. Second, there was the Department of Operations Research around Gijs de Leve at the Centre for Mathematics and Computer Science in Amsterdam (at that time called Mathematical Centre). This Centre was viewed as world-wide leading in the field of Markovian decision processes. In fact the cross-fertilization of static noncooperative games and Markovian decision processes resulting in stochastic games started when De Leve invited Tijs to visit the Mathematical Centre during November 1977. Especially Vrieze and Federgruen were encouraged by Tijs visit to apply the techniques from Markovian decision processes to game theory and in particular to stochastic games (Federgruen (1978a, 1978b), Tijs and Vrieze (1980), Vrieze and Tijs (1980), Vrieze (1981)). The third Dutch centre which ought to be mentioned is the school of Jaap Wessels at the

Mathematics Department of Eindhoven University of Technology. Also at this centre a strong group on Markovian decision processes was present and especially Van der Wal was active in the field of stochastic games (Van der Wal (1977, 1980)).

In later years more researchers on stochastic games used the theory of Markovian decision processes as their point of view. The following papers are typical examples of this approach: Hordijk and Kallenberg (1981), Parthasarathy et al. (1984), Raghavan et al. (1985), Filar and Schultz (1986), Thuijsman and Vrieze (1987), Thuijsman (1989) and Van der Neut (1989).

In this paper we will discuss the state of the theory of two-person zero-sum with the total reward criterion. In Section 2 the model is explained. Section 3 deals with criteria for stochastic games. In Section 4 we present some general theorems on total reward stochastic games Section 5 contains three subclasses of games which have been solved and in Section 6 we finish with some concluding remarks.

2. The stochastic game model

A two-person zero-sum stochastic game can be considered to be an extension of a matrix game in the following way: the game is played in periods, at each period the game is in one of finitely many states, in each state the players have to choose one action among finitely many. A pair of actions in a state determine a competitive payoff and a probability vector according to which the next state is selected. Then a states of a stochastic game can be represented by a matrix $M(s)$ in the following way. Let $m_s\,(n_s)$ be the number of actions available to player 1 (player 2) in state s. Then $M(s)$ is an $m_s \times n_s$-matrix for which cell (i,j) contains the payoff $r(s,i,j)$ (to player 1 from player 2) and the probability vector $p_s(i,j) = (p(1|s,i,j), p(2|s,i,j),...,p(z|s,i,j))$, where z is the number of states. The interpretation is that $p(t|s,i,j)$ denotes the probability that the system moves to state t if in state s player 1 chooses i and player 2 chooses j. A stochastic game is defined by a finite collection of such matrices $M(1)$, $M(2),..., M(z)$, also called states. Usually $\Sigma_t p(t|s,i,j) = 1$ but we will not require that in this paper.

If in each state the same player has only one action available, the stochastic game reduces to a Markovian decision process We will consider games with infinitely many periods numbered 0,1,2,.... The specification of a starting state determines a specific play of the game.

A strategy for a player (notation π_k for player k, $k = 1,2$) is a scheme which tells him for each period and each state at that period what action to choose as a function of the history up to that period. As usual in game theory, randomization (or mixing) is allowed. The subclass of stationary strategies (notation σ_k for player k, $k = 1,2$) makes no use of the history nor of the period number. Then σ_k has the form $\sigma_k = (\sigma_k(1), \sigma_k(2), \ldots, \sigma_k(z))$ where $\sigma_k(s)$ is a mixed action in state s, $s = 1,2,...,z$.

A pair of strategies determine a stochastic process dependent of the starting state. S_τ, $A_{1\tau}$ and $A_{2\tau}$ will denote the stochastic variables associated with respectively the state at period τ, the action of player 1 at period τ and the action of player 2 at period τ, $\tau=0,1,2,\ldots$. A realisation of these variables is denoted by s_τ, $a_{1\tau}$, $a_{2\tau}$. To each pair (π_1,π_2) and for each period τ, we can associate the well-defined stochastic variable H_τ, the outcome of which are possible histories $(s_0,a_{10},a_{20},s_1,\ldots,a_{2\tau})$ up to period τ. By the Kolmogorov extension theorem this sequence H_τ, $\tau=0,1,2,\ldots$ can be uniquely extended to a stochastic variable H_∞ defined on the set of "infinite" realisations of the game: $s_0,a_{10},a_{20},s_1,\ldots$.

Now, a pair of strategies π_1 and π_2 determine a stochastic stream of payoffs associated to a fixed starting state s_0. Such a stream can be evaluated in different ways, leading to different criteria, making use of H_τ, $\tau=0,1,2,\ldots$ or of H_∞.

Let $Cr(s,\pi_1,\pi_2)$ denote the outcome of a certain criterion. Then a stochastic game is said to have a value if $\sup_{\pi_1}\inf_{\pi_2} Cr(s,\pi_1,\pi_2)=\inf_{\pi_1}\sup_{\pi_2} Cr(\pi_1,\pi_2)=:v_s$ for all $s\in S$. π_1^ϵ is called ϵ-optimal if $Cr(s,\pi_1^\epsilon,\pi_2)\geqq v_s-\epsilon$ for all $s\in S$ and π_2^ϵ is called ϵ-optimal if $Cr(s,\pi_1,\pi_2^\epsilon)\leqq v_s+\epsilon$ for all $s\in S$ ($\epsilon\geqslant 0$). 0-optimal strategies are called optimal.

The expected payoff at period τ will be denoted by $E_{s,\pi_1,\pi_2}[r(S_\tau,A_{1\tau},A_{2\tau})]$ and this payoff can be computed from H_τ. For a pair of stationary strategies we have an explicit expression for this expectation, namely (in vector notation)

$$E_{\pi_1,\pi_2}[r(S_\tau,A_{1\tau},A_{2\tau})]=P^\tau(\sigma_1,\sigma_2)\,r(\sigma_1,\sigma_2)$$

where

$$r(\sigma_1,\sigma_2)=(r(1,\sigma_1,\sigma_2),\ r(2,\sigma_1,\sigma_2),\ldots,r(z,\sigma_1,\sigma_2))$$

with $r(s,\sigma_1,\sigma_2)$ being the expected payoff at state s. $P(\sigma_1,\sigma_2)$ denotes the stochastic $z\times z$-matrix with (s,t)-th element equal to the probability that the system moves to state t when in state s the players make their choices according to $\sigma_1(s)$ and $\sigma_2(s)$. $P^\tau(\sigma_2,\sigma_2)$ is the τ-fold multiplication of $P(\sigma_1,\sigma_2)$ and it can be verified that the (s,t)-th element of $P^\tau(\sigma_1,\sigma_2)$ equals the probability that at period τ the system is in state t if it starts in state s and the players use σ_1 and σ_2.

3. Criteria for stochastic games

Three criteria are commonly applied to stochastic games. The discounted reward criterion, for $\beta\in(0,1)$, based on H_τ, $\tau=0,1,2,\ldots$ gives

$$D_\beta(s,\pi_1,\pi_2):=\sum_{\tau=0}^{\infty}\beta^\tau E_{s\pi_1\pi_2}[r(S_\tau,A_{1\tau},A_{2\tau})]. \tag{1}$$

On the other hand, based on H_∞ one can define

$$D_\beta^\infty(s,\pi_1,\pi_2):=E_{s\pi_1\pi_2}^\infty\Big[\sum_{\tau=0}^{\infty}\beta^\tau r(S_\tau,A_{1\tau},A_{2\tau})\Big]. \tag{2}$$

Since the rewards are uniformly bounded and $\beta^\tau\to 0$ as $\tau\to\infty$ it easily follows that $D_\beta(s,\pi_1,\pi_2)=D_\beta^\infty(s,\pi_1,\pi_2)$ for all π_1 and π_2. In his initializing paper of 1953, Shapley (1953) solved the stochastic game with the discounted reward criterion.

For the average reward criterion we have to take care of limit properties of sequences of payoffs. Viewed from a worst case point of player 1, one can define

$$A(s,\pi_1,\pi_2):=\liminf_{T\to\infty}\frac{1}{T+1}\sum_{\tau=0}^{T}E_{s\pi_1\pi_2}[r(S_\tau,A_{1\tau},A_{2\tau})] \qquad (3)$$

and

$$A^{\infty}(s,\pi_1,\pi_2):=E_{s\pi_1\pi_2}[\liminf_{T\to\infty}\frac{1}{T+1}\sum_{\tau=0}^{T}r(S_\tau,A_{1\tau},A_{2\tau})] \qquad (4)$$

In general (3) and (4) do not give the same outcome. However for stationary strategies the outcomes are the same. Moreover in 1980 Mertens and Neyman (1981) showed that stochastic games with the average reward criterion have a value which is the same for both the criteria (3) and (4).

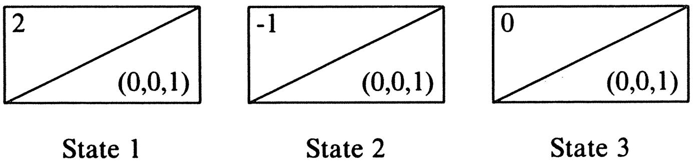
A third criterion in stochastic games is the total reward criterion defined by

$$T(s,\pi_1,\pi_2):=\liminf_{T\to\infty}\frac{1}{T+1}\sum_{\tau=0}^{T}\sum_{n=0}^{\tau}E_{s\pi_1\pi_2}[r(S_n,A_{1n},A_{2n})] \qquad (5)$$

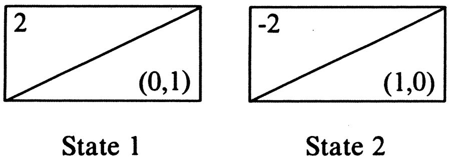
and

$$T^{\infty}(s,\pi_1,\pi_2):=E^{\infty}_{s\pi_1\pi_2}[\liminf_{T\to\infty}\frac{1}{T+1}\sum_{\tau=0}^{T}\sum_{n=0}^{\tau}r(S_n,A_{1n},A_{2n})] \qquad (6)$$

When $\lim_{T\to\infty}\Sigma_{\tau=0}^{T}E_{s\pi_1\pi_2}[r(S_\tau,A_{1\tau},A_{2\tau})]$ exists, it can be verified in (5) that this limit equals $T(s,\pi_1,\pi_2)$. The definition of the total rewards as the liminf of averages of partial sums is inspired by the fact that for stationary strategies the limit in (5) always exists as can be found in Vrieze and Thuysman (1987) (cf. also the second example below). Furthermore in that paper the application of the total reward criterion is motivated as a refinement of the average reward criterion. Consider for instance the trivial game

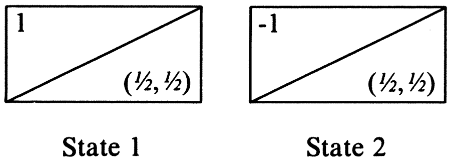


with average reward value (0,0,0) and total reward value $(1,-1,0)$. Hence starting state 1 is more favourable for player 1, which can not be deviated from the average reward. Another motivating example is

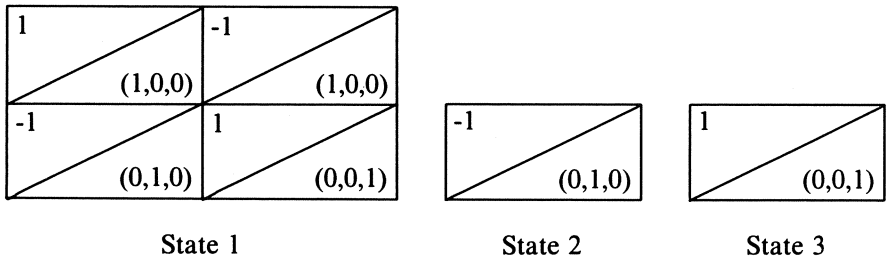


Average reward (0,0), while starting in state 1 (in state 2) means that player 1 (player 2) owns 2 half of the time and half of the time he owns 0.

Two further remarks concerning the total reward value have to be made. Firstly, the meaning of criterion (6) is not clear. Consider the following example:

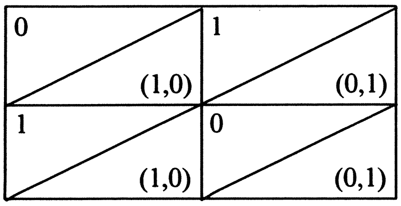


For all pairs of strategies (in fact there is only one pair) we have $T^{\infty}(1,\pi_1,\pi_2)=T^{\infty}(2,\pi_1,\pi_2)=-\infty$, since with probability one a realisation will occur for which $\liminf_{T\to\infty}\Sigma_{\tau=0}^{T}\,r(s_\tau,a_{1\tau},a_{2\tau})=-\infty$. The "normal" outcome for this game seems to be $(1,-1)$. Secondly, for criterion (5) not every game needs to have a value. Consider for instance the famous "big match" of Blackwell and Ferguson (1968):

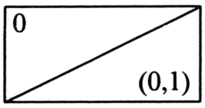


For state 1 the average reward value equals 0. It can be verified that $-\infty=\sup_{\pi_1}\inf_{\pi_2}T(1,\pi_1,\pi_2)\neq\inf_{\pi_2}\sup_{\pi_1}T(1,\pi_1,\pi_2)=0$

The next example shows that the total reward value may be infinite even if the average reward value is 0 for all states.



State 1



State 2

This phenomenon is due to the fact that for this game player 2 has no optimal stationary strategy (though there are ϵ-optimal stationary strategies and even optimal Markov strategies).

The above remarks lead to the conclusion that criterion (6) is not applicable to stochastic games and that criterion (5) can be examined for games for which properties 1 and 2 below hold:

Property 1. The value of the stochastic game equals 0 for each starting state.

Property 2. Both players possess optimal stationary strategies.

These two properties combined induce that both $\sup_{\pi_1} \inf_{\pi_2} T(s,\pi_1,\pi_2)$ and $\inf_{\pi_2} \sup_{\pi_1} T(s,\pi_1,\pi_2)$ are finite for each starting state s (cf. Vrieze and Thuijsman (1987)).

4. Properties of total reward stochastic games

From now on we assume Properties 1 and 2 to hold. If one player fixes a stationary strategy, then the other player has a pure stationary strategy as a best reply with respect to the total reward criterion. Most proofs of theorems in total reward games make use of this fact.

The following theorem relates total reward games to discounted games (Thuijsman (1989), p. 73).

Theorem *If for a zero-sum stochastic game the total reward value v_T exists and if both players have total reward optimal stationary strategies, then $v_T = \lim_{\beta\uparrow 1} v_\beta$, where v_β, $\beta \in (0,1)$, equals the β-discounted reward value.*

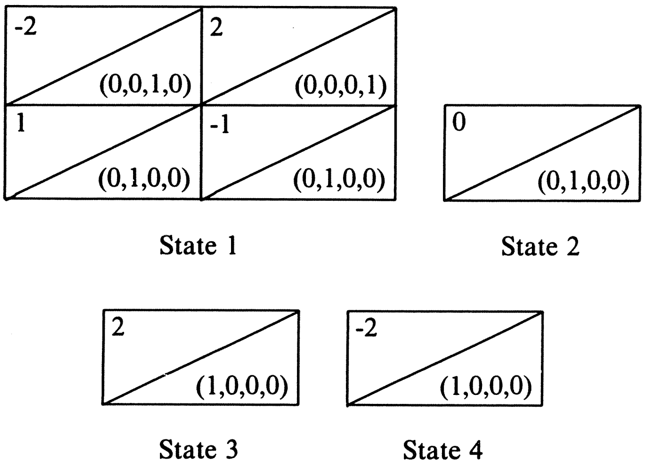
The proof of this theorem is based on the next inequality, which holds for any stochastic game: $\liminf_{\beta\uparrow 1} D_\beta(s,\pi_1,\pi_2) \geqq T(s,\pi_1,\pi_2)$. The theorem can be weakened in the sense that only the existence of ϵ-optimal stationary strategies are needed.

The following theorem gives a sufficient (but not necessary) condition for the value of the total reward game to exist.

Theorem *If for a stochastic game both players possess uniform β-discounted optimal strategies σ_1^* and σ_2^* (i.e. β-discounted optimal for all β close to 1), then* *the total reward value exists and σ_1^* and σ_2^* are total reward optimal.*

The proof of this theorem is based on the fact that uniform, discount optimal strategies are also average optimal.

In Thuijsman and Vrieze (1987) an example is elaborated showing that analogous to average reward games, it may happen that history dependent strategies are needed to assure ϵ-optimality. The game



has similar properties as the big match mentioned above, and it turns out that in state 1 player 1 essentially needs history dependent strategies to assure himself the total reward value 0 of that starting state.

This similarity of complexity of total reward games and average reward games is also reflected by the fact that a total reward game can be reformulated as an average reward game by enlarging the state space (becoming infinite) in a suitable way and redefining the payoffs and transitions (cf. Thuijsman and Vrieze (1987) or Thuijsman (1989), p. 85).

5. Some classes of total reward stochastic games

5.1. Contracting stochastic games

A contracting stochastic game is a stochastic game for which for each pair of stationary strategies (σ_1,σ_2) it holds that $\lim_{n\to\infty} P^n(\sigma_1,\sigma_2)=0$, related to the fact that the corresponding stochastic process is dying out. Usually this contraction assumption is given as the requirement of the existence of a vector $\mu\in\mathbb{R}^z$, $\mu>0$, and a number $\alpha\in(0,1)$ such that $\Sigma_t p(t|s,a_1,a_2)\,\mu_t \leqq \alpha\mu_s$, all s,a_1 and a_2. For games with a finite state space these two conditions are equivalent. Contracting stochastic games can be solved by using the contracting property of the value operator (cf. Van der Wal (1978)) leading to the existence of the value and of the existence of total reward optimal stationary strategies of the players. Two important subclasses of contracting stochastic games can be discerned. The first contain the stopping games, i.e. games for which $\Sigma_t p(t|s,a_1,a_2)<1$ for all s,a_1 and a_2. For each cell there is a strictly positive probability that the game will stop if that cell occurs. In fact, already Shapley (1953) considered stopping games, though often his paper is referred to as an introduction to discounted stochastic games. Discounted stochastic games form the second subclass of contracting stochastic games that ought to be mentioned. A discounted stochastic game can be reformulated as a stopping stochastic game by redefining the transition probabilities as $\beta p(t|s,a_1,a_2)$ giving constant stopping probabilities of $1-\beta$.

5.2. Nonnegative stochastic games

Consider the class of games for which $r(s,a_1,a_2)\geqq 0$ for all s,a_1 and a_2 and for which the properties 1 and 2 hold. We will give an abbreviated proof that games out of this class are solvable.

Properties 1 and 2 ensure that $v_1(s):=\lim_{\beta\uparrow 1} v_\beta(s)$ exists and is finite. The limit for β to 1 of the value equation of the discounted version leads to

$$v_1(s)= \underset{A_s\times B_s}{\text{val}} [r(s,.,.) + \sum_t p(t|s,.,.)v_1(t)] \tag{7}$$

Let $\sigma_2^*(s)$ be optimal for player 2 in (7) and let $\sigma_2^*=(\sigma_2^*(1),\sigma_2^*(2),\ldots,\sigma_2^*(z))$. By iteration we derive for each σ_1 from (7) (in matrix notation)

$$v_1 \geqq T(\sigma_1,\sigma_2^*) + Q(\sigma_1,\sigma_2^*)v_1 \geqq T(\sigma_1,\sigma_2^*) \tag{8}$$

(here $Q(\sigma_1,\sigma_2^*)$ is the Cesaro-limit of $P(\sigma_1,\sigma_2^*)$). The second inequality of (8) follows from the nonnegativity assumption which gives $v_\beta\geqq 0$ and hence $v_1\geqq 0$.

So σ_2^* assures player 2 that the payoff is at most v_1. Let $\sigma_{1\beta}^*$ be an optimal β-discounted stationary strategy for player 1. Take σ_2 arbitrary. In the ergodic classes of $P(\sigma_{1\beta}^*,\sigma_2)$ the average payoff is either 0 or strictly positive. In the first case $D_\beta(\sigma_{1\beta}^*,\sigma_2)=0$ and hence $v_\beta=0$ and $v_1=0$ and also $T(\sigma_{1\beta}^*,\sigma_2)=0$. In the second case $T(\sigma_{1\beta}^*,\sigma_2)=\infty$. We may conclude that in the recurrent states $\sigma_{1\beta}^*$ yields at least $v_1(s)$. For the transient states, by using the value equation for the discounted version, we can show that for these states $T(s,\sigma_{1\beta}^*,\sigma_2)\geqq v_1(s)-\epsilon$, where $\epsilon\to 0$ if $\beta\to 1$. It then can be concluded that $\sigma_{1\beta}^*$ for β close to 1 guarantees player 1 a payoff v_1 up to ϵ. Combining with (8) gives that v_1 is the total reward value of the game, that player 2 possesses optimal stationary strategies and that player 1 possesses ϵ-optimal stationary strategies.

An example in which player 1 for state 1 with total reward value 1 has no optimal stationary strategy is the following.

0 (0,0,1)	0 (1,0,0)
0 (0,1,0)	1 (0,1,0)

State 1

0 (0,1,0)

State 2

1 (0,1,0)

State 3

5.3. Recursive games

The last class we will examine is the class of recursive games. Recursive games, introduced by Everett (1957) and reconsidered by Orkin (1972) are total reward games with stopping probability 1 in every cell where a nonzero-payoff comes up. In the cells with payoff 0 the stopping probability is 0. Two types of realisations can occur for such games. Either once there is a nonzero-payoff and the game stops at that period, or the play goes on for ever with payoff 0 as the total reward of that play. Notice that the nonnegative game above is in fact a recursive game, since from both the cells with nonzero payoff the play jumps to the absorbing state 2 with payoff 0 from that period on, which has the same effect as stopping.

For recursive games Properties 1 and 2 always apply since for every pair of strategies the average payoff equals 0.

Everett (1957) considered recursive games with arbitrary state space. Orkin (1972) considered our model with a finite state space, however his prove is incomplete and recently Van der Neut (1989) gave an alternative proof, making extensive use of the underlying Markov chain properties. He showed that an ϵ-optimal stationary strategy for player 1 can be constructed in the following way: if for a state s $v_1(s)>0$, then take a β-discounted optimal action in such a state; if for a state s $v_1(s) \leqq 0$, then take an optimal action in the limit matrix game of the β-discounted value equation (cf. (7)). Analogously, ϵ-optimal stationary strategies for player 2 can be constructed.

7. Concluding remarks

In all the classes treated in section 5 it turns out that the total reward value equals the limit of the β-discounted reward value. Whether this is always the case is an open question. Also not solved is the question whether games for which Properties 1 and 2 hold always possess a total reward value. There are good indications that this will be the case. With the similarity between total reward games and average reward games (cf. Section 4) in mind an approach could run along the same lines as the proof of Mertens and Neyman (1980). The main difficulty one has to master is that the reformulated total reward game becomes an average reward game with a countable number of states.

References

D. Blackwell and T.S. Ferguson (1986), The big match, *Annals of Mathematical Statistics 39,* 159-163.

H. Everetts (1957), Recursive games. In: M. Dresher, A.W. Tucker and P. Wolfe (eds.) *Contributions to the Theory of Games III,* Annals of Math. Studies 39, Princeton University Press, 47-78.

A. Federgruen (1978a), On *n*-person stochastic games with denumerable state space, *Advances in Applied Prob. 10,* 452-471.

A. Federgruen (1978b), *Markovian Control Problems,* Ph.D. dissertation, Mathematical Centre, Amsterdam.

J.A. Filar and T.A. Schultz (1986), Nonlinear programming and stationary strategies in stochastic games, *Math. Programming 35,* 243-247.

A. Hordijk and L.C.M. Kallenberg (1981), Linear programming and Markov games I and II, In: O. Moeschlin and D. Pallaschke (eds.), *Game Theory and Mathematical Economics*, North-Holland, Amsterdam, 291-320.

J.F. Mertens and A. Neyman (1981), Stochastic games, *Int. Journal of Game Theory 10,* 53-66.

M. Orkin (1972), Recursive matrix games, *Journal of Applied Prob. 9,* 813-820.

T. Parthasarathy, S.H. Tijs and O.J. Vrieze (1984), Stochastic games with state independent transitions and separable rewards. In: C. Hammer and D. Pallaschke (eds.), *Selected Topics in Operations Research and Mathematical Economics,* Springer, Berlin, 262-271.

T.E.S. Raghavan, S.H. Tijs and O.J. Vrieze (1985), On stochastic games with additive reward and transition structure, *JOTA 47,* 451-464.

L.S. Shapley (1953), Stochastic games, *Proc. of the Nat. Ac. of Sciences USA 39,* 1095-1100.

F. Thuijsman (1989), *Optimality and Equilibria in Stochastic Games,* Ph.D. Dissertation, University of Limburg, Maastricht.

F. Thuijsman and O.J. Vrieze (1987), The bad match, a total reward stochastic game, *OR Spektrum 9,* 93-99.

Tijs and O.J. Vrieze (1980), Perturbation theory for games in normal form and stochastic games, *JOTA 30,* 549-567.

M. van der Neut (1989), ϵ-Optimal strategies in recursive games (in Dutch), University of Limburg, Maastricht.

J. van der Wal (1977), Discounted Markov games: successive approximations and stopping times, *Int. Journal of Game Theory 6,* 11-22.

J. van der Wal (1980), *Stochastic Dynamic Programming,* Ph.D. Dissertation, Technical University of Eindhoven, Eindhoven.

O.J. Vrieze (1981), Linear programming and undiscounted stochastic games in which one player controls transitions, *OR Spektrum 3,* 29-35.

O.J. Vrieze and F. Thuijsman (1987), Stochastic games and optimal stationary strategies, a survey. In: W. Domschke, W. Krabs, J. Lehn and P. Spellucci (eds.), *Methods of Operations Research 57,* 513-529.

O.J. Vrieze and S.H. Tijs (1980), Relations between the game parameters, value and optimal strategy space in stochastic games and construction of games with given solution, *JOTA 31,* 501-513.

Comparing fragmentary and integral production and inventory control concepts

P.J. Weeda
Department of Mechanical Engineering
University of Twente, Enschede

This contribution reflects a research development from an application of generalized Markov programming to a quantitative comparison of production and inventory planning and control concepts in use nowadays. The first topic has been part of the research the author has done under direction of the jubilaris at the Mathematical Centre. The second topic was started by the author about five years ago at the Department of Mechanical Engineering at the University of Twente. Recently the first dissertation on this subject has appeared.

Introduction

The field of goods flow management or logistics has experienced a vastly increasing interest during the past decade due to the recognition of its contribution to the profitability of enterprises. This paper adresses the part of the goods flow in manufacturing enterprises, extending from the supply of raw materials up to and including final product inventories. The second part of the goods flow, distribution and transport to the customers is not considered here.

In many manufacturing companies the goods flow is a complicated process involving a large assortment of final products with complex structures. During this process a product-dependent number of production facilities has to be passed in a product-dependent sequence. A well controlled goods flow is characterized by the way a compromise is obtained between conflicting goals as for example: low inventories, high service degree, smooth utilization of capacity, short throughput times, short reaction times and a fast realization of changes. These goals are extremely hard to achieve, mainly because of three reasons:

- the contrariety of the goals themselves,
- the complexity of the product(ion) structures,
- the uncertainties in market demand, production process and supply.

Hence a proper choice of production (and inventory) control concept is an extremely difficult one. Insight in the consequences of the application of such concepts in specific production situations is more than required.

The outline of this paper is as follows. Primarily, the compared production control concepts are reviewed. Secondly, a description of the basic production situations for the comparison is given. Thirdly the adopted method of comparison is explained. Finally, the results of the simulation experiments are discussed within a framework of characteristics and conjectures. Some specific numerical results are presented in order to show the kind of numerical results obtained by the simulation experiments. For a complete picture of the results and the conclusions the reader is referred to the dissertation [2].

Although this paper is about concepts for planning and control of production and inventory, the term 'production control concept' will be used.

THE COMPARED PRODUCTION CONTROL CONCEPTS

The production control concepts investigated are specific versions of

- Statistical Inventory Control (SIC),
- Base Stock Control (BSC),
- Manufacturing Resource Planning (MRP).

Mostly three types of production planning and control are distinguished with respect to the planning term they encompass. One may consider production planning and control in the long term, medium term or short term. Up until now, the comparison of the concepts has focussed on short term production planning and control aspects.

Short term production control is associated with the determination of production orders for final products (factory order generation), for non-final products (material order generation) and the assignment of production orders to capacity (scheduling). The three production control concepts differ among others in the way these three functions are performed.

Another important aspect of production control concepts are protective measures against various types of uncertainty. Finally the role of the computer is important, because of the required investments in information technology.

For the concept SIC, inventories are controlled independently. When to order and how much is specified by the replenishment policy, for example the well-known (s,Q)-policy. The values of the policy parameters for a given item are obtained in a way completely independent of the existing information about the inventory levels of other related items. For this reason, such a concept may be called a fragmentary control concept. It controls each inventory as though it is the only one in the world. The production control model for a single item, intensively studied in [1] may be called fragmentary for the same reason. A fragmentary inventory control concept like SIC implies also neglection of dependent demand relationships, existing between final products and their composing items.

Since SIC is an inventory concept by nature, no attention is paid to the question how replenishment orders are going to be produced. Production scheduling criteria are unspecified. Usually a FIFO priority rule is tacidly assumed. Another shortcoming of SIC is the fact that protective measures are restricted to safety stocks. Safety stock is incorporated in the reorder level specified by the replenishment policy.

Despite of these shortcomings SIC can be further 'dressed' to a more complete production control concept. It might even function quite reasonable, as simulation results indicate. In any case, expenses in hard- and software are negligable compared with more sophisticated production and inventory control concepts.

The major difference between BSC and SIC is the fact that inventory is replaced by the notion of echelon stock. The echelon stock of an item is defined as the number of units in stock of the item including the number of units of the item in successor stocks downstream the goods flow. The echelon position is defined to be equal to the echelon stock plus the number on order minus the number committed. The latter quantity includes final product demand received but not yet satisfied, multiplied by the multiplicity of the item in the final product. In this way, information on final product demand is not delayed as it is if SIC is applied. Moreover, BSC provides coordination between stock levels upstream the goods flow by means of the notion of echelon position.

The meaning of the three initials MRP is twofold. Originally it stands for Material Requirements Planning, a planning technique by means of which size and timing of production and supply orders of the items composing a final product are computed, given the delivery times and sizes of customer orders for final products as specified by a Master Production Schedule (MPS).

More recently Manufacturing Resource Planning has been developed. It is the integral planning and control concept most known. It encompasses planning and control of the goods flow on different terms. Moreover, it aims at an improved coordination of all departments involved in the goods flow. The short term part of Manufacturing Resource Planning consists of the Master Production Schedule, Material Requirements Planning, Capacity Requirements Planning and Shop Floor Control.

Contrary to SIC, and BSC to a less extent, MRP exploits dependent-demand relationships between final products and their composing items. The sizes of the item requirements are determined by explosion of final product demand. Their timing is obtained by backward scheduling, by means of which the release dates of the composing items are obtained by 'offsetting' the throughput time from their due dates. The gross requirements obtained by explosion, are balanced against the available stock of each item. This results in net requirements, which are compensated by planned orders, due to arrive at the appropriate time.

Capacity requirements implied by the timing and size of planned orders are compared with the available capacity. If necessary, material and capacity requirements planning is repeated until capacity utilization is sufficiently balanced.

Protection against uncertainties can be accomplished in MRP in three ways:

- including a safety stock in the net requirements,
- including a safety time in the throughput time by advancing due dates,
- overplanning of requirements on the final product level.

Contrary to SIC and BSC the role of the computer is essential in using

MRP. The basic characteristics of the three concepts are summarized in table 1 below.

concept	basic characteristics	factory order generation	scheduling	material order generation	protection against uncertainty	role of computer
SIC	* assumes independent demand * independently controlled stocks	make-to-stock	unspecified	supply-to-stock	safety stock	optional
BSC	* upstream co-ordination by echelon stocks	make-to-echelon-stock	unspecified	supply-to-echelon-stock	safety stock	optional
MRP	* exploits dependent demand relationships (explosion) * backward scheduling from due date on MPS	make due to expected net requirement	due date based	net calculation	safety time safety stock overplanning	essential

Table 1. Basic characteristics of Statistical Inventory Control, Base Stock Control and Manufacturing Resource Planning.

The production situations investigated

In this section the simulation model of the production situations is described on which the concepts are applied. Customer orders for final products arrive in accordance with a renewal process. The customer order sizes are mutually independent and identically distributed random variables. The lead time desired by customers is also a random variable. However in the simulations performed sofar, the customer order arrival process has been the only source of uncertainty. No uncertainty is assumed to exist with respect to customer order sizes, customer desired lead time, production process and supply.

Although within limits of size, any product network structure can be constructed by the simulation software. Sofar however, the experiments have been restricted to the structures depicted in figure 1.

Each vertex in the network reflects a relation between an item and its predecessor. To each vertex an explosion factor is assigned, specifying the number of units of a predecessor in the item considered. Values may be assigned to supplied items. To each vertex an added value may be assigned. In this way a value structure to each product network can be established. In the simulations performed sofar, explosion factors equal one. Moreover, supplied items have value one and all added values are zero. Lot sizes are equal to average demand per control interval.

The three network structures compared have some consequences regarding the uncertainty in the order arrivals. Primarily, the mean order arrival rates are taken to be equal for each structure. Since the convergent structure has only one final product, its mean arrival rate will be eight times those of the particular final products in the divergent and mixed structures. Concerning the

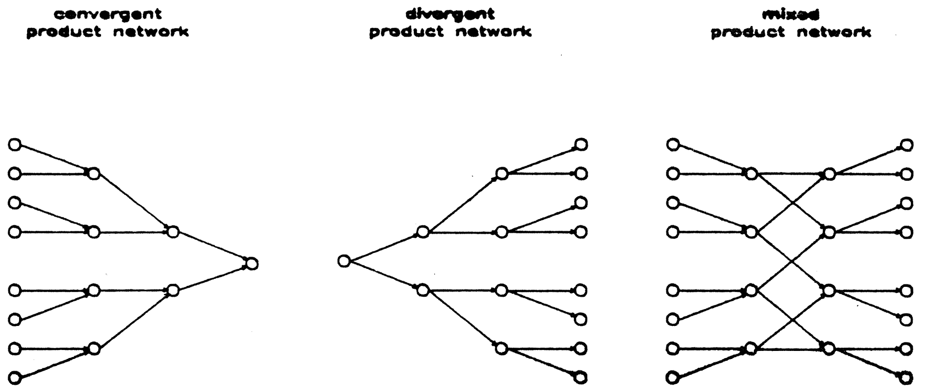


Figure 1. Three basic product structures.

variance in the arrival rate, one may take the view that this variance should be equal for the individual final products. This is guaranteed if the arrival times are Erlang 8-distributed for the single final item of the convergent structure, while for the other two product networks the arrival times of the particular final products are exponentially distributed. A serious alternative is to use one basic Poisson process for all three structures with identical random seed. For the divergent and mixed structure, an arrival of this basic process has probability 1/8 being an arrival of a particular final product. Both ways have been investigated.

An important consideration is the way non-supplied items are allocated to the production facilities. In most of the investigated production situations, the items on the same level in the network structure are assigned to one production facility as indicated in figure 2. The processing times are chosen such that the utilization of the production facilities are all equal. Set-up and transportation times have been equal to zero in the simulations performed sofar.

Although MRP has capacity utilization planning facilities only a rudimentary form of this kind of planning can be used in the simulation model. In make-to-order situations the release dates of the orders can be advanced by a fixed time interval in order to smooth short term capacity requirements. The corresponding time interval is called slack time. In make-to-stock situations a slack time is not appropriate since release dates are determined by the inventory policy.

Comparing concepts

The main objective of this research is to quantify the benefits of the concepts as a function of the production situation. The concepts are compared for identical production situations by means of computer simulation. In order to garantee a fair comparison an optimization is carried out with respect to the control variables for each concept and situation. Control variables are the safety stocks of the composing items. Safety stocks are absent on final product level, since final products are produced on order. In the simulations performed

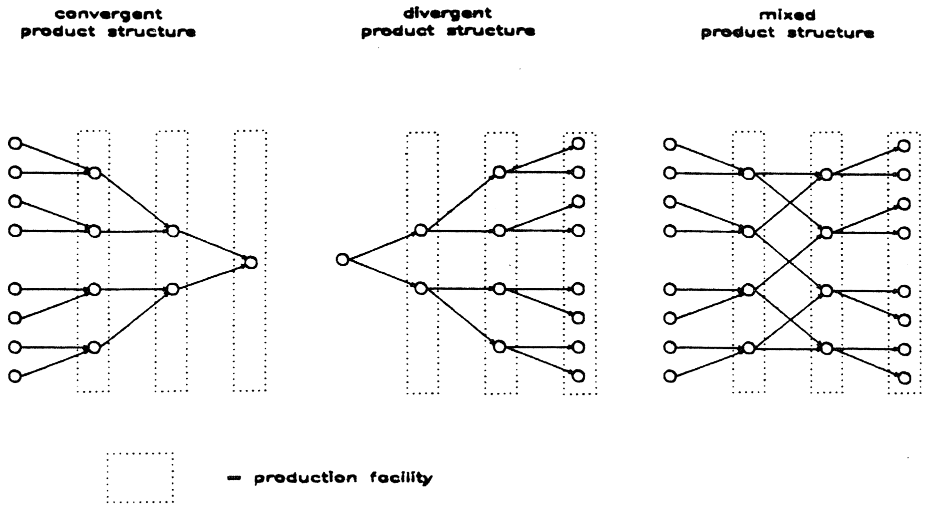


Figure 2. Three basic product(ion) structures.

the data have been chosen in such a way that items on the same level have identical characteristics. As a consequence the number of control variables is restricted to three safety stock levels in the three investigated product networks.

Optimization is performed with respect to two criteria. These criteria are service level and aggregate stock. The selected performance indicator for service level has been the fraction of customer orders delivered in time, accounted over all final products. By aggregate stock is meant average stock on hand, expressed in units of average demand per control interval. The value of an item equals the sum of the values of the composing items and the added value.

Since two performance indicators are involved, each simulation run with fixed sizes of the safety stocks results in a (performance) point in two-dimensional space. A performance point is called effective if neither a decrease in aggregate stock without decreasing the service level nor an increase in service level without increasing aggregate stock can be realized. For each concept and situation a set of effective points is obtained by enumeration.

In order to compare the performance of the concepts by a single number, the stock ratio is introduced. It is defined as the ratio between aggregate stock averaged over five specific service levels respectively given by .9, .95, .98, .99 and 1.0 for each concept.

Topics and conjectures

Sofar, the impact on the performance of the following topics has been investigated:

- control variables, like slack time, control frequency and priority rules,
- modifications in the market characteristics, like increased lumpiness in market demand, uncertainty in market demand rate and commercial lead time,

- characteristics of the product network, like degree of convergence (divergence) and added value,
- uncertainty in supply lead time,
- production to stock on final product level.

In order to provide a clear arrangement of results and conclusions, a framework of characteristics and conjectures has been developed. The results are discussed in view of the conjectures. For a complete discussion of these matters the reader is referred to [2]. Concept characteristics summarize advantages and shortcomings of the concepts. The following concept characteristics are employed:

- upstream coordination,
- lateral coordination,
- futurity,
- state-dependent physical safety stocks,
- scheduling criteria.

Upstream coordination is a form of coordination with the aim of tuning availability and requirements between an item and its successor(s). MRP provides this type of coordination. BSC provides it to a smaller extent. It is absent in SIC.

Lateral coordination is a form of coordination promoting simultaneous availability of items with a common successor. Its presence in the control concepts is similar to upstream coordination.

Futurity indicates to what extent information about the future is incorporated in a control concept. Futurity is only present in MRP.

State dependent safety stock reflects the dependence of the physical safety stock of an item on the stock levels of its successors. To a certain extent the stock of an item provides protection against stock-out of its predecessors. This concept characteristic is only present in BSC.

Scheduling criteria for priority determination of production orders. In contrast with MRP, scheduling criteria are not specified by SIC and BSC. For these concepts the FIFO priority rule is adopted.

The most important conjectures are summarized below:

- The degree of upstream coordination will increase in sequence of SIC, BSC and MRP. The advantage of upstream coordination will increase for an increasing number of levels in the product structure, increasing lumpiness and increasing demand rate uncertainty. It is expected to decrease for increasing control frequency and increasing supply lead time uncertainty.
- The degree of lateral coordination will increase in sequence of SIC, BSC and MRP. The advantage of lateral coordination will increase for increasing degree of convergence. It is expected to decrease for increasing supply uncertainty.
- The advantage of futurity will increase for increasing lumpiness, increasing demand rate uncertainty and increasing commercial lead time.
- The combination of futurity and upstream coordination results in the possibility of using due date based priority rules. For increasing degree of capacity sharing and decreasing control frequency, their impact will become more apparent.

Some specific results and conclusions

In this section a restricted selection of numerical results are presented. For a complete overview, the reader is once again referred to [2]. The impact of modification in the market characteristic lumpiness has been chosen.

Primarily the results of the basic production situation for the concepts SIC, BSC and MRP has been chosen. The basic production situation has been described above. It is shortly reviewed here. The three product(ion) networks depicted in figure 1 are investigated. The market characteristics are Erlang 8-arrivals for the single final product of the convergent structure and independent Poisson processes for the eight final products of the divergent and mixed structures. Customer lead time and order size are deterministic and equal for each final product. The results for the basic production situations are presented in table 2.

	Convergent	Divergent	Mixed
STR(SIC,MRP)	1.06	1.32	1.34
STR(BSC,MRP)	0.84	1.26	1.25

Table 2. Stock ratios for the basic production situations.

The stock ratio STR(SIC,MRP) expresses the average aggregated stock for SIC, summed over five specific service levels, divided by the same quantity for MRP. STR(BSC,MRP) expresses the same quantity for BSC with respect to MRP.

For the convergent product network the necessary stock for high delivery performance appears to increase in the sequence BSC, MRP, SIC. For the divergent and mixed product networks this sequence is MRP, BSC, SIC with the note that the difference between BSC and SIC is much smaller than the difference between MRP and the other two concepts.

The explanation of the fact that BSC outperforms MRP in the convergent case, has to be considered very carefully. The grid unit used for the optimization procedure is partly responsible for this result. Stock levels are expressed in units of the demand lot size. For BSC the demand lot size for an item equals the customer lot size multiplied by its multiplicity in the final product, in agreement with the notion of echelon stock position. For MRP and SIC the demand lot size for an item equals its average demand per time unit. For an item on level 2 of the convergent product network, this means for example that the demand lot size for MRP and SIC is eight times larger than for BSC. This factor equals two in the divergent and mixed cases. Hence a better finetuning of the stock levels in the optimization procedure can be obtained for BSC, if its demand lot size is smaller than the demand lot size for SIC and MRP.

In order to investigate the impact of increasing lumpiness in market demand the customer lot size has been multiplied by a factor eight. In order to keep the mean demand rate unchanged the arrival rate of customer orders is decreased by a factor eight. In order to keep the variance of the demand rate also unchanged compared with the basic production situation, the Erlang 8 arrival process for the convergent product network is replaced by a Poisson process. The Poisson process for customer orders arrivals in case of the mixed and divergent structure is replaced by an arrival process with hyperexponentially distributed arrival times. The results for increased lumpiness are depicted in table 3.

	Convergent	Divergent	Mixed
STR(SIC,MRP)	1.05	2.38	2.47
STR(BSC,MRP)	0.91	2.29	2.31

Table 3. Stock ratios for increasing lumpiness.

Table 3 shows that for the convergent network, increased lumpiness does not result in a significant improvement for MRP compared with SIC. For MRP compared with BSC there is a small improvement for MRP, caused by the fact that the demand lot size and customer lot size are equal now, because of the enlarged customer order lot size. The concept characteristic of BSC, state-dependent safety stock, can be the only reason for the fact that BSC still outperforms MRP for increased lumpiness in case of the convergent product network. Since demand lot sizes are equal for both concepts, a better finetuning of stock levels is the optimization procedure for BSC is out of the question in this case.

For the divergent and mixed product networks, the concept characteristics of MRP obviously garantee a better handling of lumpiness, compared with SIC and BSC.

Conclusions on the conjectures

The degree of upstream coordination is expected to increase in sequence of ROP, BSC and MRP. The results for the basic production situation show an improvement in performance in sequence of ROP, BSC, MRP, except for the convergent network where BSC is slightly better than MRP, taking into account the already discussed lot size effect.

Simulations with pure stock production instead of production on order on final product level, confirm the conjecture that an increasing number of levels favours upstream coordination. This is shown by the improvement of the performance of MRP. However an improvement of BSC with respect to SIC is absent. Hence BSC does not prove the expectations about its upstream coordination with respect to SIC. The results for increased lumpiness, exhibited in the preceeding section, yield similar conclusions.

The conjecture concerning the advantage of upstream coordination for increasing demand rate uncertainty, appears not to be monotonic. For increasing demand uncertainty the differences between the concepts increase, followed by a decrease. This throws up the additional conjecture that for large demand rate uncertainty, no control concept prevails over the others.

The conjecture that increasing control frequency reduces differences in upstream coordination, cannot be established. Increased control frequency causes similar stock reductions for the control concepts, resulting in an increase of the relative difference between the concepts.

The simulation results with respect to increased supply lead time uncertainty do not support the conjecture of decreasing advantage of upstream coordination. The performance of MRP shows a relative improvement with respect to SIC, a fact which can be explained by the difference between the replenishment mechanisms of MRP and SIC. However, the relative performance of MRP with respect to SIC increases in sequence of convergent, mixed, divergent. This confirms the conjecture that increased supply lead time uncertainty reduces lateral coordination. Indeed, joint availability of supplied items is reduced. The divergent product network with a single supplied item is in favour.

Other simulation results do not sufficiently confirm the advantage of lateral coordination. For example, it is expected to increase with increasing degree of convergence in the product network. However, the results for the divergent and mixed product network do not differ significantly.

Contrary to SIC and BSC, MRP has futurity. The advantage of futurity is expected to decrease for decreasing commercial lead time. This conjecture is confirmed by the simulation results, showing a relative improvement of SIC compared with MRP.

ACKNOWLEDGEMENT
Beste Gijs, veel dank voor de wetenschappelijke vorming die je mij hebt bijgebracht en voor je persoonlijke inzet die mij zeer heeft geïnspireerd!

REFERENCES

1. G. DE LEVE, H.C. TIJMS and P.J. WEEDA, *Generalized Markovian Decision Processes*, Applications, Mathematical Centre Tract 5, Amsterdam, 1970.
2. P.E.T. STRIEKWOLD, *A quantitative comparison of production and inventory control concepts*, Ph.D. Thesis, Department of Mechanical Engineering, University of Twente, Dec. 1988.
3. H. GRUENWALD, P.E.T. STRIEKWOLD and P.J. WEEDA, *A framework for quantiative comparison of production control concepts*, Int. J. Prod. Res., 1989, Vol. 27, No. 2, 281-292.

Capacity analysis of automatic transport systems in an assembly factory

W.H.M. Zijm
Centre for Quantitative Methods
Nederlandse Philips Bedrijven B.V.

and

Department of Mathematics and Computing Science
Eindhoven University of Technology

We describe a case study concerning the capacity analysis of a completely automated transport system in a flexible assembly environment. Basically, the system is modelled as a network of queues, however, due to its complex nature, product-form network theory is not applicable. Instead, we present an aggregation/decomposition procedure, i.e. we study a number of key elements of the system in detail after which the results are implemented in the overall model for the transport system. Among these key elements are a buffer transfer system (the bottleneck of the system), modelled as a sequence of deterministic, preemptive (identical) repeat priority queues, a set of elevators, modelled as vacation servers, and several conveyor belts, modelled as pure delays (infinite servers). In this short review, we focus on modelling aspects. In our view, modelling is an extremely important element of any practical application; nevertheless, the level of attention paid to modelbuilding in the literature is still limited.

1. Introduction and background

Changing market requirements for consumer products in the last twenty years have had a dramatic impact on the design and control of manufacturing processes, as well as on the logistic control of the materials flow in an industrial organization while furthermore new marketing strategies had to be developed. The old mass production philosophy for a market with a low product diversity is now rapidly replaced by new production control concepts which caused fundamental changes in the layouts and the organization of manufacturing processes. A high product diversity and the ever decreasing commercial product life cycles rule out the traditional mass production concepts, characterized by large batch sizes and high work-in-process and final inventory levels. Instead, we design factory layouts which can be characterized by small workcell organization structures (instead of long, dedicated assembly lines) where material is supplied via highly flexible automated transport systems which often are completely computer controlled. Parts may be transported one by one (i.e. we have transport batches of one single item), on product carriers which are coded; guided by a central control system each part follows its own route through the system. In this way, one attempts to avoid large stockpiles on the shopfloor.

Complex automated transport systems however require a thorough analysis in the design phase, in order to make sure that a desired capacity can indeed be realized. Queueing and blocking phenomena may reduce this transport capacity considerably, leading to idleness of workstations and hence ultimately to a severe loss of manufacturing efficiency. Both discrete simulation techniques and queueing network analysis have proved to be valuable tools in analyzing these systems (e.g. Solberg [1981], Coffman et al. [1988]).

In this paper, we describe a case study carried out in a vacuum cleaner factory of Philips in the Netherlands in which an approximate queueing network model has been developed to analyze a materials transport system. The system discussed here is actually a simplication of the one studied in the course of the consultancy project carried out in the factory, but the main objective of this paper is to outline some basic ideas, without getting stuck in technical details. Therefore, we focus on modelling aspects, also because we believe that the choice of the right model, which on the one hand reflects the important aspects of the system appropriately and which still can be analyzed on the other hand, is the key element in any practical application.

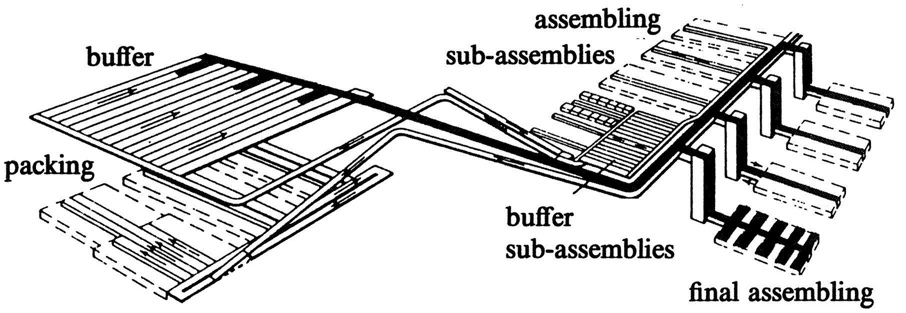


FIGURE 1. Layout of the vacuum cleaner assembly system.

In this section, we start with a brief description of the relevant transport and assembly operations in an assembly department for vacuum cleaners. The production environment can be characterized as follows.

- *Products*. The product range may be grouped into several main types. Within one type, items may vary on minor details; these minor variations do not play a role in our study and are therefore ignored.
- *Production*. Products are assembled in 40 workstations, grouped into four clusters or *workcells*. Within specified time periods, each workcell is dedicated to the assembly of exactly one product type.
- *Flow of material*. Material is supplied to the workstations in *kits*. A kit is a rectangular box, containing all the subassemblies (called *components* in the sequel) needed to assemble exactly one vacuum cleaner of a specific type.

Kits are loaded in a so called *kit preparation department* and then placed in a central buffer. Upon request of a work*cell*, a kit leaves the buffer system and proceeds, via a system of conveyor belts and elevators) on its way to that particular workcell. At the workcell, a kit waits in a small local buffer until one station in the cell is ready to serve the kit. After completing service, the vacuum cleaner is placed in the kit again which leaves the workstation (via *another* conveyor system) on its way to the packing area.

- *Control mechanism.* After completing an assembly operation, a kit leaves the system. At that moment, a signal is transferred to the central buffer area to release a new kit with components of a vacuum cleaner of the same type (an overall control mechanism assures that such kits are present indeed in the central buffer). If there are still kits waiting at the workcell, another assembly operation is started.

We study the behavior of the kit transportation system from the moment kits have received a call until the moment their service is completed. This includes the central buffer area (as far as 'called' kits are concerned), the conveyor systems and the elevators between central buffer area and workcells, and the workcells themselves (see Figure 1). Note that we are dealing with a *closed loop system* since each kit leaving a workcell is immediately 'replaced' by a new kit, containing components of the same product type. This situation leads in a natural way to a multiple type closed queueing network model, at least in principle (cf. Figure 2).

Unfortunately, the classical product-form network theory (Baskett et al. [1975], Reiser and Lavenberg [1981]) is not applicable. The assembly workstations operate according to a FCFS (First Come First Serve) discipline, with nonexponential service times. Transport times of the conveyor belts are deterministic while the elevators provide a kind of 'gated' service. Finally, certain traffic control rules imply (changing) priorities between several job types in the central buffer area.

In the next sections, we show how to overcome these difficulties. We propose models for key elements in the system. By selecting the right model for the central buffer system, we show that a detailed analysis of that system is possible, followed by a decomposition of the complete network into four cyclic queueing systems, one for each product type. Next, the elevators are handled after which an exponentialization approach completes the analysis.

2. The buffer transfer system

Figure 3 depicts the situation in the central buffer system. Kits are stored in a number of buffer lanes such that to each lane only one product type is allocated. After having received a call, a kit may try to depart from the buffer lane to the 'main road' by means of a special transfer, which, *as a part of the main road*, is located in front of the buffer lane. This transfer acts as a deterministic single server, giving absolute priority to kits already on the main road. Therefore, in order to allow the first kit to leave a buffer lane, a sufficiently large zone must be free at the main road (Figure 3 shows a free zone of length x for transfer 3).

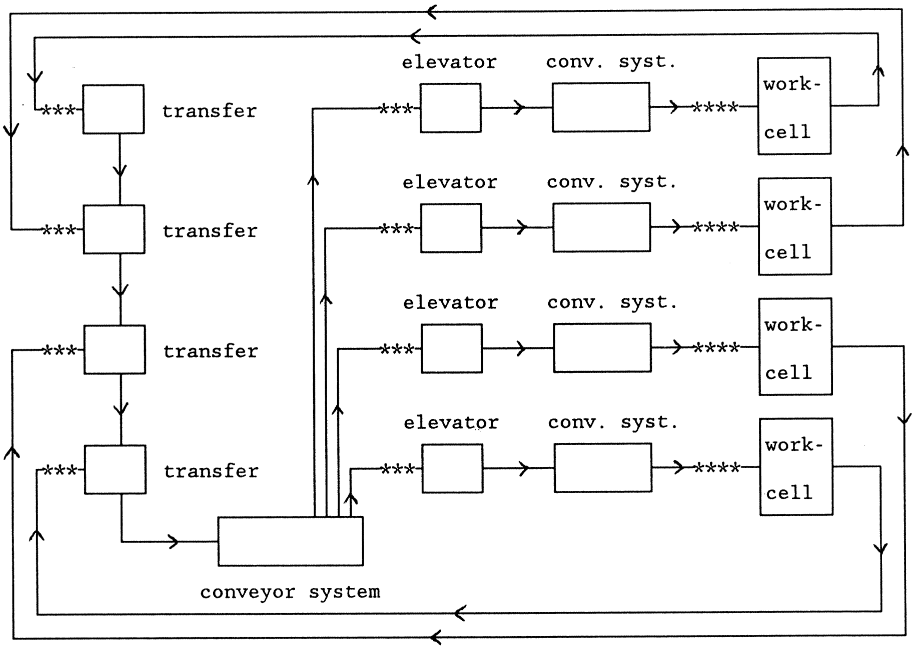


FIGURE 2. Queueing network model of the assembly system.

Let D be the service time for lower priority kits (coming from the buffer lane) and d the service time for higher priority kits (already on the main road. Modelling a transfer now as a nonpreemptive priority server would lead to a service requirement for higher priority kits equal to the time to travel a distance x, $t(x)$ say, since this is the time during which the transfer is blocked for use by lower priority kits. Clearly, $t(x) \geqslant D+d$. However, it is possible that two or more kits, already on the main road, to be in the free zone of length x *at the same time*. Moreover, free zones for subsequent transfers may overlap.

In order to overcome these difficulties, we model each transfer as a *preemptive (identical) repeat priority server* (compare e.g. Jaiswall [1968]). Instead of considering free zones, let the first lower priority kit in the buffer lane, which has received a call, enter the transfer as soon as this transfer is free. If, during this service, the kit is interrupted (physically, such an event would represent a collision) then the kit is returned to the head of the queue of lower priority

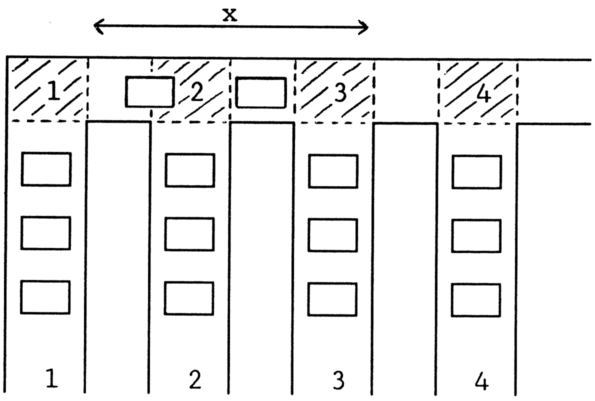


FIGURE 3. Buffer transfer system.

kits and *has to start its service all over again.* Some reflection shows that this way of modelling yields exactly the same results, in terms of throughput times, etc., as the original (physical) model. In particular, if the control mechanism requires a free zone of length x, then the service times of higher priority kits (already on the main road) in our preemptive repeat priority model have to be chosen equal to $t(x)-D$.

Now suppose that kits in buffer lane i receive calls according to a Poisson process (which was approximately true) with rate λ_i ($i = 1,2,3,4$). Clearly, the arrival of kits on the main road at transfer 2 is not a Poisson process (note that the departure process of the deterministic transfer 1 is not even a renewal process). However, observe that service of lower priority kits at transfer 2 is interrupted by time periods equal to a *busy period* of the preceding transfer, while the alternating idle periods of this transfer 1 are exponentially distributed with parameter λ_i. Hence, when a service of a lower priority job at transfer 2 is interrupted $\mathbf{N}_2$ times, we may define an 'aggregate' service time $\mathbf{S}_2$ (the time period starting with the first trial and ending with the final, successful, completion of the service of a lower priority job), which satisfies

$$\mathbf{S}_2 \,|\, \mathbf{N}_2 = \sum_{j=1}^{\mathbf{N}_2} (\mathbf{y}_{2j} + \mathbf{BP}_{1j}) + D \qquad (1)$$

where the $\mathbf{y}_{2j}$ are independent identically distributed random variables (the durations of the unsuccessful trials). The same holds for the $\mathbf{BP}_{1j}$ which are busy periods of transfer 1 (cf. Figure 4). The first two moments of $\mathbf{BP}_{1j}$ are easily calculated by taking derivatives in a well-known equation for the Laplace-Stieltjes transform of the random variable $\mathbf{BP}$ (see for example p. 229 in Cooper [1981]). Also, first and second moments of $\mathbf{y}_{2j}$ and $\mathbf{N}_2$ are easily calculated by exploiting the above mentioned exponentiality of the idle periods of transfer 1. It follows that the first two moments of $\mathbf{S}_2$ can be calculated. Next, a busy period of transfer 2 can be characterized completely (in terms of the first two moments) which determines in its turn the interruptions at transfer 3, etc. By iteration, we find aggregate service times for kits in each buffer lane.

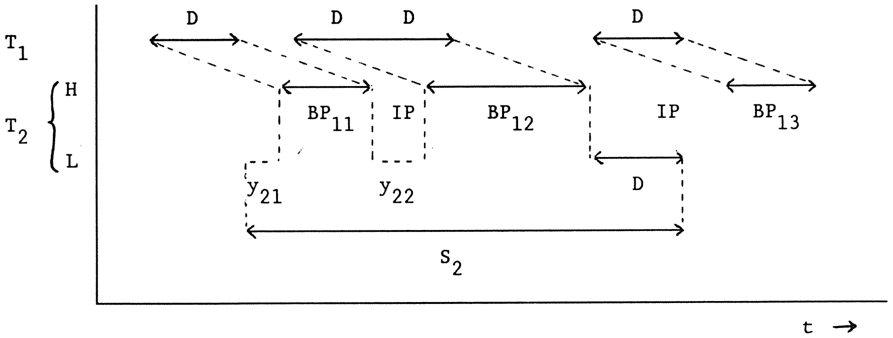


FIGURE 4. Trials and interruptions, leading to the 'aggregate' service time $\mathbf{S}_2$.

The above analysis is only slightly complicated by the observation that the first lower priority kit in a busy period of transfer 2 has an aggregate service time somewhat different from (1), caused by the fact that such a kit, having just received a call, may have to wait for a period equal to the residual lifetime of a busy period $\mathbf{BP}_1$ of transfer 1, even before starting its first trial. Call this residual lifetime $\mathbf{r}_1$ then with probability $\lambda_1 D$ the conditional aggregate service time of this first lower priority job equals

$$\mathbf{S}_2 \,|\, \mathbf{N}_2 = \mathbf{r}_1 + \sum_{j=1}^{\mathbf{N}_2} (\mathbf{y}_{2j} + \mathbf{BP}_{1j}) + D. \tag{2}$$

A rigorous analysis can be found in Repkes and Zijm [1988].

3. Decomposition of the network; modelling of elevators and workcells

Recall that each buffer lane stores only kits with components of one specific product type. Figure 2 shows that different product types are interfering only at the central buffer transfer system (described in Section 2) and on the conveyor belt thereafter. Since the latter one only acts as a pure delay (all arriving jobs start their, deterministic, service immediately) it is naturally modelled as a deterministic infinite server. This means in particular that also on the conveyor belt jobs do not interfere (which is clearly the case in reality). But the definition and subsequent analysis of the aggregate transfer service times implies in particular that all product types (kits coming from a certain buffer lane) can now be handled separately. For instance, note that, by incorporating busy periods of transfer 1 in the aggregate service times of kits in buffer lane 2, we only have to consider these latter kits in determining their average throughputtimes. Therefore, we end up with a decomposition leading to four cyclic networks (one for each product type) instead of the network depicted in

Figure 2. In the network for product type i, the transfer service time is taken equal to S_i, which can be obtained from the analysis outlined in Section 2.

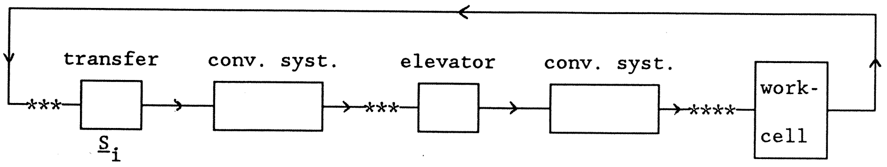


FIGURE 5. Queueing network for one product type after decomposition.

The elevators are continuously moving chain-elevators with product carriers (each carrier can hold at most one kit) at a certain distance which determines the entrance cycle time c, i.e. every c seconds a kit is allowed to enter the elevator. However, since the chain is continuously moving, it may happen that an arriving kit finding no other waiting kits in front of the elevator still has to wait for a certain *residual cycle time*, before a carrier is positioned in front of it and entrance is allowed. Therefore, a vacation server model seems to be appropriate. We do not study these models here in detail, the interested reader is referred to Cooper [1981], see also Doshi [1986].

The continuous movement of the elevator chain is needed because several other kits, which already entered the elevator and are on their way downstairs, would otherwise be delayed. Typically, we see two or three kits in an elevator at the same time, each on a carrier, at a distance which, measured in seconds, is a multiple of c. Hence, actually only the entrance mechanism of the elevator is modelled as a (single) vacation server (operating according to a FCFS discipline), the latter part of the elevator can be integrated with the next conveyor system and be modelled as a pure delay, i.e. a deterministic infinite server, again.

Finally, we arrive at the workcells. Due to minor variations within one type range and to differences in speed between workers at the final assymbly workstations (the vacuum cleaners are manually assembled), the service times are not deterministic but have a coefficient of variation typically between 0.1 and 0.25. Since the call rate at each buffer lane is exactly the departure rate at the associated workcell, which stems from a superposition of 10 departure processes at the parallel workstations, we found that the assumption of the Poisson arrival process for the calls at the central buffer system is approximately justified. The workcells are modelled as multi-server queueing systems with a correctly tuned Coxian distribution.

4. Analysis of the cyclic network: the exponentialization method

In the cyclic network of Figure 5 we have now characterized all service stations. In particular, we have shown how to calculate aggregate service times for each product type at the central buffer transfer by incorporating the influence of other types. Besides a characterization of throughputtimes for each type separately (easily obtained from the Pollaczek-Khintchine formula) these aggregate service times also enable us to decompose the system into four cyclic networks, one for each product type, thus saving much computational effort (to say the least). The problem left is the analysis of these networks, which still is non-trivial, in particular since the different stations do not have exponential service times.

The problem is solved by applying an exponentialization method outlined in detail by Yao and Buzacott [1986] (similar methods have been proposed by Marie [1979]). The basic idea is to find a set of *state-dependent exponential service rates* at each station such that the *marginal steady state probabilities* at each workstation (workcell) equals the steady state probabilities which are obtained by analyzing that station (workcell) as an $M/G/c$ queue (see e.g. Chapter 4 in Tijms [1986]). The network with state-dependent service rates is analyzed by exploiting a specialized Mean Value Analysis technique due to Reiser [1981]. As a result, we obtain average throughputtimes and throughputs for all product types, dependent of the number of kits in the system (per product type).

The analysis of the vacuum cleaner assembly/transport system is now carried out as follows. Given a number of kits per product type in the system, we start with an estimate for the call rates λ_i to analyze the buffer transfer system (cf. Section 2). Next, we decompose and calculate performance indices for all four cyclic networks using the exponentialization method. The resulting throughputs for each product type are then taken as our next set of call rates λ_i, etc. We iterate until convergence occurs.

5. Conclusions

In this paper, it has been shown how a complicated assembly/transport system can be analyzed appropriately by exploiting a variety of queueing-theoretical models and techniques Numerical results justify the use of these models as a design tool for such systems. For details, see Repkes and Zijm [1988].

Acknowledgement

The author wishes to express his gratitude to Han Repkes. This paper is partly based on, and partly extends some joint work (Repkes and Zijm [1989]).

References

F. Basskett, K.M. Chandy, R.R. Muntz and F.G. Palacios [1975], *Open, closed and mixed networks of queues with different classes of customers*, Journal of the Association of Computing Machinery 22 (2), pp. 248-260.

E.G. Coffman, E. Gelenbe and E.N. Gilbert [1988], *Analysis of a conveyor queue in a flexible manufacturing system*, European Journal of Operational Research 35, pp. 382-392.

R.B. Cooper [1981], *Introduction to queueing theory*, Edward Arnold, New York.

B.T. Doshbi [1986], *Queueing systems with vacations - a survey*, Queueing Systems 1, pp. 29-66.

N.K. Jaiswall [1968], *Priority Queues*, Academic Press, New York.

R. Marie [1979], *An approximate analytical method for general queueing networks*, IEEE Transactions of Software Engineering SE-5, pp. 530-538.

M. Reiser [1981], *Mean-value analysis and convolution method for queue-dependent servers in closed queueing networks*, Performance Evaluation 1, 7-18.

M. Reiser and S.S. Lavenberg [1980], *Mean Value Analysis of closed multichain queueing networks*, Journal of the Association of Computing Machinery 22, pp. 313-322.

J.P. Repkes and W.H.M. Zijm [1989], *Queueing network analysis of an assembly factory*, CQM Report, in preparation.

J.J. Solberg [1981], *Capacity planning with a stochastic workflow model*, AIIE Transactions 13, pp. 116-122.

H.C. Tijms [1986], *Stochastic modelling and analysis: a computational approach*, Wiley, Chichester.

D.D. Yao and J.A. Buzacott [1986], *The exponentialization approach to flexible manufacturing system models with general processing times*, European Journal of Operational Research 24, pp. 410-416.

MATHEMATICAL CENTRE TRACTS

1 T. van der Walt. *Fixed and almost fixed points.* 1963.

2 A.R. Bloemena. *Sampling from a graph.* 1964.

3 G. de Leve. *Generalized Markovian decision processes, part I: model and method.* 1964.

4 G. de Leve. *Generalized Markovian decision processes, part II: probabilistic background.* 1964.

5 G. de Leve, H.C. Tijms, P.J. Weeda. *Generalized Markovian decision processes, applications.* 1970.

6 M.A. Maurice. *Compact ordered spaces.* 1964.

7 W.R. van Zwet. *Convex transformations of random variables.* 1964.

8 J.A. Zonneveld. *Automatic numerical integration.* 1964.

9 P.C. Baayen. *Universal morphisms.* 1964.

10 E.M. de Jager. *Applications of distributions in mathematical physics.* 1964.

11 A.B. Paalman-de Miranda. *Topological semigroups.* 1964.

12 J.A.Th.M. van Berckel, H. Brandt Corstius, R.J. Mokken, A. van Wijngaarden. *Formal properties of newspaper Dutch.* 1965.

13 H.A. Lauwerier. *Asymptotic expansions.* 1966, out of print; replaced by MCT 54.

14 H.A. Lauwerier. *Calculus of variations in mathematical physics.* 1966.

15 R. Doornbos. *Slippage tests.* 1966.

16 J.W. de Bakker. *Formal definition of programming languages with an application to the definition of ALGOL 60.* 1967.

17 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 1.* 1968.

18 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 2.* 1968.

19 J. van der Slot. *Some properties related to compactness.* 1968.

20 P.J. van der Houwen. *Finite difference methods for solving partial differential equations.* 1968.

21 E. Wattel. *The compactness operator in set theory and topology.* 1968.

22 T.J. Dekker. *ALGOL 60 procedures in numerical algebra, part 1.* 1968.

23 T.J. Dekker, W. Hoffmann. *ALGOL 60 procedures in numerical algebra, part 2.* 1968.

24 J.W. de Bakker. *Recursive procedures.* 1971.

25 E.R. Paërl. *Representations of the Lorentz group and projective geometry.* 1969.

26 European Meeting 1968. *Selected statistical papers, part I.* 1968.

27 European Meeting 1968. *Selected statistical papers, part II.* 1968.

28 J. Oosterhoff. *Combination of one-sided statistical tests.* 1969.

29 J. Verhoeff. *Error detecting decimal codes.* 1969.

30 H. Brandt Corstius. *Exercises in computational linguistics.* 1970.

31 W. Molenaar. *Approximations to the Poisson, binomial and hypergeometric distribution functions.* 1970.

32 L. de Haan. *On regular variation and its application to the weak convergence of sample extremes.* 1970.

33 F.W. Steutel. *Preservation of infinite divisibility under mixing and related topics.* 1970.

34 I. Juhász, A. Verbeek, N.S. Kroonenberg. *Cardinal functions in topology.* 1971.

35 M.H. van Emden. *An analysis of complexity.* 1971.

36 J. Grasman. *On the birth of boundary layers.* 1971.

37 J.W. de Bakker, G.A. Blaauw, A.J.W. Duijvestijn, E.W. Dijkstra, P.J. van der Houwen, G.A.M. Kamsteeg-Kemper, F.E.J. Kruseman Aretz, W.L. van der Poel, J.P. Schaap-Kruseman, M.V. Wilkes, G. Zoutendijk. *MC-25 Informatica Symposium.* 1971.

38 W.A. Verloren van Themaat. *Automatic analysis of Dutch compound words.* 1972.

39 H. Bavinck. *Jacobi series and approximation.* 1972.

40 H.C. Tijms. *Analysis of (s,S) inventory models.* 1972.

41 A. Verbeek. *Superextensions of topological spaces.* 1972.

42 W. Vervaat. *Success epochs in Bernoulli trials (with applications in number theory).* 1972.

43 F.H. Ruymgaart. *Asymptotic theory of rank tests for independence.* 1973.

44 H. Bart. *Meromorphic operator valued functions.* 1973.

45 A.A. Balkema. *Monotone transformations and limit laws.* 1973.

46 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 1: the language.* 1973.

47 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 2: the compiler.* 1973.

48 F.E.J. Kruseman Aretz, P.J.W. ten Hagen, H.L. Oudshoorn. *An ALGOL 60 compiler in ALGOL 60, text of the MC-compiler for the EL-X8.* 1973.

49 H. Kok. *Connected orderable spaces.* 1974.

50 A. van Wijngaarden, B.J. Mailloux, J.E.L. Peck, C.H.A. Koster, M. Sintzoff, C.H. Lindsey, L.G.L.T. Meertens, R.G. Fisker (eds.). *Revised report on the algorithmic language ALGOL 68.* 1976.

51 A. Hordijk. *Dynamic programming and Markov potential theory.* 1974.

52 P.C. Baayen (ed.). *Topological structures.* 1974.

53 M.J. Faber. *Metrizability in generalized ordered spaces.* 1974.

54 H.A. Lauwerier. *Asymptotic analysis, part 1.* 1974.

55 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 1: theory of designs, finite geometry and coding theory.* 1974.

56 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 2: graph theory, foundations, partitions and combinatorial geometry.* 1974.

57 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 3: combinatorial group theory.* 1974.

58 W. Albers. *Asymptotic expansions and the deficiency concept in statistics.* 1975.

59 J.L. Mijnheer. *Sample path properties of stable processes.* 1975.

60 F. Göbel. *Queueing models involving buffers.* 1975.

63 J.W. de Bakker (ed.). *Foundations of computer science.* 1975.

64 W.J. de Schipper. *Symmetric closed categories.* 1975.

65 J. de Vries. *Topological transformation groups, 1: a categorical approach.* 1975.

66 H.G.J. Pijls. *Logically convex algebras in spectral theory and eigenfunction expansions.* 1976.

68 P.P.N. de Groen. *Singularly perturbed differential operators of second order.* 1976.

69 J.K. Lenstra. *Sequencing by enumerative methods.* 1977.

70 W.P. de Roever, Jr. *Recursive program schemes: semantics and proof theory.* 1976.

71 J.A.E.E. van Nunen. *Contracting Markov decision processes.* 1976.

72 J.K.M. Jansen. *Simple periodic and non-periodic Lamé functions and their applications in the theory of conical waveguides.* 1977.

73 D.M.R. Leivant. *Absoluteness of intuitionistic logic.* 1979.

74 H.J.J. te Riele. *A theoretical and computational study of generalized aliquot sequences.* 1976.

75 A.E. Brouwer. *Treelike spaces and related connected topological spaces.* 1977.

76 M. Rem. *Associons and the closure statement.* 1976.

77 W.C.M. Kallenberg. *Asymptotic optimality of likelihood ratio tests in exponential families.* 1978.

78 E. de Jonge, A.C.M. van Rooij. *Introduction to Riesz spaces.* 1977.

79 M.C.A. van Zuijlen. *Emperical distributions and rank statistics.* 1977.

80 P.W. Hemker. *A numerical study of stiff two-point boundary problems.* 1977.

81 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 1.* 1976.

82 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 2.* 1976.

83 L.S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the AUTOMATH system.* 1979.

84 H.L.L. Busard. *The translation of the elements of Euclid from the Arabic into Latin by Hermann of Carinthia (?), books vii-xii.* 1977.

85 J. van Mill. *Supercompactness and Wallman spaces.* 1977.

86 S.G. van der Meulen, M. Veldhorst. *Torrix I, a programming system for operations on vectors and matrices over arbitrary fields and of variable size.* 1978.

88 A. Schrijver. *Matroids and linking systems.* 1977.

89 J.W. de Roever. *Complex Fourier transformation and analytic functionals with unbounded carriers.* 1978.

90 L.P.J. Groenewegen. *Characterization of optimal strategies in dynamic games.* 1981.

91 J.M. Geysel. *Transcendence in fields of positive characteristic.* 1979.

92 P.J. Weeda. *Finite generalized Markov programming.* 1979.

93 H.C. Tijms, J. Wessels (eds.). *Markov decision theory.* 1977.

94 A. Bijlsma. *Simultaneous approximations in transcendental number theory.* 1978.

95 K.M. van Hee. *Bayesian control of Markov chains.* 1978.

96 P.M.B. Vitányi. *Lindenmayer systems: structure, languages, and growth functions.* 1980.

97 A. Federgruen. *Markovian control problems; functional equations and algorithms.* 1984.

98 R. Geel. *Singular perturbations of hyperbolic type.* 1978.

99 J.K. Lenstra, A.H.G. Rinnooy Kan, P. van Emde Boas (eds.). *Interfaces between computer science and operations research.* 1978.

100 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 1.* 1979.

101 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 2.* 1979.

102 D. van Dulst. *Reflexive and superreflexive Banach spaces.* 1978.

103 K. van Harn. *Classifying infinitely divisible distributions by functional equations.* 1978.

104 J.M. van Wouwe. *Go-spaces and generalizations of metrizability.* 1979.

105 R. Helmers. *Edgeworth expansions for linear combinations of order statistics.* 1982.

106 A. Schrijver (ed.). *Packing and covering in combinatorics.* 1979.

107 C. den Heijer. *The numerical solution of nonlinear operator equations by imbedding methods.* 1979.

108 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 1.* 1979.

109 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 2.* 1979.

110 J.C. van Vliet. *ALGOL 68 transput, part I: historical review and discussion of the implementation model.* 1979.

111 J.C. van Vliet. *ALGOL 68 transput, part II: an implementation model.* 1979.

112 H.C.P. Berbee. *Random walks with stationary increments and renewal theory.* 1979.

113 T.A.B. Snijders. *Asymptotic optimality theory for testing problems with restricted alternatives.* 1979.

114 A.J.E.M. Janssen. *Application of the Wigner distribution to harmonic analysis of generalized stochastic processes.* 1979.

115 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 1.* 1979.

116 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 2.* 1979.

117 P.J.M. Kallenberg. *Branching processes with continuous state space.* 1979.

118 P. Groeneboom. *Large deviations and asymptotic efficiencies.* 1980.

119 F.J. Peters. *Sparse matrices and substructures, with a novel implementation of finite element algorithms.* 1980.

120 W.P.M. de Ruyter. *On the asymptotic analysis of large-scale ocean circulation.* 1980.

121 W.H. Haemers. *Eigenvalue techniques in design and graph theory.* 1980.

122 J.C.P. Bus. *Numerical solution of systems of nonlinear equations.* 1980.

123 I. Yuhász. *Cardinal functions in topology - ten years later.* 1980.

124 R.D. Gill. *Censoring and stochastic integrals.* 1980.

125 R. Eising. *2-D systems, an algebraic approach.* 1980.

126 G. van der Hoek. *Reduction methods in nonlinear programming.* 1980.

127 J.W. Klop. *Combinatory reduction systems.* 1980.

128 A.J.J. Talman. *Variable dimension fixed point algorithms and triangulations.* 1980.

129 G. van der Laan. *Simplicial fixed point algorithms.* 1980.

130 P.J.W. ten Hagen, T. Hagen, P. Klint, H. Noot, H.J. Sint, A.H. Veen. *ILP: intermediate language for pictures.* 1980.

131 R.J.R. Back. *Correctness preserving program refinements: proof theory and applications.* 1980.

132 H.M. Mulder. *The interval function of a graph.* 1980.

133 C.A.J. Klaassen. *Statistical performance of location estimators.* 1981.

134 J.C. van Vliet, H. Wupper (eds.). *Proceedings international conference on ALGOL 68.* 1981.

135 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part I.* 1981.

136 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part II.* 1981.

137 J. Telgen. *Redundancy and linear programs.* 1981.

138 H.A. Lauwerier. *Mathematical models of epidemics.* 1981.

139 J. van der Wal. *Stochastic dynamic programming, successive approximations and nearly optimal strategies for Markov decision processes and Markov games.* 1981.

140 J.H. van Geldrop. *A mathematical theory of pure exchange economies without the no-critical-point hypothesis.* 1981.

141 G.E. Welters. *Abel-Jacobi isogenies for certain types of Fano threefolds.* 1981.

142 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 1.* 1981.

143 J.M. Schumacher. *Dynamic feedback in finite- and infinite-dimensional linear systems.* 1981.

144 P. Eijgenraam. *The solution of initial value problems using interval arithmetic; formulation and analysis of an algorithm.* 1981.

145 A.J. Brentjes. *Multi-dimensional continued fraction algorithms.* 1981.

146 C.V.M. van der Mee. *Semigroup and factorization methods in transport theory.* 1981.

147 H.H. Tigelaar. *Identification and informative sample size.* 1982.

148 L.C.M. Kallenberg. *Linear programming and finite Markovian control problems.* 1983.

149 C.B. Huijsmans, M.A. Kaashoek, W.A.J. Luxemburg, W.K. Vietsch (eds.). *From A to Z, proceedings of a symposium in honour of A.C. Zaanen.* 1982.

150 M. Veldhorst. *An analysis of sparse matrix storage schemes.* 1982.

151 R.J.M.M. Does. *Higher order asymptotics for simple linear rank statistics.* 1982.

152 G.F. van der Hoeven. *Projections of lawless sequences.* 1982.

153 J.P.C. Blanc. *Application of the theory of boundary value problems in the analysis of a queueing model with paired services.* 1982.

154 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part I.* 1982.

155 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part II.* 1982.

156 P.M.G. Apers. *Query processing and data allocation in distributed database systems.* 1983.

157 H.A.W.M. Kneppers. *The covariant classification of two-dimensional smooth commutative formal groups over an algebraically closed field of positive characteristic.* 1983.

158 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 1.* 1983.

159 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 2.* 1983.

160 A. Rezus. *Abstract AUTOMATH.* 1983.

161 G.F. Helminck. *Eisenstein series on the metaplectic group, an algebraic approach.* 1983.

162 J.J. Dik. *Tests for preference.* 1983.

163 H. Schippers. *Multiple grid methods for equations of the second kind with applications in fluid mechanics.* 1983.

164 F.A. van der Duyn Schouten. *Markov decision processes with continuous time parameter.* 1983.

165 P.C.T. van der Hoeven. *On point processes.* 1983.

166 H.B.M. Jonkers. *Abstraction, specification and implementation techniques, with an application to garbage collection.* 1983.

167 W.H.M. Zijm. *Nonnegative matrices in dynamic programming.* 1983.

168 J.H. Evertse. *Upper bounds for the numbers of solutions of diophantine equations.* 1983.

169 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 2.* 1983.

CWI TRACTS

1 D.H.J. Epema. *Surfaces with canonical hyperplane sections.* 1984.

2 J.J. Dijkstra. *Fake topological Hilbert spaces and characterizations of dimension in terms of negligibility.* 1984.

3 A.J. van der Schaft. *System theoretic descriptions of physical systems.* 1984.

4 J. Koene. *Minimal cost flow in processing networks, a primal approach.* 1984.

5 B. Hoogenboom. *Intertwining functions on compact Lie groups.* 1984.

6 A.P.W. Böhm. *Dataflow computation.* 1984.

7 A. Blokhuis. *Few-distance sets.* 1984.

8 M.H. van Hoorn. *Algorithms and approximations for queueing systems.* 1984.

9 C.P.J. Koymans. *Models of the lambda calculus.* 1984.

10 C.G. van der Laan, N.M. Temme. *Calculation of special functions: the gamma function, the exponential integrals and error-like functions.* 1984.

11 N.M. van Dijk. *Controlled Markov processes; time-discretization.* 1984.

12 W.H. Hundsdorfer. *The numerical solution of nonlinear stiff initial value problems: an analysis of one step methods.* 1985.

13 D. Grune. *On the design of ALEPH.* 1985.

14 J.G.F. Thiemann. *Analytic spaces and dynamic programming: a measure theoretic approach.* 1985.

15 F.J. van der Linden. *Euclidean rings with two infinite primes.* 1985.

16 R.J.P. Groothuizen. *Mixed elliptic-hyperbolic partial differential operators: a case-study in Fourier integral operators.* 1985.

17 H.M.M. ten Eikelder. *Symmetries for dynamical and Hamiltonian systems.* 1985.

18 A.D.M. Kester. *Some large deviation results in statistics.* 1985.

19 T.M.V. Janssen. *Foundations and applications of Montague grammar, part 1: Philosophy, framework, computer science.* 1986.

20 B.F. Schriever. *Order dependence.* 1986.

21 D.P. van der Vecht. *Inequalities for stopped Brownian motion.* 1986.

22 J.C.S.P. van der Woude. *Topological dynamix.* 1986.

23 A.F. Monna. *Methods, concepts and ideas in mathematics: aspects of an evolution.* 1986.

24 J.C.M. Baeten. *Filters and ultrafilters over definable subsets of admissible ordinals.* 1986.

25 A.W.J. Kolen. *Tree network and planar rectilinear location theory.* 1986.

26 A.H. Veen. *The misconstrued semicolon: Reconciling imperative languages and dataflow machines.* 1986.

27 A.J.M. van Engelen. *Homogeneous zero-dimensional absolute Borel sets.* 1986.

28 T.M.V. Janssen. *Foundations and applications of Montague grammar, part 2: Applications to natural language.* 1986.

29 H.L. Trentelman. *Almost invariant subspaces and high gain feedback.* 1986.

30 A.G. de Kok. *Production-inventory control models: approximations and algorithms.* 1987.

31 E.E.M. van Berkum. *Optimal paired comparison designs for factorial experiments.* 1987.

32 J.H.J. Einmahl. *Multivariate empirical processes.* 1987.

33 O.J. Vrieze. *Stochastic games with finite state and action spaces.* 1987.

34 P.H.M. Kersten. *Infinitesimal symmetries: a computational approach.* 1987.

35 M.L. Eaton. *Lectures on topics in probability inequalities.* 1987.

36 A.H.P. van der Burgh, R.M.M. Mattheij (eds.). *Proceedings of the first international conference on industrial and applied mathematics (ICIAM 87).* 1987.

37 L. Stougie. *Design and analysis of algorithms for stochastic integer programming.* 1987.

38 J.B.G. Frenk. *On Banach algebras, renewal measures and regenerative processes.* 1987.

39 H.J.M. Peters, O.J. Vrieze (eds.). *Surveys in game theory and related topics.* 1987.

40 J.L. Geluk, L. de Haan. *Regular variation, extensions and Tauberian theorems.* 1987.

41 Sape J. Mullender (ed.). *The Amoeba distributed operating system: Selected papers 1984-1987.* 1987.

42 P.R.J. Asveld, A. Nijholt (eds.). *Essays on concepts, formalisms, and tools.* 1987.

43 H.L. Bodlaender. *Distributed computing: structure and complexity.* 1987.

44 A.W. van der Vaart. *Statistical estimation in large parameter spaces.* 1988.

45 S.A. van de Geer. *Regression analysis and empirical processes.* 1988.

46 S.P. Spekreijse. *Multigrid solution of the steady Euler equations.* 1988.

47 J.B. Dijkstra. *Analysis of means in some non-standard situations.* 1988.

48 F.C. Drost. *Asymptotics for generalized chi-square goodness-of-fit tests.* 1988.

49 F.W. Wubs. *Numerical solution of the shallow-water equations.* 1988.

50 F. de Kerf. *Asymptotic analysis of a class of perturbed Korteweg-de Vries initial value problems.* 1988.

51 P.J.M. van Laarhoven. *Theoretical and computational aspects of simulated annealing.* 1988.

52 P.M. van Loon. *Continuous decoupling transformations for linear boundary value problems.* 1988.

53 K.C.P. Machielsen. *Numerical solution of optimal control problems with state constraints by sequential quadratic programming in function space.* 1988.

54 L.C.R.J. Willenborg. *Computational aspects of survey data processing.* 1988.

55 G.J. van der Steen. *A program generator for recognition, parsing and transduction with syntactic patterns.* 1988.

56 J.C. Ebergen. *Translating programs into delay-insensitive circuits.* 1989.

57 S.M. Verduyn Lunel. *Exponential type calculus for linear delay equations.* 1989.

58 M.C.M. de Gunst. *A random model for plant cell population growth.* 1989.

59 D. van Dulst. *Characterizations of Banach spaces not containing l^1.* 1989.

60 H.E. de Swart. *Vacillation and predictability properties of low-order atmospheric spectral models.* 1989.

61 P. de Jong. *Central limit theorems for generalized multilinear forms.* 1989.

62 V.J. de Jong. *A specification system for statistical software.* 1989.

63 B. Hanzon. *Identifiability, recursive identification and spaces of linear dynamical systems, part I.* 1989.

64 B. Hanzon. *Identifiability, recursive identification and spaces of linear dynamical systems, part II.* 1989.

65 B.M.M. de Weger. *Algorithms for diophantine equations.* 1989.

66 A. Jung. *Cartesian closed categories of domains.* 1989.

67 J.W. Polderman. *Adaptive control & identification: Conflict of conflux?* 1989.

68 H.J. Woerdeman. *Matrix and operator extensions.* 1989.

69 B.G. Hansen. *Monotonicity properties of infinitely divisible distributions.* 1989.

70 J.K. Lenstra, H.C. Tijms, A. Volgenant (eds.). *Twenty-five years of operations research in the Netherlands: Papers dedicated to Gijs de Leve.*