M

Machine Learning

A term used in the artificial intelligence community to indicate automated improvement based on experience or empirical data in accomplishing a given task such as optimizing an objective function.

See

► Artificial Intelligence

MAD

Mean absolute deviation.

Maintenance

Maintenance is the support of successful system operation during long periods of usage by means of: (1) regular or sample check-ups; (2) planned or preventive replacement of the system's units; (3) failure diagnosis; and/or (4) spare units supply. Operations research models for a system maintenance analysis are represented mainly by optimization models for the improvement of system and equipment reliability.

For (1) and (2), one usually uses methods of controlled stochastic processes. For (3), one uses special methods based on mathematical logic, while (4) is considered in the scope of optimal redundancy and inventory control.

See

► Airline Industry Operations Research
► Inventory Modeling

References

Ushakov, I. A. (Ed.). (1994). *Handbook of reliability engineering*. New York: Wiley.

Makespan

► Scheduling and sequencing

Malcolm Baldrige Award

► Total Quality Management

Manhattan Metric

► Location Analysis

S.I. Gass, M.C. Fu (eds.), *Encyclopedia of Operations Research and Management Science*,
DOI 10.1007/978-1-4419-1153-7,

Manpower Planning

David J. Bartholomew
The London School of Economics and Political Science, London, UK

Introduction

Manpower (or, human resource) planning is concerned with the quantitative aspects of the supply of and demand for people in employment. At one extreme this might include the whole working population of a country, but it has been most successful when applied to smaller, more homogeneous systems like individual firms or professions. The term manpower planning appears to date from the 1960s though many of the ideas can be traced back much further. In recent years terms such as Workforce Panning and Personnel Planning have been used in the same sense. A history of the subject up to the 1980s,from a U.K. perspective, will be found in Smith and Bartholomew (1988). The literature of the subject is very scattered reflecting the diverse disciplinary origins of the practitioners, but most of the technical material is to be found in the journals of operations research, probability, and statistics. There was an initial surge of publication in the late 1960s and early 1970s and since then book length treatments include Grinold and Marshall (1977), Vajda (1978), and Bennison and Casson (1984). Bartholomew, Forbes and McClean (1991) gives a thorough coverage of the technical material and contains an extensive bibliography. Since then there has been a period of consolidation. The earlier theoretical work has largely proved adequate for practical needs, though there have been developments in closely related areas. See, for example, Kalamatianou and McLean (2003).

The essence of manpower planning is summed up in the aphorism that its aim is to have the right numbers of people of the right kinds in the right places at the right time. The basic approach is first to classify the members of a system in relevant ways. These will often be on the basis of such things as grade, salary level, sex, qualifications, and job title. The state of the system at any point in time can then be described by the numbers in these categories, often referred to as the stocks. Over time, changes occur as individuals join, leave the system or move within it. The numbers making these transitions are called the flows. The factors giving rise to change may be predictable or unpredictable but will include such things as individual decisions to leave, changes in demand for goods, management decisions on promotion or organizational structure and so on. The operations researcher's role is to describe and model the system as a basis for optimizing its performance.

Stochastic Models

The presence of uncertainty in so many aspects of the functioning of a manpower system means that any adequate model has to be stochastic. Two probability processes, in particular, have proved to be both flexible and realistic. These are the absorbing Markov chain and the renewal process. The former is appropriate in systems where the stocks are free to vary over time under the impact of constant flow rates, or probabilities. The art of successful application is to define the classification of individuals that all those within a category have approximately the same probability of moving to any other category. Loss from the system corresponds to absorption, and the theory of Markov chains can then be used to predict future stock numbers for various sets of transition probabilities. Later work has extended these methods by allowing the intervals between transitions to be random variables in which case a semi-Markov process or a Markov renewal process results.

When the numbers in the categories are fixed, as they often are when the categories are grades or based on job function, a different approach must be used. Transitions cannot then be regarded as generated by fixed probabilities, but arise in response to the occurrence of vacancies. The result is a replacement, or renewal process, where movement is driven by wastage (or the creation of new places). It was shown in Bartholomew, Forbes and McClean (1991) that the flows of vacancies could be modelled by a Markov chain in a manner very similar to that used for the modeling of the flows of people.

If a system is relatively small or if the rules governing its operation are complex, the only realistic way to model it may be to use a computer-based simulation model. The term simulation is commonly

used in two distinct senses in this context. Primarily it means that each individual movement is generated in the model by a random mechanism. Secondly, it is sometimes used of any algorithm for computing the aggregate properties of a system treated deterministically.

Forecasting and Control

Broadly speaking all models may be used in two modes for forecasting or control. In the early stages of a study one usually wishes to forecast the future state of the system if current trends continue. Next, it will usually be desirable to carry out a sensitivity analysis to explore the consequences of variations from present conditions. This leads on to questions of control where the question is how those parameters under management control should be chosen to achieve some desired goal. The distinction between forecasting and control can be illustrated using a simple form of the Markov model. According to this model successive vectors of expected stocks are related by an equation of the form

$$\boldsymbol{n}(T+1) = \boldsymbol{n}(T)\boldsymbol{P} + \boldsymbol{R}$$

where T represents time, $\boldsymbol{P}$ is a matrix of transition probabilities, and $\boldsymbol{R}$ is a vector of recruitment numbers. In forecasting mode, estimated or guessed values of $\boldsymbol{P}$ and $\boldsymbol{R}$ could be used to predict future values of $\boldsymbol{n}(T)$. In principle, $\boldsymbol{P}$ and $\boldsymbol{R}$ could both depend on T. In control mode, one would be asking how some or all of the elements of $\boldsymbol{P}$ and $\boldsymbol{R}$ should be chosen to attain a given $\boldsymbol{n}$ within a specified time. This gives rise to questions of attainability (whether the problem is solvable) and maintainability (whether an $\boldsymbol{n}$ can be maintained once it is reached). These matters have led to an interesting set of theoretical questions about the solvability of such problems in deterministic or stochastic environments. At a more practical level it has led to the formulation of optimization problems expressed in goal programming and/or network analysis terms (Gass 1991; Klingman and Phillips 1984).

The wastage flow (also known as attrition or turnover) is an important element in a manpower system both because it is highly variable and, largely, beyond the control of management. It has been intensively studied mainly through the survivor function or, equivalently, the frequency distribution of completed length of service. In practice the analysis is complicated by the fact that the data are usually censored and sometimes truncated also. This work has three main objectives: measurement, prediction, and gaining insight into the factors determining wastage.

The demand side of the manpower equation has proved to be less tractable. Demand for people is equivalent to the supply of jobs and this depends on technological, political, social, and economic factors many of which may be specific to particular organizations or industries. To take only one example, the demand for qualified medical manpower will depend on such varied things as demographic changes, the willingness of government or users of the service to pay, and the appearance and spread of new diseases like AIDS. The methods used have been, and have to be, as diverse as the fields of application. Because of the considerable uncertainties involved it is important to monitor constantly the changing environment and to adjust plans accordingly. A once-and-for-all plan has no place in manpower planning.

See

- ▶ Goal Programming
- ▶ Markov Chains
- ▶ Markov Processes
- ▶ Network

References

Bartholomew, D. J., Forbes, A. F., & McClean, S. I. (1991). *Statistical techniques for manpower planning* (2nd ed.). Chichester: Wiley.

Bennison, M., & Casson, J. (1984). *The manpower planning handbook*. London: McGraw-Hill.

Gass, S. I. (1991). Military manpower planning models. *Computers and OR, 18*(1), 65–73.

Grinold, R. C., & Marshall, K. T. (1977). *Manpower planning models*. New York/Amsterdam: North-Holland.

Kalamatianou, A. G., & McClean, S. (2003). The perpetual student: Modeling duration of undergraduate studies based on lifetime-type educational data. *Data Analysis, 9*, 311–330.

Klingman, D., & Phillips, N. (1984). Topological and computational aspects of preemptive multicriteria military personnel assignment problems. *Management Science, 30*, 1362–1375.

Smith, A. R., & Bartholomew, D. J. (1988). Manpower planning in the United Kingdom: An historical review. *Journal of Operational Research Society, 9*, 235–248.
Vajda, S. (1978). *Mathematics of manpower planning*. Chichester: Wiley.

Manufacturing

- ▶ Flexible Manufacturing Systems
- ▶ Operations Management
- ▶ Production Management

MAP

Markov arrival process.

See

- ▶ Matrix-Analytic Stochastic Models

Marginal Value

The marginal value is the extra cost of producing one extra unit of output. Similarly, marginal revenue is the extra revenue resulting from selling an extra unit of goods. From the economics of a firm, when marginal revenue equals marginal costs, the firm is in an equilibrium optimal condition in terms of maximizing profits. Depending on the application, the dual variables of a linear-programming problem can be interpreted as marginal values. The economic interpretation of the dual variables is complicated by alternate optimum solutions (corresponding to different bases) that may yield different values of the dual variables. Thus, there may be two or more marginal values for the same constraint. Such multiple values must be interpreted with care.

See

- ▶ Dual Linear-Programming Problem
- ▶ Duality Theorem

Marketing

Yoram (Jerry) Wind[1], Eric T. Bradlow[1], Jehoshua Eliashberg[1], Gary L. Lilien[2], Jagmohan Raju[1], Arvind Rangaswamy[2] and Berend Wierenga[3]
[1]University of Pennsylvania, Philadelphia, PA, USA
[2]The Pennsylvania State University, University Park, PA, USA
[3]Erasmus University Rotterdam, Rotterdam, The Netherlands

Introduction

Marketing offers a rich and unique domain for applications of operations research (OR) methods, models, and approaches. Not only does the marketing area offer opportunities to develop and apply OR models and methods to increasingly important decisions affecting ALL companies, nonprofits, governments, societies, and individuals, but also unique opportunities to further the much needed collaboration between academics and practitioners, and for bridging the silos between marketing and the other management disciplines and functions.

Since customers (individuals or groups) are at the heart of the marketing system, OR modeling approaches help characterize, understand, and predict their behaviors. For consumers and organizational buyers, that behavior involves the search for solutions to a want or desire, the screening or evaluation of alternatives, the selection of a best alternative, the act of purchase, the post-purchase feedback to the firm as well as to other customers and learning that affects future purchasing behavior. In fact, such applications of OR to marketing problems have become even more prevalent, with website morphing (Hauser et al. 2009), Netzer's work on optimal email campaigns, and optimal in-store movement using the traveling salesman paradigm (Hui et al. 2009).

Firms and other non profit organizations (such as museums, politicians, government organizations) capitalize on that knowledge or model of individual behavior by focusing on such decisions as product/service design, pricing, distribution, promotion, advertising, personal selling, and the likely customer responses to them. In addition, at a higher level, these

decisions must be integrated and coordinated with the activities of other management functions (finance, manufacturing, R&D, etc.) and linked to other product and market decisions of the organization, including the critical resource allocation decisions among products, markets, distribution options, and businesses. Such critical decisions are evaluated based on their return on investment (ROI) under alternative scenarios reflecting different views of the future.

The external scenarios range from pessimistic views of recurring financial crisis, catastrophic natural disasters, continued terrorist activities and political unrest around the world, through continuation of the status quo, to optimistic scenarios of growth and prosperity driven by the fast growing economies of Asia and a recovery of the West. For marketers, these scenarios lead to consideration of strategic alternatives derived from a narrow view of modeling, e.g., the impact of a specific marketing activity (such as advertising expenditures) through an integrated view of all marketing touch points and product/service/solutions/customer experience, to the design of full strategy integration across the various management functions, incorporating multiple short and long-term performance measures.

Background

The American Marketing Association defines marketing as: "... the process of planning and executing the conception, pricing, promotion, and distribution of ideas, goods, and services to create exchanges that satisfy individual and organizational objectives."

As a management function, marketing includes such activities as advertising, sales and marketing research. Or, more simply put, marketing's organizational role is the interface between a firm and its customers. It is also a critical participant in cross-functional processes aimed at developing and launching new products and services that create customer value, i.e., products and services that customers want.

As a philosophy, marketing views the need to understand, anticipate and meet customer needs as the key to organizational success. As such, the customer is the final arbitrator of the value of any product or service offering. Marketing philosophy also extends the concept of customer orientation to internal customers and other stakeholders.

Thus, marketing is concerned with anticipating and understanding human needs and wants and translating those needs and wants into the demand (as economists use the term) for products and services. Those needs and wants are satisfied with products and services that are increasingly being developed in collaboration with empowered consumers. Businesses that exemplify this view include Build-a-Bear, Dell, and others offering opportunities for customization of the products and services, as well as firms that now scrape blogs, discussion forums, and other user-generated content to bring the digital voice of the customer into the firm, and help determine the appropriate responses (Ghose and Han 2011).

Products and services have functional as well as image characteristics. They are made available to the customer through a variety of channels ranging from physical retail stores to online websites, to mail order to social network platforms (e.g., Facebook). In order to effect an exchange, individuals have to be aware of, emotionally engaged, and understand the product (through advertising or other communication media), find the product worth their money (by comparing the product's total cost — its purchase price adjusted by any promotional offerings plus the cost of maintaining, using and disposing of the product — with the benefits promised in terms of performance and image), and participate in the exchange process.

While historically marketing models of behavior saw a product's value as consisting of the sum of the utilities of the features and benefits of which it is comprised (Green et al. 1973), and that is still a significant part of marketing modeling, newer conceptual models also take into account the perceived value of others, the recommendation of others, and the ability to share those experiences with others via one's social network (Stephen and Toubia 2010), or via the network externalities generated by other adopters.

Exchanges have typically been aggregated to the context of a market segment, which consists of the customers sharing a particular and similar need and who are willing to engage in exchange to satisfy that need. However, it is no longer uncommon to see exchange activities take place between the firm and individual consumers rather than at the level of a market segment (which represents higher level of aggregation). In essence, technology has allowed marketing in the 21st century to be infinitely tailored

because of the wealth of individual-level data that is now tractable due to the advances in the interactive media, and consumers' motivation and ability to customize the offerings.

OR Marketing Model Types

OR in marketing helps decision makers by harnessing measurement models and theoretical models and embedding them within a decision model (or more generally, within a decision-support system). The corresponding models are called **measurement models**, **stylized theoretical models**, and **decision-making models**, respectively (although it may be equally helpful to interpret these categories as classification dimensions for interpreting the multiple purposes of models).

Measurement Models — The purpose of measurement models is to describe and predict a current or anticipated either an individual consumer or the market reaction to a product or service as a function of various independent variables. The phrase "market reaction" here should be interpreted broadly. It is not necessarily units demanded but could be some other related variables. For example, in Guadagni and Little's (1983) model, the dependent (reaction) variable is the probability that the individual will purchase a given brand on a given purchase occasion. Choice models often have several independent variables including whether the brand was on sale (deal) at a given purchase occasion, regular price of the brand, deal price (if any), brand loyalty of the individual, etc. In addition, sometimes the focus of such models may be on certain variables preceding the steady-state demand (e.g., awareness, first-trial, repeat purchase). These examples suggest that measurement models can deal with individual (disaggregate) demand or aggregate (segment or market-level) demand as well as transitory or steady-state demand. Note that advances in measurement models can be due to better data (e.g., scanner data) or better estimation methods and procedures (maximum-likelihood methods for generalized logit models, for example). In traditional marketing problems such as customer satisfaction and customer-defined quality, OR measurement models have greatly enhanced the relatively simplistic survey-based approaches to the measurement of these constructs. Relying on advances in structural equation modeling. as well as the new area of empirical industrial organization, allows researchers to address more realistic and rich problems, such as competitive pricing behavior in markets with a large number of products (e.g., Sudhir 2001).

Stylized Theoretical Models — The purpose of stylized theoretical models is to explain and provide insights into marketing phenomena: a stylized theoretical model typically begins with a set of assumptions that describes a particular marketing environment. Some of these assumptions may be purely mathematical, but are also intuitively logical with the objective of making the analysis tractable. Others are substantive assumptions with real empirical grounding. Two well-known theoretical modeling efforts are Bell, Keeney and Little (1975), who show what functional forms of market share models are consistent with a certain set of reasonable criteria, and Basu et al. (1985), who show what form of sales force compensation plan is optimal under a set of assumptions about firm and salesperson objectives and behavior.

Such stylized theoretical models have helped improve the ability to design optimal product lines, issues related to specialization versus vertical integration (McGuire and Staelin 1983), aligning the incentives between manufacturers and retailers (Jeuland and Shugan 1983), designing pricing strategies for traditional goods, and also information goods. Stylized models have helped improve how companies offer short-term price discounts (Raju et al. 1990), how such short-term price discounts pass through to the consumer, and how retailers might improve their private label offerings. As marketing systems evolved, especially with the advent of new technologies, such stylized models have significantly improved understanding of new platforms and mechanisms for interactions between buyers and sellers. Stylized theoretical models have also helped in the understanding of the role brands play in a competitive market, including the symbolic role that brands play in social interactions, and how firms may improve their advertising and communications strategies (Chen et al. 2009).

While the emphasis in this work is on developing stylized theoretical models, most work in this area also rigorously tests the ability of these models to predict firm and market behavior. Recent empirical work in

Marketing, Table 1 The frontiers of decision models (DM)

	DM frontiers today	DM frontiers tomorrow
Time Scale	Days and weeks, if not months	Moving toward real time in data entry, data access, data analysis, implementation, and feedback
Focus of DM	Support strategic decisions	Support both strategic and operational decisions
Mode of Operation	Individual and PC-centric	Organization and Network centric – support multiple employees in multiple locations on multiple devices
Decision Domain	Marketing	Marketing and other functions, such as Supply Chain and Finance
Company Interface	Loosely coupled to company's IT systems	Woven into IT-supported company's operations and decision processes
Intervention Opportunities	Discrete, Problem-driven	Continuous, Process-driven
DM Goal	Support analysis and optimization	Support robust and adaptive organizational decision processes
DM System Design	As a tool to understand information and enhance decisions	As tool to enhance productivity and success of business models
DM System Operation	Interactive (User interacts with model)	Interactive as well as autonomous (embedded)
DM Outputs	Recommended actions; What if analyses	Visualization of markets and their behavior (e.g., Dashboard), Extended reality (e.g., Business model simulation), Explanation (Why?), Automated implementation (e.g., create alerts, automate actions)
DM Implementation Sequence	Intervention Opportunity → Implementation of decisions → Integration with IT Systems	Integration with IT → Intervention Opportunity → Implementation of decisions

the structural economics also contains stylized theoretical models where observed outcomes are assumed to arise from equilibrium actions taken by agents, modulo stochastic error (Dube et al. 2010). In this manner, joint theory and empirical work has begun to play a larger role. Distinguishing features of stylized theoretical models, especially the ones that use economic modeling and game-theory as tools, are that they explicitly recognize that companies must make decisions in a competitive environment and recognize that they compete with other firms who also are capable of making sound decisions. It is through this explicit recognition that these models are able to provide companies with a theoretically sound and empirically grounded means of improving strategic marketing decisions.

Decision-Making Models — These models are designed to directly help marketing managers make better decisions. They incorporate measurement models as building blocks, but go beyond measurement models in recommending specific actions (e.g., optimal marketing-mix decisions) for the manager. The techniques used to derive the optimal policies vary across applications, and include calculus, dynamic programming, optimal control, and calculus of variations techniques, as well as linear and integer programming. These models have been developed for each marketing variable and for the entire marketing mix program (i.e., a product and service offering including pricing, distribution, etc.). Little's BRANDAID is a classical example of such a model. Lilien et al. (2011) elaborate on the impact such models have had.

Since 2000, many enhanced decision-making models have been developed that are embedded inside enterprise information systems. Examples include revenue management systems used by airlines and hotels and recommender systems used by web sites such as Amazon.com and netflix.com. Table 1, adapted from Lilien and Rangaswamy (2006), summarizes the many ways that decision models are evolving to provide enterprises with real-time and automated decision making capabilities.

The Emergence of Marketing Science

By most accounts, OR in marketing began its growth in the 1960s and 1970s. The literature used a variety of OR methods to address marketing problems: those

problems included product design/development decisions, distribution system decisions, sales force management decisions, advertising and mass communication decisions, and promotion decisions (Kotler 1971). The OR tools that were most prevalent in the 1960s and earlier included mathematical programming, simulation, stochastic processes applied to models of consumer choice behavior, response function analysis, and various forms of dynamic modeling (difference and differential equations, usually of the first order). Some uses of game theory were reported, but most models that included competition used decision analysis, risk analysis, or market simulation games.

Nearly three times the number of marketing articles appeared in the OR literature in the 1970s as appeared in the period from 1952 though 1969. In addition to the increase in the number of articles, reviews by Lilien and Kotler (1983) showed that a number of new areas had begun to emerge. These included descriptive models of marketing decisions, the impact of and interaction of marketing on organizational design, subjective decision models, strategic planning models, models for public and non-profit organizations, organizational buying models, and the emergence of the concept of the Marketing Decision Support System (MDSS). In addition, while the number of published articles rose dramatically, the impact on organizational performance did not appear to be equally significant, raising questions about effective implementation. Much of the literature of the 1970s pointed to the need to expand the domain of application. The "limitations" sections of some of the papers in the 1970s pointed out that many important phenomena that were being overlooked (such as competition, dynamics, and interactions amongst marketing decision variables) were both important and inherently more complex to model. Hence, the level of model complexity and the insightfulness of the analyses in marketing seemed destined to escalate in the 1980s and beyond.

The 1980s saw another more-than doubling of the number of published OR articles in marketing compared to the earlier decade. Two of the areas that produced much of this growth were stylized theoretical models and process-oriented models. The shortening of product life cycles and the impact of competitive reactions in the market place preclude most markets from approaching steady state or equilibrium. Areas of special research focus in that decade included extensive focus on consumer choice models (focusing on the dynamics and heterogeneity of the choice process and the implications for decision making) and the new product area (where the moves and countermoves of competitors keep the marketplace in a constant state of flux).

The 1990s saw new trends in marketing science (and in marketing in general), with the electronic marketplace changing the locus and the nature of the transaction. The concept of the physical marketplace is being replaced by that of market space, and marketing science has found new territories to develop theories and applications. Most of this, of course, is due to the applied nature of the marketing discipline in which solutions to problems emanate from the data and the problem at hand. As the physical marketplace is being replaced by the physical in conjunction with digital marketplace, OR methods that allow for cross-channel optimization are being developed.

The first decade of the 21st century has seen the marketspace/customer centricity trend continue, as customers have gained increased influence and power in all areas of marketing. User-generated content and customers as co-producers and co-marketers are increasingly accepted. Understanding and monitoring these new market structures are central to the new view of marketing. Markets are now made up of customer-networks, and models for understanding and managing such networks are being developed. And the study of the role of the marketing manager and the related decision support systems has evolved from Little's (1979) perspective to a domain of mainstream interest both to academics and practitioners (see Wierenga 2011).

Another important trend is the emergence of two-sided and multi-sided platforms, wherein a business builds a platform that enables many distinct audiences to engage with the business as well as interact with each other, to create economic value, often in the presence of network externalities (Eisenmann et al. 2006). Typically, value appropriation occurs through cross-subsidies, wherein the costs of acquiring one group (e.g., consumers) are subsidized by another group (e.g., advertisers), and the platform itself retains part of the value created. eBay (buyers and sellers), Amazon.com (consumers and affiliates), HMO (patients and doctors), and credit-card payment systems (merchants and consumers) are

often given as examples of platforms. Even many traditional businesses are transforming into platforms that connect players in a complex eco-system (e.g., iPhone as a device connecting application developers with consumers). Cross-subsidies between audiences creates complex transaction flows that offer opportunities for OR modelers to help in carefully managing prices, revenues, and subsidies to optimize business performance.

Trends of OR Use in Marketing

The OR literature in marketing is vast, as reviewed in Lilien and Rangaswamy (2008). Models have been used to explore most facets of marketing and the marketplace, and increasingly marketing research is integrated with appropriate modeling. Some key trends include the following:

1. **OR in marketing is having important impact both on academic development in marketing and in marketing practice**. During the 1980s two new and important journals were started that emphasize the OR approach: *Marketing Science* and the *International Journal of Research in Marketing* (IJRM). Both are healthy, popular, and extremely influential, especially among academics. Another journal, *Quantitative Marketing and Economics* was started in 2003. Together, they reflect the developments of marketing models.
2. **Digital marketing represents vast area of opportunity for OR**. By transforming the market place into marketspace, the revolution in the marketplace brings a host of modeling opportunities and challenges, such as: How are new products and ideas generated, diffused, and discussed in a digital environment? How can a firm manage the natural conflict in physical and electronic distribution channels? How can firms offer different prices to different groups of customers in an electronically linked world? How and when word-of-mouth among consumers evolves? When marketing, manufacturing and the customer are interlinked in the digital environment, what opportunities emerge in the marketing-manufacturing interface? Digital marketing has other major implications, such as the development of new markets (on-line auctions, electronic bargaining) and the possibility of involving customers directly in the development of information products (Dell stores, IBM Jam, and others). More recently, it has become feasible to model large-scale social networks consisting of millions of nodes and billions of links, such as for example to link in near real-time a TV event (e.g., Super Bowl ad) with the Twitter and Facebook feeds triggered by the ad, to potential impact on market outcomes. These provide opportunities to apply OR modeling for analyzing flows of information and influence in such networks to link those to consequences for the firm (e.g., profit) or adverse spread of word of mouth in the marketplace.
3. **New data sources are having a major impact on marketing modeling**. One of the most influential developments of the 1980s and 1990s has been the impact of scanner data on the marketing models field. Scanner data and the closely related single source data (of communication and consumption data) have enabled marketing scientists to develop and test models with much more precision than ever before. Indeed, the very volume of new data has helped spawn tools to help manage the flow of new information inherent in such data. Data mining methods applied to some of the new, massive direct-response data bases has resulted in much more precise customer targeting and promotion-selection procedures. Two new data sources are providing opportunities for OR modelers in marketing: (1) Large integrated data warehouses created by companies to feed enterprise systems, such as CRM, are creating opportunities for developing more fine-grained models that integrate traditional demand side modeling undertaken by marketing modelers with supply side modeling issues such as inventory management, multi-channel logistics, and the like. (2) User-generated data (e.g., online product reviews posted by consumers, social media activities such as twitter feeds) that provide information in real-time about market sentiments offer opportunities for modelers to develop new tools for supporting marketing decision makers. New models for text analysis and synthesis (e.g., to convert reviews into numeric scores representing valence and volume

of sentiments) developed by computer scientists represent a start, but many new opportunities exist in this nascent area to translate huge volumes of raw data into insights for action. Traditional quantitative data sources have been employed by marketing modelers extensively, but more and more attention is now being given to analyzing qualitative and textual data through data and text mining as well as sentiment analysis software packages developed in computer sciences. While the 1990s presented the land of promise for these methods, the 2000s saw it materialize. Thus, the number of people in the information systems area working on traditional marketing problems has increased dramatically, blurring the lines between these related disciplines.

4. **Stylized theoretical modeling is still a mainstream research tradition in marketing**. Stylized models allow researchers to state explicitly as set of assumptions or axioms and then derive theoretical propositions with respect to the phenomena being considered. Such propositions provide valuable managerial insights.
5. **Competition and interaction are major thrusts of marketing models today**. The saturation of markets and the economic fights for survival has changed the focus of interest in marketing models, probably forever. A key-word search of past volumes of *Marketing Science*, *Journal of Marketing Research*, and *Management Science* (marketing articles only) reveals multiple entries for "competition," "competitive strategy," "non-cooperative games," "competitive entry," "late entry," and "market structure." These terms are largely missing in a comparable search in the 1960s and early 1970s.
6. **Marketing research and modeling are facing new challenges**. Both marketing research and modeling, especially as applied to new product development, have to be reformed to address such issues as global scope, electronically interconnected product development sites, the potential for mass customization and rapid prototyping/testing. These issues drive the development of models that incorporate nontraditional customer information, including trade show-participant feedback, user co-development, lead user methods, data and text mining, and Internet panels. Similarly, advertising and marketing mix modeling face comparable challenges, which have led to numerous efforts to develop single source data, related modeling, experiments, and dashboards. Another challenge is to develop models, beyond those developed for the consumer package good industry, that capture adequately various idiosyncratic characteristics of industries such as financial services, entertainment, life sciences, and B2B industries.
7. **Beyond Marketing Analytics—Marketing Engineering**. Marketing analytics, a term that refers to any systematic analysis of marketplace behavior and transactions is giving way to advance marketing analytics or marketing engineering, a term Lilien and Rangaswamy (2006) have popularized to refer to the use of decision models for making marketing decisions. Many of these decisions are now being automated, with decision models making routine pricing and promotion decisions in low-risk stable environments. But the confluence of new data sources, theories, hardware and software, and computer networks has now put these decision models on the desktop of marketing executives everywhere. The use of OR in marketing through marketing engineering is accelerating because of at least six trends (Lilien and Rangaswamy 2008):
   - Investments in infrastructure firms need to maintain extensive, integrated corporate information warehouses (also called data warehouses).
   - The use of On-Line Analytic Processing (OLAP — or just-in-time OR!) to integrate modeling capabilities with data bases.
   - Deploying intelligent systems to automate many modeling tasks.
   - Developing computer simulations for decision training and for exploring multiple options.
   - Installing groupware systems to support group decision making.
   - Enhancing user interfaces to make the use of even complex modeling systems accessible to a wide range of users.
8. **Marketing Management Support Systems and Artificial Intelligence**. A marketing management support systems (MMSS) is defined as any device, combining information technology, analytical

capabilities, marketing data, and marketing knowledge, made available to marketing decision makers with the objective to improve the quality of marketing management (Wierenga and Van Bruggen 2000). Marketing models, with their origin in OR constitute the analytical part of MMSS. However, in marketing there are also many weakly-structured problem areas, where qualitative considerations and judgment are more important. Here, the knowledge and the expertise of the marketer are key resources. Therefore, marketing management support systems not only include the primarily quantitative, data-driven decision-support systems, but also support technologies that are aimed at supporting marketing decision making in weakly structured areas.

9. **Expert Systems**. Marketing expert systems have been developed for many domains of marketing, e.g., (i) to find the most suitable type of sales promotion; (ii) to recommend the execution of advertisements (positioning, message, presenter) (Burke et al. 1990); (iii) to screen new product ideas, and (iv) to automate the interpretation of scanner data, including writing reports. For an overview, see Wierenga and Van Bruggen (2000, Chapter 5). An example of a system especially developed for supporting a particular marketing function is BRANDFRAME. This system supports the decision making of a product or brand manager, which is a typical marketing job. More recently, expert systems in marketing are less often stand-alone systems, but are woven into the company's overall IT systems (Lilien and Rangaswamy 2008).
10. **Neural Networks and Predictive Modeling**. As mentioned earlier, in marketing companies can work more and more with data about individual customers. As a consequence of this development, customer relationship management systems (CRM) became important. An essential element of CRM is the customer database that contains information about each individual customer. This information may refer to socio-economic characteristics (age, gender, education, income), earlier interactions with the customer (e.g., offers made and responses to these offers, complaints, service), and information about the purchase history of the customer (i.e., how much purchased and when). This data can be used to predict the response of customers to a new offer or to predict customer retention/churn. Such predictions are very useful, for example, for selecting the most promising prospects for a mailing or for selecting customers in need of special attention because they have a high likelihood of leaving the company (campaign optimization). A large set of techniques is available for this kind of predictive modeling. Prominent examples are neural networks and classification and regression trees. Both techniques are rooted in artificial intelligence. CRM is a quickly growing area of marketing. Companies want to achieve maximum return on their often large investments in customer databases. (Van Bruggen and Wierenga 2010).
11. **Analogical Reasoning and Case-Based Reasoning (CBR)**. Analogical reasoning plays an important role in human perception and decision making. When confronted with a new problem, people seek similarities with earlier situations and use previous solutions as the starting point for dealing with the problem at hand. Analogical reasoning is also the principle behind the field of case-based reasoning (CBR) in Artificial Intelligence. A CBR system comprises a set of previous cases from the domain under study and a set of search criteria for retrieving cases for situations that are similar (or analogous) to the target problem. Applications of CBR can be found in weakly-structured domains such as architecture, engineering, law, and medicine. By their nature, many marketing problems have a good fit with CBR. A recent application uses CBR as a decision-support technology for designing creative sales promotion campaigns (Van Bruggen and Wierenga 2010).
12. **Adaptive Experimentation**. While OR applications in marketing have been focused on models, given the increased uncertainty, complexity and speed of change of the business environment, it is unlikely that one can model optimal strategies. The alternative to the search for a silver bullet is the adoption of an adaptive experimentation philosophy (Wind 2007) that allows experimentation with a number of innovative strategies, facilitates learning, helps create an innovative organizational culture that reduces the pressures for risk averse decisions, encourages relevant measurement and provides

a competitive advantage. As sophisticated firms such as Google and most direct response companies Increasingly engage in adaptive experimentation, a new role for many of the OR marketing models (including marketing mix models) is in suggesting hypotheses that guide the experimental variables and design. Adaptive experimentation is consistent with the philosophy of OR and should be considered in any portfolio of approaches to aid decision makers in making better decisions.

13. **Documentation of the Impact of OR Marketing Models on the Organization is Now Mainstream**. The emergence of the INFORMS Society for Marketing Science Practice Prize and work by Lilien (2011) and Wierenga (2011) have underscored the need to study how marketing integrated with the concepts of OR can become a mainstream research domain for marketing academics while having a greater impact on the operations of firms. According to a *Business Week* article in 2010, the Fortune 1,000 companies spend over $1 trillion in marketing annually. Yet, according to a McKinsey report (2009), most of these companies do not use marketing models to improve their marketing investment related decision making, even though the small percentage of companies that do (17% of B2C and 7% of B2B) seem to realize considerable benefits from their use. In a controlled experimental study, Lilien shows that the managers using decision models realize measurable improvements in decision performance when compared to managers who have access to the same data, but without a decision-support model to optimally interpret the data. Research is ongoing on what factors influence companies to deploy marketing models, under what conditions their impact is maximized, and how decision tools should be designed to enhance their usability and impact.

Concluding Remarks

OR/marketing models and approaches have had significant impact on academic research and practice. Marketing science has also been used to address important societal problems, e.g., Bradlow (2009) discusses the use of marketing science to aid in creatively solving problems related to the financial crisis. Developments in constructing, testing and applying new marketing science models will continue to benefit management and society.

See

- ▶ Advertising
- ▶ Data Mining
- ▶ Decision Analysis
- ▶ Electronic Commerce
- ▶ Game Theory
- ▶ Linear Programming
- ▶ Operations Management
- ▶ Retailing

References

Basu, A., Lal, R., Srinivasan, V., & Staelin, R. (1985). Sales force compensation plans: An agency theoretic perspective. *Marketing Science, 4*, 267–291.

Bell, D. E., Keeney, R. L., & Little, J. D. C. (1975). A market share theorem. *Journal of Marketing Research, 12*, 136–141.

Bradlow, E. T. (2009). Marketing science and the financial crisis. *Marketing Science, 28*(2), 201.

Burke, R. R., Rangaswamy, A., Wind, J., & Eliashberg, J. (1990). A knowledge-based system for advertising design. *Marketing Science, 9*(3), 212–229.

Chen, Y., Yogesh, J., Raju, J. S., & Zhang, J. (2009). A theory of combative advertising. *Marketing Science, 28*, 1–19.

Dube, J. P., Hitsch, G. J., & Chintagunta, P. K. (2010). Tipping and concentration in markets with indirect network effects. *Marketing Science, 29*(2), 216–249.

Eisenmann, T., Parker, G., & Van Alstyne, M. W. (2006). Strategies for two-sided markets. *Harvard Business Review*, October 2006, 1–10.

Ghose, A., & Han, S. (2011). An empirical analysis of user content generation and usage behavior on the mobile internet. *Management Science, 57*(9), 1671–1691.

Green, P. E., Wind, Y., & Carroll, D. (1973). *Multi-attribute decisions in marketing: A measurement approach*. Hinsdale: The Dryden Press.

Guadagni, P., & Little, J. D. C. (1983). A Logit model of brand choice calibrated on scanner data. *Marketing Science, 2*, 203–238.

Hauser, J. R., Urban, G. L., Liberali, G., & Braun, M. (2009). Website morphing. *Marketing Science, 28*(2), 202–223.

Hui, S., Bradlow, E., & Fader, P. (2009). The traveling salesman goes shopping: The systematic deviations of grocery paths from TSP-optimality. *Marketing Science, 28*(3), 566–572.

Jeuland, A. P., & Shugan, S. M. (1983). Managing channel profits. *Marketing Science, 2*(3), 239–272.

Kotler, P. (1971). *Marketing decision making: A model building approach*. New York: Holt, Rinehart and Winston.

Lilien, G. L. (2011). Bridging the academic-practitioner divide in marketing decision models. *The Journal of Marketing, 75*, 196–210.
Lilien, G. L., & Kotler, P. (1983). *Marketing decision making: A model building approach*. New York: Harper and Row.
Lilien, G. L., & Rangaswamy, A. (2006). Marketing decision support models: The marketing engineering approach. In R. Grover & M. Vriens (Eds.), *The handbook of marketing research: Uses, misuses, and future advances*. Thousand Oaks, CA: Sage Publications.
Lilien, G. L., & Rangaswamy, A. (2008). Marketing engineering: Models that connect with practice. In B. Wierenga (Ed.), *Handbook of marketing decision models: International series in operations research & management science*. New York: Elsevier Press.
Little, J. D. C. (1979). Decision support systems for marketing managers. *Journal of Marketing, 43*(3), 9–27.
McGuire, T. W., & Staelin, R. P. (1983). An Industry equilibrium analysis of downstream vertical integrations. *Marketing Science, 2*(2), 161–192.
Raju, J. S., Srinivasan, V., & Lal, R. (1990). Effects of brand loyalty on competitive price promotional strategies. *Management Science, 36*, 276–304.
Stephen, A., & Toubia, O. (2010). Deriving value from social commerce networks. *Journal of Marketing Research, 47*(2), 215–228.
Sudhir, K. (2001). Competitive pricing behavior in the auto market: A structural analysis. *Marketing Science, 20*(1), 42–60.
Van Bruggen, G. H., & Wierenga, B. (2010). Marketing decision making and decision support: Challenges and perspectives for successful marketing management support systems. *Foundations and Trends in Marketing, 4*(4), 126.
Wierenga, B. (2011). Managerial decision making in marketing: The next research frontier. *International Journal of Research in Marketing, 28*(2), 89–101.
Wierenga, B., & Van Bruggen, G. H. (2000). *Marketing management support systems: Principles, tools, and implementation*. Boston: Kluwer Academic.
Wind, J. (2007). Marketing by experiment. *Marketing Research*, Spring 2007, 10–16.

Markov Chain Equations

William J. Stewart
North Carolina State University, Raleigh, NC, USA

Introduction

For a continuous-time Markov chain, the probability distribution at any time t, $\boldsymbol{\pi}(t)$, is calculated from the Chapman-Kolmogorov differential equation,

$$\frac{d\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t)\boldsymbol{Q}. \tag{1}$$

where the vector $\boldsymbol{\pi}(t)$ is of length n, the number of possible states in the Markov chain, and its ith component, $\pi_i(t)$, expresses the probability that the Markov chain is in state i at time t, and $\boldsymbol{Q}$ is the infinitesimal generator or transition rate matrix, a square matrix of order n whose elements satisfy

$$q_{ij} \geq 0, \quad i \neq j;$$
$$q_{ii} = \sum_{j=1,\, j\neq i}^{n} q_{ij}, \quad \text{for all } i = 1,\ 2,\ \ldots,\ n.$$

When the number of states in the Markov chain is relatively small (e.g., less than a thousand), computing numerical solutions of the chain equations is generally easy, and (1) can be solved readily by software such as MATLAB. But two difficulties arise when the number of states is large: The first is the sheer size of the matrices involved; the second is how well-conditioned or how ill-conditioned the equations are. These difficulties exist even in the simpler setting considered here when all that is required is the stationary solution of the Markov chain obtained by setting the left-hand side of (1) to zero and solving the linear system of equations that results.

It is not unusual for the number of states in a Markov chain model to exceed the millions. Such size impacts both the storage of the matrix and the number of vectors needed to compute the solution. Very large matrices cannot be stored in the usual two-dimensional array format; there is simply not enough storage space available. In addition, this would be very wasteful, since most of the matrix elements are zero. In general, each state communicates directly with only a small number of states and so the number of nonzero elements in the matrix is usually equal to a small multiple of the number of states. If the states can be ordered sequentially so that each communicates only with its closest neighbors, then the nonzero elements of Q lie close to the diagonal and a banded storage technique can be used. Otherwise, it is usual to store only the nonzero elements in a double-precision one-dimensional array and use two integer one-dimensional arrays to indicate the position of each nonzero element in the matrix. In addition to storing the transition matrix, a certain

number of double-precision vectors, of size equal to the number of states, is also needed. In the simplest numerical methods, two such vectors suffice. In other more sophisticated methods, many more (possibly in excess of 50) may be needed.

A second difficulty in solving Markov chains numerically is that of the degree of ill-conditioning of Q. In certain models, the difference in the rates at which events can occur may be many orders of magnitude, as is the case when a model allows for both human interaction and electronic transactions. These differences in magnitude may lead to ill-conditioned systems, that is to say, a small change in one of the parameters can result in a large change in the solution. It is appropriate to distinguish between numerical conditioning and numerical stability; the first has already been described and is a function of the problem itself; the second describes the behavior of an algorithm in attempting to compute solutions. A stable algorithm will not allow the error to grow out of proportion to the degree of ill-conditioning of the problem. In other words, a stable algorithm will give as good a solution as can be expected for the particular problem to be solved. A further effect of large differences in transitions rates is that they can create convergence problems for iterative solution methods.

Numerical Methods for Computing Stationary Distributions

The goal is to solve the matrix equation

$$\boldsymbol{\pi Q} = 0. \tag{2}$$

By setting $\boldsymbol{P} = \boldsymbol{Q}\Delta t + \boldsymbol{I}$, where $\Delta t \leq (\max_i |q_{ii}|)^{-1}$, this equation may be written as

$$\boldsymbol{\pi P} = \boldsymbol{\pi}. \tag{3}$$

In carrying out this operation, the continuous-time system represented by the transition rate matrix, $\boldsymbol{Q}$, is essentially converted to a discrete-time system represented by the stochastic transition probability matrix, $\boldsymbol{P}$. In the discrete-time system, transitions take place at intervals of time Δt, this parameter being chosen so that the probability of two transitions taking place in time Δt is negligible. The stationary distribution π may be computed from either of these equations.

Direct Methods — Since Eq. (2) is a homogeneous system of linear equations, one may use standard linear solution methods based on Gaussian elimination. Assume that the Markov chain is ergodic. In this case, the fact that the system of equations is homogeneous does not create any problems, because any of the n equations can be replaced by the n normalizing equation, $\sum_{j=1}^{n} \pi_j = 1$, and thereby convert it into a nonhomogeneous system with nonsingular coefficient matrix and nonzero right hand side. The solution in this case is well defined. It turns out that replacing an equation with the normalizing equation is not really necessary.

The usual approach taken is to construct an $\boldsymbol{LU}$ decomposition of $\boldsymbol{Q}$ and replace the final zero diagonal element of $\boldsymbol{U}$ with an arbitrary value. The solution computed by back substitution on $\boldsymbol{U}$ must then be normalized. Furthermore, since the diagonal elements are equal to the negated sum of the off-diagonal elements ($\boldsymbol{Q}$ is, in a restricted sense, diagonally dominant), it is not necessary to perform pivoting while computing the $\boldsymbol{LU}$ decomposition. This simplifies the algorithm considerably.

The problems of the size and nonzero structure (the placement of the nonzero elements within the matrix) still remain. Obviously this method works well when the number of states is small. It will also work well when the nonzero structure of $\boldsymbol{Q}$ fits into a narrow band along the diagonal. In these cases, a very stable variant, referred to as the GTH (Grassmann, Taskar, and Heyman) algorithm, may be used. In this variant, all subtraction is avoided by computing diagonal elements as the sum of off-diagonal elements. This is possible since the zero-row-sum property of an infinitesimal generator is invariant under the basic operation of Gaussian elimination, namely adding a multiple of one row into another. For an efficient implementation, the GTH variant requires convenient access to both the rows and the columns of the matrix. This is the case when a banded structure is used to store $\boldsymbol{Q}$, but is generally not the case with other compact storage procedures. When the number of states becomes large and the structure in not banded, the direct approach loses its appeal and one must resort to other methods.

Iterative Methods — For iterative methods, the first approach is to solve Eq. (3) in which $\boldsymbol{P}$ is a matrix of transitions probabilities. Let the initial probability distribution vector be given by $\boldsymbol{\pi}^{(0)}$. After the first transition, the probability vector is given by $\boldsymbol{\pi}^{(1)} = \boldsymbol{\pi}^{(0)}\boldsymbol{P}$; after k transitions it is given by $\boldsymbol{\pi}^{(k)} = \boldsymbol{\pi}^{(k-1)}\boldsymbol{P} = \boldsymbol{\pi}^{(0)}\boldsymbol{P}^{\boldsymbol{k}}$. If the Markov chain is ergodic, then $\lim_{k\to\infty}\boldsymbol{\pi}^{(k)} = \boldsymbol{\pi}$. This method of determining the stationary probability vector, by successively multiplying some initial probability distribution vector by the matrix of transition probabilities, is called the Power method. Observe that all that is required is a vector–matrix multiplication operation. This may be conveniently performed on sparse matrices that are stored in compact form. Because of its simplicity, this method is widely used, even though it often takes a very long time to converge. Its rate of convergence is a function of how close the subdominant eigenvalue of P is to its dominant unit eigenvalue. In models in which there are large differences in the magnitudes of transition rates, the subdominant eigenvalue can be pathologically close to one, so that for all intensive purposes the Power method fails to converge.

It is also possible to apply iterative equation solving techniques to the system of equations given by (2). The well-known Jacobi method is closely related to the Power method, and it also frequently takes very long to converge. A better iterative method is Gauss-Seidel. Unlike the previous two methods, in which the equations are only updated after each completed iteration, the Gauss-Seidel method uses the most recently computed values of the variables as soon as they become available and, as a result, almost always converges faster than Jacobi or the Power method. All three methods can be written so that the only numerical operation is that of forming the product of a sparse matrix and a probability vector, so all are equal from a computation per iteration point of view.

Block Methods — In Markov chain models, it is frequently the case that the state space can be meaningfully partitioned into subsets. Perhaps the states of a subset interact only infrequently with the states of other subsets, or perhaps the states possess some property that merits special consideration. In these cases, it is possible to partition the transition rate matrix accordingly and to develop iterative methods based on this partition. In general, such block iterative methods require more computation per iteration, but this is offset by a faster rate of convergence.

If the state space of the Markov chain is partitioned into N subsets of size $n_1, n_2, \ldots, n_N$ with $\sum_{i=1}^{N} n_i = n$, then block iterative methods essentially involve the solution of N systems of equations of size n_i, $i = 1,2,\ldots, N$, within a global iterative structure, such as Gauss-Seidel, for instance: thus the Block Gauss-Seidel method. Furthermore, these n systems of equations are nonhomogeneous and have nonsingular coefficient matrices and either direct or iterative methods may be used to solve them. It is not required that the same method be used to solve all the diagonal blocks, so that it is possible to tailor methods to the particular block structures.

If a direct method is used, then a decomposition of the diagonal block may be formed once and for all before initializing the global iteration process. In each subsequent global iteration, solving for that block then reduces to a forward and backward substitution operation. The nonzero structure of the blocks may be such that this is a particularly attractive approach. For example, if the diagonal blocks are themselves diagonal matrices, or if they are upper or lower triangular matrices or even tridiagonal matrices, then it is very easy to obtain their LU decomposition, and a block iterative method becomes very attractive.

If the diagonal blocks do not possess such a structure, and when they are of large dimension, it may be appropriate to use an iterative method to solve each of the block systems. In this case, there are many inner iterative methods (one per block) within an outer (or global) iteration. A number of tricks may be used to speed up this process. First, the solution computed for any block at global iteration k should be used as the initial approximation to the solution of this same block at iteration $k + 1$. Second, it is hardly worthwhile computing a highly accurate solution in early (outer) iterations. Only a small number of digits of accuracy should be required until the global process begins to converge. One convenient way to achieve this is to carry out only a fixed small number of iterations for each inner solution.

Iterative Aggregation/Disaggregation Methods — Related to block iterative methods, these methods are particularly powerful when the Markov chain is nearly completely decomposable, as the partitions are chosen based on how strongly the states of the Markov chain interact with one another.

Projection Methods — An idea that is basic to sparse-linear systems and eigenvalue problems is that of projection processes. Whereas iterative methods begin with an approximate solution vector that is modified at each iteration and which (supposedly) converges to a solution, projection methods create vector subspaces and search for the best possible approximation to the solution that can be obtained from that subspace. With a given subspace, for example, it is possible to extract a vector $\hat{\pi}$ that is a linear combination of a set of basis vector for that space and which minimizes $|\hat{\boldsymbol{\pi}}\boldsymbol{Q}|$ in some vector norm. This vector $\hat{\pi}$ may then be taken as an approximation to the solution of $\boldsymbol{\pi Q} = 0$. This is the basis for the Generalized Minimal Residual (GMRES) algorithm. Another popular projection method is the method of Arnoldi. The subspace most often used is the Krylov subspace, $K_m = \text{span}\{\boldsymbol{v}_1, \boldsymbol{v}_1\boldsymbol{Q}, \ldots, \boldsymbol{v}_1\boldsymbol{Q}^{m-1}\}$, constructed from a starting vector v_1 and successive iterates of the power method. The computed vectors are then orthogonalized with respect to one another. It is also possible to construct iterative variants of these methods. When the subspace reaches some maximum size, the best approximation is chosen from this subspace and a new subspace generated using this approximation as the initial starting point.

Preconditioning techniques are frequently used to improve the convergence rate of iterative Arnoldi and GMRES. This typically amounts to replacing the original system $\boldsymbol{\pi Q} = 0$ by $\boldsymbol{\pi Q M}^{-1} = 0$, where $\boldsymbol{M}$ is a matrix whose inverse is easy to compute. The objective of preconditioning is to modify the system of equations to obtain a coefficient matrix with a fast rate of convergence. It is worthwhile pointing out that preconditioning may also be used with the basic power method to improve its rate of convergence. The inverse of the matrix $\boldsymbol{M}$ is generally computed from an incomplete *LU* factorization of the matrix $\boldsymbol{Q}$.

Stochastic Automata Networks

Stochastic Automata Networks (SANs) provide a means of performing Markov chain modeling without the problem of having to store huge transition matrices. A SAN consists of a number of individual stochastic automata that operate more or less independently of each other. Each individual automaton is represented by a number of states and rules that govern the manner in which it moves from one state to the next. The state of an automaton at any time t is just the state it occupies at time t, and the state of the SAN at time t is given by the state of each of its constituent automata. An automaton may be thought of as a component in a Markov chain state descriptor.

The use of SANs is important in the performance modeling of parallel and distributed systems, since such systems are often viewed as collections of components that operate more or less independently, requiring only infrequent interaction such as synchronizing their actions or operating at different rates depending on the state of parts of the overall system. This is exactly the viewpoint adopted by SANs. Furthermore, the state space explosion problem associated with Markov chain models is mitigated by the fact that the state transition matrix is not stored, nor even generated. Instead, it is represented by a number of much smaller matrices, one for each of the stochastic automata that constitute the system, and from these all relevant information may be determined without explicitly forming the global matrix. A considerable saving in memory is realized by storing the matrix in this fashion.

The compact form in which the transition matrix that characterizes the model is kept (called the SAN Descriptor) is written as

$$\sum_{j=1}^{N+2E} \otimes_{i=1}^{N} \boldsymbol{Q}_j^{(i)},$$

where N is the number of automata in the SAN, E is the number of synchronizing events and $\boldsymbol{Q}_j^{(i)}$ is a square matrix of low dimension. In order to benefit from this compact form, the descriptor is never expanded into a single large matrix. Consequently, all subsequent operations must necessarily work with the model in its descriptor form, and hence, numerical operations on the underlying Markov chain infinitesimal generator become more costly. Research efforts directed at reducing these costs include the development of a generalized tensor algebra to permit functional transitions to be handled at the same low costs as constant transitions, design of algorithms to reduce the amount of computation involved in forming the product of a vector and a SAN descriptor, and finding suitable preconditioners with which to speed up iterative methods.

See

- Markov Chains
- Markov Processes
- Numerical Analysis
- Queueing Theory
- Stochastic Process

References

Berman, A., & Plemmons, R. J. (1994). *Nonnegative matrices in the mathematical sciences*. Philadelphia: SIAM.

Fernandes, P., Plateau, B., & Stewart, W. J. (1998). Efficient descriptor-vector multiplication in stochastic automata networks. *Journal of the Association for Computing Machinery, 45*, 381–414.

Saad, Y. (1996). *Iterative solution of sparse linear systems*. New York: PWS Publishing.

Stewart, W. J. (1976). MARCA: Markov chain analyzer. IEEE Computer Repository, No. R76 232 (See the URL: http://www.csc.ncsu.edu/faculty/WStewart).

Stewart, W. J. (1994). *An introduction to the numerical solution of Markov chains*. New Jersey: Princeton University Press.

Markov Chain Monte Carlo

Michel Wedel[1] and Peter Lenk[2]
[1]University of Maryland, College Park, MD, USA
[2]University of Michigan, Ann Arbor, MI, USA

Introduction

Markov chain Monte Carlo (MCMC) methods numerically approximate the integral or expectation, $E[g(Y)] = \int g(y) f(y|\Theta) dy$, where Y is a random variable with distribution $f(y|\Theta)$, which is parameterized by Θ, and $g(Y)$ is an integrable function of Y, where the integral is with respect to either Lebesgue measure for continuous random variables or counting measure for discrete ones. A simple way to compute $E[g(Y)]$ is through Monte Carlo (MC) simulation, which approximates the integral as an average of $g(Y)$ across a random sample from $f(y|\Theta)$: $\overline{g(Y)} = \frac{1}{n} \sum_{i=1}^{n} g(y_i)$. The estimation variance is proportional to n^{-1}, regardless of the dimension of Y, and the estimator can be made arbitrarily accurate by letting the size of the sample $n \to \infty$ by the strong law of large numbers. MCMC addresses settings where random variates for $f(y|\Theta)$ cannot be generated easily, e.g., through the inverse transform method, the acceptance-rejection method (also called rejection sampling), or importance sampling. These methods generally rely on independent and identically distributed (i.i.d.) random draws to approximate the integral.

MCMC methods relax this independence assumption to construct a Markov chain of draws $\{y_i, i = 1, \ldots, n\}$, with a stationary distribution equal to. $f(y|\Theta)$. MCMC uses recursive simulation where the random number generator for Y_i depends on the previous draw y_{i-1}, hence the name Markov chain Monte Carlo. MCMC's range of applications is astonishing, and continues to expand. A large part of these applications have been in Bayesian statistics, but MCMC originated in image processing and physics and continues to be used in these fields, as well as in biology, engineering, demography, finance and marketing. MCMC was started by the work of Metropolis et al. (1953) and Hastings (1970). Gibbs sampling as a special case developed through the work of Besag (1974), Geman and Geman (1984), and Gelfand and Smith (1990). Important extensions were developed by Albert and Chib (1993), Green (1995), Richardson and Green (1997) and Neal (2003). Texts include Gill (2008), Press (2003), Gelman et al. (2003), and Zellner (1971). Essential MCMC methods are reviewed here, while details can be found in the references above.

Discussion

Metropolis-Hastings Sampler: The Metropolis-Hastings (MH) sampler is very general and sparked the MCMC revolution. For $i = 1, \ldots, n$, it generates a candidate sample y_i from a proposal distribution $h(y|y_{i-1}, \Phi)$ and transforms it to make it behave as if it came from $f(y|\Theta)$ (The support of h is a subset of that of f). If the proposal distribution h depends on the previous value y_{i-1}, the algorithm is called Random Walk Metropolis-Hastings (rMH), while if it does not depend on previous values, it is called Independence Metropolis-Hastings (iMH). The algorithm works as follows, for $i = 1, 2, \ldots, n$

1. Initialize the chain at y_0 that is in the support of f.
2. Given that a prior value y_{i-1} has been obtained, sample a candidate value $y^* \sim h(y|y_{i-1};\Phi)$ and sample $u_i \sim U(0,1)$.
3. Calculate $\alpha(y_{i-1}, y^*) = \frac{f(y^*|\Theta)}{f(y_{i-1}|\Theta)} \cdot \frac{h(y_{i-1}|y^*,\Phi)}{h(y^*|y_{i-1},\Phi)}$.
4. Accept the candidate $y_i = y^*$, if $\alpha(y_{i-1}, y^*) > u_i$, otherwise set $y_i = y_{i-1}$.

The normalizing constants for f and h cancel in Step 3, so that they only need to be known up to such constants. The MH algorithm creates a Markov chain with transition function $q(y_{i-1}, y_i) = h(y_i|y_{i-1},\Phi)\tau(y_{i-1}, y_i)$ where $\tau(y_{i-1}, y^*) = \min[\alpha(y_{i-1}, y^*), 1]$ is the acceptance probability from Step 4. The chain is reversible because $f(y_{i-1}|\Theta)q(y_{i-1}, y_i) = f(y_i|\Theta)q(y_i, y_{i-1})$, and is therefore ergodic with stationary distribution f. This has the crucial implication that regardless of the initial value in Step 1, the draws from the Markov chain will eventually be from f. Monte Carlo is a special case if the candidate distribution h is equal to f: then $\alpha(y_{i-1}, y^*) = 1$. If $h(y|y_{i-1},\Phi)$ is symmetric in $(y - y_{i-1})$, e.g. a normal distribution with mean y_{i-1}, then the ratio in h cancels in Step 3.

The performance of MH depends on the proposal distribution. In rMH if the proposal distribution is too tight around the last value of the chain, then the candidate is highly likely to be accepted, and the Markov chain will tour the support of f very slowly, so n will have to be quite large to obtain reliable MCMC estimates. Conversely, if the variance of the proposals is too large, the MCMC algorithm will reject most of the candidate values, and the chain will hardly budge. For the estimator to be valid, the chain needs to visit areas of the support of f with non-negligible probabilities.

Convergence: Starting from an arbitrary y_0, the chain passes through a transitory period, say $i = 1, \ldots, l$ for $l < n$, where the draws are not from f. These initial draws are not used in the MCMC approximation of $E[g(Y)]$: $\overline{g(Y)} = \frac{1}{n-l} \sum_{i=l+1}^{n} g(y_i)$. In theory, under very general conditions the rate of convergence is geometric in the second eigen value of the transition function. Problems can occur if the target distribution is multimodal, and f is zero between modes, so that subsets of the support do not communicate with each other. Then the chain can become stuck in isolated regions of the support unless the proposal distribution h is sufficiently broad to bridge the gaps. In practice, it may be difficult to conclusively determine l. One procedure for monitoring convergence is to run multiple chains from different initial values and to compute multiple estimates. If the between-chain variance of the estimators is small relative to the within-chain variance, then the chain has likely converged. A host of other diagnostic measures are available, that may help identify likely convergence of the chain.

Blocked MH Sampler: Depending on the structure of f, it may be convenient to block Y into sub-vectors Y_s for $s = 1, 2, \ldots, S$. The distribution of each sub-vector is conditioned on all others to obtain the full conditional distributions: $f(y_s|y_{-s};\Theta)$, with y_{-s} denoting y with y_s omitted $h_s(y_s|y_{-s};\Phi)$ is the proposal distribution for Y_s. This leads to the following algorithm for $i = 1, 2, \ldots, n$ and $s = 1, 2, \ldots, S$:

1. Initialize y_0 in the support of f.
2. Sample a candidate value $y_s^* \sim h_s\left(y_s|y_{-s,i-1};\Phi\right)$ and $u_i \sim U(0,1)$.
3. Calculate $\alpha_{s,i} = \frac{f(y^*|\Theta)}{f(y_{i-1}|\Theta)} \cdot \frac{h_s(y_{s,i-1}|y^*;\Phi)}{h_s(y_s^*|y_{s,i-1};\Phi)}$, where y^* is identical to y_{i-1}, except for sub-vector s, which equals y_s^*.
4. Accept the candidate $y_{s,i} = y_s^*$, if $\alpha_{s,i} > u_i$, otherwise keep $y_{s,i-1}$.

This algorithm cycles through the s sub-vectors (in arbitrary order, systematically or randomly) and updates them separately though MH-steps. Not every sub-vector needs to be updated at every iteration i.

Gibbs Sampler: In many applications, some or even all of the full conditional target distributions $f(y_s|y_{-s};\Theta)$ can be sampled directly, which greatly simplifies the Blocked MH algorithm. This can be seen by substituting the full conditional distributions for the proposal distributions in Step 3 of the Blocked MH: $\alpha_{s,i} = \frac{f(y^*|\Theta)}{f(y_{i-1}|\Theta)} \cdot \frac{f(y_{s,i-1}|y^*;\Theta)}{f(y_s^*|y_{s,i-1};\Theta)}$. Because $f(y|\Theta) = f(y_s, y_{-s}|\Theta)$, and $y_{-s}^* = y_{-s,i-1}$, it holds that $\alpha_{s,i} = \frac{f\left(y_{-s,i-1}|\Theta\right)}{f\left(y_{-s,i-1}|\Theta\right)} = 1$. The algorithm for the Gibbs sampler modifies the Blocked MH by replacing Step 2 with directly drawing $y_{s,i} \sim f\left(y_{s,i}|y_{-s,i};\Theta\right)$ and skipping Steps 3 and 4 for these blocks.

Modifications of the Gibbs sampler have been proposed to speed up convergence and provide the chains with better properties. For three sub-vectors y_1, y_2, y_3, for example, the Collapsed Gibbs Sampler

draws from the unconditional joint distribution $y_{1:2,i} \sim f(y_1, y_2|\Phi)$, and the full conditional distribution $y_{3,i} \sim f(y_3|y_{1,i}, y_{2,i}; \Phi)$. The Grouped Gibbs Sampler on the other hand, groups two sub-vectors and draws from the full conditionals $y_{1:2,i} \sim f\left(y_1, y_2|y_{3,i-1}; \Phi\right)$ and $y_{3,i} \sim f(y_3|y_{1,i}, y_{2,i}; \Phi)$. The relative simplicity of the Gibbs sampling algorithms has contributed to their popularity, and many extensions, three important ones being the Auxiliary Variable, Slice and Reversible Jump Samplers.

Auxiliary Variable Sampler: Introducing an auxiliary random variable Z can simplify MCMC if there is a joint distribution $h(y, z|\theta, \Psi)$ such that $f(y|\Theta) = \int h(y, z|\theta, \Psi)dz$, and both $h(y|z, \Theta, \Psi)$ and $h(z|y, \Theta, \Psi)$ are easy to sample. Using these two full conditional distributions, it is then straightforward to sample from $h(y, z|\theta, \Psi)$, using Gibbs sampling. The Auxiliary Variable Gibbs Sampler is then, for $i = 1, 2, \ldots, n$:

1. Sample $y_i \sim h(y|z_{i-1}, \Theta, \Psi)$.
2. Sample $z_i \sim h(z|y_i, \Theta, \Psi)$.

An additional advantage is that the introduction of the augmented variable helps mixing.

Slice Sampler: A special case of the Auxiliary Variable Sampler arises if $f(y|\Theta)$ an be factored as $f(y|\Theta) \propto k(y|\Theta) \cdot h(y|\Theta)$. The auxiliary variable Z in this case is chosen such that the joint density $f(y, z) \propto I[0 < z < k(y|\Theta)] \cdot h(y|\Theta)$. The resulting sampler is called the Slice Sampler and iterates between the following full conditional distributions, for $i = 1, 2, \ldots, n$:

1. Sample $z_i \sim U(0, k(y_{i-1}|\Theta))$, from a uniform distribution on 0 and $k(y_{i-1}|\Theta)$.
2. Sample $y_i \sim h(y|\Theta)I[0 < z_i < k(y|\Theta)]$, from the distribution $h(y|\Theta)$ truncated on the set $\{y : z_i < k(y|\Theta)\}$.

Slice sampling is applicable in cases where $k^{-1}(y|\Theta)$ can be analytically obtained, and the truncated distribution $h(y|\Theta)I[0 < z_i < k(y|\Theta)]$ can be sampled from, often by using the inverse transform method. The extension to distributions that factor as $f(y|\Theta) \propto h(y|\Theta) \cdot \prod_t k_t(y|\Theta)$ is straightforward if all $k_t^{-1}(y|\Theta)$ an be obtained, now by sampling multiple $z_{i,t} \sim U(0, k_t(y_{i-1}|\Theta))$.

Reversible Jump Sampler: The above algorithms assume that the dimension of Y is constant. The Reversible Jump (RJ) sampler is an extension of MH that constructs a Markov chain that transverses spaces of different dimensions. The spaces are labeled m, and $Y^{(m)}$ is the random variable Y restricted to space m. The dimension of $Y^{(m)}$ or $\dim\left(Y^{(m)}\right)$ depends on m (In Bayesian statistics – details below – RJ is used to transverse different models where m indicates the model, and then simulate $Y^{(m)}$ given model m). The state space for the Markov chain is $\left(M, Y^{(M)}\right)$ with joint distribution $f\left(m, y^{(m)}|\Theta\right) = f\left(y^{(m)}|m, \Theta_y\right)f(m|\Theta_m)$ where $P(M = m) = f(m|\Theta_m)$ is a discrete distribution, and $f\left(y^{(m)}|m, \Theta_y\right)$ is the distribution of Y restricted to space m. RJ is a strategy to simulate $\left(M, Y^{(M)}\right)$ when a convenient random number generator for $f\left(m, y^{(m)}|\Theta\right)$ does not exist.

As with MH, the goal is to construct a reversible Markov chain with stationary distribution $f\left(m, y^{(m)}|\Theta\right)$. Reversible moves between any $\left(m, y^{(m)}\right)$ and $\left(m', y^{(m')}\right)$ require a bijective mapping, which does not exist when the spaces have different dimensions. The trick is to augment $y^{(m)}$ with a random variable $u^{(m)}$ so that $\dim\left(y^{(m)}\right) + \dim\left(u^{(m)}\right)$ is constant across all m: $\dim\left(y^{(m)}\right) + \dim\left(u^{(m)}\right) = \dim\left(y^{(m')}\right) + \dim\left(u^{(m')}\right)$. RJ requires a bijective, differentiable function $\left(y^{(m')}, u^{(m')}\right) = T_{m,m'}\left(y^{(m)}, u^{(m)}\right)$ that uniquely maps $\left(y^{(m)}, u^{(m)}\right)$ to $\left(y^{(m')}, u^{(m')}\right)$ with reverse mapping $T_{m',m} = T_{m,m'}^{-1}$. Given the current state $\left(m, y^{(m)}\right)$ of the Markov chain, candidate values are generated by: (1) selecting a new value m' according to the proposal distribution $q\left(m'|m, \Psi\right)$; (2) generating $u^{(m)}$ from $h_{m,m'}\left(u^{(m)}|y^{(m)}, \Phi\right)$; and (3) computing the candidate $\left(y^{(m')}, u^{(m')}\right) = T_{m,m'}\left(y^{(m)}, u^{(m)}\right)$. For the Markov chain to be reversible, the implied distribution of $u^{(m')}$, $h_{m',m}\left(u^{(m')}|y^{(m')}, \Phi\right)$, is required to move from $\left(y^{(m')}, u^{(m')}\right)$ to $\left(y^{(m)}, u^{(m)}\right)$ using the reverse mapping $T_{m',m}$. Implementation details of the RJ are as much art as science, because the construction of $\left\{T_{m,m'}\right\}$ for all m and m' and the selection of proposal distributions are tailored specifically for each application. The RJ algorithm for $i = 1, 2, \ldots, n$ is:

1. Initialize the chain at $\left(m_0, y_0^{m_0}\right)$ in the support of f.
2. Given m_{i-1} and $y_{i-1}^{(m_{i-1})}$ are obtained, set $m = m_{i-1}$ and $y = y_{i-1}^{(m_{i-1})}$ and
   a. Sample $m' \sim q(m'|m, \Psi)$;
   b. Sample $u \equiv u^{(m)} \sim h_{m,m'}\left(u^{(m)}|y, \Phi\right)$;
   c. Compute proposal $y' \equiv y^{(m')}$ from $(y', u') = T_{m,m'}(y, u)$.

3. Calculate
$$\alpha(y,y') = \frac{f\left(m',y'|\Theta\right)}{f(m,y|\Theta)} \cdot \frac{h_{m',m}\left(u'|y',\Phi\right)}{h_{m,m'}(u|y,\Phi)} \cdot \frac{q(m|m',\Psi)}{q(m'|m,\Psi)} \cdot \left|\frac{\partial T_{m,m'}(y,u)}{\partial y \partial u}\right|.$$
4. Sample $v_i \sim U(0,1)$ and accept the candidate $\left(m_i, y_i^{(m_i)}\right) = (m', y')$ if $\alpha(y,y') > v_i$, otherwise set $\left(m_i, y_i^{(m_i)}\right) = \left(m_{i-1}, y_{i-1}^{(m_{i-1})}\right)$.

In step 3, $\left|\frac{\partial T_{m,m'}(y,u)}{\partial y \partial u}\right|$ is the Jacobian of the transformation $T_{m,m'}$, which is needed because it is a deterministic function for the change in variables from $\left(y^{(m)}, u^{(m)}\right)$ to $\left(y^{(m')}, u^{(m')}\right)$. As in the MH algorithm, the distributions $f\left(m, y^{(m)}|\Theta\right)$, $q\left(m'|m,\Psi\right)$, and $h_{m,m'}\left(u^{(m)}|y,\Phi\right)$ only need to be known up to normalizing constants which cancel in step 3. It should be noted that while $\alpha(y,y')$ in its general form provided in step 3 is somewhat complex, in a wide range of practical applications it simplifies considerably, for example when the proposal distributions are symmetric (see above), when $\dim\left(y^{(m')}\right) > \dim\left(y^{(m)}\right)$, in which case the mapping reduces to $\left(y^{(m')}\right) = T\left(y^{(m)}, u^{(m)}\right)$, and when moves are limited to $m' \in \{(m_{i-1}-1), m_{i-1}, (m_{i-1}+1)\}$.

***Example*:** In Bayesian statistics the parameters of a model are considered random variables, reflecting a priori uncertainty on the part of the researcher that is reduced a posteriori after the data are observed. Inference focuses on their posterior distribution, which summarizes all information about the parameters. According to Bayes Theorem, the posterior distribution is proportional to the prior distribution of the parameters times the distribution of the data given the parameters. Bayesian estimation and inference has gained great popularity in business, in particular in marketing and finance, because even without strictly accepting the (attractive) fundamental properties of Bayesian inference, pragmatic Bayesians have found great value in MCMC algorithms to estimate complex models, especially as uninformative prior distributions can be used. Simpler illustrative examples follow.

***Example 1*:** The Weibull distribution is used in duration analysis applications to bankruptcy in finance, and in customer relationship management (CRM) in marketing. The observations $\{x_j\}$ for $j = 1,\ldots,J$ are a random sample of durations from a Weibull distribution: $f(x|\theta,\delta) = \theta\delta x^{\delta-1}\exp(-\theta x^{\delta})$ for $x > 0$. The prior distributions of the parameters are Gamma distributions: $p(\theta) = \frac{s_0^{r_0}}{\Gamma(r_0)}\theta^{r_0-1}\exp(-s_0\theta)$ and $p(\delta) = \frac{a_0^{b_0}}{\Gamma(b_0)}\delta^{a_0-1}\exp(-b_0\delta)$. The joint posterior distribution of the parameters is: $\pi\left(\theta,\delta|\{x_j\}\right) \propto p(\theta)p(\delta)\prod_{j=1}^{J} f\left(x_j|\theta,\delta\right)$, which does not have a convenient random number generator and can be sampled with MH within Gibbs. The full conditional distribution of θ given the data and δ_{i-1} is also a Gamma distribution: $\pi\left(\theta|\delta_{i-1},\{x_j\}\right) \propto \theta^{r_0+n-1}\exp\left(-\theta\left[s_0 + \sum_{j=1}^{J} x_j^{\delta_{i-1}}\right]\right)$. The full conditional distribution of δ given the data and θ_i does not have a known distributional form: $\pi\left(\delta|\theta_i,\{x_j\}\right) \propto \delta^{a_0+n-1}\left[\prod_{j=1}^{J} x_j^{\delta-1}\right]\exp\left(-b_0\delta - \theta_i\sum_{j=1}^{J} x_j^{\delta}\right)$. Thus, rMN can be used to generate the candidate δ^*. The MCMC algorithm to approximate the posterior distribution of the parameters is, for $i = 1,2,\ldots,n$:

1. Initialize the chain at (θ_0,δ_0).
2. Draw θ_i from a Gamma distribution $\theta_i = G\left(r_0 + n, s_0 + \sum_{j=1}^{J} x_j^{\delta_{i-1}}\right)$.
3. Sample $u_i \sim U(0,1)$, and generate a candidate δ^* from a log-normal distribution: $g(\delta^*|\delta_{i-1},\sigma) \propto \frac{1}{\delta^*}\exp\left[-\frac{1}{2\sigma^2}\left(\ln(\delta^*) - \ln(\delta_{i-1})\right)^2\right]$.
4. Compute $\alpha(\delta_{i-1},\delta^*) = \frac{\pi\left(\delta^*|\theta_i,\{x_j\}\right)\delta_{i-1}}{\pi\left(\delta_i|\theta_i,\{x_j\}\right)\delta^*}$.
5. Accept $\delta_i = \delta^*$ if $\alpha(\delta_{i-1},\delta^*) > u_i$, otherwise set $\delta_{i-1} = \delta_{i-1}$.

Extensions involve the parameterization of θ in terms of predictor variables $\theta_j = w_j\beta$, and the case where the durations are censored by the observation time; the estimations of the models in question involve extensions of the algorithms above.

***Example 2*:** Change-point regression models are popular in finance to describe financial time series data with a structural change, and used in marketing in models of stochastic preference and market shares. Here, the data $\{x_t\}$ are observed for time points $t = 1,\ldots,T$, and assumed to follow a binomial distribution: $f(x_t|\pi_t) = \pi_t^{x_t}(1-\pi_t)^{1-x_t}$; for $x_t \in \{0,1\}$. Two regression functions are

separated in time by an unknown switch-point $\tau : \pi_t = \Phi\left(w_t^{'}\beta_k\right)$, with $\beta_k = \beta_1$ for $t \leq \tau$, and $\beta_k = \beta_2$ for $t > \tau$. Φ is the Normal CDF used as an inverse link function, w_t is a vector of regressors, and $\beta_k \sim N(b_0, B_0)$ the prior distributions of its coefficients. The 'switch-point' has a uniform discrete prior on a subset of the observed timepoints: $\tau \sim U(c, d)$. The MCMC algorithm simplifies through the introduction of an auxiliary variable $z_t \sim N\left(w_t^{'}\beta_k, 1\right)$, with, $x_t = I(z_t > 0)$, and $I(\cdot)$ the indicator function. The MCMC algorithm to approximate the posterior distribution of the parameters is, for $i = 1, 2, \ldots, n$:

1. Sample for: $k = 1, 2:\ \beta_{k,i} \sim N(b_{k,i}, B_{k,i})$, with $b_{k,i} = B_{k,i}\left(B_0^{-1}b_0 + \sum_{t=L_{k,i}}^{t=U_{k,i}} w_t^{'}z_{t,i}\right)$, $B_{k,i} = \left(B_0^{-1} + \sum_{t=L_{k,i}}^{t=U_{k,i}} w_t w_t^{'}\right)$, and $L_{k,i} = 1 + (k-1)\tau_i$, $U_{k,i} = \tau + (k-1)(T - \tau_i)$.
2. Sample, for $L_{k,i} < t < U_{k,i}$: $z_{t,i} \sim N\left(w_t^{'}\beta_{k,i}, 1\right)I(z_{t,i} < 0)$ if $x_t = 0$, and $z_{t,i} \sim N\left(w_t^{'}\beta_{k,i}, 1\right)I(z_{t,i} > 0)$ if $x_t = 1$.
3. Sample τ_i using $Pr(\tau_i = r) = \frac{\prod_{t\leq r} f\left(x_t|w_t^{'}\beta_{1,i}\right)\prod_{t>r} f\left(x_t|w_t^{'}\beta_{2,i}\right)}{\sum_{s=c}^{d}\prod_{t\leq s} f\left(x_t|w_t^{'}\beta_{1,i}\right)\prod_{t>s} f\left(x_t|w_t^{'}\beta_{2,i}\right)}$.

Extensions of this MCMC procedure for multiples witch points are available, and extensions to an unknown number of switch points require RJMCMC.

***Example 3*:** Mixture models are used in finance to describe financial returns during different economic regimes, and are popular in marketing to identify unobserved heterogeneity in response-based market segmentation. The data $\{x_j\}$ are observed for individuals $j = 1, \ldots, J$, and assumed to follow a mixture Normal distribution with K classes and probabilities δ_k for which $0 < \delta_k < 1$ and $\sum_{k=1}^{m} \delta_k = 1$. Thus, $x_j \sim \sum_{k=1}^{m} \delta_k N\left(w_j^{'}\beta_k, \sigma_k^2\right)$. Here, β_k are class-specific regression coefficients associated with the vector of regressors w_t, with prior distributions $\beta_k \sim N(b_0, B_0)$. Further Inverse Gamma and Dirichlet priors are specified for: $\sigma_k^2 \sim IG\left(\frac{a_0}{2}, \frac{A_0}{2}\right)$ and $\delta_{1:m} \sim D(c_0, \ldots, c_0)$. The MCMC algorithm simplifies by introducing an auxiliary variable with a multinomial prior distribution: $z_j \sim M(\delta_{1:m})$ that indicates the membership of individual j in class k, that is $z_j = 1, \ldots, m$. The MCMC algorithm is, for $i = 1, 2, \ldots, n$:

1. Sample, for $k = 1, \ldots, m : \beta_{k,i} \sim N(b_{k,i}, B_{k,i})$, with $b_{k,i} = B_{k,i}\left(B_0^{-1}b_0 + \sum_{\{j:z_j=k\}} w_j^{'}x_j\right)$, and $B_{k,i} = \left(B_0^{-1} + \sum_{\{j:z_j=k\}} w_j w_j^{'}\right)$.
2. Sample, for $k = 1, \ldots, m :$ $\sigma_{k,i}^2 \sim IG\left(\frac{a_0+n_k}{2}, \frac{A_0+\sum_{\{j:z_j=k\}}\left(x_j - w_j^{'}\beta_{k,i}\right)^2}{2}\right)$, with $n_k = \sum_{\{j:z_j=k\}} 1$.
3. Sample $\delta_{1:m} \sim D(c_0 + n_1, \ldots, c_0 + n_m)$.
4. Sample z_j using $Pr(z_j = k) = \frac{\delta_k f\left(x_j|w_t^{'}\beta_{k,i}, \sigma_{k,i}^2\right)}{\sum_s \delta_s f\left(x_j|w_t^{'}\beta_{s,i}, \sigma_{s,i}^2\right)}$.

This sampler, like that for many mixture models, suffers from "label switching," a problem in which the class parameters switch across the class labels during the iterations. Several solutions are available, including ordering the mixture probabilities or post-processing of the draws.

Furthermore, the above algorithm can be extended to include the number of classes $m = 1, \ldots, m_{\max}$, using RJMCMC. A step is added to the algorithm in which two randomly chosen classes (k_1 and k_2) are merged (k_*), or one randomly chosen class is split. A splitting decision is usually made with probability $\eta_m = 0.5$, a merging decision with $(1 - \eta_m)$, for, $m = 2, \ldots, (m_{\max} - 1)$, and $\eta_1 = 0$ and $\eta_{m_{\max}} = 1$. The merge move involves matching of moments of the class-distributions, involving the computation of β_{k_*} such that the mean $\mu_{k_*} = w^{'}\beta_{k_*}$ of the new class matches that of k_1 and k_2, as does the variance $\sigma_{k_*}^2$:

M1. Randomly select $k_1 \propto 1/m$ and find k_2 'most similar' to k_1.
M2. Compute $\delta_{k_*} = \delta_{k_1} + \delta_{k_2}$.
M3. Match $\mu_{k_*} = \frac{\delta_{k_1}}{\delta_{k_*}}\mu_{k_1} + \frac{\delta_{k_2}}{\delta_{k_*}}\mu_{k_2}$.
M4. Compute $\sigma_{k_*}^2 = \frac{\delta_{k_1}}{\delta_{k_*}}\left(\mu_{k_1}^2 + \sigma_{k_1}^2\right) + \frac{\delta_{k_2}}{\delta_{k_*}}\left(\mu_{k_2}^2 + \sigma_{k_2}^2\right) - \mu_{k_*}^2$.
M5. Recompute z_j using step 4 above.

The split move operates as follows, and again involves matching of the first two moments of the class-distributions, of the old and new classes:

S1. Randomly select $k_* \propto 1/m$, and draw the auxiliary variables $u_{1:3} \sim Beta(a, b)$.
S2. Compute $\delta_{k_1} = u_1\delta_{k_*}$, and $\delta_{k_2} = (1 - u_1)\delta_{k_*}$.
S3. Match $\mu_{k_1} = \mu_{k_*} - u_2\sigma_{k_*}\sqrt{\frac{\delta_{k_2}}{\delta_{k_1}}}$, and $\mu_{k_2} = \mu_{k_*} + u_2\sigma_{k_*}\sqrt{\frac{\delta_{k_1}}{\delta_{k_2}}}$.

S4. Compute $\sigma_{k_1}^2 = u_3\left(1-u_2^2\right)\sigma_{k_*}^2 \frac{\delta_{k*}}{\delta_{k_1}}$, and $\sigma_{k_2}^2 = (1-u_3)\left(1-u_2^2\right)\sigma_{k_*}^2 \frac{\delta_{k*}}{\delta_{k_2}}$.

S5. Recompute z_j using step 4 above.

The split/merge proposal is accepted with probability $\min(\alpha(y,y'),1)$, computed as outlined in the RJ algorithm above (and the split is rejected if k_2 is not 'most similar' to k_1 to ensure reversibility). Here $h_m\left(u^{(m)}|y,\Phi\right) = Beta\,(a,b)$, and $q\left(m'|m,\Psi\right) = P\left(m'|m_{i-1}\right) = 0.5$ in the RJ algorithm described above. The split/merge moves are reversible, as $T_{m,m'}$ is defined in S2-S4, and $T_{m,m'}^{-1}$ in M2-M3. The split/merge moves may be combined with "birth/death" moves, randomly chosen with probabilities 0.5/0.5. In a birth move the parameters of a new class are drawn at random from proposal distributions on the appropriate support (e.g., $\delta_{k_*} \sim Beta, \beta_k \sim MVN, \sigma_k^{-2} \sim Gamma$), and the weights are rescaled so that they sum to one. In a death move an empty class is deleted, and the remaining weights are rescaled (Richardson and Green 1997).

See

- Acceptance-Rejection Method
- Importance Sampling
- Inverse Transform Method
- Markov Chains
- Monte Carlo Simulation
- Reversible Markov Chain/Process
- Simulation of Stochastic Discrete-Event Systems
- Simulation Optimization
- Variance Reduction Techniques in Monte Carlo Methods

References

Albert, J. H., & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association, 88*(422), 669–679.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society, Series B, 41*, 143–168.

Gelfand, A. E., & Smith, A. F. M. (1990). Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association, 85*, 398–409.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis*. London: Chapman and Hall.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 6*, 721–741.

Gill, J. (2008). *Bayesian methods: A social and behavioral sciences approach*. New York: Chapman &Hall.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika, 82*(4), 711–732.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains, and their applications. *Biometrika, 57*, 97–109.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics, 21*, 1087–1091.

Neal, R. M. (2003). Slice sampling. *Annals of Statistics, 31*(3), 705–767.

Press, S. J. (2003). *Subjective and objective Bayesian statistics* (2nd ed.). New York: Wiley.

Richardson, S., & Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society, Series B, 59*, 731–792.

Zellner, A. (1971). *An introduction to Bayesian inference in econometrics*. New York: Wiley.

Markov Chains

Carl M. Harris
George Mason University, Fairfax, VA, USA

Introduction

A Markov chain is a Markov process $\{X(t), t \in T\}$ whose state space S is discrete, while its time domain T may be either continuous or discrete. Only considered here is the countable state-space problem. Classic texts treating Markov chains include Breiman (1986), Çinlar (1975), Chung (1967), Feller (1968), Heyman and Sobel (2004), Isaacson and Madsen (1976), Iosifescu (1980), Karlin and Taylor (1975), Kemeny and Snell (1976), Kemeny, Snell and Knapp (1976), and Meyn and Tweedie (2009).

As a stochastic process of the Markov type, chains possess the Markov or lack-of-memory (memoryless) property, which means that the probabilities of future events are completely determined by the present state of the process and the probabilities of its behavior from the present point on. In other words, the past behavior of the process provides no additional information in

determining the probabilities of future events if the current state of the process is known. Thus, the discrete process $\{X(t), t \in T\}$ is a Markov chain if, for any $n > 0$, any $t_1 < t_2 < \ldots < t_n < t_{n+1}$ in the time domain T, any states $i_1, i_2, \ldots, i_n$ and any state j in the state space S,

$$\Pr\{X(t_{n+1}) = j | X(t_1) = i_1, \ldots, X(t_n) = i_n\} = \Pr\{X(t_{n+1}) = j | X(t_n) = i_n\}.$$

The conditional transition probabilities on the right-hand side of this equation can be simplified by mapping the n time points directly into the nonnegative integers and renaming state i_n as i. Then the probabilities are only a function of the pair (i, j) and the transition number n. Oftentimes, it is assumed that the transition probabilities are stationary, i.e., time invariant, resulting in a square (possibly infinite) matrix $\boldsymbol{P} = [p_{ij}]$ (viz., the single-step transition matrix), which gives all conditional probabilities of moving to state j in a transition, given that the chain is currently in state i. (Any matrix with the property that its rows are nonnegative numbers summing to one is called a stochastic matrix, whether or not it is associated with a particular Markov chain).

Examples of Markov Chains

1. *Random Walk*. In its simplest form, an object moves to the left one space at each transition time with probability p or to the right with probability $1 - p$. The problem can be kept finite by requiring reflecting barriers at fixed left-and right-hand points, say M and N, such that the transition probabilities send the chain back to states $M + 1$ and $N - 1$, respectively, whenever it reaches M or N. One important variation on this problem allows the object to stay put with non-zero probability.
2. *Gambler's Ruin*. A gambler makes repeated independent bets and wins \$1 on each bet with probability p or loses \$1 with probability $1 - p$. The gambler starts with an initial stake and will play repeatedly until all money is lost or until the fortune increases to \$$M$. Let X_n equal the gambler's wealth after n plays. The stochastic process $\{X_n, n = 0, 1, 2, \ldots\}$ is a Markov chain with state space $\{0, 1, 2, \ldots, M\}$. The Markov property follows from the assumption that outcomes of successive bets are independent events. The Markov model can be used to derive performance measures of interest for this situation, such as the probability of losing all the money, the probability of reaching the goal of \$$M$, and the expected number of bets before the game terminates. All these performance measures are functions of the gambler's initial state x_0, probability p and goal \$$M$. (The gambler's fortune is thus a random walk with absorbing boundaries 0 and M). The gambler's ruin problem is a simplification of more complex systems that experience random rewards, risk, and possible ruin, such as insurance companies.
3. *Coin Toss Sequence*. Consider a series of independent tosses of a fair coin. One Markov chain is obtained by associating state 1, 2, 3 or 4 at time n depending on whether the outcomes of tosses $n - 1$ and n are (H,H), (H,T), (T,H) or (T,T), respectively.

Define the n-step transition probability $p_{ij}^{(n)}$ as the probability that the chain moves from state i to state j in n steps, and write

$$P_{ij}^{(n)} = \Pr\{X_{m+n} = j | X_m = i\} \quad \text{for all } m \geq 0 \;\; n > 0.$$

Then it follows that the n-step transition probabilities can be computed using the Chapman-Kolmogorov equations

$$P_{ij}^{(n+m)} = \sum_{k=0}^{\infty} p_{ik}^{(n)} \, p_{kj}^{(m)} \text{ for all } n, m, i, \; j \geq 0.$$

In particular, for $m = 0$,

$$P_{ij}^{(n)} = \sum_{k=0}^{\infty} p_{ik}^{(n-1)} p_{kj} = \sum_{k=0}^{\infty} p_{ik} p_{kj}^{(n-1)}, \qquad n = 2, 3, \ldots; i, j \geq 0.$$

Denoting the matrix of n-step probabilities by $\boldsymbol{P}^{(n)}$, it follows that $\boldsymbol{P}^{(n)} = \boldsymbol{P}^{(n-k)} \boldsymbol{P}^{(k)} = \boldsymbol{P}^{(n-1)} \boldsymbol{P}$ and that $\boldsymbol{P}^{(n)}$ can be calculated as the nth power of the original single-step transition matrix $\boldsymbol{P}$.

To calculate the unconditional distribution of the state at time n requires specifying the initial probability distribution of the state, namely, $\Pr\{X_0 = i\} = p_i, i \geq 0$. Then the unconditional distribution of X_n is given by

$$\Pr\{X_n = j\} = \sum_{i=0}^{\infty} \Pr\{X_n = j | X_0 = i\} \Pr\{X_0 = i\}$$
$$= \sum_{i=0}^{\infty} p_i p_{ij}^{(n)}$$

which is equivalent to multiplying the row vector $\boldsymbol{p}$ by the jth column of $\boldsymbol{P}$.

Properties of a Chain

The ultimate long-run behavior of a chain is fully determined by the location and relative size of the entries in the single-step transition matrix. These probabilities determine which states can be reached from which other ones and how long it takes on average to make those transitions. More formally, state j is said to be reachable from state i, written $i \to j$, if it is possible for the chain to proceed from i to j in a finite number of transitions, i.e., if $p^{(n)} > 0$ for some $n \geq 0$. If, in addition, $j \to i$, then the two states are said to communicate with each other, written as $i \leftrightarrow j$. If every state is reachable from every other state in the chain, the chain is said to be irreducible, i.e., the chain is not reducible into subclasses of states that do not communicate with each other.

Furthermore, the period of state i is defined as the greatest common divisor, $d(i)$, of the set of positive integers n such that $p_{ii}^{(n)} > 0$ (with $d(i) \equiv 0$ when $p_{ii}^{(n)} = 0$ for all $n \geq 1$). If $d(i) = 1$, then i is said to be aperiodic; otherwise, it is periodic with period $d(i)$. Clearly, any state with $p_{ii}^{(n)} \geq 0$ is an aperiodic state. All states in a single communicating class must have the same period, and the full Markov chain is said to be aperiodic if all of its states have period 1.

For each pair of states (i, j) of a Markov chain, define $f_{ij}^{(n)}$ as the probability that a first return from i to j occurs in n transitions and f_{ij} as the probability of ever returning to j from i. If $f_{ij} = 1$, the expectation m_{ij} of this distribution is called the mean first passage time from i to j. When $j = i$, write the respective probabilities as $f_i^{(n)}$ and f_i, and the expectation as m_i, which is called the mean recurrence time of i. If $f_i = 1$ and $m_i < \infty$, state i is said to be positive recurrent or nonnull recurrent; if $f_i = 1$ and $m_i = \infty$, state i is said to be null recurrent; if $f_i < 1$, state i is said to be transient.

A major result that follows from the above is that if $i \leftrightarrow j$ and i is recurrent, then so is j. Furthermore, if the chain is finite, then all states cannot be transient and at least one must be recurrent; if all the states in the finite chain are recurrent, then they are all positive recurrent. More generally, all the states of an irreducible chain are either positive recurrent, null recurrent, or transient.

Example: Reflecting Random Walk

Consider such a chain with movement between its four states governed by the single-step transition matrix

$$\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 2/3 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{1}$$

All the states communicate since there exists a path with non-zero probability from state 1 back to state 1 hitting all the other states in the interim. All the states are recurrent and aperiodic, as well.

If the random walk were infinite instead and without reflecting barriers (on either side), then the chain would be recurrent if and only if it is equally probable to go from right to left from each state; for otherwise the system would drift to $+\infty$ or $-\infty$ without returning to any finite starting point.

Limiting Behavior

The major characterizations of the stochastic behavior of a chain are typically stated in terms of its long-run or limiting behavior. Define the probability that the chain is in state j at the nth transition as $\pi_j^{(n)}$, with the initial distribution written as $\pi_j^{(0)}$. A discrete Markov chain is said to have a stationary distribution $\boldsymbol{\pi} = (\boldsymbol{\pi_0}, \boldsymbol{\pi_1}, \ldots)$ if these (legitimate) probabilities satisfy the vector–matrix equation $\boldsymbol{\pi} = \boldsymbol{\pi P}$. When written out in simultaneous equation form, the problem is equivalent to solving

$$\pi_j = \sum_i \pi_i p_{ij}, \quad j = 0, 1, 2, \ldots, \text{ with}$$
$$\sum_i \pi_i = 1.$$

The chain is said to have a long-run, limiting, equilibrium, or steady-state probability distribution $\boldsymbol{\pi} = (\boldsymbol{\pi_0}, \boldsymbol{\pi_1}, \ldots)$ if

$$\lim_{n\to\infty} \pi_j^{(n)} = \lim_{n\to\infty} \Pr\{X_n = j\} = \pi_j, \;\; j = 0, 1, 2, \ldots$$

A Markov chain that is irreducible, aperiodic and positive recurrent is said to be ergodic, and the following theorem relates these properties to the existence of stationary and/or limiting distributions.

Theorem: If $\{X_n\}$ is an irreducible, aperiodic, time-homogeneous Markov chain, then limiting probabilities

$$\pi_j = \lim_{n\to\infty} \Pr\{X_n = j\}, \;\; j = 0, 1, 2, \ldots$$

always exist and are independent of the initial state probability distribution. If all the states are either null recurrent or transient, then $\pi_j = 0$ for all j and no stationary distribution exits; if all the states are instead positive recurrent (thus the chain is ergodic), then $\pi_j > 0$ for all j the set $\{\pi_j\}$ also forms a stationary distribution, with $\pi_j = 1/m_j$.

It is important to observe that the existence of a stationary distribution does not imply that a limiting distribution exists. An example is the simple Markov chain

$$\boldsymbol{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

For this chain, it is easy to show that the vector $\boldsymbol{\pi} = (1/2, 1/2)$ solves the stationary equation. However, since the chain is oscillating between states 1 and 2, there will be no limiting distribution. The chain clearly has period 2, which violates the sufficient conditions for the above ergodic theorem. Combined with the earlier discussion, this implies that an irreducible finite-state chain needs to be aperiodic to be ergodic. Note that the stationary distribution (1/2, 1/2) still has meaning because it gives the fraction of time the chain spends in each state in the limit, even though there is periodic oscillation.

More on the Reflecting Random Walk

The example Markov chain with single-step transition matrix given by (1) is ergodic, so its steady-state probabilities are found by solving $\boldsymbol{\pi} = \boldsymbol{\pi P}$, written out as the simultaneous system

$$\begin{aligned}
\pi_1 &= \frac{1}{3}\pi_2 \\
\pi_2 &= \pi_1 + \frac{1}{3}\pi_2 + \frac{1}{3}\pi_3 \\
\pi_3 &= \frac{1}{3}\pi_2 + \pi_4 \\
\pi_4 &= \frac{2}{3}\pi_3.
\end{aligned}$$

When these equations are solved and normalized (to sum to 1), a unique solution is found $\boldsymbol{\pi} = (1/9, 3/9, 3/9, 2/9)$. Furthermore, the limiting n-step matrix, $\lim_{n\to\infty} \boldsymbol{P}^n$, would have identical rows all equal to the vector $\boldsymbol{\pi}$.

More on the Gambler's Ruin Problem

For the Gambler's Ruin, there are three classes of states, $\{0\}$, $\{1, 2, \ldots, M-1\}$, and $\{M\}$. After a finite time, the gambler will either reach the goal of M units or lose all the money. Of particular interest is the probability that the gambler's fortune will grow to M before all the resources are lost, denoted here by p_i, $i = 0, 1, \ldots, M$. It is not too difficult to show that

$$p_i = \begin{cases} \dfrac{1 - [(1-p)/p]^i}{1 - [(1-p)/p]^M} & \text{if } p \neq \dfrac{1}{2} \\ \dfrac{i}{M} & \text{if } p = \dfrac{1}{2}. \end{cases}$$

More on the Coin Toss Sequence Problem

For the coin toss sequence example, the single-step transition matrix is given by

$$\boldsymbol{P} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}.$$

This particular matrix is very special since its columns also add up to 1; such a matrix is said to be doubly stochastic. It can be shown that any doubly stochastic transition matrix coming from a recurrent and aperiodic finite chain has the discrete uniform stationary probabilities $\pi_j = 1/M$.

Concluding Remarks

For continuous-time Markov chains, the analog for the single-step transition matrix is the transition rate matrix (infinitesimal generator), where the matrix entries of probabilities are replaced by rates of exponentially distributed random variables. The holding time in a state in a continuous-time Markov chain is exponentially distributed, the analog to the geometric holding time in a state of a discrete-time Markov chain. Well-known examples of continuous-time Markov chains include birth-death processes (analog to random walk), the Poisson process, and many queueing systems with exponentially distributed interarrival and service times, e.g., Jackson queueing networks.

See

- ▶ Birth-Death Process
- ▶ Markov Processes
- ▶ Matrix-Analytic Stochastic Models
- ▶ Networks of Queues
- ▶ Poisson Process
- ▶ Queueing Theory
- ▶ Stochastic Process

References

Breiman, L. (1986). *Probability and stochastic processes, with a view toward applications* (2nd ed.). Palo Alto, CA: The Scientific Press.

Chung, K. L. (1967). *Markov chains with stationary transition probabilities*. New York: Springer-Verlag.

Çinlar, E. (1975). *Introduction to stochastic processes*. Englewood Cliffs, NJ: Prentice-Hall.

Feller, W. (1968). *An introduction to probability theory and its applications* (3rd ed., Vol. 1). New York: Wiley.

Heyman, D. P., & Sobel, M. J. (2004). *Stochastic models in operations research, volume I: Stochastic processes and operating characteristics*. New York: Dover.

Iosifescu, M. (1980). *Finite Markov processes and their application*. New York: Wiley.

Isaacson, D. L., & Madsen, R. W. (1976). *Markov chains: Theory and applications*. New York: Wiley.

Karlin, S., & Taylor, H. M. (1975). *A first course in stochastic processes* (2nd ed.). New York: Academic.

Kemeny, J. G., & Snell, J. L. (1976). *Finite Markov chains*. New York: Springer-Verlag.

Kemeny, J. G., Snell, J. L., & Knapp, A. W. (1976). *Denumerable Markov chains* (2nd ed.). New York: Springer-Verlag.

Meyn, S., & Tweedie, R. L. (2009). *Markov chains and stochastic stability* (2nd ed.). New York: Cambridge University Press.

Markov Decision Processes

Chelsea C. White III
Georgia Institute of Technology, Atlanta, GA, USA

Introduction

The finite-state, finite-action Markov decision process (MDP) is a model of sequential decision making under uncertainty. MDPs have been applied in such diverse fields as health care, highway maintenance, inventory, machine maintenance, cash-flow management, and regulation of water reservoir capacity (Derman 1970; Hernandez-Lermer 1989; Ross 1995; White 1969). After defining an MDP and providing a simple illustrative example, various solution procedures for several different types of MDPs are presented, all of which are based on dynamic programming (Bertsekas 2007; Howard 1971; Puterman 2005; Sennott 1999).

Problem Formulation

Let $k \in \{0,\ 1,\ \ldots,\ K-1\}$ represent the kth stage or decision epoch, i.e., when the kth decision must be selected; $K < \infty$ represents the planning horizon of the Markov decision process. Let s_k be the state of the system to be controlled at stage k. This state must be a member of a finite set S, called the state space, where $s_k \in S$, $k = 0, 1, \ldots, K$. The state process $\{s_k, k = 0, 1, \ldots, K\}$ makes transitions according to the conditional probabilities

$$p_{ij}(a) = \Pr\{s_{k+1} = j | s_k = i,\ a_k = a\},$$

where a_k is the action selected at stage k. The action selected must be a member of the finite action space A, which is allowed to depend on the current state value, i.e., $a_k \in A(i)$ when $s_k = i$, thus allowing a_k to be selected on the basis of the current state s_k for all k. Let δ_k be a mapping from the state space into the action space satisfying $\delta_k(s_k) \in A(s_k)$. Then δ_k is called a policy and a sequence of policies $\pi = \{\delta_0, \ldots, \delta_{K-1}\}$ is known as a strategy.

Let $r(i, a)$ be the one-stage reward accrued at stage $k = 0, 1, \ldots, K-1$, if $s_k = i$ and $a_k = a$. Assume $\bar{r}(i)$ is the terminal reward accrued at stage K (assuming $K < \infty$) if

$s_k = i$. The total discounted reward over the planning horizon accrued by strategy $\pi = \{\delta_0, \ldots, \delta_{K-1}\}$ is then given by

$$\sum_{k=0}^{K-1} \beta^k r(s_k, a_k) + \beta^K \bar{r}(s_k)$$

where $a_k = \delta_k(s_k)$, $k = 0, 1, \ldots, K-1$, where β is the nonnegative real-valued discount factor. The problem objective is to select a strategy that maximizes the expected value of the total discounted reward, with respect to the set of all strategies. Any such strategy is called an optimal strategy.

Example — An inspector must decide at each stage, on the basis of a machine's current state of deterioration, whether to replace the machine, repair it, or do nothing. Assume that the machine can be in one of M states, i.e., the state space is $S = \{1, \ldots, M\}$, where 1 represents the perfect machine state, M represents the failed machine state, and $1 < m < M$ represents an imperfect but functioning state of the machine. Each week the machine inspector can choose to let the machine produce (the do-nothing decision $a = 1$), completely replace the machine (the replace decision $a = R$), or perform some sort of maintenance on the machine, $1 < a < R$. Thus, the action space is $A = \{1, \ldots, R\}$. Generally, these problems are expressed in terms of costs rather than rewards, which can be formulated as $r(i, a) = -c(i, a)$, where $c(i, a)$ be the cost accrued over the following week if at the beginning of the week the machine is in state i and the machine inspector selects action a. Let β be the current value of a dollar to be received next week. Assume the transition probabilities $p_{ij}(a)$ are known for all $i, j \in S$, $a \in A$, where generally $p_{i1}(\mathrm{R}) = 1$ and $p_{ij}(1) = 0$ if $j < i$.

Dynamic Programming Formulation (Finite Stage Case)

To formulate the MDP as a dynamic program for the finite planning horizon case, let $f_k(i)$ be the optimal expected total discounted reward accrued from stage k through the terminal stage K, assuming $s_k = i$. Note that $f_k(i)$ should differ from $f_{k+1}(s_{k+1})$ only by the reward accrued at stage k. In fact, it is easily shown that f_k and f_{k+1} are related by the dynamic programming optimality equation

$$f_k(i) = \max_{a \in A(i)} \left\{ r(i, a) + \beta \sum_{j \in S} p_{ij}(a) f_{k+1}(j) \right\},$$

which has boundary condition $f_K(i) = \bar{r}(i)$. Note also that an optimal strategy $\pi^* = \{\delta_0^*, \ldots, \delta_{K-1}^*\}$ necessarily and sufficiently satisfies

$$f_k(i) = r[i, \delta_k^*(i)] + \beta \sum_j p_{ij}[\delta_k^*(i)] f_{k+1}(j)$$

for all $k = 0, 1, \ldots, K-1$. Thus, the action that should be taken at stage k, given $s_k = i$, is any action that achieves the maximum in

$$\max_{a \in A(i)} \left\{ r(i, a) + \beta \sum_j p_{ij}(a) f_{k+1}(j) \right\}.$$

The Infinite Horizon Discounted Reward Case

For the infinite horizon setting where $K = \infty$, there may exist strategies that could generate an infinite reward. However, if the discount factor β is strictly less than 1, no such strategy exists, which can be verified by noting that

$$\sum_{k=0}^{\infty} \beta^k r(s_k, a_k) \le \sum_{k=0}^{\infty} \beta^k \max_{(i,a)} |r(i, a)| = \frac{\max_{(i,a)} |r(i, a)|}{1 - \beta}.$$

Not surprisingly, the dynamic program for the infinite horizon case can be related to the dynamic program for the finite horizon case. Defining m as the number of stages to go until the terminal stage of the finite horizon case, the dynamic program for the finite horizon problem can then be rewritten as

$$g_{m+1}(i) = \max_{a \in A(i)} \left\{ r(i, a) + \beta \sum_j p_{ij}(a) g_m(j) \right\}$$

where $f_k(i) = g_{K-k}(i)$. Now the optimal expected total discounted reward should be $g(i) = \lim_{m \to \infty} g(i)$ for initial state i, which should satisfy

$$g(i) = \max_{a \in A(i)} \left\{ r(i,a) + \beta \sum_j p_{ij}(a) g(i) \right\} \quad (1)$$

if the limit and maximization operators can be interchanged. It so happens that this interchange is possible under the conditions considered here, and hence the optimal expected total discounted reward uniquely satisfies (1). It can also be shown that an optimal strategy exists that is stage invariant and that this strategy, or equivalently, policy, satisfies

$$g(i) = r[i, \delta^*(i)] + \beta \sum_j p_{ij}[\delta^*] g(j) \quad (1a)$$

for all $i \in S$.

Solution Procedures

Three different computational approaches for determining g and δ^* in (1) are presented.

Linear Programming — The following linear program can solve the infinite-horizon discounted MDP:

$$\text{minimize} \sum_{i \in S} g(i)$$
$$\text{subject to} \quad g(i) - \beta \sum_j p_{ij}(a) g(j) \geq r(i,a)$$

where the constraint inequality must be satisfied for all $i \in S$ and $a \in A(i)$, $i \in S$.

Successive Approximations — This procedure, in its simplest form, involves determining $g_m(i)$ for large m, using the iteration equation

$$g_m(i) = \max_{a \in A(i)} \left\{ r(i,a) + \beta \sum_j p_{ij}(a) g_{m-1}(j) \right\},$$

where $g_0(i)$ can be arbitrarily selected; however, it is generally beneficial to select g_0 as close to g as possible if there is some way of estimating g *a priori*.

Policy Iteration — This computational procedure involves the following iterative approach:

Step 0: Select δ

Step 1: Determine g_δ where g_δ, satisfy

$$g_\delta(i) = r[i, \delta(i)] + \beta \sum_j p_{ij}[\delta(i)] g_\delta(j).$$

Note that

$$g_\delta = (I - \beta P_\delta)^{-1} r_\delta$$

where $P_\delta = \{p_{ij}[\delta(i)]\}$, $g_\delta = \{g_\delta(i)\}$, $r_\delta = \{r[i, \delta(i)]\}$, I is the identity matrix, and the inverse is guaranteed to exist since $\beta < 1$.

Step 2: Determine δ' that satisfies

$$r[i, \delta'(i)] + \beta \sum_j p_{ij}[\delta'(i)] g_\delta(j) = \max_{a \in A(i)} \left\{ r(i,a) + \beta \sum_j p_{ij}(a) g_\delta(j) \right\}.$$

Step 3: Set $\delta = \delta'$ and return to Step 1 until g_δ and $g_{\delta'}$ are sufficiently close.

Note that each of the above solution procedures is far more efficient than exhaustive enumeration. Combining policy iteration and successive approximations can lead to efficient computational procedures for large-scale infinite-horizon discounted MDPs.

Markov Decision Processes without Discounting (The Average Reward Case)

Assume that the criterion is

$$\lim_{K \to \infty} \left(\frac{1}{K+1} \right) E \left\{ \sum_{k=0}^{K} r(S_k, a_k) \right\}$$

which is the expected average reward criterion. When the system operates under stationary policy δ, it can be shown that there exist values $v_\delta(i)$, $i \in S$, and a state independent gain γ_δ, which satisfy

$$\gamma_\delta + v_\delta(i) = r[i, \delta(i)] + \sum_j p_{ij}[\delta(i)] v_\delta(i) \quad (2)$$

if P_δ is ergodic. Let γ^*, δ^* and v be such that

$$\gamma^* + v(i) = \max_{a \in A(i)} \left\{ r(i,a) + \beta \sum_j p_{ij} v(j) \right\} = r[i, \delta^*(i)] + \sum_j p_{ij}[\delta^*(i)] v(j)$$

where P_δ is assumed ergodic for all δ. Then, γ^* is the value of the criterion generated by an optimal strategy and δ^* is an optimal strategy. The following is a policy iteration procedure for determining γ^*, δ^* and v, where it is necessary only to know v up to a positive constant due to the sum-to-one characteristic of the probabilities.

Algorithm. Step 0: Choose δ.

Step 1: Solve equation (2) for v_δ and γ_δ, where for some i, $v_\delta(i) = 0$.

Step 2: Determine a policy δ' that achieves the maximum in

$$\max_{a \in A(i)} \left\{ r(i,a) + \sum_j p_{ij} v_\delta(i) \right\}.$$

Step 3: Set $\delta = \delta'$ and go to Step 1 until γ_δ and $\gamma_{\delta'}$ are sufficiently close.

Concluding Remarks

The discussion has focused on the MDP setting where the state and action spaces are finite; the reward is separable with respect to stage; all rewards, the discount factor, and all transition probabilities are known precisely and the current state can be accurately made available to the decision maker before selection of the current alternative. The references treat more general settings. Much research effort is devoted to improving the computational tractability of large-scale MDPs so as to improve both the validity and tractability of this modeling tool. One such approach is approximate dynamic programming, which is treated in detail in Volume II of Bertsekas (2007).

See

- Approximate Dynamic Programming
- Dynamic Programming
- Markov Chains
- Markov Processes

References

Bertsekas, D.P. (2007). *Dynamic programming and optimal control* (Vols. I & II, 3rd edn.). (Vol. II, 4th edn., 2012). Nashua, NH: Athena Scientific.

Derman, C. (1970). *Finite state Markovian decision processes*. New York: Academic.

Hernandez-Lermer, O. (1989). *Adaptive Markov control processes*. New York: Springer.

Howard, R. (1971). *Dynamic programming and Markov processes*. Cambridge, MA: MIT Press.

Puterman, M. L. (2005). *Markov decision processes: Discrete stochastic dynamic programming*. New York: John Wiley & Sons.

Ross, S. M. (1995). *Introduction to stochastic dynamic programming*. New York: Academic.

Sennott, L. I. (1999). *Stochastic dynamic programming and the control of queueing systems*. New York: John Wiley & Sons.

White, D. J. (1969). *Markov decision processes*. Chichester, UK: John Wiley.

Markov Processes

Douglas R. Miller
George Mason University, Fairfax, VA, USA

Introduction

A Markov process is a stochastic process $\{X(t), t \in T\}$ with state space S and time domain T that satisfies the Markov property, which is also known as lack of memory. In general, probabilities of behavior of a stochastic process at future times usually depend on the behavior of the process at times in the past. The Markov property means that probabilities of future events are completely determined by the present state of the process: if the current state of the process is known, then the past behavior of the process provides no additional information in determining the probabilities of future events. Mathematically, the process $\{X(t), t \in T\}$ is Markov if, for any $n > 0$, any $t_1 < t_2 < \ldots, < t_n < t_{n+1}$ in the time domain T, and any states $x_1, x_2, \ldots, x_n$ and any set A in the state space S,

$$\Pr\{X(t_{n+1}) \in A | X(t_1) = x_1, \ldots, X(t_n) = x_n\} = \Pr\{X(t_{n+1}) \in A | X(t_n) = x_n\}.$$

The conditional probabilities on the right-hand side of this equation are the transition probabilities of the Markov process; they play a key role in the study of Markov processes. The transition probabilities of the process are presented as a transition function $p(s,x;t,A) = \Pr\{X(t) \in A \,|\, X(s) = x\}$, $s < t$, for $s, t \in T$, $x \in S$, and $A \subset S$. The initial distribution of the

process is $q(A) = \Pr\{X(0) \in A\}$, for $A \subset S$. The distribution of a Markov process is uniquely determined by an initial distribution $q(\cdot)$ and a transition function $p(\cdot,\ldots,\cdot)$: for $0 = t_0 < t_1 < \ldots < t_n$ in the time domain, and subsets A_1, A_2,..., A_n of the state space S,

$$\begin{aligned}&\Pr\{X(t_1) \in A_1, \ldots, X(t_n) \in A_n\}\\ &= \int_{x_0 \in S} q(dx_0) \int_{x_1 \in A_1} p(t_0, x_0; t_1, dx_1) \cdots\\ &\int_{x_{n-1} \in A_{n-1}} p(t_{n-2}, x_{n-2}; t_{n-1}, dx_{n-1}) p(t_{n-1}, x_{n-1}; t_n, A_n).\end{aligned}$$

An equivalent interpretation of the Markov property is that the past behavior and the future behavior of the process are conditionally independent given the present state of the process: for any $m > 0$, any $n > 0$, any $t_{-m} < \ldots, < t_{-1} < t_0 < t_1 < \ldots, < t_n$ in the time domain, and any state x_0 and any sets $A_1, A_2, \ldots, A_m$ and $B_1, B_2, \ldots, B_n$ in the state space S,

$$\begin{aligned}\Pr\{X(t_{-m}) \in A_m, \ldots, X(t_{-1}) \in A_1, X(t_1) \in B_1,\\ \ldots, X(t_n) \in B_n | X(t_0) = x_0\}\\ = \Pr\{X(t_{-m}) \in A_m, \ldots, X(t_{-1}) \in A_1 | X(t_0) = x_0\}\\ \cdot \Pr\{X(t_1) \in B_1, \ldots, X(t_n) \in B_n | X(t_0) = x_0\}.\end{aligned}$$

A Markov process has stationary transition probabilities if the transition probabilities are time invariant, i.e., for s, $t > 0$, $\Pr\{X(s + t) \in A \mid X(s) = x\} = \Pr\{X(t) \in A | X(0) = x\}$. In this case the transition function takes the simplified form $p_t(x, A) = \Pr\{X(t) \in A \mid X(0) = x\}$. Most Markov process models assume stationary transition probabilities.

Classification of Markov Processes

There is a natural classification of Markov processes according to whether the time domain T and the state space S are denumerable or non-denumerable. This yields four general classes. Denumerable time domains are usually modeled as the integers or non-negative integers. Non-denumerable time domains are usually modeled as the continuum ($\mathcal{R}$ or $[0, \infty]$). Denumerable state spaces can be modeled as the integers, but it is often useful to retain other descriptions of the states rather than simply enumerating them. Non-denumerable state spaces are usually modeled as a one or higher dimensional continuum. Roughly speaking, discrete is equivalent to denumerable and continuous is equivalent to non-denumerable. In 1907, Markov considered a discrete time domain and a finite state space; he used the word "chain" to denote the dependence over time, hence the term Markov chain for Markov processes with discrete time and denumerable states. See Maistrov (1974) for some historical discussion and see Appendix B of Howard (1971) for a reprint of one of Markov's 1907 papers. There is no universal convention for the scope of definition of Markov chain. Chung (1967) and most elementary operations research/management science textbooks (e.g., Hillier and Lieberman 2009) define Markov processes with denumerable state spaces to be Markov chains. Iosifescu (1980) and the Romanian school use the convention that Markov chain applies to discrete time and any state space, while Markov process applies to continuous time and any state space. The terminology varies in popular texts: Karlin and Taylor (1975, 1981) and Ross (1995) agree with Chung; Breiman (1968) and Çinlar (1975) agree with the Romanians. The terms discrete-time Markov chain (DTMC) and continuous-time Markov chain (CTMC) are sometimes used to clarify the situation.

Here are four examples of Markov processes representing the four classes with respect to discrete or continuous time and denumerable or continuous state space.

(a) *Gambler's Ruin* (*discrete time/denumerable states*). A gambler makes repeated bets. On each bet he wins \$1 with probability p or loses \$1 with probability $1 - p$. Outcomes of successive bets are independent events. He starts with a certain initial stake and will play repeatedly until he loses all his money or until he increases his fortune to \$$M$. Let X_n equal the gambler's wealth after n plays. The stochastic process $\{X_n, n = 0, 1, 2, \ldots\}$ is a discrete time Markov chain (DTMC) with state space $\{0, 1, 2, \ldots, M\}$. The Markov property follows from the assumption that outcomes of successive bets are independent events. The Markov model can be used to derive performance measures of interest for this situation: for example, the probability he loses all his money, the probability he reaches his goal of \$$M$, and the expected number of times he makes a bet. All these performance measures are

functions of his initial stake x_0, probability p and goal M. (The gambler's fortune is a random walk with absorbing boundaries 0 and M.) The gambler's ruin is a simplification of more complex systems that experience random rewards, risk, and possible ruin; for example, insurance companies.

(b) *A Maintenance System* (*continuous time/denumerable states*). A system consists of two machines and one repairman. Each machine operates until it breaks down. The machine is then repaired and put back into operation. If the repairman is busy with the other machine, the just broken machine waits its turn for repair. So, each machine cycles through the states: operating (O), waiting (W), and repairing (R). Labeling the machines as "1" and "2" and using the corresponding subscripts, the states of the system are (O_1,O_2), (O_1,R_2), (R_1,O_2), (W_1,R_2) and (R_1,W_2). Assume that all breakdown instances and repairs are independent of each other and that the operating times until breakdown and the repair times are random with exponential distributions. The mean operating times for the machines are $1/\alpha_1$ and $1/\alpha_2$, respectively (so the machines break down at rates α_1 and α_2). The mean repair times for the machines are $1/\beta_1$ and $1/\beta_2$, respectively (so the machines are repaired at rates β_1 and β_2). Letting $X_i(t)$ equal the state of machine i at time t, the stochastic process $\{(X_1(t), X_2(t)), 0 \leq t\}$ is a continuous time Markov chain (CTMC) on a state space consisting of five states. The Markov property follows from the assumption about independent exponential operating times and repair times. (The exponential distribution is the only continuous distribution with lack-of-memory.) For this type of system there are several performance measures of interest: for example, the long-run proportion of time both machines are broken or the long-run average number of working machines. This maintained system is a simplified example of more complex maintained systems.

(c) *Quality Control System (discrete time/continuous states)*. A manufacturing system produces a physical part that has a particularly critical length along one dimension. The specified value for the length is α. However, the manufacturing equipment is imprecise. Successive parts produced by this equipment vary randomly from the desired value, α. Let X_n equal the size of the nth part produced. The noise added to the system at each step is modeled as $D_n \sim \text{Normal}(0, \delta^2)$. The system can be controlled by attempting to correct the size of the $(n+1)$st part by adding $c_n = -\beta\,(x_n - \alpha)$ to the current manufacturing setting after observing the size x_n of the nth part; however, there is also noise in the control so that, in fact, $C_n \sim \text{Normal}\,(c_n,(\gamma c_n)^2)$ is added to the current setting. This gives $X_{n+1} = X_n + C_n + D_n$. The process $\{X_n,\ n = 0, 1, 2,\ldots\}$ is a discrete-time Markov process on a continuous state space. The Markov property will follow if all the noise random variables $\{D_n\}$ are independent and the control random variables $\{C_n\}$ depend only on the current setting (X_n) of the system. Performance measures of interest for this system include the long-run distribution of lengths produced (if the system is stable over the long-run). There is also a question of determining the values of β for which the system is stable and then finding the optimal value of β.

(d) *Brownian Motion* (*continuous time/continuous states*). In 1828, English botanist Robert Brown observed random movement of pollen grains on the surface of water. The motion is caused by collisions with water molecules. The displacement of a pollen grain as a function of time is a two-dimensional Brownian motion. A one-dimensional Brownian motion can be obtained by scaling a random walk: Consider a sequence of independent, identically-distributed random variables, Z_i, with $\Pr\{Z_i = +1\} = \Pr\{Z_i = -1\} = 1/2$, $i = 1, 2,\ldots$. Let $S_n = \sum_{i=1}^{n} Z_i$, $n = 0, 1, 2,\ldots$. Then, let $X_n(t) = n^{-1/2} S_{[nt]}, 0 \leq t \leq 1$, $n = 1, 2,\ldots$, where $[nt]$ is the greatest integer $\leq nt$. As $n \to \infty$, the sequence of processes $\{X_n(t), 0 \leq t \leq 1\}$ converges to $\{W(t), 0 \leq t \leq 1\}$, standard Brownian motion or the Wiener process; see Billingsley (1968). The Wiener process is a continuous-time, continuous-state Markov process. The sample paths of the Wiener process are continuous. Diffusions are the general class of continuous-time, continuous-state Markov processes with continuous sample paths. Diffusion models are useful approximations to discrete processes analogous to how the Wiener process is an approximation to the above random

walk process $\{S_n,\ n = 0,\ 1,\ 2,\ldots\}$; see Glynn (1990). Geometric Brownian motion $\{Y(t), 0 \le t\}$ is defined as $Y(t) = \exp(\sigma W(t))$, $0 \le t$; it is a diffusion. Geometric Brownian motion has been suggested as a model for stock price fluctuations; see Karlin and Taylor (1975). A performance measure of interest is the distribution of the maximum value of the process over a finite time interval.

There are various performance measures that can be derived for Markov process models. Some specific performance measures were mentioned for the above examples. Some general behavioral properties and performance measures are now described. The descriptions are for a discrete-time Markov chain $\{X_n,\ n = 0,\ 1,\ 2,\ldots\}$ but similar concepts apply to other classes of Markov processes. A Markov chain is strongly ergodic if X_n converges in distribution as $n \to \infty$, independent of the initial state x_0. A Markov chain is weakly ergodic if $n^{-1}\sum_{i=1}^{n} X_i$ converges to a constant as $n \to \infty$, independent of the initial state x_0. Also as $n \to \infty$, under certain conditions and for real-valued functions f: $S \to \mathbb{R}$, $f(X_n)$ converges in distribution, $n^{-1}\sum_{i=1}^{n} f(X_i)$ converges to a constant, and $n^{-1/2}\sum_{i=1}^{n}[f(X_i) - Ef(X_i)]$ is asymptotically normal. Markov process theory identifies conditions for ergodicity, conditions for the existence of limits, and provides methods for evaluation of limits when they exist. For example, in the above maintained system example, $f(\cdot)$ might be a cost function and the performance measure of interest is long-run average cost. The above performance is long-run (or infinite-horizon, or steady-state, or asymptotic) behavior. Short-run (or finite-horizon, or transient) behavior and performance is also of interest. For a subset A of the state space S, the first passage time T_A is the time of the first visit of the process to A: $T_A = \min\{n\colon X_n \in A\}$. The hitting probability $\Pr\{T_A < \infty\}$, the distribution of T_A, and $E(T_A)$ are of interest. In the gambler's ruin example, the gambler wants to know the hitting probabilities for sets $\{0\}$ and $\{M\}$. Transient analysis of Markov processes investigates these and other transient performance measures. The analysis of performance measures takes on different forms for the four different classes of Markov processes.

Evaluation of performance measures for Markov process models of complex systems may be difficult. Standard numerical analysis algorithms are sometimes useful, and specialized algorithms have been developed for Markov models; for example, see Grassmann (1990). Workers in the field of computational probability have developed and evaluated numerical solution techniques for Markov models by exploiting special structure and probabilistic behavior of the system or by using insights gained from theoretical probability analysis. In this spirit, Neuts (1981) has developed algorithms for a general class of Markov chains. A structural property of Markov chains called reversibility leads to efficient numerical methods of performance evaluation; see Keilson (1979), Kelly (1979), and Whittle (1986). There is a relationship between discrete-time and continuous-time Markov chains called uniformization or randomization that can be used to calculate performance measures of continuous-time Markov chains; see Keilson (1979) and Gross and Miller (1984). For Markov chains with huge state spaces, Monte Carlo simulation can be used as an efficient numerical method for performance evaluation; see, for example, Hordijk, Iglehart and Schassberger (1976) and Fox (1990).

There are classes of stochastic processes related to Markov processes. There are stochastic processes that exhibit some lack of memory but are not Markovian. Regenerative processes have lack of memory at special points (regeneration points) but at other times the process has a memory; see Çinlar (1975). A semi-Markov process is a discrete-state continuous-time process that makes transitions according to a DTMC but may have general distributions of holding times between transitions; see Çinlar (1975). It is sometimes possible to convert a non-Markovian stochastic process into a Markov process by expanding the state description with supplementary variables; that is, $\{X(t), 0 \le t\}$ may be non-Markovian but $\{(X(t), Y(t)),\ 0 \le t\}$ is Markovian. Supplementary variables are often elapsed times for phenomena with memory; in this way very general discrete state stochastic systems can be modeled as Markov processes with huge state spaces. The general model for discrete-event dynamic systems is the generalized semi-Markov process (GSMP); see Whitt (1980) and Cassandras and Lafortune (2008).

The index set T of a stochastic process $\{X(t), t \in T\}$ may represent "time" or "space" or both, leading to temporal processes, spatial processes, or spatial-temporal processes when the index set is time, space, or space-time, respectively. Stochastic processes with multi-dimensional index sets are called random

fields. The Markov property can be generalized to the context of multi-dimensional index sets resulting in Markov random fields; see Kelly (1979), Kindermann and Snell (1980) and Whittle (1986). Markov random fields have many applications. They are models for statistical mechanical systems (interacting particle systems). They are useful in texture analysis and image analysis; see Chellappa and Jain (1993).

See

- ▶ Hidden Markov Models
- ▶ Markov Chain Monte Carlo
- ▶ Markov Chains
- ▶ Markov Decision Processes
- ▶ Markov Random Field
- ▶ Monte Carlo Simulation
- ▶ Regenerative Process
- ▶ Regenerative Simulation
- ▶ Reversible Markov Chain/Process

References

Billingsley, P. (1968). *Convergence of probability measures*. New York: Wiley.

Breiman, L. (1968). *Probability*. Reading, MA: Addison-Wesley.

Breiman, L. (1986). *Probability and stochastic processes, with a view toward applications* (2nd ed.). Palo Alto, CA: The Scientific Press.

Cassandras, C. G., & Lafortune, S. (2008). *Discrete event systems: Modeling and performance analysis* (2nd ed.). New York: Springer.

Chellappa, R., & Jain, A. (Eds.). (1993). *Markov random fields: Theory and application*. San Diego: Academic Press.

Chung, K. L. (1967). *Markov chains with stationary transition probabilities*. New York: Springer-Verlag.

Çinlar, E. (1975). *Introduction to stochastic processes*. Englewood Cliffs, NJ: Prentice-Hall.

Feller, W. (1968). *An introduction to probability theory and its applications, volume I* (3rd ed.). New York: Wiley.

Feller, W. (1971). *An introduction to probability theory and its applications, volume II* (2nd ed.). New York: Wiley.

Fox, B. L. (1990). Generating Markov-chain transitions quickly. *ORSA Journal on Computing, 2*, 126–135.

Glynn, P. W. (1989). A GSMP formalism for discrete event systems. *Proceedings of the IEEE, 77*, 14–23.

Glynn, P. W. (1990). Diffusion approximations. In D. P. Heyman & M. J. Sobel (Eds.), *Handbooks in OR and MS* (Vol. 2, pp. 145–198). Amsterdam: Elsevier Science.

Grassmann, W. K. (1990). Computational methods in probability. In D. P. Heyman & M. J. Sobel (Eds.), *Handbooks in OR and MS* (Vol. 2, pp. 199–254). Amsterdam: Elsevier Science.

Gross, D., & Miller, D. R. (1984). The randomization technique as a modelling tool and solution procedure for transient Markov processes. *Operations Research, 32*, 343–361.

Heyman, D. P., & Sobel, M. J. (1982). *Stochastic models in operations research, volume I: Stochastic processes and operating characteristics*. New York: McGraw-Hill.

Hillier, F. S., & Lieberman, G. J. (2009). *Introduction to operations research* (9th ed.). New York: McGraw-Hill.

Hordijk, A., Iglehart, D. L., & Schassberger, R. (1976). Discrete-time methods for simulating continuous-time Markov chains. *Advances in Applied Probability, 8*, 772–778.

Howard, R. A. (1971). *Dynamic probabilistic systems, volume I: Markov models*. New York: Wiley.

Iosifescu, M. (1980). *Finite Markov processes and their application*. New York: Wiley.

Isaacson, D. L., & Madsen, R. W. (1976). *Markov chains: Theory and applications*. New York: Wiley.

Karlin, S., & Taylor, H. M. (1975). *A first course in stochastic processes* (2nd ed.). New York: Academic Press.

Karlin, S., & Taylor, H. M. (1981). *A second course in stochastic processes*. New York: Academic Press.

Keilson, J. (1979). *Markov chain models – Rarity and exponentiality*. New York: Springer-Verlag.

Kelly, F. P. (1979). *Reversibility and stochastic networks*. New York: Wiley.

Kemeny, J. G., & Snell, J. L. (1976). *Finite Markov chains*. New York: Springer-Verlag.

Kemeny, J. G., Snell, J. L., & Knapp, A. W. (1976). *Denumerable Markov chains* (2nd ed.). New York: Springer.

Kindermann, R., & Snell, J. L. (1980). *Markov random fields and their applications*. Providence, RI: American Mathematical Society.

Maistrov, L. E. (1974). *Probability theory: A historical sketch*. New York: Academic Press.

Neuts, M. F. (1981). *Matrix-geometric solutions in stochastic models*. Baltimore: The Johns Hopkins University Press.

Parzen, E. (1962). *Stochastic processes*. San Francisco: Holden-Day.

Ross, S. M. (1995). *Stochastic processes* (2nd ed.). New York: Wiley.

Snell, J. L. (1988). *Introduction to probability*. New York: Random House.

Whitt, W. (1980). Continuity of generalized semi-Markov processes. *Mathematical Methods of Operations Research, 5*, 494–501.

Whittle, P. (1986). *Systems in stochastic equilibrium*. New York: Wiley.

Markov Property

When the behavior of a stochastic process $\{X(t), t \in T\}$ at times in the future depends only on the present state of the process (past behavior of the process affects the future behavior only through the present state of the process); viz., for any $n > 0$, any set of time points $t_1 < t_2 < \ldots < t_n < t_{n+1}$ in the time domain T, and any

states $x_1, x_2, \ldots, x_n$ and any set A in the same space, $\Pr\{X(t_{n+1}) \in A | X(t_1) = x_1, \ldots, X(t_n) = x_n\} = \Pr\{X(t_{n+1}) \in A | X(t_n) = x_n\}$.

See

- Markov Chains
- Markov Processes

Markov Random Field

A random field that satisfies a generalization of the Markov property.

See

- Markov Processes
- Random Field

Markov Renewal Process

When the times between successive transitions of a Markov chain are independent random variables indexed on the to and from states of the chain.

See

- Markov Chains
- Markov Processes
- Networks of Queues
- Renewal Process

Markov Routing

The process of assigning customers to nodes in a queueing network according to a Markov chain over the set of nodes, where $p(j, k)$ is the probability that a customer exiting node j proceeds next to node k, with $1 - \sum p(j, k)$ being the probability a customer leaves the network from node j (the sum is over all nodes of the network, including leaving the network altogether).

See

- Networks of Queues

Markovian Arrival Process (MAP)

- Matrix-Analytic Stochastic Models

Marriage Problem

Given a group of m men and m women, the marriage problem is to couple the men and women such that the total happiness of the group is maximized when the assigned couples marry. The women and the men determine an $m \times m$ table of happiness coefficients, where the coefficient a_{ij} represents the happiness rating for the couple formed by woman i and man j if they marry. The larger the a_{ij}, the higher the happiness. The problem can be formulated as an assignment problem whose solution matches each woman to one man. This result, which is due to the fact that the assignment problem has a solution in which the variables can take on only the values of 0 or 1, is sometimes used to prove that monogomy is the best form of marriage.

See

- Assignment Problem

Martingale

A stochastic process (with finite expectation) for which the conditional expectation of future values is equal to the present value. For example, for a discrete-time process $\{X_0, X_1, X_2, \ldots\}$,

$$E[X_{n+1} | X_0, X_1, \ldots X_n] = X_n.$$

Master Problem

The transformed extreme-point problem that results when applying the Dantzig-Wolfe decomposition algorithm.

See

► Dantzig-Wolfe Decomposition Algorithm

Matching

Richard W. Eglese
Lancaster University, Lancaster, UK

Introduction

Matching problems form an important branch of graph theory. They are of particular interest because of their application to problems found in Operations Research. Matching problems also form a class of integer-linear programming problems which can be solved in polynomial time. A good description of the historical development of matching problems and their solutions is contained in the preface of Lovasz and Plummer (2009).

Given a simple non-directed graph $G = [V, E]$ (where V is a set of vertices and E is a set of edges), then a matching is defined as a subset of edges M such that no two edges of M are adjacent. A matching is said to *span* a set of vertices X in G if every vertex in X is incident with an edge of the matching. A perfect matching is a matching which spans V. A maximum matching is a matching of maximum cardinality, i.e. a matching with the maximum number of members in the set.

A graph is called a bipartite graph if the set of vertices V is the disjoint union of sets V_1 and V_2 and every edge in E has the form (v_1, v_2) where v_1 is a member of V_1 and v_2 is a member of V_2.

Matching on Bipartite Graphs

The first type of matching problems consists of those which can be formulated as matching problems on a bipartite graph. For example, suppose V_1 represents a set of workers and V_2 represents a set of tasks to be performed. If each worker is able to perform a subset of the tasks and each task may be performed by some subset of the workers, the situation may be modeled by constructing a bipartite graph G, where there is an edge between v_1 in V_1 and v_2 in V_2 if and only if worker v_1 can perform task v_2. If it is assumed that each worker may only be assigned one task and each task may only be assigned to be carried out by one worker, the problem is an assignment problem. To find the maximum number of tasks which can be performed, the maximum matching on G must be found. If a measure of effectiveness can be associated with assigning a worker to a task, then the question may be asked as to how the workers should be assigned to tasks to maximize the total effectiveness. This is a maximum weighted matching problem. If costs are given in place of measures of effectiveness, the minimum cost assignment problem can be solved as a maximum weighted matching problem after replacing each cost by the difference between it and the maximum individual cost. This assumes all workers or all tasks must be assigned.

Both forms of assignment problem can be solved by a variety of algorithms. For example, a maximum matching on a bipartite graph can be found by modeling the problem as a network flow problem and finding a maximum flow on the model network. A more efficient algorithm is due to Hopcraft and Karp (1973). A well-known algorithm for solving the maximum weighted matching problem (for which the maximum matching problem can be considered a special case) on a bipartite graph is often referred to as the Hungarian method and was introduced by Kuhn (1955, 1956). Kuhn casts the procedure in terms of a primal-dual linear program. The algorithm can be implemented so as to produce an optimal matching in $O(m^2 n)$ steps, where n is the number of vertices and m is the number of edges in the graph. The details are given in Lawler (1976). Although this is an efficient algorithm, it may be necessary to find faster implementations for problems of large size or when the algorithm is used repeatedly as part of a more complex procedure. Various methods have been proposed including those due to Jonker and Volgenant (1986) and Wright (1990).

Job Scheduling

Another example of a problem which can be modeled as a matching problem arises from job scheduling (Coffman and Graham 1972). Suppose n jobs are to be processed and there are two machines available. All jobs require an equal amount of time to complete and can be processed on either machine. However there are precedence constraints which mean that some jobs must be completed before others are started. What is the shortest time required to process all n jobs?

This example can be modeled by constructing a graph G with n vertices representing the n jobs and where an edge joins two vertices if and only if they can be run simultaneously. An optimum schedule corresponds to one where the two machines are used simultaneously as often as possible. Therefore the problem becomes one of finding the maximum matching on G, from which the shortest time can be derived. In this case though, the graph G is no longer bipartite and so an algorithm for solving the maximum matching problem on a general graph is required.

The first efficient algorithm to find a maximum matching in a graph was developed by Edmonds (1965a). Most successful algorithms to find a maximum matching have been based on Edmonds' ideas. Gabow (1976) and Lawler (1976) show how to implement the algorithm in a time of $O(n^3)$. It is possible to modify the algorithm for more efficient performance on large problems. For example, Even and Kariv (1975) present an algorithm running in a time of $O(n^{5/2})$ and Micali and Vazirani (1980) describe an algorithm with running time of $O(mn^{1/2})$.

Arc Routing

There is a close connection between arc routing problems and matching. Suppose a person must deliver mail along all streets of a town. What route will traverse each street and return to the starting point in minimum total distance? This problem is known as the Chinese Postman Problem as it was first raised by the Chinese mathematician Meigu Guan (1962). It may be formulated as finding the minimum length tour on a non-directed graph G whose edges represent the streets in the town and whose vertices represent the junctions, where each edge must be included at least once. Edmonds and Johnson (1973) showed that this problem is equivalent to finding a minimum weighted matching on a graph whose vertices represent the set of odd nodes in G and whose edges represent the shortest distances in G between the odd nodes. Odd nodes are vertices where an odd number of edges meet. This minimum weighted matching problem can be solved efficiently by the algorithm introduced by Edmonds (1965b) for maximum weighted matching problems where the weights on each edge are the distances multiplied by minus one. The Chinese Postman Problem is therefore easier to solve than the Traveling Salesman Problem where a polynomially bounded algorithm has not yet been established.

For large problems, faster versions of the weighted matching algorithm have been developed by Galil, Micali and Gabow (1982) and Ball and Derigs (1983) which require $O(mn \log n)$ steps. A starting procedure which significantly reduces the computing time for the maximum matching problem is described by Derigs and Metz (1986) and involves solving the assignment problem in a related bipartite graph.

b-Matchings

Given an integer b_i for each vertex v_i of V, a b-matching of G is defined as a subset M of edges, such that at each vertex v_i, the number of edges of M incident on v_i is less than or equal to b_i. A matching is therefore a special case of a b-matching where $b_i = 1$ for all i. Efficient algorithms for b-matching problems are described in Gerards (1995), which also provides a good survey of matching in general.

Lower bounds for Vehicle Routing problems can be obtained by relaxing the subtour elimination and vehicle capacity constraints to give a perfect b-matching problem. Miller (1995) shows that this approach can be used in a branch-and-bound frame-work for this application.

See

- ▶ Assignment Problem
- ▶ Branch and Bound
- ▶ Chinese Postman Problem
- ▶ Dual Linear-Programming Problem
- ▶ Graph Theory
- ▶ Hungarian Method

- ▶ Integer and Combinatorial Optimization
- ▶ Maximum-Flow Network Problem
- ▶ Network
- ▶ Transportation Problem
- ▶ Traveling Salesman Problem
- ▶ Vehicle Routing

References

Ball, M. O., & Derigs, U. (1983). An analysis of alternate strategies for implementing matching algorithms. *Networks, 13*, 517–549.

Coffman, E. G., Jr., & Graham, R. L. (1972). Optimal scheduling for two processor systems. *Acta Informatica, 1*, 200–213.

Derigs, U., & Metz, A. (1986). On the use of optimal fractional matchings for solving the (integer) matching problem. *Computing, 36*, 263–270.

Edmonds, J. (1965a). Paths, trees, and flowers. *Canadian Journal of Mathematics, 17*, 449–467.

Edmonds, J. (1965b). Maximum matching and a polyhedron with (0,1) vertices. *Journal of Research National Bureau of Standards, Section B, 69B*, 125–130.

Edmonds, J., & Johnson, E. L. (1973). Matching, Euler tours and the Chinese postman. *Math Programming, 5*, 88–124.

Even, S., & Kariv, O. (1975). An O($n^{5/2}$) algorithm for maximum matching in general graphs. 16th *Annual symposium on foundations of computer science*, IEEE Computer Society Press, New York, pp. 100–112.

Gabow, H. N. (1976). An efficient implementation of Edmond's algorithm for maximum matching on graphs. *Journal of the Association for Computing Machinery, 23*, 221–234.

Galil, Z., Micali, S., & Gabow, H. (1982). Priority queues with variable priority and an O (*EV* log *V*) algorithm for finding a maximal weighted matching in general graphs. *23rd Annual symposium on foundations of computer science*, IEEE Computer Society Press, New York, pp. 255–261.

Gerards, A. M. H. (1995). Matching. In M. O. Ball, T. L. Magnanti, C. L. Monma, & G. L. Nemhauser (Eds.), *Network models, handbooks in operations research and management science* (Vol. 7, pp. 135–224). Amsterdam: Elsevier.

Gondran, M., & Minoux, M. (1984). *Graphs and algorithms*. Chichester: Wiley.

Guan, M. (1962). Graphic programming using odd and even points. *Chinese Mathematics, 1*, 273–277.

Hopcroft, J. E., & Karp, R. M. (1973). An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing, 2*, 225–231.

Jonker, R., & Volgenant, A. (1986). Improving the Hungarian assignment algorithm. *Operations Research Letters, 5*, 171–175.

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

Kuhn, H. W. (1956). Variants of the Hungarian method for assignment problems. *Naval Research Logistics Quarterly, 3*, 253–258.

Lawler, E. L. (1976). *Combinatorial optimization, networks and matroids*. New York: Holt, Rinehart and Winston.

Lovasz, L., & Plummer, M. D. (2009). *Matching theory*. Providence, RI: AMS Chelsea.

McHugh, J. A. (1990). *Algorithmic graph theory*. London: Prentice-Hall.

Micali, S., & Vazirani, V. V. (1980). An O($V^{1/2}$E) algorithm for finding maximum matching in general graphs. *21st Annual symposium on foundations of computer science*, IEEE Computer Society Press, New York, pp. 17–27.

Miller, D. L. (1995). A matching based exact algorithm for capacitated vehicle routing problems. *ORSA Journal of Computing, 7*, 1–9.

Wright, M. B. (1990). Speeding up the Hungarian algorithm. *Computers and Operations Research, 17*, 95–96.

Material Handling

Meir J. Rosenblatt
Washington University in St. Louis, St. Louis, MO, USA
Technion – Israel Institute of Technology, Haifa, Israel

Introduction

Material handling is concerned with moving raw materials, work-in-process, and finished goods into the plant, within the plant, and out of the plant to warehouses, distribution networks, or directly to the customers. The basic objective is to move the right combination of tools and materials (raw materials, parts and finished products) at the right time, to the right place, in the right form, and in the right orientation. And to do it with the minimum total cost.

It is estimated that 20% to 50% of the total operating expenses within manufacturing are attributed to material handling (Tompkins et al. 1996). Material handling activities may account for 80% to 95% of total overall time spent between receiving a customer order and shipping the requested items (Rosaler and Rice 1994). This indicates that improved efficiencies in material handling activities can lead to substantial reductions in product cost and production lead-time; better space and equipment utilization, improved working conditions and safety, improvements in customer service; and, eventually to higher profits and larger market share. Material handling adds to the product cost but contributes nothing to the value added of the products.

Design of material handling systems play a critical role in just-in-time (JIT) manufacturing. Under JIT, production is done in small lots so that production lead-times are reduced and inventory holding costs are minimized, requiring the frequent conveyance of material. Thus, successful implementation of JIT needs a fast and reliable material handling system as a prerequisite. A major, related development with great impact on the material handling process has been the extensive implementation of Total Quality Management (TQM) plans.

Production lot-sizing decisions have a direct impact on the assignment of storage space to different items (products) and consequently on the material handling costs. Therefore, lot sizing decisions must take into account not only setup and inventory carrying costs but also warehouse and material handling costs. In other words, production lot sizing, warehouse storage assignment, and material handling equipment decisions must be made simultaneously.

Also, in a flexible manufacturing environment, where batches of products may have several possible alternative routes, the choice of routing-mix can have a significant effect on shop throughput and work-in-process inventory. However, for such a system to be efficient, an appropriate material handling system needs to be designed. This design issue is especially important when expensive machines are being used. Major waste can be caused by a material handling system that is inappropriate and becomes a bottleneck.

Finally, it should be recognized that (computer-aided) facility layout determines the overall pattern of material flow within the plant and, therefore, has a significant impact on the material handling activities and costs. It is estimated that effective facilities planning and layout can reduce material handling costs by at least 10% to 30% (Tompkins et al. 1996). However, an effective layout requires an effective material handling system. Therefore, it is critical that these decisions are made simultaneously.

Material Handling Equipment

There are several ways of classifying material handling equipment: (1) type of control (operator controlled vs. automated); (2) where the equipment works (on the floor vs. suspended overhead); (3) travel path (fixed vs. flexible). The fixed vs. flexible travel path classification is used here as in Barger (1987). Flexible path equipment can be moved along any route and in general is operator-controlled. Trucks are a common mode of operations. There are several types of trucks depending on the type of handling that is needed, and the following are the most common:

Counterbalanced fork trucks — used both for storage at heights of 20 feet or more, as well as for fast transportation);

Narrow-aisle trucks — mainly used for storage applications;

Walkie Pallet trucks — mainly used for transportation over short hauls; and

Manual trucks — mainly used for short hauls and auxiliary services.

There are three important types of fixed-path equipment:

Conveyors — Conveyors are one of the largest families of material handling equipment. They can be classified based on the load-carrying surface involved: roller, belt, wheel, slat, carrier chain; or on the position of the conveyor: on-floor or overhead;

Automatic Guided Vehicles (**AGVs**) — these are electric vehicles with on-board sensors that enable them to automatically track along a guide path which can be an electrified guide wire or a strip of (reflective) paint or tape on the floor. The AGVs follow their designated path using their sensors to detect the electromagnetic field generated by the electric wire or to optically detect the path marked on the floor. AGVs can transport materials between any two points connected by a guide path — without human intervention. Most of today's AGVs are capable of loading and unloading materials automatically. Most applications of AGVs are for load transportation, however, they could also be used in flexible assembly operations to carry the product being assembled through the various stages of assembly. While AGVs have traditionally been fixed path vehicles, advances in technology permit them to make short deviations from their guide path. Such flexibility may considerably increase their usefulness; and

Hoists, Monorails, and Cranes — Hoists are a basic type of overhead lifting equipment and can be suspended from a rail, track, crane bridge or beam.

A hoist consists of a hook, a rope or chain used for lifting, and a container for the rope/chain. Monorails consist of individual wheeled trolleys that can move along an overhead track. The trolleys may be either powered or non-powered. Cranes have traditionally found wide application in overhead handling of materials, especially where the loads are heavy. Besides the overhead type, there are types of cranes that are wall or floor mounted, portable ones and so on. Types such as stacker cranes are useful in warehouse operations.

Interaction with Automated Storage and Retrieval Systems (AS/RS)

AS/RS consist of high-density storage spaces, computer-controlled handling and storage equipment (operated with minimal human assistance) and may be connected to the rest of the material handling system via some conveying devices such as conveyors and AGVs. Several types of AS/RS are available including: Unit Load, Miniload, Man-On-Board, Deep Lane and Carousels. The AS/RS systems help achieve very efficient placement and retrieval of materials, better inventory control, improved floor space utilization, and production scheduling efficiency. They also provide greater inventory accountability and reduce supervision requirements. Normally, stacker cranes that can move both horizontally and vertically at the same time are used for material handling. Typically, a crane operates in a single aisle, but can be moved between aisles (Rosenblatt et al. 1993). Items to be stored or retrieved are brought to/picked from the AS/RS by a conveyor or an AGV. Such integration can be used to automate material handling throughout the plant and warehouse. A great deal of research has been done on scheduling jobs and assigning storage space in the AS/RS (Hausman et al. 1976).

Issues in Material Handling System Design

Unit load concept — Traditional wisdom is that materials should be handled in the most efficient, maximum size using mechanical means to reduce the number of moves needed for a given amount of material. While reducing the number of trips required is a good objective, the drawback of this approach is that it tends to encourage the acceptance of large production lots, large material handling equipment, and large space requirements. Small unit loads allow for more responsive, less expensive, and less consuming material handling systems. Also, the trend toward continuous manufacturing flow processes and the strong drive for automation necessitate the use of smaller unit loads (Apple and Rickles 1987).

Container size and standardization — This is an issue related to the unit load concept. Container size has an obvious correlation with the size of unit load. Hence, not surprisingly, the current trend is to employ smaller containers. The benefits of smaller containers include compact and more efficient workstations, improved scheduling flexibility due to smaller transfer batch size, smaller staging areas, and lighter duty handling systems. Another consideration that strongly influences the optimal container size is the range of items served by one container. In warehouse operations, unless items vary widely in their physical characteristics, the cost of employing two or more container sizes is almost always higher than in the one-size case (Roll et al. 1989). Use of standard containers eliminates the need for container exchanges between operation sites.

Capacity of the system or number of pieces of equipment — The margins in the design of material handling system require a careful examination of the relative costs of acquiring and maintaining of work centers and handling equipment. In the design of the material handling system for an expensive job shop, enough excess capacity should be provided so that the handling system never becomes the bottleneck.

OR Models in Material Handling

Operations Research (OR) tools have been applied to model and study a variety of problems in the area of material handling. One example, dealing with the initial design phase of material handling, used a graph-theoretic modeling framework (Kouvelis and Lee 1990). Other examples include conveyor systems problems using queueing theory, and transfer lines where dynamic programming techniques were applied. Most of the theoretical work has focused on AGVs and AS/RS. The design and control of AGVs are

extremely complex tasks. The design decisions include determining the optimal number of AGVs (Maxwell and Muckstadt 1982), as well as determining the optimal flow paths (Kim and Tanchoco 1993). Factors to be considered in the design decisions include hardware considerations, impacts on facilities layout, material procurement policy, and production policy. Resulting problems tend to be intractable for any realistic scenario, and hence, heuristics and simulation are the most used techniques in addressing design issues. Control problems including dispatching and routing tasks require real time decisions, making it difficult to obtain optimal solutions. Researchers have attempted to solve simplified problems, for example, by examining static versions instead of dynamic systems (Han and McGinnis 1989), and using simple single-loop layouts (Egbelu 1993).

In the study of warehousing in general, and AS/RS in particular, many different measures of effectiveness of warehouse designs have been considered. The most common ones are throughput as measured by the number of orders handled per day, average travel time of a crane per single/dual command, and average waiting time per customer/order (Hausman et al. 1976). Researchers have considered either simulation or optimization models, usually of the nonlinear integer form, to solve these problems. Yet others have combined optimization and simulation techniques to obtain solutions that are both cost effective and operationally feasible (reasonable service time) (Rosenblatt et al. 1993).

Since factories are increasingly automated, numerical control of machine tools and flexible manufacturing systems is common. Material handling systems frequently involve the use of robots. In the absence of an effective material handling system, an automated factory would be reduced to a set of islands of automation. In the integrated and fiercely competitive global economy, material handling systems play a crucial role in the battle to cut costs and improve productivity and service levels.

See

- ▶ Facilities Layout
- ▶ Flexible Manufacturing Systems
- ▶ Integer and Combinatorial Optimization
- ▶ Inventory Modeling
- ▶ Job Shop Scheduling
- ▶ Just-in-Time (JIT) Manufacturing
- ▶ Simulation of Stochastic Discrete-Event Systems
- ▶ Total Quality Management

References

Apple, J. M., & Rickles, H. M. (1987). Material handling and storage. In J. A. White (Ed.), *Production handbook*. New York: Wiley.

Barger, B. F. (1987). Materials handling equipment. In J. A. White (Ed.), *Production handbook*. New York: Wiley.

Egbelu, P. J. (1993). Positioning of automated guided vehicles in a loop layout to improve response time. *European Journal of Operational Research, 71*, 32–44.

Han, M. H., & McGinnis, L. F. (1989). Control of material handling transporter in automated manufacturing. *IIE Transactions, 21*, 184–190.

Hausman, W. H., Schwarz, L. B., & Graves, S. C. (1976). Optimal assignment in automatic warehousing systems. *Management Science, 22*, 629–638.

Kim, K. H., & Tanchoco, J. M. A. (1993). Economical design of material flow paths. *International Journal of Production Research, 31*, 1387–1407.

Kouvelis, P., & Lee, H. L. (1990). The material handling systems design of integrated manufacturing system. *Annals of Operations Research, 26*, 379–396.

Maxwell, W. L., & Muckstadt, J. A. (1982). Design of automated guided vehicle systems. *IIE Transactions, 14*, 114–124.

Mulcahy, D. (1999). *Materials handling handbook*. New York: McGraw-Hill.

Roll, Y., Rosenblatt, M. J., & Kadosh, D. (1989). Determining the size of a warehouse container. *International Journal of Production Research, 27*, 1693–1704.

Rosaler, R. C., & Rice, J. O. (Eds.). (1994). *Standard handbook of plant engineering* (2nd ed.). New York: McGraw-Hill.

Rosenblatt, M. J., Roll, Y., & Zyser, V. (1993). A combined optimization and simulation approach to designing automated storage/retrieval systems. *IIE Transactions, 25*, 40–50.

Tompkins, J. A., White, J. A., Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M. A., & Trevino, J. (1996). *Facilities planning* (2nd ed.). New York: Wiley.

Material Requirements Planning

A material requirements planning (MRP) system is a collection of logical procedures for managing, at the most detailed level, inventories of component assemblies, subassemblies, parts and raw materials in a manufacturing environment. It is an information system and simulation tool that generates proposals for production schedules that managers can evaluate in terms of their feasibility and cost effectiveness.

See

- ▶ Hierarchical Production Planning
- ▶ Production Management

Mathematical Model

A mathematical description of (usually) a real-world problem. In operations research/management science, mathematical models take on varied forms (e.g., linear programming, queueing, Markovian systems), many of which can be applied across application areas. The basic OR/MS mathematical model can be described as the decision problem of finding the maximum (or minimum) of a measure of effectiveness (objective function) $E = F(X, Y)$, where X represents the set of possible solutions (alternative decisions) and Y the given conditions of the problem. Although a rather simple model in its concept, especially since it involves the optimization of a single objective, this mathematical decision model underlies most of the problems that have been successfully formulated and solved by OR/MS methodologies.

See

- ▶ Decision Problem
- ▶ Deterministic Model
- ▶ Stochastic Model

Mathematical Optimization Society

The Mathematical Optimization Society (MOS) is an international organization dedicated to the support and development of the application, computational methods, and theory of mathematical optimization. The society sponsors the triennial International Symposium on Mathematical Optimization and other meetings throughout the world. Until 2010, its name was the Mathematical Programming Society (MPS), which was founded in 1973.

Mathematical Programming

Mathematical programming is a major discipline in operations research/management science and, in general, is the study of how one optimizes the use and allocation of limited resources. Here the programming refers to the development of a plan or procedure for dealing with the problem. It is considered a branch of applied mathematics as it deals with the theoretical and computational aspects of finding the maximum (minimum) of a function $f(\boldsymbol{x})$ subject to a set of constraints of the form $g_i(\boldsymbol{x}) \leq b_i$. The linear-programming model is the prime example of such a problem.

Mathematical-Programming Problem

A constrained optimization problem usually stated as Minimize (Maximize) $f(\boldsymbol{x})$ subject to $g_i(\boldsymbol{x}) \leq 0$, $i = 1,\ldots, m$. Depending on the form of the objective function $f(\boldsymbol{x})$ and the constraints $g_i(\boldsymbol{x})$ the problem will have special properties and associated algorithms.

See

- ▶ Convex-Programming Problem
- ▶ Fractional Programming
- ▶ Geometric Programming
- ▶ Integer and Combinatorial Optimization
- ▶ Integer-Programming Problem
- ▶ Linear Programming
- ▶ Nonlinear Programming
- ▶ Quadratic Programming
- ▶ Separable-Programming Problem

Mathematical-Programming System (MPS)

An integrated set of computer programs that are designed to solve a range of mathematical-programming problems is often referred to as a mathematical-programming system (MPS). Such systems solve linear programs, usually by some form of the simplex method, and often

have the capability to handle integer-variable problems and other nonlinear problems such as quadratic-programming problems. To be effective, an MPS must have procedures for input data handling, matrix generation of the constraints, reliable optimization, user and automated control of the computation, sensitivity analysis of the solution, solution restart, and output reports.

Matrices and Matrix Algebra

Alan Tucker
The State University of New York at Stony Brook, Stony Brook, NY, USA

Introduction

A matrix is an $m \times n$ array of numbers, typically displayed as

$$\boldsymbol{A} = \begin{bmatrix} 4 & 3 & 8 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix},$$

where the entry in row i and column j is denoted as a_{ij}. Symbolically, $\boldsymbol{A} = (a_{ij})$, for $i = 1,\ldots, m$ and $j = 1,\ldots, n$. A vector is a one-dimensional array, either a row or a column. A column vector is an $m \times 1$ matrix, while a row vector is a $1 \times n$ matrix. For a matrix $\boldsymbol{A}$, its ith row vector is usually denoted by $\boldsymbol{a}_i'$ and its jth column by $\boldsymbol{a}_j$. Thus an $m \times n$ matrix can be decomposed into a set of m row n-vectors or a set of n column m-vectors. Matrices are a natural generalization of single numbers, or scalars. They arise directly or indirectly in most problems in operations research and management science.

The word matrix in Latin means womb. The term was introduced by J.J. Sylvester in 1848 to describe an array of numbers that could be used to generate (give birth to) a variety of determinants. A few years later, Cayley introduced matrix multiplication and the basic theory of matrix algebra quickly followed. A more general theory of linear algebra and linear transformations pushed matrices into the background until the 1940s and the advent of digital computers. During the 1940s, Alan Turing, father of computer science, introduced the LU decomposition and John von Neumann, father of the digital computer, working with Herman Goldstine, started the development of numerical matrix algebra and introduced the condition number of a matrix. Curiously, at the same time Cayley and Sylvester were developing matrix algebra, another Englishman, Charles Babbage, was building his analytical engine, the forerunner of digital computers, which are critical to the use of modern matrix models.

Basic Operations and Laws of Matrix Algebra

The language for manipulating matrices is matrix algebra. Matrix algebra is a multivariable extension of single-variable algebra. The basic building block for matrix algebra is the scalar product. The scalar product $\boldsymbol{a} \cdot \boldsymbol{b}$ of $\boldsymbol{a}$ and $\boldsymbol{b}$ is a single number (a scalar) equal to the sum of the products $a_i\ b_i$, i.e., $\boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i$, where both vectors have the same dimension n. Observe that the scalar product is a linear combination of the entries in vector $\boldsymbol{a}$ and also a linear combination of the entries of vector $\boldsymbol{b}$.

The product of an $m \times n$ matrix $\boldsymbol{A}$ and a column n-vector $\boldsymbol{b}$ is a column vector of scalar products $\boldsymbol{a}_i' \cdot \boldsymbol{b}$, of the rows $\boldsymbol{a}_i'$ of $\boldsymbol{A}$ with $\boldsymbol{b}$. For example, if

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

is a 2×3 matrix and

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

is a column 3-vector, then

$$\boldsymbol{Ab} = \begin{bmatrix} \boldsymbol{a_1'} \cdot \boldsymbol{b} \\ \boldsymbol{a_2'} \cdot \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} a_{11}b_1 + a_{12}b_2 + a_{13}b_3 \\ a_{21}b_1 + a_{22}b_2 + a_{23}b_3 \end{bmatrix},$$

so that $\boldsymbol{Ab}$ is a linear combination of $\boldsymbol{A}$. Moreover, for any scalar numbers r, q, any $m \times n$ matrix $\boldsymbol{A}$, and any column n-vectors $\boldsymbol{b}$, $\boldsymbol{c}$:

$$\boldsymbol{A}(r\boldsymbol{b} + q\boldsymbol{c}) = r\boldsymbol{A}\boldsymbol{b} + q\boldsymbol{A}\boldsymbol{c}.$$

The product of a row m-vector $\boldsymbol{c}$ and an $m \times n$ matrix $\boldsymbol{A}$ is a row vector of scalar products $\boldsymbol{c} \cdot \boldsymbol{a}_j$, of $\boldsymbol{c}$ with the columns $\boldsymbol{a}_j$ of $\boldsymbol{A}$. For example, if

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix},$$

is a 2×3 matrix and $\boldsymbol{c} = [c_1, c_2]$ is a row 2-vector, then

$$\begin{aligned} \mathbf{cA} &= [\mathbf{c} \cdot \mathbf{a_1}, \mathbf{c} \cdot \mathbf{a_2}, \mathbf{c} \cdot \mathbf{a_3}] \\ &= [a_{11}c_1 + a_{21}c_2, a_{12}c_1 + a_{22}c_2, a_{13}c_1 + a_{23}c_2]. \end{aligned}$$

If $\boldsymbol{A}$ is an $m \times r$ matrix and $\boldsymbol{B}$ is an $r \times n$ matrix, then the matrix product $\boldsymbol{AB}$ is an $m \times n$ matrix obtained by forming the scalar product of each row $\boldsymbol{a}_i'$ in $\boldsymbol{A}$ with each column $\boldsymbol{b}_j$ in $\boldsymbol{B}$. That is, the (i, j)th entry in $\boldsymbol{AB}$ is $\boldsymbol{a}_i' \cdot \boldsymbol{b}_j$. Column j of $\boldsymbol{AB}$ is the matrix–vector product $\boldsymbol{A}b_j$ and each column of $\boldsymbol{AB}$ is a linear combination of the columns of $\boldsymbol{A}$. Row i of $\boldsymbol{AB}$ is vector–matrix product $\boldsymbol{a}_i'\boldsymbol{B}$ and each row of $\boldsymbol{AB}$ is a linear combination of the rows of $\boldsymbol{B}$. The matrix–vector product $\boldsymbol{Ab}$ is a special case of the matrix-matrix product in which the second matrix has just one column; the analogous statement holds for the vector–matrix product $\boldsymbol{bA}$.

Matrix multiplication is not normally commutative. Otherwise it obeys all the standard laws of scalar multiplication.

Associative Law. *Matrix addition and multiplication are associative*:$(\boldsymbol{A} + \boldsymbol{B}) + \boldsymbol{C} = \boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C})$ *and* $(\boldsymbol{AB})\boldsymbol{C} = \boldsymbol{A}(\boldsymbol{BC})$.

Commutative Law. *Matrix addition is commutative:* $\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A}$. *Matrix multiplication is not commutative* (except in special cases): $\boldsymbol{AB} \neq \boldsymbol{BA}$.

Distributive Law. $\boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{AB} + \boldsymbol{AC}$ *and* $(\boldsymbol{B} + \boldsymbol{C})\boldsymbol{A} = \boldsymbol{BA} + \boldsymbol{CA}$.

Law of Scalar Factoring. $r(\boldsymbol{AB}) = (r\boldsymbol{A})\boldsymbol{B} = \boldsymbol{A}(r\boldsymbol{B})$.

For $n \times n$ matrices $\boldsymbol{A}$, there is an identity matrix $\boldsymbol{I}$ with ones on the main diagonal and zeros elsewhere, with the property that $\boldsymbol{AI} = \boldsymbol{IA} = \boldsymbol{A}$. Furthermore, the transpose of an $m \times n$ matrix $\boldsymbol{A}$, denoted by $\boldsymbol{A}^T$, is an $n \times m$ matrix such that the rows of $\boldsymbol{A}$ are the columns of $\boldsymbol{A}^T$.

If matrices are partitioned into submatrices in a regular fashion, say, a 4×4 matrix $\boldsymbol{A}$ is partitioned into four 2×2 submatrices,

$$\boldsymbol{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

and a 4×4 matrix $\boldsymbol{B}$ is similarly partitioned, then the matrix product $\boldsymbol{AB}$ can be computed in terms of the partitioned submatrices:

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A_{11}B_{11}} + \mathbf{A_{12}B_{21}} & \mathbf{A_{11}B_{12}} + \mathbf{A_{12}B_{22}} \\ \mathbf{A_{21}B_{11}} + \mathbf{A_{22}B_{21}} & \mathbf{A_{21}B_{12}} + \mathbf{A_{22}B_{22}} \end{bmatrix}.$$

Solving Systems of Linear Equations

Matrices are intimately tied to linear systems of equations. For example, the system of linear equations

$$\begin{aligned} 4x_1 + 2x_2 + 2x_3 &= 100 \\ 2x_1 + 5x_2 + 2x_3 &= 200 \\ 1x_1 + 3x_2 + 5x_3 &= 300 \end{aligned} \tag{1}$$

can be written as

$$\boldsymbol{Ax} = \boldsymbol{b}, \text{ where } \boldsymbol{A} = \begin{bmatrix} 4 & 2 & 2 \\ 2 & 5 & 2 \\ 1 & 3 & 5 \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}. \tag{2}$$

Essentially, the only way to solve an algebraic system with more than one variable is by solving a system of linear equations. For example, nonlinear systems must be recast as linear systems to be numerically solved. Since operations research and management science is concerned with complex problems involving large numbers of variables, matrix systems are pervasive in OR/MS.

Observe that the system of equations given by (1) can be approached from the row point of view as a set of simultaneous linear equations and solved by row operations using Gaussian elimination or Gauss-Jordan elimination. The result of elimination will be either no solution, a unique solution or an infinite number of solutions. In linear programming, one typically wants to find a vector $\boldsymbol{x}$ maximizing or

minimizing a linear objective function $\boldsymbol{c} \cdot \boldsymbol{x}$ subject to a system $\boldsymbol{Ax} = \boldsymbol{b}$ of linear constraints. The simplex method finds an optimal solution by a sequence of pivots on the augmented matrix $[\boldsymbol{Ab}]$. A pivot on non-zero entry (i, j) consists of a collection of row operations (multiplying a row by a scalar or subtracting a multiple of one row from another row) producing a transformed augmented matrix $[\boldsymbol{A}' \ \boldsymbol{b}']$ in which entry (i, j) equals 1 and all other entries in the jth column are 0. The pivot step can be accomplished by premultiplying $\boldsymbol{A}$ by a pivot matrix $\boldsymbol{P}$, which is an identity matrix with a modified ith column.

The system of equations given by (1) can also be approached from the column point of view as the following vector equation:

$$x_1 \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix} + x_3 \begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}. \qquad (3)$$

Writing the system as (3) raises questions such as which right-hand side vectors $\boldsymbol{b}$ are expressible as linear combinations of the columns of $\boldsymbol{A}$? The set of such $\boldsymbol{b}$ vectors is called the range of the matrix $\boldsymbol{A}$. For a square matrix, the system $\boldsymbol{Ax} = \boldsymbol{b}$ will have a unique solution if and only if no column vector of $\boldsymbol{A}$ can be written as a linear combination of other columns of $\boldsymbol{A}$, or equivalently, if and only if $\boldsymbol{x} = \boldsymbol{0}$ is the only solution to $\boldsymbol{Ax} = \boldsymbol{0}$, where $\boldsymbol{0}$ denotes a vector of all zeroes. When this condition holds, the columns are said to be linearly independent. When $\boldsymbol{Ax} = \boldsymbol{0}$ has non-zero solutions (whether $\boldsymbol{A}$ is square or not), the set of such nonzero solutions is called the kernel of $\boldsymbol{A}$. Kernels, ranges and linear independence are the building blocks of the theory of linear algebra. This theory plays an important role in the uses of matrices in OR/MS. For example, if $\boldsymbol{x}^*$ is a solution to $\boldsymbol{Ax} = \boldsymbol{b}$ and $\boldsymbol{x}^{\mathrm{o}}$ is in the kernel of $\boldsymbol{A}$ (i.e., $\boldsymbol{Ax}^{\mathrm{o}} = \boldsymbol{0}$), then $\boldsymbol{x}^* + \boldsymbol{x}^{\mathrm{o}}$ is also a solution of $\boldsymbol{Ax} + \boldsymbol{b}$, since $\boldsymbol{A}(\boldsymbol{x}^* + \boldsymbol{x}^{\mathrm{o}}) = \boldsymbol{Ax}^* + \boldsymbol{Ax}^{\mathrm{o}} = \boldsymbol{b} + \boldsymbol{0} = \boldsymbol{b}$, and one can show that all solutions to $\boldsymbol{Ax} = \boldsymbol{b}$ can be written in the form of a particular solution $\boldsymbol{x}^*$ plus some kernel vector $\boldsymbol{x}^{\mathrm{o}}$. In a linear program to maximize or minimize $\boldsymbol{c} \cdot \boldsymbol{x}$ subject to $\boldsymbol{Ax} = \boldsymbol{b}$, once one finds one solution $\boldsymbol{x}^*$ to $\boldsymbol{Ax} = \boldsymbol{b}$, improved solutions will be obtained by adding appropriate kernel vectors to $\boldsymbol{x}^*$.

Matrix Inverse

The inverse $\boldsymbol{A}^{-1}$ of a square matrix $\boldsymbol{A}$ has the property that $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{AA}^{-1} = \boldsymbol{I}$. The inverse can be used to solve $\boldsymbol{Ax} = \boldsymbol{b}$ as follows: $\boldsymbol{Ax} = \boldsymbol{b} \Rightarrow \boldsymbol{A}^{-1}(\boldsymbol{Ax}) = \boldsymbol{A}^{-1}\boldsymbol{b}$, but $\boldsymbol{A}^{-1}(\boldsymbol{Ax}) = (\boldsymbol{A}^{-1}\boldsymbol{A})\boldsymbol{x} = (\boldsymbol{I})\boldsymbol{x} = \boldsymbol{x}$. Thus $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$.

The square matrix $\boldsymbol{A}$ has an inverse if any of the following equivalent statements hold:

1. For all $\boldsymbol{b}$, $\boldsymbol{Ax} = \boldsymbol{b}$ has a unique solution;
2. The columns of $\boldsymbol{A}$ are linearly independent;
3. The rows of $\boldsymbol{A}$ are linearly independent.

The matrix $\boldsymbol{A}^{-1}$ is found by solving a system of equations as follows. The product $\boldsymbol{AA}^{-1} = \boldsymbol{I}$ implies that if $\boldsymbol{x}_j$ is the jth column of $\boldsymbol{A}^{-1}$ and $\boldsymbol{i}_j$ is the jth column of $\boldsymbol{I}$ ($\boldsymbol{i}_j$ has 1 in the jth entry and zeroes elsewhere), then $\boldsymbol{x}_j$ is the solution to the matrix system $\boldsymbol{Ax}_j = \boldsymbol{i}_j$. An impressive aspect of matrix algebra is that even when a matrix system $\boldsymbol{Ax} = \boldsymbol{b}$ has no solution, i.e., in (3) no linear combination of the columns of $\boldsymbol{A}$ equals $\boldsymbol{b}$, there is still a "solution" $\boldsymbol{y}$ in the sense of a linear combination $\boldsymbol{Ay}$ of the columns of $\boldsymbol{A}$ that is as close as possible to $\boldsymbol{b}$, i.e., the Euclidean distance in n-dimensional space between the vectors $\boldsymbol{Ay}$ and $\boldsymbol{b}$ is minimized. There is even an inverse-like matrix $\boldsymbol{A}^*$, called the pseudoinverse or generalized inverse, such that $\boldsymbol{y} = \boldsymbol{A}^*\boldsymbol{b}$. The matrix $\boldsymbol{A}^*$ is given by the matrix formula $\boldsymbol{A}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$, where $\boldsymbol{A}^T$ is the transpose of $\boldsymbol{A}$, obtained by interchanging rows and columns.

Eigenvalues and Eigenvectors

A standard form of a dynamic linear model is $\boldsymbol{p}' = \boldsymbol{Ap}$, where $\boldsymbol{A}$ is an $n \times n$ matrix and $\boldsymbol{p}$ is a n-column vector of populations or probabilities (in the case of probabilities, it is the convention to use row vectors: $\boldsymbol{p}' = \boldsymbol{pA}$). For some special vectors $\boldsymbol{e}$, called eigenvectors, $\boldsymbol{Ae} = \lambda\boldsymbol{e}$, where λ is a scalar called an eigenvalue. That is, premultiplying $\boldsymbol{e}$ by $\boldsymbol{A}$ has the effect of multiplying $\boldsymbol{e}$ by a scalar. It follows that $\boldsymbol{A}^n\boldsymbol{e} = \lambda^n\boldsymbol{e}$. This special situation is very valuable because it is obviously much easier to compute $\lambda^n\boldsymbol{e}$ than $\boldsymbol{A}^n\boldsymbol{e}$.

Most $n \times n$ matrices have n different (linearly independent) eigenvectors. If the vector $\boldsymbol{p}$ as a linear combination $\boldsymbol{p} = a\,\boldsymbol{e}_1 + b\,\boldsymbol{e}_2$ of, say, two eigenvectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, with associated eigenvalues λ_1, λ_2, then by the linearity of matrix–vector products, $\boldsymbol{Ap}$ and $\boldsymbol{A}^2\boldsymbol{p}$ can be calculated as

$$\boldsymbol{Ap} = \boldsymbol{A}(a\boldsymbol{e}_1 + b\boldsymbol{e}_2) = a\boldsymbol{Ae}_1 + b\boldsymbol{Ae}_2 = a\lambda_1\boldsymbol{e}_1 + b\lambda_2\boldsymbol{e}_2$$

and

$$\begin{aligned}\boldsymbol{A}^2\boldsymbol{p} &= \boldsymbol{A}^2(a\boldsymbol{e}_1 + b\boldsymbol{e}_2) = a\boldsymbol{A}^2\boldsymbol{e}_1 + b\boldsymbol{A}^2\boldsymbol{e}_2 \\ &= a\lambda_1^2\boldsymbol{e}_1 + b\lambda_2^2\boldsymbol{e}_2.\end{aligned}$$

More generally,

$$\begin{aligned}\boldsymbol{A}^k\boldsymbol{p} &= \boldsymbol{A}^k(a\boldsymbol{e}_1 + b\boldsymbol{e}_2) = a\boldsymbol{A}^\kappa\boldsymbol{e}_1 + b\boldsymbol{A}^k\boldsymbol{e}_2 \\ &= a\lambda_1^k\boldsymbol{e}_1 + b\lambda_2^k\boldsymbol{e}_2.\end{aligned}$$

If $|\lambda_1| > |\lambda_i|$, for $i \geq 2$, then for large k, λ^{k_1} will become much larger in absolute value than the other λ^{k_j}, and so $\boldsymbol{A^k p}$ approaches a multiple of the eigenvector associated with the eigenvalue of largest absolute value. For ergodic Markov chains, this largest eigenvalue is 1 and the Markov chain converges to a steady-state probability $\boldsymbol{p}^*$ such that $\boldsymbol{p}^* = \boldsymbol{p}^*\boldsymbol{A}$.

Matrix Norms

The norm $|\boldsymbol{v}|$ of a vector $\boldsymbol{v}$ is a scalar value that is nonnegative, satisfies scalar factoring, i.e., $|r\ \boldsymbol{v}| = r\ |\boldsymbol{v}|$, and the triangle inequality, i.e., $|\boldsymbol{u} + \boldsymbol{v}| \leq |\boldsymbol{u}| + |\boldsymbol{v}|$. There are three common norms used for vectors:

1. The Euclidean, or l_2, norm of $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]$ is defined as $|\boldsymbol{v}|_e = \sqrt{\boldsymbol{v}_1^2 + \boldsymbol{v}_2^2 + \cdots + \boldsymbol{v}_n^2}$.
2. The sum, or l_1, norm of $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]$ is defined $|\boldsymbol{v}|_s = |\boldsymbol{v}_1| + |\boldsymbol{v}_2| + \cdots + |\boldsymbol{v}_n|$.
3. The max, or l_∞, norm of $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]$ is $|\boldsymbol{v}|_m = \max\{|\boldsymbol{v}_1|, |\boldsymbol{v}_2|, \ldots, |\boldsymbol{v}_n|\}$.

The matrix norm $||\boldsymbol{A}||$ is the (smallest) bound such that $|\boldsymbol{Ax}| \leq ||\boldsymbol{A}||\ |\boldsymbol{x}|$, for all $\boldsymbol{x}$. Thus

$$||\boldsymbol{A}|| = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{|\boldsymbol{Ax}|}{|\boldsymbol{x}|}. \tag{4}$$

It follows that $|\boldsymbol{A}^k\boldsymbol{x}| \leq ||\boldsymbol{A}||^k\ |\boldsymbol{x}|$.

The Euclidean, sum, and max norms of the matrix are defined by using the Euclidean, sum, and max vector norms, respectively, in (4). When $\boldsymbol{A}$ is a square, symmetric matrix ($a_{ij} = a_{ji}$), the Euclidean norm $||\boldsymbol{A}||_e$ equals the absolute value of the largest eigenvalue of $\boldsymbol{A}$. When $\boldsymbol{A}$ is not symmetric, $||\boldsymbol{A}||_e$ equals the positive square root of the largest eigenvalue of $\boldsymbol{A}^T\boldsymbol{A}$. The sum and max norms of $\boldsymbol{A}$ are very simple to find and for this reason are often preferred over the Euclidean norm: $||\boldsymbol{A}||_s = \max_j\{|\boldsymbol{A}_j|_s\}$ and $||\boldsymbol{A}||_m = \max_i\{|\boldsymbol{A}_i'|_s\}$, where $\boldsymbol{A}_j$ denotes the jth column of $\boldsymbol{A}$ and $\boldsymbol{A}_i'$ denotes the ith row of $\boldsymbol{A}$. In words, the sum norm of $\boldsymbol{A}$ is the largest column sum (summing absolute values), and the max norm of $\boldsymbol{A}$ is the largest row sum.

Norms have many uses. For example, in a linear growth model $\boldsymbol{p}' = \boldsymbol{Ap}$, the kth iterate $\boldsymbol{p}^{(k)} = \boldsymbol{A}^k\boldsymbol{p}$ is bounded in norm by $|\boldsymbol{p}^{(k)}| \leq ||\boldsymbol{A}||^k\ |\boldsymbol{p}|$. One can show that if the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ is perturbed by adding a matrix $\boldsymbol{E}$ of errors to $\boldsymbol{A}$, and if $\boldsymbol{x}^*$ is the solution to the original system $\boldsymbol{Ax} = \boldsymbol{b}$ while $\boldsymbol{x}^* + \boldsymbol{e}$ is the solution to $(\boldsymbol{A} + \boldsymbol{E})\boldsymbol{x} = \boldsymbol{b}$, then the relative error $|\boldsymbol{e}|/|\boldsymbol{x}^* + \boldsymbol{e}|$ is bounded by a constant $c(\boldsymbol{A})$ times the relative error $||\boldsymbol{E}||/||\boldsymbol{A}||$, i.e., $|\boldsymbol{e}|/|\boldsymbol{x}^* + \boldsymbol{e}| \leq c(\boldsymbol{A})\ ||\boldsymbol{E}||/||\boldsymbol{A}||$. The constant $c(\boldsymbol{A}) = ||\boldsymbol{A}||\ ||\boldsymbol{A}^{-1}||$ and is called the condition number of $\boldsymbol{A}$.

A famous linear input–output model due to Leontief has the form $\boldsymbol{x} = \boldsymbol{Ax} + \boldsymbol{b}$. Here $\boldsymbol{x}$ is a vector of production of various industrial activities, $\boldsymbol{b}$ is a vector of consumer demands for these activities, and $\boldsymbol{A}$ is an inter-industry demand matrix in which entry a_{ij} tells how much of activity i is needed to produce one unit of activity j. Here, $\boldsymbol{Ax}$ is a vector of the input for the different activities needed to produce the output vector $\boldsymbol{x}$. The model $\boldsymbol{x} = \boldsymbol{Ax} + \boldsymbol{b}$ can be shown to have a solution if $||\boldsymbol{A}||_s < 1$, i.e., if the columns sums are all less than one. This condition has the natural economic interpretation that all activities must be profitable, i.e., the value of the inputs to produce a dollar's worth of any activity must be less than one dollar.

Algebraically, $\boldsymbol{x} = \boldsymbol{Ax} + \boldsymbol{b}$ is solved as follows:

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{Ax} + \boldsymbol{b} \rightarrow \boldsymbol{x} - \boldsymbol{AX} = \boldsymbol{b} \rightarrow (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} \\ &= b \rightarrow \boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{A})^{-\boldsymbol{I}}\boldsymbol{b}.\end{aligned}$$

When $||\boldsymbol{A}|| \leq 1$, the geometric series $\boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2 + \boldsymbol{A}^3 + \ldots$, converges to $(\boldsymbol{I} - \boldsymbol{A})^{-1}$, guaranteeing not only the existence of a solution to $\boldsymbol{x} = \boldsymbol{Ax} + \boldsymbol{b}$ but also a solution with nonnegative entries, since when $\boldsymbol{A}$ has nonnegative entries, then all the powers of $\boldsymbol{A}$ will have nonnegative entries implying that $(\boldsymbol{I} - \boldsymbol{A})^{-1}$ has nonnegative entries and hence so does $\boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b}$.

See

- ▶ Analytic Hierarchy Process
- ▶ Gaussian Elimination
- ▶ Gauss-Jordan Elimination Method
- ▶ Linear Programming
- ▶ LU Matrix Decomposition
- ▶ Markov Chains
- ▶ Simplex Method (Algorithm)

References

Lay, D. C. (1993). *Linear algebra and its applications*. Reading, MA: Addison Wesley.
Strang, G. (2009). *Introduction to linear algebra* (4th ed.). Wellesley, MA: Wellesley-Cambridge Press.

Matrix Game

- ▶ Game Theory

Matrix Geometric

When the solution to a stochastic model is (vector) proportional to a geometric distribution whose parameter is a matrix instead of the usual scalar.

See

- ▶ Matrix-Analytic Stochastic Models

Matrix-Analytic Stochastic Models

Marcel F. Neuts
The University of Arizona, Tucson, AZ, USA

Introduction

A rich class of models for queues, dams, inventories, and other stochastic processes has arisen out of matrix/vector generalizations of classical approaches. Three specific examples are presented: matrix-analytic solutions for M/G/1-type queueing problems, matrix-geometric solutions to GI/M/1-type queueing problems, and the Markov arrival process (MAP) generalization of the renewal point process.

Matrix-Analytic M/G/1-Type Queues

The unifying structure that underlies these models is an imbedded Markov renewal process whose transition probability matrix is of the form:

$$\tilde{\boldsymbol{Q}}(x) = \begin{bmatrix} \boldsymbol{B}_0(x) & \boldsymbol{B}_1(x) & \boldsymbol{B}_2(x) & \boldsymbol{B}_3(x) & \boldsymbol{B}_4(x) & \cdots \\ \boldsymbol{C}_0(x) & \boldsymbol{A}_1(x) & \boldsymbol{A}_2(x) & \boldsymbol{A}_3(x) & \boldsymbol{A}_4(x) & \cdots \\ \boldsymbol{0} & \boldsymbol{A}_0(x) & \boldsymbol{A}_1(x) & \boldsymbol{A}_2(x) & \boldsymbol{A}_3(x) & \cdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_0(x) & \boldsymbol{A}_1(x) & \boldsymbol{A}_2(x) & \cdots \\ . & . & . & . & . & \cdots \end{bmatrix}$$

where the elements are themselves matrices of probability mass functions. If the matrix

$$\boldsymbol{A} = \sum_{k=0}^{\infty} \boldsymbol{A}_k(\infty)$$

is irreducible and has the invariant probability vector $\boldsymbol{\pi}$, then the Markov renewal process is positive recurrent if and only if some natural moment conditions hold for the coefficient matrices and if

$$\rho = \boldsymbol{\pi} \sum_{k=1}^{\infty} k\boldsymbol{A}_k \boldsymbol{e} < 1 \quad \text{for } \boldsymbol{e} = (1, \ldots, 1)^T.$$

The quantity ρ is the generalized form of the traffic intensity for the elementary queueing models.

The state space is partitioned in levels $\boldsymbol{i}$, which are the sets of m states (i, j), $1 \leq j \leq m$. The crucial object in studying the behavior of the Markov renewal process away from the boundary states in the level $\boldsymbol{0}$ is the fundamental period, the first passage time from a state in $\boldsymbol{i}$ + 1 to a state in $\boldsymbol{i}$. The joint transform matrix $\tilde{\boldsymbol{G}}(z; s)$ of that first passage time, measured in the number of transitions to lower levels (completed services in queueing applications) and in real time, satisfies a nonlinear matrix equation of the form

$$\tilde{\boldsymbol{G}}(z;s) = z\sum_{k=0}^{\infty}\tilde{\boldsymbol{A}}(s)\left[\tilde{\boldsymbol{G}}(z;s)\right]^k.$$

This equation can be analyzed by methods of functional analysis, which leads to many explicit matrix formulas for moments. In terms of the matrix $\tilde{\boldsymbol{G}}(z; s)$, the boundary behavior of the Markov renewal process can be studied in an elementary manner. In queueing applications, the analysis leads to equations for the busy period and the busy cycle. Waiting-time distributions under the first-come, first-served discipline are obtained as first passage time distributions. Extensive generalizations of the Pollaczek-Khinchin integral equation for the classical M/G/1 queue have been obtained (see Neuts 1986b).

Applications of Markov renewal theory lead to a matrix formula for the steady-state probability vector $\boldsymbol{x}_0$ for the states in level $\mathbf{0}$ in the imbedded Markov chain. Next, a stable numerical recurrence due to Ramaswami (1988) permits computation of the steady-state probability vector $\boldsymbol{x}_i$ of the other levels i, $i \geq 1$.

There is an interesting duality between the random walks on the infinite strip of states (i, j), $-\infty < i < \infty$, $1 \leq j \leq m$, that underlie the Markov renewal processes of M/G/1 type and those of GI/M/1-type (which lead to matrix-geometric solutions). That duality is investigated in Asmussen and Ramaswami (1990) and Ramaswami (1990a).

The class of models with an imbedded Markov renewal process of M/G/1-type is very rich. It is useful in the analysis of many queueing models in continuous or discrete time that arise in communications engineering and other applications. In queueing theory, results for a variety of classical models have been extended to versatile input processes and to semi-Markovian services. These generalizations often lead to natural matrix generalizations of familiar formulas. For a discussion of what happens to the M/G/1 model when the input is changed to a Markovian arrival process (MAP — as more precisely presented in a subsequent section), see Lucantoni (1993). A treatment of cycle maxima for the MAP/G/1 queue is found in Asmussen and Perry (1992). A mathematically rigorous discussion of the complex analysis aspects of the models of M/G/1-type is found in Gail, Hantler, and Taylor (1994). Asymptotic results on the tail probabilities of queue length and waiting time distributions are discussed in Abate, Choudhury and Whitt (1994), and Falkenberg (1994).

Matrix-Geometric Solutions

Under ergodicity conditions, discrete-time Markov chains with transition probability matrix $\boldsymbol{P}$ of the form

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{B_0} & \boldsymbol{A_0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \boldsymbol{B_1} & \boldsymbol{A_1} & \boldsymbol{A_0} & \mathbf{0} & \mathbf{0} & \cdots \\ \boldsymbol{B_2} & \boldsymbol{A_2} & \boldsymbol{A_1} & \boldsymbol{A_0} & \mathbf{0} & \cdots \\ \boldsymbol{B_3} & \boldsymbol{A_3} & \boldsymbol{A_2} & \boldsymbol{A_1} & \boldsymbol{A_0} & \cdots \\ . & . & . & . & . & \cdots \end{bmatrix},$$

where the $\boldsymbol{A}_k$ are $m \times m$ nonnegative matrices summing to a stochastic matrix $\boldsymbol{A}$, and the $\mathbf{B}_k$ are nonnegative matrices such that the row sums of $\boldsymbol{P}$ are one, have an invariant probability vector $\boldsymbol{x}$ of a matrix-geometric form. That is, the unique probability vector $\boldsymbol{x}$ which satisfies $\boldsymbol{xP} = \boldsymbol{x}$, can be partitioned into row vectors $\boldsymbol{x}_i$, $i \geq 0$, which satisfy $\boldsymbol{x}_i = \boldsymbol{x}_0 \boldsymbol{R}^i$. The matrix $\boldsymbol{R}$ is the unique minimal solution to the equation

$$\boldsymbol{R} = \sum_{k=0}^{\infty}\boldsymbol{R}^k\boldsymbol{A}_k,$$

in the set of nonnegative matrices. All eigenvalues of $\boldsymbol{R}$ lie inside the unit disk. The matrix,

$$\boldsymbol{B}[\boldsymbol{R}] = \sum_{k=0}^{\infty}\boldsymbol{R}^k\boldsymbol{B}_k,$$

is an irreducible stochastic matrix. The vector $\boldsymbol{x}_0$ is determined as the unique solution to the equations

$$\begin{cases} \boldsymbol{x_0} = \boldsymbol{x_0}\boldsymbol{B}[\boldsymbol{R}] \\ 1 = \boldsymbol{x_0}(\mathbf{1} - \boldsymbol{R})^{-1}\boldsymbol{e} \end{cases}$$

where $\boldsymbol{e}$ is the column m-vector with all components equal to one. If the matrix $\boldsymbol{A}$ is irreducible and has the invariant probability vector $\boldsymbol{\pi}$, the Markov chain is positive recurrent if and only if

$$\boldsymbol{\pi} \sum_{k=1}^{\infty} k\boldsymbol{A}_k \boldsymbol{e} > 1.$$

Analogous forms of the matrix-geometric theorem hold for Markov chains with a more complicated behavior at the boundary states and for continuous Markov chains with a generator $\boldsymbol{Q}$ of the same structural form. A comprehensive treatment of the basic properties of such Markov chains and a variety of applications is given in Neuts (1981).

This result has found many applications in queueing theory. The subclass where the matrix $\boldsymbol{P}$ or the generator $\boldsymbol{Q}$ are block-tridiagonal are called quasi-birth and death (QBD) processes. These arise naturally as models for many problems in communications engineering and computer performance. The matrix-geometric form of the steady-state probability vector of a suitable imbedded Markov chain leads to explicit matrix formulas for other descriptors of queues, such as the steady-state distributions of waiting times, the distribution of the busy period and others.

In addition to its immediate applications, this construct has also generated much theoretical interest. Its generalization to the operator case was established in Tweedie (1982).

The largest eigenvalue η of the matrix $\boldsymbol{R}$ is important in various asymptotic results. Graphs of η as a function of a parameter of the queue are caudal characteristic curves. Some interesting behavioral features of the queues can be inferred from them (Neuts and Takahashi 1981; Neuts 1986a; Asmussen and Perry 1992). A matrix-exponential form for waiting-time distributions in queueing models was obtained in Sengupta (1989). Its relation to the matrix-geometric theorem was discussed in Ramaswami (1990b). A matrix-analytic treatment, covering all cases of reducibility, of the equation for $\boldsymbol{R}$, is given in Gail, Hantler and Taylor (1994).

The matrix $\boldsymbol{R}$, which is crucial to all applications of the theorem, must be computed by an iterative numerical solution of the nonlinear matrix equation

$$\boldsymbol{R} = \sum_{k=0}^{\infty} \boldsymbol{R}^k \boldsymbol{A}_k$$

A major survey and comparisons of various computational methods is found in Latouche (1993). For the block tri-diagonal case (*QBD*-processes), a particularly efficient algorithm was developed by Latouche and Ramaswami (1993).

Markovian Arrival Processes

The analytic tractability of models with Poisson or Bernoulli input is due to the lack-of-memory property, an extreme case of Markovian simplification. At the expense of performing matrix calculations, more versatile arrival processes can be used in a variety of models. The Markovian arrival process (MAP) is a point process model in which only one of a finite number of phases must be remembered to preserve many of the simplifying Markovian properties. It can be incorporated in many models which remain highly tractable by matrix-analytic methods. The MAP has found many applications in queueing and tele-traffic models to represent bursty arrival streams. Many queueing models for which traditionally Poisson arrivals were assumed are also amenable to analysis with MAP input.

It was first introduced in Neuts (1979), but a more appropriate notation was proposed by David Lucantoni in conjunction with the queueing model discussed in Lucantoni, Meier-Hellstern, and Neuts (1990). Although discrete-time versions of the MAP, as well as processes with group arrivals have been defined, their discussion requires only more elaborate notation than the single-arrival MAP in continuous time described here. Expositions of the basic properties and many examples of the MAP are found in Neuts (1989, 1992) and Lucantoni (1991).

Consider an irreducible infinitesimal generator $\boldsymbol{D}$ of dimension m with stationary probability vector θ. Write $\boldsymbol{D}$ as the sum of matrices $\boldsymbol{D}_0$ and $\boldsymbol{D}_1$, where $\boldsymbol{D}_1$ is nonnegative and $\boldsymbol{D}_0$ has nonnegative off-diagonal elements. The diagonal elements of $\boldsymbol{D}_0$ are strictly negative and $\boldsymbol{D}_0$ is nonsingular. Consider an m-state Markov renewal process $\{(J_n, X_n), n \geq 0\}$ in which each transition epoch has an associated arrival. Its transition probability matrix $\boldsymbol{F}(\cdot)$ is given by

$$\boldsymbol{F}(x) = \int_0^x \exp(\boldsymbol{D}_0 u) du \boldsymbol{D}_1, \quad \text{for } x \geq 0.$$

The most familiar MAPs are the *PH*-renewal process and the Markov-modulated Poisson Process (MMPP). These, respectively, have the pairs of parameter matrices $\boldsymbol{D}_0 = \boldsymbol{T}$, $\boldsymbol{D}_1 = \boldsymbol{T}^{\mathrm{o}}\alpha$, where $(\alpha, \boldsymbol{T})$ is the (irreducible) representation of a phase-type distribution and the column vector $\boldsymbol{T}^{\mathrm{o}} = -\boldsymbol{T}\boldsymbol{e}$, and $\boldsymbol{D}_0 = \boldsymbol{D} - \boldsymbol{\Lambda}$, $\boldsymbol{D}_1 = \boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix and $\boldsymbol{e}$ is the column m-vector with all components equal to one.

The matrix-analytic tractability of the MAP is a consequence of the matrix-exponential form of the transition probability matrix $\boldsymbol{F}(\cdot)$. It, in turn, follows from the Markov property of the underlying chain with generator $\boldsymbol{D}$, in which certain transitions are labeled as arrivals. A detailed description of that construction is found in Lucantoni (1991).

The initial conditions of the MAP are specified by the initial probability vector $\boldsymbol{\gamma}$ of the underlying Markov chain with generator $\boldsymbol{D}$. Taking $\boldsymbol{\gamma} = \boldsymbol{\theta}$, the stationary probability vector of $\boldsymbol{D}$, leads to the stationary version of the MAP. The rate γ^* of the stationary process is given by $\gamma^* = \boldsymbol{\theta}\boldsymbol{D}_1\boldsymbol{e}$. By choosing $\boldsymbol{\gamma} = (\gamma^*)^{-1}\boldsymbol{\theta}\boldsymbol{D}_1 = \theta_{\mathrm{arr}}$, the time origin is an arbitrary arrival epoch.

Computationally tractable matrix expressions are available for various moments of the MAP. These require little more than the computation of the matrix $\exp(\boldsymbol{D}t)$. A comprehensive discussion of these formulas is found in Neuts and Narayana (1992). For example, the Palm measure, $H(t) = E[N(t) \mid \text{arrival at } t = 0]$, the expected number of arrivals in an interval $(0, t]$ starting from an arbitrary arrival epoch, is given by

$$\boldsymbol{H}(t) = \boldsymbol{\lambda} * t + \theta_{\mathrm{arr}}[\boldsymbol{I} - \exp(\boldsymbol{D}t)](\boldsymbol{e}\boldsymbol{\theta} - \boldsymbol{D})^{-1}\boldsymbol{D}_1\boldsymbol{e}.$$

Other MAPs are constructed by considering selected transitions in Markov chains, by certain random time transformations or random thinning of a given MAP, and by superposition of independent MAPs. Statements and examples of these constructions are found in Neuts (1989, 1992). Specifically, the superposition of two (or more) independent MAPs is again an MAP. If two continuous-time MAPs have the parameter matrices $\{\boldsymbol{D}_k(i)\}$ for $i = 1, 2$, the parameter matrices for their superposition are given by $\boldsymbol{D}_k = \boldsymbol{D}_k(1) \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{D}_k(2) = \boldsymbol{D}_k(1) \otimes \boldsymbol{D}_k(2)$, for $k \geq 1, 2$, where $\otimes$ is the Kronecker pairwise matrix product.

See

- Markov Chains
- Markov Processes
- Matrices and Matrix Algebra
- Phase-Type Probability Distributions
- Queueing Theory

References

Abate, J., Choudhury, G. L., & Whitt, W. (1994). Asymptotics for steady-state tail probabilities in structured Markov queueing models. *Stochastic Models, 10*, 99–143.

Asmussen, S., & Perry, D. (1992). On cycle maxima, first passage problems and extreme value theory for queues. *Stochastic Models, 8*, 421–458.

Asmussen, S., & Ramaswami, V. (1990). Probabilistic interpretation of some duality results for the matrix paradigms in queueing theory. *Stochastic Models, 6*, 715–733.

Falkenberg, E. (1994). On the asymptotic behavior of the stationary distribution of Markov chains of M/G/1-type. *Stochastic Models, 10*, 75–97.

Gail, H. R., Hantler, S. L., & Taylor, B. A. (1994). Solutions of the basic matrix equations for the M/G/1 and G/M/1 Markov chains. *Stochastic Models, 10*, 1–43.

Latouche, G. (1985). An exponential semi-Markov process, with applications to queueing theory. *Stochastic Models, 1*, 137–169.

Latouche, G. (1993). Algorithms for infinite Markov chains with repeating columns. In C. D. Meyer & R. J. Plemmons (Eds.), *Linear algebra, Markov chains and queueing models* (pp. 231–265). New York: Springer-Verlag.

Latouche, G., & Ramaswami, V. (1993). A logarithmic reduction algorithm for quasi-birth-and-death processes. *Journal of Applied Probability, 30*, 650–674.

Lucantoni, D. M. (1991). New results on the single server queue with a batch Markovian arrival process. *Stochastic Models, 7*, 1–46.

Lucantoni, D. M. (1993). The BMAP/G/1 queue: A tutorial. In L. Donatiello & R. Nelson (Eds.), *Models and techniques for performance evaluation of computer and communications systems*. New York: Springer-Verlag.

Lucantoni, D. M., Meier-Hellstern, K. S., & Neuts, M. F. (1990). A single server queue with server vacations and a class of non-renewal arrival processes. *Advances in Applied Probability, 22*, 676–705.

Neuts, M. F. (1979). A versatile Markovian point process. *Journal of Applied Probability, 16*, 764–779.

Neuts, M. F. (1981). *Matrix-geometric solutions in stochastic models: An algorithmic approach*. Baltimore: The Johns Hopkins University Press. Reprinted by Dover Publications, 1994.

Neuts, M. F. (1986a). The caudal characteristic curve of queues. *Advances in Applied Probability, 18*, 221–254.

Neuts, M. F. (1986b). Generalizations of the Pollaczek-Khinchin integral equation in the theory of queues. *Advances in Applied Probability, 18*, 952–990.

Neuts, M. F. (1989). *Structured stochastic matrices of M/G/1 type and their applications*. New York: Marcel Dekker.
Neuts, M. F. (1992). Models based on the Markovian arrival process. *IEEE Transactions on Communications, Special Issue on Teletraffic, E75-B*, 1255–1265.
Neuts, M. F., & Narayana, S. (1992). The first two moment matrices of the counts for the Markovian arrival process. *Stochastic Models, 8*, 459–477.
Neuts, M. F., & Takahashi, Y. (1981). Asymptotic behavior of the stationary distributions in the GI/PH/c queue with heterogeneous servers. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte, 57*, 441–452.
Ramaswami, V. (1988). A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Stochastic Models, 4*, 183–188.
Ramaswami, V. (1990a). A duality theorem for the matrix paradigms in queueing theory. *Stochastic Models, 6*, 151–161.
Ramaswami, V. (1990b). From the matrix-geometric to the matrix-exponential. *Queueing Systems, 6*, 229–260.
Schellhaas, H. (1990). On Ramaswami's algorithm for the computation of the steady state vector in Markov chains of M/G/1-type. *Stochastic Models, 6*, 541–550.
Sengupta, B. (1989). Markov processes whose steady state distribution is matrix-exponential with an application to the GI/PH/1 queue. *Advances in Applied Probability, 21*, 159–180.
Tweedie, R. L. (1982). Operator-geometric stationary distributions for Markov chains with application to queueing models. *Advances in Applied Probability, 14*, 368–391.

MAUT

► Multi-Attribute Utility Theory

Max-Flow Min-Cut Theorem

For a maximum-flow network problem, it can be shown that the maximum flow through the network is equal to the minimum capacity of all the cuts that separate the source (origin) and the sink (destination) nodes, where the capacity of a cut is the sum of the capacities of the arcs in the cut.

See

► Maximum-Flow Network Problem

References

Ford, L. R., & Fulkerson, D. R. (1962). *Flows in networks*. Princeton, NJ: Princeton University Press.

Maximum

A function $f(x)$ is said to have a maximum on a set S when the least upper bound of $f(x)$ on S is assumed by $f(x)$ for some x^0 in S. Thus, $f(x^0) \geq f(x)$ for all x in S.

See

► Global Maximum (Minimum)

Maximum Feasible Solution

► Minimum (Maximum) Feasible Solution

Maximum Matching Problem

Involves finding in a graph a maximal set of links which meet each node at most once.

See

► Matching

Maximum-Flow Network Problem

For a directed, capacitated network with source and sink nodes, the problem is to find the maximum amount of goods (flow) that can be sent from the source to the sink.

See

► Network Optimization

References

Ford, L. R., & Fulkerson, D. R. (1962). *Flows in networks*. Princeton, NJ: Princeton University Press.

MCDM

▶ Multiple Criteria Decision Making

Measure of Effectiveness (MOE)

In a decision problem, the single objective that is to be optimized is called the measure of effectiveness (MOE). In a linear-programming problem, the MOE is the objective function. In a queueing-theory problem, frequently used MOEs include the expected steady-state queue length and the mean delay in queue.

See

▶ Mathematical Model

Measure-Valued Differentiation

▶ Weak Derivatives

Memetic Algorithms

Hybrid metaheuristic evolutionary algorithms (EAs) that combine population-based approaches such as genetic algorithms with local search improvement procedures or individual learning. Also known as Baldwinian EAs, Lamarckian EAs, cultural algorithms or genetic local search. Derived from the word "meme" that was coined by the British scientist Richard Dawkins in his book, *The Selfish Gene* (1976), to represent an evolutionary unit for cultural transmission analagous to a gene in biological evolution.

See

▶ Evolutionary Algorithms
▶ Genetic Algorithms
▶ Metaheuristics

References

Lim, M. H., Gustafson, S., Krasnogor, N., & Ong, Y. S. (2009). Editorial to the first issue. *Memetic Computing, 1*, 1–2.

Memoryless Property

For stochastic processes, lack-of-memory is synonymous with the Markov property. For a positive random variable T that models the duration of some phenomenon, lack-of-memory means that the time remaining is independent of the time already passed, i.e., $\Pr\{T > t + s \mid T > s\} = \Pr\{T > t\}$ for $s, t > 0$. The exponential distribution is the only continuous distribution with lack-of-memory, while the geometric distribution is the only discrete distribution with lack-of-memory.

See

▶ Exponential Arrivals
▶ Markov Processes
▶ Markov Property
▶ Poisson Arrivals
▶ Poisson Process
▶ Queueing Theory

Menu Planning

A diet problem in which the variables represent complete menu items such as appetizers and entrees, instead of individual foods. The problem is formulated as an integer-programming problem in which the integer binary variables represent the decision of selecting or not selecting a complete menu item.

Metagame Analysis

A problem structuring method that addresses situations of conflict and cooperation between independent actors. Based on game-theoretic concepts, it identifies explicit and implicit threats and promises between the actors to analyze the stability of alternative scenarios.

Metaheuristics

Kenneth Sörensen[1] and Fred W. Glover[2,3]
[1]University of Antwerp, Antwerp, Belgium
[2]OptTek Systems, Inc., Boulder, CO, USA
[3]University of Colorado Boulder, Boulder, CO, USA

Introduction

A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. The term is also used to refer to a problem-specific implementation of a heuristic optimization algorithm according to the guidelines expressed in such a framework. It combines the Greek prefix meta- (μετά, beyond in the sense of high-level) with heuristic (from the Greek heuriskein or ευρισχειν, to search) and was coined by Fred Glover in 1986.

Most metaheuristic frameworks have their origin in the 1980s (although in some cases roots can be traced to the mid 1960s and 1970s) and were proposed as an alternative to classic methods of optimization such as branch-and-bound and dynamic programming. As a means for solving difficult optimization problems, metaheuristics have enjoyed a steady rise in both use and popularity since the early 1980s. EU/ME – the metaheuristics community – is the EURO-sponsored working group on metaheuristics and the largest platform for communication among metaheuristics researchers worldwide. Conferences and journals devoted to metaheuristics, along with some software, are described at the end of this article.

Different metaheuristics can vary significantly in their underlying foundations. Some metaheuristics mimick a process seemingly unrelated to optimization, such as natural evolution, the cooling of a crystalline solid, or the behavior of animal swarms. Attending such variation is also a striking similarity among some methods that rely on a common foundation. For example, many methods have been proposed (and given different names) that differ in not much more than the metaphor underlying them, which is often a close variant of an original method's metaphor. In this manner, the metaheuristic framework of ant colony optimization, for instance, has spawned a steady stream of different social insect-based methods (using bees, flies, termites, etc.). Most metaheuristic frameworks advise the use of randomness, although some propose completely deterministic strategies. In optimization, metaheuristics are most often used to solve combinatorial optimization problems, although metaheuristics for other problems exist (see below).

One of the defining characteristics of a metaheuristic framework is that the resulting methods are — as the name suggests — always heuristic in nature. Exact methods for combinatorial optimization, such as branch-and-bound or dynamic programming, are subject to combinatorial explosion, i.e., for NP-hard problems the computing time required by such methods increases as an exponential function of the problem size. By relaxing the demand that the optimal solution should be found in a finite (but often prohibitively large) amount of time, optimization methods can be built that attempt to find a solution that is good enough in a computing time that is small enough. However, there are important aspects of metaheuristics that link them more closely with exact methods and that give rise to a number of hybrids that unite these two types of methods. These aspects will be discussed later.

The required quality of a solution and the maximum allowable computing time can, of course, vary greatly across optimization problems and situations. Metaheuristic frameworks, being defined in very general terms, can be adapted to fit the needs of most real-life optimization problems, from the smallest and simplest to the largest and most complex. Additionally, metaheuristics do not put any demands on the formulation of the optimization problem (like requiring constraints or objective functions to be expressed as linear functions of the decision variables), in contrast, for example, to methods for mixed-integer programming. As a result, several

commercial software vendors have implemented metaheuristics as their primary optimization engines, both in specialized software packages for production scheduling, vehicle routing (Sörensen et al. 2008) and nurse rostering (Burke et al. 2004), as well as in general-purpose simulation packages (April et al. 2003; Fu 2002; Glover et al. 1999).

However, the research field of metaheuristics is not without its critics, most of whom attack the perceived lack of a universally applicable design methodology for metaheuristics and the lack of scientific rigor in testing and comparing different implementations. The no free lunch theorems (Wolpert and Macready 1997) state that, when averaged over all problems, all optimization methods perform equally well. This suggests that no single metaheuristic can be considered as a panacea for combinatorial optimization problems, but rather that a lot of problem-specific tuning is necessary to achieve acceptable performance. Moreover, metaheuristics often have a large number of parameters and tuning them is a notoriously difficult process. Consequently, computational testing to compare different metaheuristics is very difficult and often done in an ad-hoc way, rather than by established scientific standards (Barr et al. 1995; Hooker 1995; Rardin and Uzsoy 2001) This has motivated work on self-adaptive metaheuristics that automatically tune their parameters (Cotta et al. 2008; Kramer 2008; Nonobe and Ibaraki 2001, 2002) From an alternative perspective, if a research study identifies parameter values that work well for a selected class of applications — as most studies attempt to do — then for practical purposes other researchers can consider these parameters as being constants (Of course, this doesn't prevent future experimentation from seeking better parameter values.)

Another criticism sometimes levied at metaheuristics concerns the occasional tendency to create overly intricate methods (Michalewicz and Fogel 2004) with many different operators, where the contribution of these operators to the final quality of the solutions found may be poorly understood (Watson et al. 2006). Despite some theoretical results, such as proofs for the convergence of some metaheuristics under special assumptions – usually infinite running time (Eiben et al. 1991; Mitra et al. 1985) – or attempts to explain why genetic algorithms work (such as the heavily criticized Wright et al. (2003) building block hypothesis (Holland 1975)), research papers that attempt to capture the fundamental reasons why metaheuristics work are still few and far between.

Despite these criticisms, the ability to obtain good solutions where other methods fail has made metaheuristics the method of choice for solving a majority of large real-life optimization problems, both in academic research and in practical applications.

Metaheuristic Concepts

Like all optimization methods, metaheuristics attempt to find the best (feasible) solution out of all possible solutions of an optimization problem. In order to do this, they examine various solutions and perform a series of operations on them in order to find different, better solutions.

Metaheuristics operate on a representation or encoding of a solution, an object that can be stored in computer memory and can be conveniently manipulated by the different operators employed by the metaheuristic. Since metaheuristics are most often used to solve combinatorial optimization problems, representations too are generally combinatorial in nature (i.e., they are able to represent only a finite number of solutions). Representations used in the metaheuristics literature are quite diverse (see, e.g., Talbi (2009) for an overview) and range from vector-representations (binary, integer) over permutations to more complex representations such as trees and other graphs. Many metaheuristic algorithms use a combination of different representation types, such as a vector of permutations. Contrary to exact algorithms, metaheuristics do not require the encoding of solutions to be a bijection, i.e., several solutions may share the same encoding and a single solution may be encoded in different ways. Often, an encoding is chosen on the grounds of being convenient to manipulate, although sometimes a time-consuming decoding procedure may be required to obtain the actual solution (such as the encoding used in Prins (2004)).

Although many different metaheuristics have been proposed, their mechanisms for obtaining good solutions primarily operate by manipulating solutions in three ways: by iteratively making small changes to a current solution (local search metaheuristics), by

constructing solutions from their constituting parts (constructive metaheuristics), and by iteratively combining solutions into new ones (population-based metaheuristics). Each of these manipulation mechanisms gives rise to a class of metaheuristic frameworks that are discussed separately below. It is important to note that these classes are not mutually exclusive, and many metaheuristic algorithms combine ideas from each of them. Also, in some instances the transitions from one solution to another are achieved by solving specially generated subproblems.

Local Search Metaheuristics

Local search metaheuristics find good solutions by iteratively making small changes, called moves, to a single solution, called the current solution. The set of solutions that can be obtained by applying a single move to a given solution is called the neighborhood of that solution. In each iteration, a solution from the neighborhood of the current solution is selected to become the new current solution. The sequence of moves defines a trajectory through the search space. Hence, local search metaheuristics are also known under the names of neighborhood search methods or trajectory methods.

For almost all problem representations, different move types can be defined, resulting in different neighborhood structures. The rule used to select the new current solution is called the move strategy or search strategy and determines the aggressiveness of the search. Metaheuristics that use the steepest descent or steepest ascent strategy select the best move from the neighborhood and are often called hill-climbers. Other move strategies include selecting the first move that improves upon the current solution (called the mildest ascent/descent or first-improving strategy), as well as selecting a random improving solution.

In general, the set of allowable moves is restricted to those that result in solutions that are both feasible and improve upon the current solution. Some metaheuristics allow infeasible moves in a strategy that is called strategic oscillation. In this strategy, the search is usually only allowed to temporarily remain in the infeasible region of the search space. A striking example of the utility of this strategy is shown in Glover and Hao (2010).

A solution whose neighborhood does not contain any better solutions is called a local optimum (as opposed to a global optimum, i.e., a best possible solution to the optimization problem). When the current solution is a local optimum, the metaheuristic utilizes a strategy to escape to other regions of the search space. It is this strategy that distinguishes metaheuristics from simple heuristics and from each other. The metaheuristic's name therefore usually refers to the strategy to prevent the search from becoming ensnared within regions whose local optima may be substantially inferior to a global optimum.

The simplest strategy to escape to potentially more fertile regions is to either start the search again from a new, usually random, solution or to make a relatively large change (called a perturbation) to the current solution. These strategies are respectively called multi-start local search (MLS) and iterated local search (ILS) (Lourenco et al. 2003).

A number of metaheuristics define different move types and change the move type used once a local optimum has been reached. The rationale for this strategy is that a local optimum relative to a specific move type can often be improved by performing local search with a different move type. The global optimum on the other hand is a local optimum with respect to every possible move type. Metaheuristics that use this strategy are commonly called variable neighborhood search (VNS) (Mladenović and Hansen 1997) algorithms, but using more than one neighborhood is far more common in the metaheuristics literature and not restricted to algorithms labeled VNS (Sörensen et al. 2008).

Using memory structures is a third commonly encountered way for metaheuristics to avoid remaining trapped in a local optimum and to guide the search in general so as to find good solutions more quickly. Algorithms that use memory structures are commonly grouped under the umbrella term tabu search (Glover 1989, 1990, 1996) algorithms (sometimes also called adaptive memory programming algorithms). Different memory structures may be used to explicitly remember different aspects about the trajectory through the search space that the algorithm has previously undertaken and different strategies may be devised to use this information to direct the search (Glover and Laguna 1993) to promising areas of the search space. Often-used memory structures include the tabu list (from which the name of the metaheuristic

framework derives) that records the last encountered solutions (or some attributes of them) and forbids these solutions (or attributes) from being visited again as long as they are on the list. Some variants record move attributes rather than solution attributes on the tabu list, for the purpose of preventing moves from being reversed. The tabu list is usually organized in a first-in, first-out (FIFO) fashion, i.e., the current solution replaces the oldest one on the list. The length of the tabu list is called the tabu tenure. Frequency memory records how often certain attributes have been encountered in solutions on the search trajectory, which allows the search to avoid visiting solutions that display the most often encountered attributes or to visit solutions with attributes seldom encountered. Such memory can also include an evaluative component that allows moves to be influenced by the quality of solutions previously encountered that contain various attributes or attribute combinations. Other memory structures such as an elite set of the best solutions encountered so far are also common. Another example of the use of memory can be found in a metaheuristic called guided local search (GLS) (Voudouris and Tsang 1999). GLS introduces an augmented objective function that includes a penalty factor for each potential element. When trapped in a local optimum, GLS increases the penalty factor for all elements of the current solution, making other elements (and therefore other moves) more attractive and allowing the search to escape from the local optimum. Similarly, some variants of tabu search use penalties to determine the tabu status of moves, though drawing more strongly on memory.

Contrary to most other local search metaheuristics, simulated annealing uses a random move strategy, emulating the annealing process of a crystalline solid. At each iteration, this strategy selects a random solution x' from the neighborhood of the current solution x and accepts x' as the new current solution with probability $e^{-[f(x')-f(x)]/T}$, where $f(\cdot)$ is the objective function value (to be maximized) of the solution and T is an endogenous parameter called the temperature. The acceptance probability increases as the increase in solution quality is higher (or the decrease is lower). The temperature is initially set to a high value, which leads to higher acceptance probabilities, and then gradually lowered as the search progresses (although it may be increases again at certain moments during the search). The function that describes the evolution of T throughout the different iterations is called the cooling schedule. Simulated annealing was first described in Kirkpatrick et al. (1983), based upon an algorithm by Metropolis et al. (1953).

Relaxation induced local search (RINS) (Danna et al. 2005) is a metaheuristic that constructs a promising neighborhood using information contained in the continuous relaxation of the mixed integer programming (MIP) model of the optimization problem. Because it does not need problem-specific information to construct its neighborhood, RINS can be more easily built into general-purpose MIP solvers [11] and is currently available in the latest versions of LINDO/LINGO and CPLEX. Contrary to other metaheuristics, RINS requires the problem to be formulated as a MIP which makes it less general than other metaheuristics.

Constructive Metaheuristics

Constructive metaheuristics constitute a separate class from local search metaheuristics in that they do not operate on complete solutions, but rather construct solutions from their constituent elements, starting from an empty set and adding one element during each iteration, an operation that is also called a move. After each iteration except the last, the algorithm therefore operates on a partial solution (e.g., a traveling salesperson tour that does not visit all cities), of which it may not be possible to determine the objective function value or the feasibility status. Constructive metaheuristics are often adaptations of greedy algorithms, i.e., algorithms that add the best possible element at each iteration, a myopic strategy that may result in suboptimal solutions.

GRASP, the acronym for greedy randomized adaptive search procedure (Feo and Resende 1995), uses randomization to overcome this drawback of purely greedy algorithms by adding some randomness to the selection process. Several variants of GRASP have been proposed, founded on the following basic idea. At each iteration, a restricted candidate list, which contains the α best elements that can be added, is updated. From the restricted candidate list, a random element is selected for addition to the partial solution, after which the list is updated to reflect the new situation. The parameter α determines the greediness of the search: if α equals 1, the search is completely greedy, whereas if α is equal to the

number of elements that can be added, the search is completely random. A particularly useful advance in GRASP algorithms has occurred by blending them with the path relinking strategy of tabu search. Notable examples of this approach include Commander et al. (2008); Nascimento et al. (2010); Resende et al. (2010).

Rather that using randomness to outperform a greedy heuristic, more strategic ways of performing constructive (or destructive) moves, once again making use of memory, are examined in Fleurent and Glover (1999); Glover et al. (2000). Another approach is embodied in a look-ahead strategy (Pearl 1984), which evaluates the elements that can be added by considering not only the next move, but several moves into the future. The pilot method (Duin and Voß 1999), for example, uses a (usually greedy) constructive heuristic to determine a pilot solution for each potential move, i.e., the value of a potential element is evaluated by determining the objective function value of the solution that results from applying the heuristic to generate a complete solution from the current partial solution with this element added. The idea of looking ahead has a long history, having been proposed in probing strategies for integer programming in (Lemke and Spielberg 1967).

Ant colony optimization (ACO) (Dorigo et al. 1996, 2006) is an umbrella term for a set of related constructive metaheuristics that build solutions by imitating the foraging behavior of ants. Perhaps because of the appeal of its imagery, this class of approaches has received and continues to receive widespread attention in the popular press (e.g., Anonymous 2010). Ant colony optimization introduces an external parameter for each potential element called the pheromone level (a pheromone is a chemical factor that triggers a social response in the same species), initially set to zero for all elements. The metaheuristic uses multiple parallel artificial agents (called ants) that each construct a solution by an iterative constructive process in which elements are selected based on a combination of the value of that element and its pheromone level. Once all ants have constructed a solution, the pheromone level of all elements is updated in a way that reflects the quality of the solution found by that ant (the elements of better solutions receive more pheromone). Each ant then constructs a new solution, but elements that were present in high-quality solutions will now receive a higher probability of being selected by the ants. Periodically, the pheromone level of all elements is reduced to reflect evaporation. The process of constructing solutions in the way described above is repeated, and the best solution found is reported at the end.

To improve the quality of the final solutions, most constructive metaheuristics include a local search phase after the construction phase.

Population-Based Metaheuristics

The main mechanism that allows population-based metaheuristics to find good solutions is the combination of existing solutions from a set, usually called the population. The fundamental reasoning behind this class of metaheuristics is that good solutions can be found by exchanging solution attributes between two or more (usually high-quality) solutions. The most important members of this class are called evolutionary algorithms because they mimic the principles of natural evolution. Following Michalewicz and Fogel (2004), here the term evolutionary algorithms is used as an umbrella term to encompass the wide range names given to metaheuristics based on evolution. This includes genetic algorithms (Goldberg et al. 1989; Holland 1975), genetic/evolutionary programming (Koza 1992), evolutionary computation (Fogel 2006), evolution strategies (Beyer and Schwefel 2002), and many others. The literature on evolutionary algorithms is larger than that on other metaheuristics, and this field has spawned several dedicated journals and conferences.

Typical of the field of evolutionary algorithms is that its researchers tend to adopt the vocabulary of the metaphor on which the algorithms are based. The descriptions of these algorithms therefore are stated in terms of chromosomes (instead of solutions), fitness (instead of objective function value), genotype (instead of encoding), etc. The driving force behind most evolutionary algorithms is selection and recombination. Selection ensures that predominantly high-quality solutions in the population are selected for recombination, usually by biasing the probability of each solution in the population to be selected towards its objective function value. Recombination utilizes specialized operators to combine the attributes of two or more solutions into new ones. The new solutions are then added to the population by a process called

reinsertion, possibly subject to feasibility or minimum quality demands, to replace (usually low-quality) solutions. In a large majority of cases, all operators (selection, recombination and reinsertion) make heavy use of randomness. A large number of evolutionary algorithms additionally include a mutation operator that (again, randomly) changes a solution after it has been recombined. Most evolutionary algorithms iterate the selection, recombination, mutation, and reinsertion phases a number of times, and report the best solution in the population.

Scatter search and path relinking (Glover et al. 2000, 2003) are both population-based metaheuristics for continuous (or mixed-integer) and combinatorial optimization respectively, proposed as a deterministic alternative for the highly stochastic evolutionary algorithms. Scatter search encodes solutions as real-valued vectors (or rounded real-valued vectors for integer values) and generates new solutions by considering convex or concave linear combinations of these vectors. Path relinking, on the other hand, generalizes this idea, making it applicable to combinatorial optimization problems, by generating paths between high-quality solutions. Paths consist of elementary moves such as the ones used in local search metaheuristics and essentially link one solution (called the initiating solution) to a second solution (called the guiding solution) in the solution space. Contrary to local search metaheuristics, path relinking uses a move strategy that chooses the move to execute based on the fact that this move will bring the solution closer to the guiding solution. In both scatter search and path relinking, the selection of both initiating and guiding solution from a population (called the reference set) is done in a deterministic way, as are the mechanisms for updating the reference set once new solutions have been generated.

Hybrid Metaheuristics

Metaheuristics that combine aspects or operators from different metaheuristics paradigms are called hybrid metaheurstics. The term has lost much of its discriminatory power, however, since such combinations of operators from different metaheuristic frameworks have become the norm rather than the exception. Indeed, there is a tendency in the metaheuristics research field to look at metaheuristics frameworks as providing general ideas or components to build optimization algorithms, rather than to consider them as recipes that should be closely followed (Michalewicz and Fogel 2004). In this spirit, many metaheuristics use specialized heuristics to efficiently solve subproblems produced by the metaheuristic method (e.g., Gendreau et al. 1994). Also, a large number of local search metaheuristics use a construction phase to find an initial solution (or a set of initial solutions) from which to start the neighborhood search. In fact the original description of the GRASP metaheuristic (Feo and Resende 1995) prescribes a local search phase to follow the greedy randomized construction phase.

Memetic algorithms (Moscato 1989) are the only class of hybrid metaheuristics that has been given a specific name. Metaheuristics belonging to this class combine recombination operators from the class of evolutionary algorithms with local search (meta) heuristics. Although the name is commonly used, many evolutionary algorithms either replace or complement their mutation operator with a local search phase and can also be considered memetic.

Metaheuristics and Exact Methods

A more recent development has been a special focus on combining ideas from different metaheuristics, usually local search, with exact methods such as branch-and-bound or branch-and-cut. Sometimes called matheuristics, the resulting method usually integrates existing exact procedures to solve subproblems and guide the higher-level heuristic (Dumitrescu and Stützle 2009; Raidl and Puchinger 2008). In a similar way, ideas and operators from constraint programming techniques are integrated with metaheuristics (Van Hentenryck and Michel 2009). The links between metaheuristics and exact methods provide examples of additional forms of combinations:

1. There exist exact methods for solving various special classes of optimization problems, such as linear programming and certain graph (or matroid) problems, that can be incorporated to solve subproblems produced by a metaheuristic method. Such subproblems can be generated by a decomposition strategy, a restriction strategy or a relaxation strategy (see Glover and Klingman (1988); Rego (2005)).
2. Exact methods for more complex problems can sometimes solve small instances of these problems effectively. A metaheuristic may operate by constructing collections of such small instances as

a strategy for generating structured moves that transition from a given solution to a new one (see, e.g., Glover (2005)).
3. An exact method can be run for a very long time to obtain optimal solutions (to at least some instances of a problem class), and these optimal solutions can be used in the learning approach called target analysis (Glover 1990; Glover and Laguna 1997) as a way to produce improved decision rules for both metaheuristics and exact methods.
4. Metaheuristics can be integrated with exact methods to improve the performance of the exact methods (Friden et al. 1989; Glover 1990; Puchinger et al. 2009).
5. By not demanding that the optimal solution be found, metaheuristics can, for example, employ a truncated optimization method in place of (or in conjunction with) generating subproblems that are structured to be easier to solve.

Metaheuristics for Different Optimization Problems

Continuous Optimization

Although metaheuristics are predominantly used for combinatorial optimization, many of them have been adapted for continuous optimization. Some metaheuristics are very naturally defined over continuous search spaces. Notable examples include scatter search (Glover et al. 2000), particle swarm optimization (Kennedy et al. 1995) and an evolutionary approach called differential evolution (Storn and Price 1997). Other, especially constructive and local search approaches, require a considerable adaptation from their original formulation. Nonetheless, algorithms for continuous optimization based on tabu search (Chelouah and Siarry 2000; Glover 1994), GRASP (Hirsch et al. 2007), variable neighborhood search (Liberti and Drazič 2005), and others, have been proposed.

Multi-objective Optimization

Many real-life problems have multiple objectives, for which the notion of optimality is generally replaced with the notion of dominance. A solution is said to dominate another solution if its quality is at least as good on every objective and better on atleast one. In multi-objective optimization, the set of non-dominated solutions is called the Pareto set and the projection of this set onto the objective function space is called the Pareto front or Pareto frontier. The aim of multi-objective metaheuristics, i.e., metaheuristics specifically designed to solve multi-objective optimization problems, is to approximate the Pareto front as closely as possible (Zitzler et al. 2004). The outcome of any multi-objective algorithm is therefore generally a set of mutually non-dominated solutions, the Pareto set approximation. To measure the quality of such an approximation, many different measures exist (Jaszkiewicz 2004). Although adaptations to the multi-objective paradigm of both tabu search and simulated annealing exist (Czyżak et al. 1998; Hansen 1997), most multi-objective metaheuristics are of the evolutionary type (Jones et al. 2002), a fact generally attributed to the observation that these algorithms naturally operate on a set of solutions. Evolutionary multi-objective metaheuristics include the vector evaluated genetic algorithm (VEGA) (Schaffer 1985), the non-dominated sorting algorithm (NDSA) (Srinivas and Deb 1994), the multi-objective genetic algorithm (MOGA) (Fonseca and Fleming 1993) and the improved strength pareto evolutionary algorithm (SPEA2) (Zitzler and Thiele 1999).

Stochastic Optimization

Stochastic combinatorial optimization problems include uncertain, stochastic or dynamic information in their parameters. Metaheuristics for such problems therefore need to take into account that the objective function value is a random variable and that the constraints are violated with some probability. Evaluating a solution's objective function value and/or its feasibility can be done either exactly (if a closed-form expression is available), by approximation or by Monte Carlo simulation. Metaheuristicsusing each of these possibilities have been proposed to solve different stochastic problems (Bianchi et al. 2009; Ribeiro and Resende 2010).

Research in Metaheuristics

Conferences

The premier conference on metaheuristics is MIC, the Metaheuristics International Conference.

Other conferences on metaheuristics include the yearly EU/ME meeting on a specific metaheuristics-related topic, organized by EU/ME in collaboration with a local research group, and the Hybrid Metaheuristics conference series that focuses on combinations of different metaheuristics and the integration of AI/OR techniques. The Learning and Intelligent Optimization conferences aim at exploring the boundaries between machine learning, artificial intelligent, mathematical programming and algorithms for optimization.

A large number of conferences focus exclusively on evolutionary algorithms, including Parallel Problem Solving From Nature (PPSN), the Genetic and Evolutionary Computation Conference (GECCO), EvoStar (a multi-conference comprising EuroGP, EvoCOP, EvoBIO,and EvoApplications), Evolutionary Multi-Criterion Optimization (EMO), and the IEEE Congress on Evolutionary Computation (CEC).

The Ants conference series is dedicated to research in swarm intelligence methods.

Journals

The field of metaheuristics has several dedicated journals: the well-established *Journal of Heuristics* and the newer *International Journal of Metaheuristics* and *International Journal of Applied Metaheuristic Computing* (IJAMC). However, a large majority of articles on metaheuristics are published in general OR/MS journals.

Several journals are devoted exclusively to evolutionary algorithms: *Evolutionary Computation*, *IEEE Transactions on Evolutionary Computation*, *Genetic Programming and Evolvable Machines*, and the *Journal of Artificial Evolution and Applications*.

The journal *Swarm Intelligence* is currently the main journal for advances in the swarm intelligence area.

Metaheuristics Software

Several vendors of commercial optimization software have included (albeit to a limited extent) metaheuristics in their packages. Frontline Systems' Risk Solver Platform and its derivatives, an extension of the Microsoft Excel Solver, include a hybrid evolutionary solver. Tomlab/GENO is a package for static or dynamic, single- or multi-objective optimization based on a real-coded genetic algorithm. Both LINDO/LINGO and CPLEX include the relaxation induced neighborhood search (RINS) metaheuristic.

Open source metaheuristics software frameworks have recently appeared in the COIN-OR library. These include METSlib, an object oriented metaheuristics optimization framework, and Open Tabu Search (OTS), a framework for constructing tabu search algorithms.

Besides these solvers for combinatorial optimization, most commercial (stochastic) simulation packages today include an optimization tool (Fu 2002). Autostat, included in AutoMod, and Simrunner, included in ProModel, both use evolutionary algorithms. A variety of companies in the simulation industry, as well as general management service and consulting firms like Rockwell Software, Dassault Systemes, Flextronics, Halliburton, HP, Planview and CACI, employ OptQuest, which uses tabu search and scatter search.

See

- ▶ Artificial Intelligence
- ▶ COIN-OR Computational Infrastructure for Operations Research
- ▶ Heuristics
- ▶ Integer and Combinatorial Optimization
- ▶ Multi-attribute Utility Theory
- ▶ Neural Networks
- ▶ Simulated Annealing
- ▶ Simulation Optimization
- ▶ Tabu Search

References

Anonymous (2010). Riders on a swarm. *The Economist*, 12 August 2010.

April, J., Glover, F., Kelly, J., & Laguna, M. (2003). Practical introduction to simulation optimization. In S. Chick, T. Sanchez, D. Ferrin, & D. Morrice, (Eds.), *Proceedings of the 2003 Winter Simulation Conference* 2003.

Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C., & Stewart, W. R. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics, 1*(1), 9–32.

Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies–a comprehensive introduction. *Natural Computing, 1*(1), 3–52.

Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing, 80*(2), 239–287.

Burke, E., De Causmaecker, P., Petrovic, S., Berghe, G. V., et al. (2004). Variable neighborhood search for nurse rostering problems. In M. G. C. Resende & A. Viana (Eds.), *Metaheuristics: Computer decision-making* (pp. 153–172). Boston: Kluwer Academic.

Chelouah, R., & Siarry, P. (2000). Tabu search applied to global optimization. *European Journal of Operational Research, 123*(2), 256–270.

Commander, C., Festa, P., Oliveira, C. A. S., Pardalos, P. M., Resende, M. G. C., & Tsitselis, M. (2008). Grasp with path-relinking for the cooperative communication problem on ad hoc networks. In D. A. Grundel, R. A. Murphey, P. M. Pardalos, & O. A. Prokopyev (Eds.), *Cooperative networks: Control and optimization* (pp. 187–207). Cheltenham: Edward Elgar Publishing.

Cotta, C., Sevaux, M., & Sörensen, K. (2008). *Adaptive and multilevel metaheuristics*. Berlin: Springer-Verlag.

Czyżak, P., et al. (1998). Pareto simulated annealing-a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis, 7*(1), 34–47.

Danna, E. (2004). Integrating local search techniques into mixed integer programming. 4OR. *A Quarterly Journal of Operations Research., 2*(4), 321–324.

Danna, E., Rothberg, E., & Le Pape, C. (2005). Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming, 102*(1), 71–90.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, 26*(1), 29–41.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine, 1*(4), 28–39.

Duin, C., & Voß, S. (1999). The pilot method: A strategy for heuristic repetition with application to the Steiner problem in graphs. *Networks, 34*(3), 181–191.

Dumitrescu, I., & Stützle, T. (2009). Usage of exact algorithms to enhance stochastic local search algorithms. In V. Maniezzo, T. Stützle, & S. Voß (Eds.), *Matheuristics: Hybridizing metaheuristics and mathematical programming, volume 10 of annals of information systems* (Vol. 10). New York: Springer-Verlag.

Eiben, A., Aarts, E., & Van Hee K. (1991). Global convergence of genetic algorithms: A Markov chain analysis. *Parallel problem solving from nature*, (pp. 3–12).

Feo, T. A., & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization, 6*(2), 109–133.

Fleurent, C., & Glover, F. (1999). Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. *INFORMS Journal on Computing, 11*(2), 198–204.

Fogel, D. B. (2006). *Evolutionary computation: Toward a new philosophy of machine intelligence*. New York: Wiley-IEEE Press.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the fifth international conference on genetic algorithms*, (pp. 416–423), Citeseer.

Friden, C., Hertz, A., & de Werra, D. (1989). *TABARIS: An exact algorithms based on tabu search for finding a maximum independent set in a graph*. Working paper, Swiss Federal Institute of Technology, Lausanne.

Fu, M. C. (2002). Optimization for simulation: Theory vs practice. *INFORMS Journal on Computing, 14*(3), 192–215.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science, 40*(10), 1276–1290.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research, 13*, 533–549.

Glover, F. (1989). Tabu search-part I. *ORSA Journal on Computing, 1*(3), 190–206.

Glover, F. (1990). Tabu search-part II. *ORSA Journal on Computing, 2*(1), 4–32.

Glover, F. (1994). Tabu search nonlinear and parametric optimization (with links to genetic algorithms). *Discrete Applied Mathematics, 49*, 231–255.

Glover, F. (1996). Tabu search and adaptive memory programming: Advances, applications and challenges. In R. Barr, R. Helgason, & J. L. Kennington (Eds.), *Interfaces in computer science and operations research*. Boston: Kluwer Academic.

Glover, F. (2005). Adaptive memory projection methods for integer programming. In C. Rego & B. Alidaee (Eds.), *Metaheuristic optimization via memory and evolution* (pp. 425–440). Boston: Kluwer Academic.

Glover, F., & Hao, J. K. (2010). The case for strategic oscillation. *Annals of Operations Research*. DOI:10.1007/s10479-009-0597-1.

Glover, F., & Klingman, D. (1988). Layering strategies for creating exploitable structure in linear and integer programs. *Mathematical Programming, 40*(1), 165–181.

Glover, F., & Laguna, M. (1993). Tabu search. In C. R. Reeves (Ed.), *Modern heuristic techniques for combinatorial problems* (pp. 70–141). New York: John Wiley & Sons.

Glover, F., & Laguna, M. (1997). *Tabu search*. Boston: Kluwer Academic.

Glover, F., Kelly, J., & Laguna, M. (1999). New advances wedding simulation and optimization. In D. Kelton, (ed.), *Proceedings of the 1999 Winter Simulation Conference*.

Glover, F., Laguna, M., & Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics, 39*(3), 653–684.

Glover, F., Laguna, M., & Marti, R. (2003). Scatter search and path relinking: Advances and applications. In *Handbook of metaheuristics*, (pp. 1–35).

Goldberg, D. E., et al. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading Menlo Park: Addison-Wesley.

Hansen, M. P. (1997). Tabu search for multiobjective optimization: MOTS. In *Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM'97), Cape Town, South Africa*, (pp. 574–586), Citeseer.

Hirsch, M. J., Meneses, C. N., Pardalos, P. M., & Resende, M. G. C. (2007). Global optimization by continuous GRASP. *Optimization Letters, 1*(2), 201–212.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Hooker, J. N. (1995). Testing heuristics: We have it all wrong. *Journal of Heuristics, 1*(1), 33–42.

Jaszkiewicz, A. (2004). Evaluation of multiobjective metaheuristics. In X. Gandibleux, M. Sevaux, K. Sörensen, & V. T'kindt (Eds.), *Metaheuristics for multiobjective optimization* (Lecture notes in economics and mathematical systems, Vol. 535, pp. 65–90). Berlin: Springer-Verlag.

Jones, D. F., Mirrazavi, S. K., & Tamiz, M. (2002). Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research, 137*(1), 1–9.

Kennedy, J., Eberhart, R. C. et al. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, *4*, 1942–1948.

Kirkpatrick, S., Gelatt, C. D., Jr., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*. Cambridge: The MIT press.

Kramer, O. (2008). *Self-adaptive heuristics for evolutionary computation*. Berlin: Springer-Verlag.

Lemke, C., & Spielberg, K. (1967). Direct search algorithms for zero–one and mixed integer programming. *Operations Research, 15*, 892–914.

Liberti, L., & Drazič, M. (2005) Variable neighbourhood search for the global optimization of constrained NLPs. In *Proceedings of GO*, (pp. 1–5).

Lourenco, H., Martin, O., & Stützle, T. (2003). Iterated local search.In *Handbook of metaheuristics*, (pp. 320–353).

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., et al. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics, 21*(6), 1087.

Michalewicz, Z., & Fogel, D. B. (2004). *How to solve it: Modern heuristics*. New York: Springer-Verlag.

Mitra, D., Romeo, F., & Sangiovanni-Vincentelli, A. (1985). Convergence and finite-time behavior of simulated annealing. In *1985 24th IEEE Conference on Decision and Control*, Vol. 24.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research, 24*(11), 1097–1100.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report*, 826.

Nascimento, M. C. V., Resende, M. G. C., & Toledo, F. M. B. (2010). Grasp heuristic with path-relinking for the multi-plant capacitated lot sizing problem. *European Journal of Operational Research, 200*, 747–754.

Nonobe, K., & Ibaraki, T. (2001). An improved tabu search method for the weighted constraint satisfaction problem. *INFOR, 39*(2), 131–151.

Nonobe, K., & Ibaraki, T. (2002). Formulation and tabu search algorithm for the resource constrained project scheduling problem. In C. C. Ribeiro & P. Hansen (Eds.), *Essays and surveys in metaheuristics* (pp. 557–588). Boston: Kluwer Academic.

Pearl, J. (1984). *Heuristics–intelligent search strategies for computer problem solving*. Reading, MA: Addison-Wesley.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research, 31*(12), 1985–2002.

Puchinger, J., Raidl, G. R., & Pirkwieser, S. (2009). Metaboosting: Enhancing integer programming techniques by metaheuristics. In V. Maniezzo, T. Stützle, & S. Voß (Eds.), *Matheuristics: Hybridizing metaheuristics and mathematical programming* (Annals of information systems, Vol. 10). New York: Springer-Verlag.

Raidl, G. R., & Puchinger, J. (2008). Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In C. Blum, M. J. Blesa Aguilera, A. Roli, & M. Sampels (Eds.), *Hybrid metaheuristics: An emerging approach to optimization* (Studies in computational intelligence, Vol. 114). Berlin: Springer-Verlag.

Rardin, R. L., & Uzsoy, R. (2001). Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics, 7*(3), 261–304.

Rego, C. (2005). RAMP: A new metaheuristic framework for combinatorial optimization. In C. Rego & B. Alidaee (Eds.), *Metaheuristic optimization via memory and evolution: Tabu search and scatter search* (pp. 441–460). Boston: Kluwer Academic.

Resende, M. G. C., Martí, R., Gallego, M., & Duarte, A. (2010). Grasp and path relinking for the max-min diversity problem. *Computers and Operations Research, 37*, 498–508.

Ribeiro, C. C., & Resende, M. G. C. (2010). Path-relinking intensification methods for stochastic local search algorithms. Research technical report, AT&T Labs.

Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, (pp. 93–100). L. Erlbaum Associates.

Sörensen, K., Sevaux, M., & Schittekat, P. (2008). *"Multiple neighbourhood search" in commercial VRP packages: Evolving towards self-adaptive methods, volume 136 of lecture notes in economics and mathematical systems, chapter adaptive, self-adaptive and multi-level metaheuristics* (pp. 239–253). London: Springer-Verlag.

Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation, 2*(3), 221–248.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359.

Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. Hoboken, NJ: Wiley.

Van Hentenryck, P., & Michel, L. (2009). *Constraint-based local search*. Cambridge: The MIT Press.

Voudouris, C., & Tsang, E. (1999). Guided local search and its application to the traveling salesman problem. *European Journal of Operational Research, 113*(2), 469–499.

Watson, J. P., Howe, A. E., & Darrell Whitley, L. (2006). Deconstructing nowicki and Smutnicki's i-TSAB tabu search algorithm for the job-shop scheduling problem. *Computers and Operations Research, 33*(9), 2623–2644.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(67).
Wright, A., Vose, M., & Rowe, J. (2003). Implicit parallelism. In *Genetic and evolutionary computation–GECCO 2003*, (pp. 211–211). Springer.
Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the srength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257.
Zitzler, E., Laumanns, M., & Bleuler, S. (2004). A tutorial on evolutionary multiobjective optimization. In X. Gandibleux, M. Sevaux, K. Sörensen, & V. T'kindt (Eds.), *Metaheuristics for multiobjective optimization* (Lecture notes in economics and mathematical systems, Vol. 535, pp. 3–38). Berlin: Springer-Verlag.

Metamodeling

For simulation models, the objective is to provide an explicit input-output relationship through a fitted mathematical function, e.g., using statistical regression, splines, neural networks, or kriging. Differs from the use of the term in computer science.

See

► Response Surface Methodology
► Simulation Metamodeling

Method of Stages

An analysis method that extends the birth-and-death-type analysis to queueing systems with Erlangian service or interarrival times. Since an Erlang random variable can be represented as the sum of independent and identically distributed exponential random variables, the method of stages increases the state space to coincide with the underlying exponential random variables and the resulting system of equations is generally solved using generating functions.

See

► Queueing Theory

Military Operations Other Than War

Dean S. Hartley III
Oak Ridge National Laboratory, Oak Ridge, TN, USA

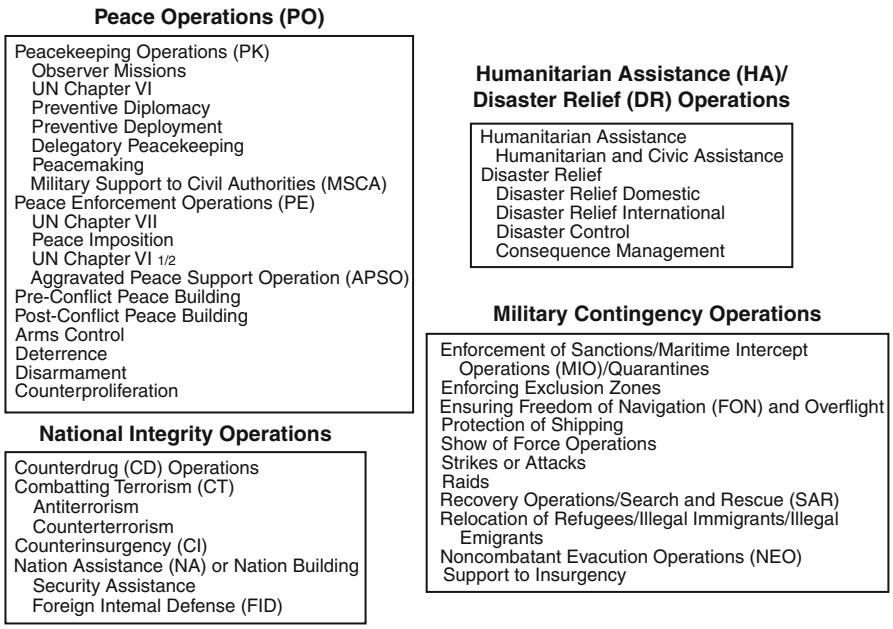
Introduction

Operations Other Than War (OOTW) suffer from an identity crisis. Sometimes called Military Operations Other Than War (MOOTW), sometimes known as Low Intensity Conflict (LIC), sometimes called Stability Support Operations (SSO), and sometimes designated as Small Scale Contingencies (SSC), these operations have caused both theoretical and practical problems for the military.

- These operations range in size from airlifting several fire trucks from Tennessee to Florida to fighting the 1998 Summer fires to the Bosnia Peacekeeping operation involving tens of thousands of U.S. military personnel and tens of thousands of other nations' military personnel, hardly a small-scale contingency.
- They include operations to provide stability to foreign countries, such as Haiti; however, they also include support to insurgencies, a "stability support operation" only in the negative.
- They include Non-combatant Evacuation Operations (NEOs) in which armed force may be needed to support the evacuation; they include operations such as Somalia that result in a number of U.S. military deaths in combat, low intensity conflict providing cold comfort to families of the dead; and they include operations such as fire-fighting that can be defined as conflict only by stretching the definition.

These operations cannot even be distinguished from other operations by time frame or geographic impact:

- Their time span ranges from the one-day cruise missile strike against Iraq to the 17-year peacekeeping operation in the Sinai (or the 45-year peacekeeping operation in Korea).
- Their geographic impact ranges from the purely local issues of disaster relief in Hawaii for Typhoon Iniki to the global geopolitical concerns stirred by peacekeeping in Bosnia.



Military Operations Other Than War, Fig. 1 Types of OOTW

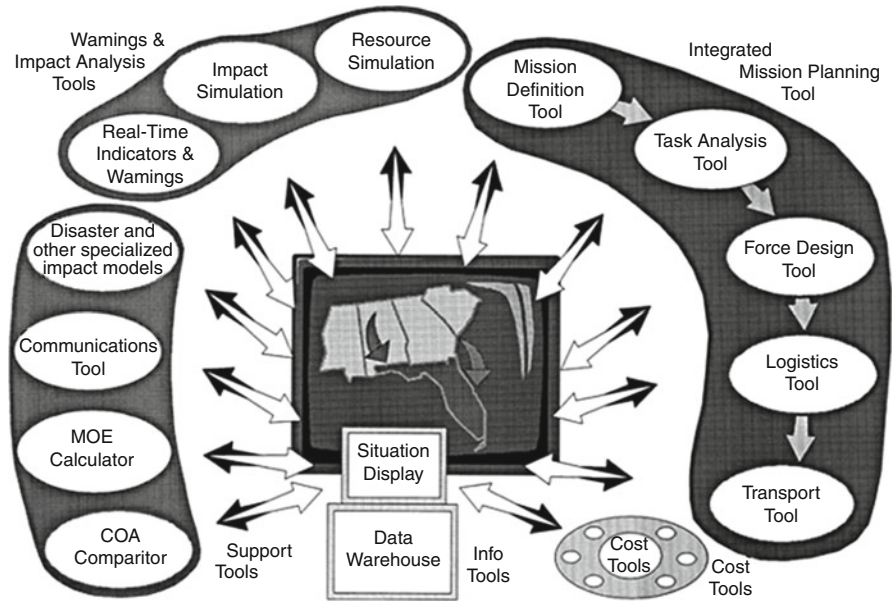
Clumsy as the OOTW designation may be, it is accurate: operations (as opposed to training activities) that are not war are included and operations that are part of a war are not included. Strictly speaking, the people who are using the designation are Department of Defense people and the operations so designated are military operations, leading to a preference for the term MOOTW; however, henceforth the shorter term OOTW will be used, because most of these operations are not led by the military, but by the State Department or some other agency. Figure 1 organizes OOTWs into categories.

The discussion of OOTWs suffering from an identity crisis is more than just a pleasant exercise in rhetoric. The underlying diversity of activities subsumed in the category creates a problem in defining standing operating procedures (SOPs) for dealing with them. The subordinate role of the military creates problems in planning for and executing them. The variability of participation of other federal agencies, other governments, the United Nations, non-governmental organizations (NGOs), and private volunteer organizations (PVOs) exacerbates the problem. Their ad hoc nature means that they are not included in the military's budget; the accounting systems are not designed to capture the costs; and recovering the resulting costs is problematic. These problems would be less troublesome if OOTWs were infrequent; however, since 1990 they have been undertaken at a rate of 20–35 per year! Over the past several years, there has been an increasing recognition of the need for analysis tools to support military planning and execution of OOTWs. Analysis tools to support decision making for large-scale military combat operations (such as major regional contingencies) are relatively mature (Battle Modeling). In contrast, OOTW analysis tools are embryonic or non-existent. The increasing U.S. military involvement in OOTWs during the post-Cold-War era has led to the need to develop OOTW analysis tools.

Questions

The analytical requirements are characterized by the questions that must be answered. The questions fall into five groups:

- Those that are non-mission-related (e.g., what force structure, equipment and plans are needed for the future?).
- Those that support a decision to engage (or not to engage) in a mission (e.g., what impacts will an OOTW have on other operations and how much will it cost?).



Military Operations Other Than War, Fig. 2 OOTW Analysis tool category

- Those needed to plan a mission (e.g., what is the right force structure and what transport support will we provide to reporters, NGO/PVOs, etc.?).
- Those that occur during a mission (e.g., which course of action will most quickly accomplish the mission?).
- Those related to the termination of a mission (e.g., how do we define success and what are its Measures of Effectiveness (MOEs)?).

The question groups are identical to the question groups for combat analysis. Most of the individual questions are also identical. In general, the analysis techniques required to answer the questions are the same. The problem lies in the application: standard applications make assumptions that are valid for combat analysis and invalid for OOTW analysis.

The question of force structure for a mission provides a simple example of the difference between combat analysis and OOTW analysis. For a combat mission, combat troops and equipment are determined first and the balance of the force structure is composed of the troops and materiel required to support them. Analysis procedures and tools are structured to support this situation. For an OOTW, however, the primary forces may be engineers for disaster reconstruction, medical personnel for disease control, some other support function, or combat troops, depending on the particulars of the mission. The implied force structure consists of the troops and materiel to support these forces and may (or may not) include combat troops to protect them. Not only are combat analysis procedures and tools set up backwards for OOTW analysis, but also OOTW analysis involves multiple possible permutations, requiring significantly more flexibility.

Nature of The Analysis Tools

Generally, the desirable tools are decision support tools, are simple (e.g., menu driven, point and click), are deployable, are joint (multi-service), are rigorous, use non-parochial data, have available data, and are capable of rapid turnaround. Analysis tools range from complex simulations of political, economic, sociological, military interactions to database tools, to spreadsheets, to checklists, with the emphasis on small tools. Figure 2 shows the categories of OOTW analysis tools.

Warnings and Impact Analysis Tools

These tools are among the most difficult (scientifically) to create, but are essential to the analysis of OOTWs. Three tools are included in this group.

- The real-time indicators and warnings tool serves to filter and interpret world news in the light of

possible future OOTWs: there are several attempts being made to create such a tool, such as the Protocol for the Assessment of Nonviolent Direct Action (PANDA) (Bond and Vogele 1995).

- The impact simulation models the significant relationships included in and surrounding an OOTW to permit prediction of the results of actions, whether human or environmental: the commercial computer game, Sim City™, is an example of an impact simulation. Unfortunately, the nature of social interactions is a matter for debate and consequently the proper mathematical expressions of these interactions and the best methods for modeling them are undecided. While at least two candidate simulations exist, Spectrum (National Simulation Center 1996) and the Deployable Exercise Support system/Civil Affairs Module (DEXES/CAM) (Woodcock 1996), these are regarded with some misgivings by working analysts, apparently because of lack of transparency or because they are used for training. The Situational Influence Assessment Module (SIAM) uses another technique to address social interactions. It is an influence diagram-based model, not a simulation model, but may be useful in this category.
- The resource simulation models the changes in resource consumption and sequestration over the course of an OOTW: this need may well be satisfied by the Joint Warfare Simulation (JWARS).

Integrated Mission Planning Tool

The five separate tools that comprise this group should ultimately be seamlessly integrated, although the initial integration may be loose. Each tool feeds its successor, while permitting reentry for iterative planning. These tools are relatively simple (scientifically); however, to be useful in an OOTW context, they require careful definition with respect to applicability to joint, coalition (multi-country) and non-military component analysis. The tools are a mission definition tool, a task analysis tool, a force design tool, a logistics tool, and a transportation tool.

- The mission definition tool should provide a reality check to ensure that the complete implications of the mission are fully understood. The Conceptual Model of Peace Operations (CMPO), a peace operations influence diagram-based checklist, is an example (Davis 1996).
- The object of the task analysis tool is to support an accurate and complete analysis of the mission tasks. The tool needed is a decision support tool that connects missions to strategies to tasks, both explicit and implied, in the OOTW domain. It should identify both those tasks that are central to the mission and any contingent tasks that might be implied by reasonable shifts in mission definition. It should also support replanning as the situation changes. Lidy (1998) has produced the data to support such a tool.
- The object of the force design tool is to support the designation of U.S. forces required for an operation in an OOTW context. The tool needed is a decision support tool that connects the tasks to generic resources and connects generic resources to actual available resources, including U.S. military, U.S. non-military, foreign government, NGO/PVO, and contractor resources. Data requirements include task capability for all resources (or the facility for user input of unique resources) and availability data (based on reserve commitments, etc.). It should provide for restrictions on choices based on cultural issues. Processing should include selection of military resources and substitution of other resources. The tool should also support replanning as the situation changes.
- The object of the logistics analysis tool is to support the logistics analysis of the mission in an OOTW context. The tool needed is a decision support tool that derives the logistics requirements from the total force structure. It should allow for supply from outside sources and provide for supply of non-military personnel. It should support replanning as the situation changes. Recent work has investigated the availability and utility of existing tools of this type (Brundage et al. 1998).
- The object of the transport analysis tool is to support the transportation analysis for mission arrival, sustainment, and departure in an OOTW context. The tool needed is a decision support tool that plans the transport requirements, based on all appropriate constraints. It must support replanning when the situation changes after some transport has been accomplished. The Joint

Flow and Analysis System for Transportation (JFAST) and the Model for Intertheater Deployment by Air and Sea (MIDAS) are examples of this type tool.

Support Tools

This group contains three specific tools and a cluster of several tools related by type. The *COA* comparitor permits the development of courses of action (COAs) through several levels of alternatives: an influence diagram/decision tree methodology would support this type analysis. The MOE calculator supports the calculation and tracking of MOE values. The communications tool supports planning the communications system within the complex context of OOTWs. The cluster of disaster impact tools (e.g., hurricanes, volcanoes, earthquakes, fires, and nuclear accident) supports the estimate of the situation in several technical areas, such as engineering and health. The Consequence Assessment Tool Set (CATS) supports some of these functions.

Cost Models

Seven tools make up this group. Their object is to calculate the cost information for various aspects of OOTWs: incremental costs of notional OOTWs, to support long-term analysis; probable incremental costs, to support the decision on engaging in a particular OOTW; relative (full) costs, to support the selection of the mission plan; costs incurred, to support cost recovery from other U.S. agencies and from foreign organizations and governments; incremental costs of a particular OOTW, to support the Congressional Budget process; costs of a particular OOTW, including equipment depreciation, readiness losses, increased reserve recruitment and training costs, and perhaps other costs, to support future acquisition, budgeting and training decisions; and actual costs of a completed OOTW, to support improved estimates of future operations and reports to Congress on actual costs. Work is underway to address analysis tools (Institute for Defense Analyses 1998; Hartley and Packard 1998b).

Information Tools

There are two tools in this category. The situation display presents the information concerning the situation in a manner designed to maximize understanding: the Virtual Information Center (VIC) project represents a first attempt at creating this type tool (Sovereign 1998). The data warehouse either stores or provides links to (as appropriate) all pertinent data. The data and their useability are critical to good analysis in the OOTW domain, as well as in the combat domain. However, the data required for OOTW analysis and the display requirements are in an embryonic state when compared to the state of affairs of combat analysis.

Tool Definition Process

Analysis of OOTWs is a new field and is in a state of flux. The first concerted effort to address the need for analytic tools is documented in Hartley (1996). Follow-on efforts are documented in Staniec (1998), Hartley and Packard (1998a), Brundage et al. (1998), Lidy (1998), Sovereign (1998), Hartley and Packard (1998b), and Hartley and Packard (1999).

See

- ▶ Analytic Hierarchy Process
- ▶ Battle Modeling
- ▶ Cost Analysis
- ▶ Crime and Justice
- ▶ Econometrics
- ▶ Economics and Operations Research
- ▶ Global Models
- ▶ Health Care Management
- ▶ Influence Diagrams
- ▶ Logistics and Supply Chain Management
- ▶ Military Operations Research
- ▶ Operations Management
- ▶ Production Management
- ▶ Public Policy Analysis
- ▶ Simulation of Stochastic Discrete-Event Systems
- ▶ Supply Chain Management
- ▶ System Dynamics

References

Bond, D., & Vogele, W. B. (1995). *Profiles of international "Hotspots"*. Harvard, Cambridge, MA: Center for International Affairs.

Brundage, W., et al. (1998). *Analysis of U.S. involvement in multiple small scale contingencies — Failed state*, OSD (PA&E).

Davis, D. F. (1996). Peace operations analysis with Bayesian belief networks. In *13th International symposium on military operational research (ISMOR)*, England.

Hartley, D. S., III. (1996). *Operations other than war: Requirements for Analysis Tools Research Report, K/DSRD-2098*. Oak Ridge, TN: Lockheed Martin Energy Systems, Inc.

Hartley, D. S., III, & Packard, S. L. (1998a). *OOTW tool requirements in relation to JWARS, K/DSRD-3076*. Oak Ridge, TN: Lockheed Martin Energy Systems, Inc.

Hartley, D. S., III, & Packard, S. L. (1998b). *OOTW cost tools, Y/DSRD-3099*. Oak Ridge, TN: Lockheed Martin Energy Systems, Inc.

Hartley, D. S., III, & Packard, S. L. (1999). *OOTW mission planning, Y/DSRD-3117*. Oak Ridge, TN: Lockheed Martin Energy Systems, Inc.

Institute for Defense Analyses (1998). Contingency operations support tool web site, http://www.ida.org/COST.

Lidy, A. M. (1998). *United States military role in smaller scale contingencies, D2166*. Alexandria, VA: Institute for Defense Analyses.

National Simulation Center (1996). *Information Paper*. http://www.leav.army.mil/nsc/famsim/spectrum/infopapr.htm.

Sovereign, M. (1998). *Humanitarian assistance and disaster relief in the next century* (Workshop Report). Arlington, VA: CCRP, National Defense University.

Staniec, C. (1998). *MORS workshop on OOTW analysis and modeling techniques (OOTWAMT)*. Alexandria, VA: Military Operations Research Society.

Woodcock, A. E. R. (1996). Modeling and analysis of societal dynamics: The deployable exercise support (DEXES) system. In A. Woodcock & D. Davis (Eds.), *Analytical approaches to the study of future conflict*. Clemensport, Nova Scotia: The Lester B. Pearson Canadian International Peacekeeping Training Centre.

Military Operations Research

Brian R. McEnany and Robert S. Sheldon
Military Operations Research Society (MORS), Alexandria, VA, USA

Introduction

To say that Military Operations Research (MOR) is the application to military operations of the methods of operations research (OR) is strictly correct, but gives only one clue to understanding the subject. The MOR accomplishments in World War II, sketched below, pioneered and greatly influenced the early development and institutionalization of operations research generally. Also, they led to the continuation of MOR after the war, in the governments of World War II participants, in academia, in industry, in not-for-profit think tanks, and its adoption in similar institutions of other nations. The emphasis in this article is on practice and trends in the United States, with particular emphasis on the Army.

The general methods of OR apply in particular to many aspects of military applications. Such differences as exist pertain mainly to the needs of military security and classification procedures, the nature of military operations and equipment, and the concerns of strategy, operational art, and tactics that relate to the use of military forces as instruments of national policy.

Current developments in the field are described in the quarterly bulletin *Phalanx* and the journal *Military Operations Research* published by the Military Operations Research Society (MORS) and the Military Applications Society (MAS) of INFORMS. MORS also conducts annual classified symposia, as well as smaller mini-symposia and workshops (some unclassified), from which they publish proceedings and monographs.

World War II MOR Accomplishments

Although there were individual contributions to the scientific study of military operations, ranging from Archimedes to the work of Thomas A. Edison in World War I, it was in World War II that MOR became widespread and institutionalized. Solandt (1955) recalled that MOR began in the services in England as operational research in the early days of the war. The British work centered about different subjects depending on the service: in the Air Force it was the problem of how to use radar, in the Navy it was the problem of anti-submarine warfare, and in the Army it was first limited to anti-aircraft problems and again centered around radar. Professor Blackett is sometimes said to have started the work in all three services, and his account in Blackett (1962) drew on earlier papers to describe both results and methods.

Schrader (2006) describes the organization and use of OR by the U.S. Army from WWII until 1995. His detailed account of how and where OR was used represents a definitive study of the U.S. Army's use of OR in peace and war, and much of what is summarized here is based on his writings.

Cooperation between the United States and Britain over the use of OR did not begin immediately during WW II. British liaison teams visited the U.S., but it was not until late in 1940, just after the fall of France, that President Roosevelt authorized the creation of the National Defense Research Committee (NRDC) and subsequently, the Office of Scientific Research and Development (OSRD) under Professor Vannevar Bush. This office helped recruit, manage and organize the military OR effort in the U.S defense establishment during the war.

While Britain fielded OR teams and detachments with its Army and Navy during the war, only the U.S. Navy and the U.S. Army Air Force (AAF) took full advantage of the new discipline after Pearl Harbor. OR teams of scientists and businessmen, recruited and organized through the OSRD, formed the initial groups. Small detachments were sent to AAF to conduct bombing accuracy studies and assessments of tactics. A useful account of World War II MOR, centering about the AAF, is Brothers (1954). In addition to illuminating examples such as aerial bombing accuracy improvement, it gives valuable guidance on the organization of MOR groups and operating procedures. In World War II, most of the MOR practitioners were civilians (though sometimes in uniform), and they had to earn the trust of military operators over time through useful work. This, of course, is by no means unique to MOR in World War II.

The U.S. Army's Technical Services – the scientific branches (Ordnance Department, the Medical Services, Signal Corps and Chemical Warfare Service) took advantage of the expertise offered by the new multi-disciplinary teams and detachments were deployed in Europe and in the U.S. The Army ground forces, on the other hand, were reluctant to begin using operational analysts (or "Op Annies" as they were called) until 1944. Teams were primarily used to support anti-aircraft weapons development and support to U.S. Army forces in the Pacific area.

The AAF was quick to emulate its British comrades and OR teams were soon supporting the various major Air Force operations in Europe and elsewhere. The Army's technical services were slower, but before the end of the war, studies in support of radar training, development, and organization, signal work load in message centers, transportation scheduling, loading, and handling, as well as some operational studies involving introduction of new equipment and technology to units were undertaken. The ground forces lagged well behind until late in 1944 when OR teams were sent to the Pacific.

At the end of the war, the rapid demobilization of the U.S. Army dissolved its existing teams and organizations as civilian scientists quickly returned to their academic or business careers. The national offices, NRDC and OSRD, were also demobilized, but the newly organized Department of Defense (DoD) created the Weapon Systems Evaluation Group (WSEG) to carry on work begun earlier. The limited use of OR in the Army's decision-making process during the war lagged well behind the other services. In the postwar period, the civilian leadership recognized the benefit provided by the studies and analysis of weapon systems and their development. The ground Army quickly closed the gap in the postwar period.

Early in the post-war period, Morse and Kimball (1946) drew on the work of many early MOR analysts of the Operations Research Group, U.S. Navy, to give results and methods. That work, once it was declassified and slightly modified, was republished in 1951 and was very influential, not only in introducing MOR to future analysts, but also in introducing the potential applications of OR generally to a wider audience. This Morse and Kimball classic was republished by MORS in 1998.

The above work quotes a letter from Admiral King that enumerated helpful MOR applications (suggestive also of the work in other services):

(a) The evaluation of new equipment to meet military requirements.
(b) The evaluation of specific phases of operations (e.g., gun support, anti-aircraft fire) from studies of action reports.
(c) The evaluation and analysis of tactical problems to measure the operational behavior of new material.
(d) The development of new tactical doctrine to meet specific requirements.
(e) The technical aspects of strategic planning.
(f) The liaison for the fleet with the development and research laboratories, naval and extra-naval.

Morse and Kimball also gave some reasons for the emergence in World War II of the practical value of the methods of MOR. As opposed to earlier wars there were the following:

(i) more repetitive operations susceptible to analysis — strategic bombing, submarine attacks on shipping, landing operations, etc.;
(ii) increased mechanization of warfare, in that "... a men-plus-machines operation can be studied statistically, experimented with, analyzed, and predicted by the use of known scientific techniques just as a machine operation can be."
(iii) increasing tempo of obsolescence in military equipment ... When we can no longer have the time to learn by lengthy trial and error on the battlefield, the advantages of quantitative appraisal and planning become more apparent."

Post-War MOR Developments

After World War II ended, a majority of the MOR practitioners returned to non-military pursuits: universities, laboratories, industry, etc. The military services wondered how much MOR would be needed in peacetime. Each decided to institutionalize its use of MOR. An early chapter of Tidman (1984) gives an interesting account of how the Navy chose to continue MOR by establishing the Center for Naval Analysis (CNA) after World War II and by 1948, each service had a different choice or mix of civil service groups, not-for-profit groups, use of industry, etc., and their emphases varied over time. The newly organized U.S. Air Force soon created Project RAND (later RAND Corporation) in 1948 to support its research and development efforts. The newly formed DoD followed suit with establishment of WSEG. Fairly soon, as the Cold War emerged, there was general recognition that it would be necessary to increase the use of MOR. Both Tidman and Schrader appropriately addressed this topic as periods of consolidation and growth in their respective histories.

The Army rapidly demobilized after the war, as stated above, the civilian scientists quickly returned to their jobs and homes. While the Army ground forces quickly inactivated its MOR organizations, the technical services (Ordnance and Signal) retained theirs. By 1948, the Army's leadership created a relationship with John Hopkins University under Dr. Ellis Johnson to form the Operations Research Office (ORO), a relationship that was to last for 13 years. World War II had seen the introduction of radar, atomic weapons, cruise missiles, and ballistic missiles, but each type was still improving rapidly at war's end. Their implications for, and fuller integration into, military forces needed more thought. The Cold War climate also provided a sense of urgency, and MOR offices took on these problems as important foci of effort. The growing Cold War with the Soviet Union forced the Army to address more than just weapons design and tactical doctrine. ORO soon began addressing areas well beyond weapons development – entering international politics, economics, national policy and global strategy while the technical services and newly organized field force boards maintained their focus on weapons development. Several key MOR organizations were created – ORO, Combat Operations Research Group (CORG), the Human Resources Research Organization (HumRRO), and Special Operations Research Office (SORO) dealing with psychological operations. Computer modeling of complex systems met increased need to process large quantities of data. At Headquarters, Department of the Army (HQ DA), the Strategic Tactics and Analysis Group (STAG) was formed to study force structure and future forces capability through gaming and simulation. The increasing use of MOR in the combat development process fostered a need for increased numbers of military officers with MOR training and a formal Operations Research/Systems Analysis (ORSA) specialty program was created in 1967 to satisfy the growing need to form in-house MOR capabilities as the Army moved toward a competitive contractual arrangement with various commercial and academic analytic groups. The Research Analysis Corporation (RAC) took over as the primary research arm of the Army staff in 1963 while primary research efforts were funneled into academia through the Army Research Office (ARO) at Duke University.

Some of the postwar applications of MOR resembled wartime MOR, with combat operations replaced by tests or exercises. With the rise of the Continental Army Command (CONARC), MOR organizations began efforts involving war gaming and field experimentation. As technology increased and problems became more complex, recommendations soon increased the amount of field

experimentation and testing, and by 1956, the first combat development and testing command was created. However, some of the OR (or operations analysis or operations evaluation, as it was often termed) *remained* devoted to operations of supply, logistics, recruiting, and training. Moreover, much of the post-war efforts went into thinking through the implications of new weapons for new types of combat operations. It fostered an atmosphere that led to increased use of digital computing capabilities in war gaming and simulation to help solve increasingly more sophisticated and complex problems.

The Emergence of Systems Analysis

MOR also took on problems at a level higher than that of individual weapon systems or engagements between two opposing weapon systems. Even in a Cold War climate, there were significant limits on national expenditures for armed forces. It was necessary for government to decide "how much is enough" and MOR sought to aid this decision.

Applications of OR at this high level, often termed systems analysis, face difficulties far greater than the difficulties of World War II MOR, significant as the latter were. Wartime combat analysis, sometimes without recognizing it, had already faced criterion problems of sub-optimization, as Hitch (1953) points out. These become still more significant when structuring forces for the future, seeking to be prepared to deal with contingencies still beset with great uncertainty.

Hitch (1955) gives an understanding of the relative difficulty of systems analysis by comparing the World War II problem of improving bomber accuracy with the postwar problems of weapon system development and force composition. In the former problem, difficult as it seemed at the time, known were the types of aircraft involved, how many there were, much about their characteristics, the kind of bombs available, and much about enemy targets and their defenses. These become variables when considering an uncertain future that may sometimes hold a multiplicity of potential opposing forces.

The difficulties are in the problems, as Hitch went on to point out. Despite these difficulties, governments must make decisions and systems analysis, with all of its limitations, has much to offer. MOR analysts developed judgment in cutting problems down to size, and Quade (1954) collected some of the helpful approaches in an influential volume. Quade and Boucher (1968) and Miser and Quade (1988) give refinements and extensions to non-defense analysis.

The Institutionalization and Impact of Systems Analysis

Hitch and McKean (1960) did much to introduce cost-effectiveness studies as instruments of defense systems analysis. In the Kennedy administration in 1961, Secretary of Defense McNamara brought Hitch into the Office of the Secretary of Defense (OSD) as Comptroller to install a system of planning-programming-budgeting (PPB), and Enthoven, as Hitch's assistant, started an office of systems analysis. Although the titles and organizational placement have changed over the years, OSD has continued both PPB and systems analysis.

These new offices had great impact. The government sought to create similar offices in other departments (Bureau of the Budget 1965). Within the DoD, the new OSD offices played an important role in departmental decisions. As its emphasis on, and requests for, quantitative analysis increased, the military services organized and enlarged their MOR offices to meet the demand.

The above developments came at a time when computer capabilities were rapidly increasing. Many MOR offices sought to use the new capabilities in producing cost-effectiveness studies required for systems analysis. Computer simulation models began to proliferate in the effort to understand what new or proposed weapon systems would contribute to the future battlefields. Because this effort contributed to studies with great impact on weapon systems acquisition, it has continued to grow.

Wartime Combat OR in Korea and Vietnam

Although its successes in World War II led service leadership to gradually incorporate MOR into its decision-making process, MOR efforts came to emphasize future weapon system acquisition as described above. For more details of what is summarized here, see Schrader (2008).

KOREA: Right from the beginning, the Army leadership was admonished to deploy MOR teams to Japan and Korea. As in WWII, the analysis of current operations, organizations and tactics were prominent. Increased interest in new organizations, counter-measures, winter operations, clothing, airborne operations, and psychological warfare were undertaken. By the end of 1953, the efforts of the deployed MOR teams had validated WWII experience and demonstrated that MOR could be successfully applied to land warfare. Between Korea and Vietnam, multiple MOR organizations provided analyses and supported the combat development process. Major war gaming and simulation centers were created to study the impact of new weapons, organizations, and future force structure and tactics. Centers grew within The U.S. Army Training and Doctrine Command (TRADOC) at Fort Leavenworth and White Sands to assist in defining new organizations, doctrine and tactics while HQ DA continued to rely upon the successor to STAG, the Concepts Analysis Agency (CAA), to evaluate future force structures. Individual weapons research and evaluation continued to expand at Aberdeen Proving Grounds where the Ballistics Research laboratory (BRL) studied, evaluated, assessed and developed new, improved weapons.

VIETNAM: While a shooting war was underway in Southeast Asia (SEA), the rise of PPB at the Pentagon split Army MOR activities. It developed additional in-house capability to support the centralization of decision-making begun under Secretary McNamara and the Office, Secretary of the Army (OSA). More MOR trained personnel were needed to support the PPB System (PPBS) and a formal specialty program was created in 1967 for military officers. This was coupled with use of civilian contractors and Federally Funded Research and Development Centers (FFRDCs). That is not to say that MOR activities were totally devoted to PPBS. Of particular interest was the lengthy analysis and assessment of the air mobility concept and organization of the air assault division prior to the war in SEA. Multiple organizations, field boards and MOR offices significantly supported the vast testing and experimentation of the air mobility concept.

The war in SEA renewed interest in the study of current operations, battlefield performance of weapons, equipment, organizations and tactics. RAC, the successor to ORO, deployed teams to to collect data along with HumRRO, SORO and Combat Development Command (CDC). HQ Military Assistance Command, Vietnam (MACV) established an in-theater analysis and assessment capability. Quantitative methods were employed extensively at Field Force and Division level. Manually assisted war games were run to help develop alternate strategies and think through potential issues. Efforts were focused upon counter-insurgency operations and suffered from lack of large amounts of quantitative data needed to adequately analyze it. Still, as one division commander noted, the "judicious use of operational analysis and analytic techniques when melded with military judgment were quite effective in improving performance of many activities."

In Chapter I of Hughes (1989), Thomas observed that combat OR both in Korea and later in Vietnam was very similar to that of World War II. Despite the postwar increase in modeling and computer capabilities, it did not make nearly as much contribution in Korea or Vietnam as might have been expected. "Though the menu of available techniques increased with time, much that had been learned in World War II was forgotten and relearned in later conflicts." The 1960s and 1970s were a time of great growth in the analytic community. MOR efforts greatly expanded force planning and management with a commensurate need to expand the number of MOR-trained officer personnel. A whole new set of challenges faced the Army after Vietnam as the MOR community assisted in helping the Army reorganize, revitalize, and reorient itself prior to the First Gulf War.

Contributions After Vietnam and the Gulf War

The period after Vietnam was a time of recovery and reorganization for the U.S. Army (see Schrader 2009 for more details). The multi-year conflict had severely damaged the Army's equipment modernization process and MOR efforts concentrated upon providing the analytic underpinning for major changes in weapon systems, equipment, organizations, doctrine and training. In light of two major studies affecting MOR organizations, competitive contracting was more formalized

(RAC was disestablished) and MOR assets became more concentrated into fewer organizations – CAA, TRADOC Analysis Command (TRAC), the Operational Test and Evaluation Command (OPTEC) and the Army's Material Systems Analysis Agency (AMSAA) ultimately concentrated the efforts of the majority of civilian and military MOR specialists and performed the majority of all studies. MOR became more integrated into the Army's decision-making process as new technology, better weapon systems, and improved organizations were developed. A pyramid of responsibility was formed with CAA at the apex studying force structure and strategy, TRAC focused on battalion to Corps level studies, and AMSAA dealing with individual weapon system analysis.

The end of the Cold War in 1989 presented the Army and the MOR community with entirely new issues – much more complex and demanding that ever before – and MOR support to the material acquisition process became more important. The ever-increasing improvements in technology and computing power brought with it an expanding use of models and simulations to solve the issues facing the Army. This expansion also created issues in validation, verification and accreditation of the analytic tools used to support the decision making process.

During the First Gulf War in 1991, the efforts of 20 years of MOR involvement in conjunction with new organizations, new equipment and weapon systems, new doctrinal, and training improvements, fielded the finest fighting force in the history of the United States. Each of the major organizations actively supported the collection of data. CAA was intimately involved in the evaluation of the forces involved during the planning phase of the operation. War games and separate assessments assisted Army planners and major headquarters in preparing for the deployment and employment of forces. Multiple rapid response assessments – some as short as 12 h – were provided during Operation Desert Shield. Ultimately, a small MOR cell was deployed to support HQ Central Command (CENTCOM), but most MOR efforts were conducted in the continental U.S. (CONUS). The successful military outcome underscored the need for rapid and flexible support to deployed forces with a full range of theater level analysis capabilities.

MOR Lessons from Desert Shield/Desert Storm

The new computer and modeling capabilities seemed to have more impact in MOR for the Gulf War combat of 1991. Vandiver et al. (1992) concluded that while some of its analytic lessons were reminiscent of World War II, and some lessons were probably peculiar to wars like the Gulf War, there were trends indicative of future combat analysis:

- Computer influence on analysis is increasingly varied and pervasive.
- Software analytical tools are increasingly available to all - including non-analysts.
- The demand for good databases is growing more rapidly than the supply.
- There is growing need for coalition and joint service analysis.
- There is increasing analytical interest in operational art and campaign focus.
- There is a need to have MOR teams ready to join, and planning models and simulations in place with deployed forces.
- Teams must be ready to improvise quickly to support ongoing and planned operations in the field.
- There is less danger of central misuse of field analysis and data than formerly. The lessons of better methods of data collection and selection of more accurate measures of effectiveness learned in earlier conflicts have been absorbed by the MOR community.

Concluding Remarks

Although MOR has been a flourishing enterprise with an expanding technological menu, there are still issues to resolve, some long standing. While it is clear that MOR tools and techniques improved the material acquisition process and the PPBS, a significant fraction of the issues relate to modeling and simulation, or are frequently so characterized. Some of the more serious concerns address scientific foundations (including verification and validation); DoD organization and management (including that for MOR); management; filling a perceived need; and taking suitable advantage of technological opportunities.

See

- ▶ Air Force Operations Research
- ▶ Battle Modeling
- ▶ Center for Naval Analyses
- ▶ Cost Analysis
- ▶ Cost-Effectiveness Analysis
- ▶ Exploratory Modeling and Analysis
- ▶ Military Operations Other Than War
- ▶ Operations Research Office and Research Analysis Corporation
- ▶ RAND Corporation
- ▶ Simulation of Stochastic Discrete-Event Systems
- ▶ Systems Analysis
- ▶ War Game

References

Blackett, P. M. S. (1962). *Studies of war*. New York: Hill and Wang.

Brothers, L. A. (1954). Operations analysis in the United States Air Force. *Operations Research, 2*, 1–16.

Bureau of the Budget. (1965). Planning-programming-budgeting. Bulletin 65-5, Washington, DC.

Hitch, C. J. (1953). Sub-optimization in operations problems. *Operations Research, 1*, 87–99.

Hitch, C. J. (1955). An appreciation of systems analysis. *Operations Research, 3*, 466–481.

Hitch, C. J., & McKean, R. N. (1960). *The economics of defense in the nuclear age*. Santa Monica, CA: RAND. R-348.

Hughes, W. P., Jr. (Ed.). (1989). *Military modeling* (2nd ed). Alexandria, VA: MORS.

Miser, H. J., & Quade, E. S. (Eds.). (1988). *Handbook of systems analysis: Craft issues and procedural choices*. New York: North-Holland.

Morse, P. M., & Kimball, G. E. (1946). *Methods of operations research*. OEG Rpt. 54, Office of the Chief of Naval Operations, Navy Dept., Washington, DC.

Quade, E. S. (Ed.). (1954). *Analysis for military decisions*. Santa Monica, CA: RAND. R-387-PR.

Quade, E. S., & Boucher, W. I. (Eds.). (1968). *Systems analysis and policy planning: Applications in defense*. Santa Monica, CA: RAND. R-439-PR.

Schrader, C. R. (2006). *History of operations research in the US Army:1942-1962*. Carlise, England: Center for Military History. Pub70-102-1.

Schrader, C. R. (2008). *History of operations research in the US Army:1961-1973*. Carlise, England: Center for Military History. Pub70-105-1.

Schrader, C. R. (2009). *History of operations research in the US Army:1973-1995*. Carlise, England: Center for Military History. Pub70-110-1.

Solandt, O. (1955). Observation, experiment, and measurement in operations research. *Operations Research, 3*, 1–14.

Tidman, K. R. (1984). *The operations evaluation group, a history of naval operations analysis*. Annapolis, MD: Naval Institute Press.

Vandiver, E. B., et al. (1992). Lessons are learned from desert shield/desert storm. *PHALANX, 25*(1), 6–87.

MIMD

Multiple instruction, multiple data. A class of parallel computer architectures in which each processing element fetches and decodes its own stream of instructions, possibly different from the instruction streams for other processors.

Minimum

A real-valued function $f(x)$ is said to have a minimum on a set S when the greatest lower bound of $f(x)$ on S is assumed by $f(x)$ for some x^0 in S. Thus, $f(x^0) \leq f(x)$ for all x in S.

See

- ▶ Global Maximum (Minimum)

Minimum (Maximum) Feasible Solution

In a mathematical-programming problem, the solution that both satisfies the constraints of the problem and minimizes (maximizes) the objective function is a minimum (maximum) feasible solution. Such solutions may not be unique.

Minimum Spanning Tree Problem

Given a connected network with n nodes and individual costs associated with all edges, the problem is to find the least-cost spanning trees.

See

- Network Optimization
- Spanning Tree

Minimum-Cost Network-Flow Problem

In a directed, capacitated network with supply and demand nodes, the problem is to determine the flows of a single, homogeneous commodity from the supply nodes to the demand nodes that minimize a linear cost function. In its general form, when the network contains transshipment or intermediate nodes – nodes that are neither supply nor demand nodes – the problem is called the transshipment problem. Conservation of flow through each node is assumed. Due to its special mathematical structure, this problem has a solution in integer flows, given that the data that define the network are integers. It is a linear-programming problem whose major constraints form a node-arc incidence matrix.

See

- Conservation of Flow
- Maximum-Flow Network Problem
- Network Optimization

MIP

- Mixed-Integer Programming Problem (MIP)

MIS

Management information systems.

See

- Information Systems and Database Design in OR/MS

Mixed Network

A queueing network in which some customers can enter and leave the network while others neither enter nor leave but cycle through the nodes endlessly. A queueing network in which the routing process contains at least one closed set of states for some types of customers but not others.

See

- Closed Network
- Networks of Queues
- Open Network
- Queueing Theory

Mixed-Integer Programming Problem (MIP)

A mathematical-programming problem in which the constraints and objective function are linear, but some of the variables are constrained to be integer valued. The integer variables can either be binary or take on general integer values.

See

- Binary Variable
- Integer and Combinatorial Optimization
- Linear Programming
- Mathematical Programming

Model

An idealized — abstract and simplified — representation of a real-world situation that is to be studied and/or analyzed. Models can be classified in many ways. A mental model is an individual's conceptual, unstated, view of the situation under review; a verbal or written model is a description of one's mental model; an iconic model looks like what it is supposed to represent (e.g., an architectural model of

a building); an analogue model relates the properties of the entity being studied with other properties that are both descriptive and meaningful (e.g., the concept of time as described by the hands and markings of a clock); a symbolic or mathematical model represents a symbolic representation of the process under investigation, e.g., Einstein's equation $E = mc^2$, a linear-programming model, or a computer simulation model.

See

- ▶ Descriptive Model
- ▶ Deterministic Model
- ▶ Linear Programming
- ▶ Mathematical Model
- ▶ Normative Model
- ▶ Predictive Model
- ▶ Prescriptive Model
- ▶ Simulation of Stochastic Discrete-Event Systems
- ▶ Stochastic Model

Model Accreditation

Saul I. Gass
University of Maryland, College Park, MD, USA

Model accreditation is an official determination that a model is acceptable for a specific purpose (Williams and Sikora 1991; Ritchie 1992). Accreditation certifies that the element being accredited meets given standards. For a model, accreditation must be done with respect to the model's explicit specifications and the demonstration that the computer-based model does or does not meet the specifications. This demonstration is the responsibility of the model developers, who must show that their work passes agreed-to user and developer acceptance tests. If the modeling process was done properly and was accompanied by appropriate documentation, accreditation of the model for its specified uses should follow.

Accreditation of a model must rely on a review and evaluation of its available documentation. Such an evaluation, usually done by an independent third-party, is made against various criteria to determine the levels of accomplishment of the criteria, in particular those of verification and validation. The review is made with a specific user and uses in mind. The review should produce a report that gives guidance to the user on whether or not the model in question can be used with confidence for the designated uses, that is, the model is or is not accredited for specific uses (Gass 1993).

The ideas, if not the general process behind model accreditation, have been accepted by modeling agencies within government and private industry, most notably by the U.S. Department of Defense (2009) in the context of modeling and simulation (see also Sargent 2005).

See

- ▶ Model Evaluation
- ▶ Model Management
- ▶ Practice of Operations Research and Management Science
- ▶ Validation
- ▶ Verification
- ▶ Verification, Validation, and Testing of Models

References

DoDI. (2009). *DoD modeling and simulation (M&S) verification, validation, and accreditation* (*VV&A*). DoDI 5000.61, December 9, 2009.

Gass, S. I. (1993). Model accreditation: A rationale and process for determining a numerical rating. *European Journal of Operational Research, 66*(2), 250–258.

Ritchie, A. E. (Ed.). (1992). *Simulation validation workshop proceedings (SIMVAL II)*. Alexandria, VA: Military Operations Research Society.

Sargent, R. G. (2005). Verification and validation of simulation models. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, & J. A. Joines, (Eds.), *Proceedings of the 2005 Winter Simulation Conference*, IEEE Press.

Williams, M. K., & Sikora, J. (1991). SIMVAL Minisymposium — A Report," Phalanx, *Bulletin of the Military Operations Research Society*, *24*, 2.

Model Builder's Risk

Probability of rejecting the credibility of a model when in fact the model is sufficiently credible.

See

► Verification, Validation, and Testing of Models

Model Evaluation

Saul I. Gass
University of Maryland, College Park, MD, USA

Model evaluation or assessment is a process by which interested parties, who were not involved in a model's origins, development and implementation, can assess the model's results in terms of its structure and data inputs so as to determine, with some level of confidence, whether or not the results can be used in decision making. Model evaluation encompasses: (1) verification, validation, and quality control of the usability of the model and its readiness for use, and (2) investigations into the assumptions and limitations of the model, its appropriate uses, and why it produces the results it does.

There are three reasons for advocating evaluation of models: (1) for many models, the ultimate decision maker is far removed from the modeling process and a basis for accepting the model's results by such a decision maker needs to be established; (2) for complex models, it is difficult to assess and to comprehend fully the interactions and impact of a model's assumptions, data availability, and other elements on the model structure and results without a formal, independent evaluation; and (3) users of a complex model that was developed for others must be able to obtain a clear statement of the applicability of the model to the new user problem area (Gass 1977a).

All procedures for evaluating a model are basically information gathering activities, with the detail and level of information being a function of the purposes of the assessment and the skills of the assessors. Specific evaluation approaches are given in Gass (1977a, b), Gass (1980), U.S GAO (1979), with an evaluation case study given in Fossett et al. (1991).

A model evaluation procedure and its objectives should be tailored to the scope and purposes of the model and will vary with the model, model developers, assessors, users, and available resources. Model assessment is an expensive and involved undertaking; all models need not be assessed. Model developers and users should recognize that by applying proper modeling management procedures, the burdens that evaluators of models have to contend with are alleviated greatly (Gass 1987).

See

► Model Accreditation
► Model Management
► Practice of Operations Research and Management Science
► Project Management
► Verification, Validation, and Testing of Models

References

Fossett, C., Harrison, D., Weintrob, H., & Gass, S. I. (1991). An assessment procedure for simulations models: A case study. *Operations Research, 39*, 710–723.

Gass, S. I. (1977a). Evaluation of complex models. *Computers and Operations Research, 4*, 27–35.

Gass, S. I. (1977b). A procedure for the evaluation of complex models. *Proceedings of the First International Conference in Mathematical Modeling*, 247–258.

Gass, S. I., (Ed.). (1980). *Validation and Assessment Issues of Energy Models* (National Bureau of Standards Special Publication 569, U.S. GPO Stock No. 033-003-02155-5). Washington, DC: U.S. Government Printing Office.

Gass, S. I., (Ed.). (1980). *Validation and Assessment Issues of Energy Models* (National Bureau of Standards Special Publication 616). Washington, DC: U.S. Government Printing Office.

Gass, S. I. (1983). Decision-aiding models: Validation, assessment, and related issues for policy analysis. *Operations Research, 31*, 603–631.

Gass, S. I. (1987). Managing the modeling process: A personal perspective. *European Journal of Operational Research, 31*, 1–8.

Ritchie, A. E., (Ed.). (1992). *Simulation validation workshop proceedings*, (*SIMVAL II*). Alexandria, VA: Military Operations Research Society.

U.S GAO. (1979). *Guidelines for model evaluation*. Washington, DC: GAO/PAD-79-17.

Willemain, T. R. (1995). Model formulation: What experts think about and when. *Operations Research, 43*, 916–932.

Model Management

Ramayya Krishnan[1] and Kaushal Chari[2]
[1]Carnegie Mellon University, Pittsburgh, PA, USA
[2]University of South Florida, Tampa, FL, USA

Introduction

The term model management was coined in the mid-1970s in the context of work on decision support systems (DSS) (Sprague and Watson 1975; Will 1975). An important objective of the DSS concept was to provide an environment in which decision makers could gain materially useful insights by interactively exercising OR/MS models. However, developing such an environment required principled solutions to problems of specifying, representing and interacting with models. This focus on models, and in turn on modeling, led to the study of model management, defined broadly to encompass the study of model representation, the set of operations facilitated by such representation at various stages of the modeling life cycle, and computer-based environments that facilitate modeling.

What follows is a brief review of work in two areas that have been actively studied in model management. First, work on languages to specify models, and on the development of techniques to facilitate operations that support modelers in both the pre-solution and post-solution phases of the modeling life cycle. Second, work on the representation of a collection of models (e.g., a model library) and the development of techniques to enable model selection and configuration. As with other information technology-based fields, model management has benefitted from the growth of Internet technologies. A detailed review of the implications for model management of the growth in Internet, and in particular the World Wide Web technologies, is in Bhargava and Krishnan (1998), and Bhargava, Power, and Sun (2007).

Model Management-I

Modeling languages — The need to represent a model in a notation that is easy to validate, verify, debug, maintain and communicate motivated the development of modeling languages (Fourer 1983). Prior to their development, the only computer-executable representation of a model was in an arcane format optimized for efficient solution (e.g., the Mathematical Programming System MPS format).

Current modeling languages provide a high-level symbolic notation to specify models. Solution operations can also be declared and all the required details of binding the model instance to the data structures required by solver done transparently. Further, this has greatly increased the productivity of model-based work.

Four principles have been articulated as essential to modeling language design (Bhargava and Kimbrough 1993; Fourer 1983; Geoffrion 1992a; Krishnan and Chari 2000). These are:

- *Model data independence*: requires the mathematical structure of the model to be independent of the data used to instantiate it. This permits model data to be modified in format, dimension, units or values without any modification to the model representation.
- *Model solver independence*: requires the model representation to be independent of the representation required by the solver. This permits more than one solver to be used with a given model. Further, it recognizes the fundamental differences in the requirements placed on model representations and representations required by the solver.
- *Model paradigm independence*: requires that the modeling language allow the representation of models drawn from different paradigms (e.g., mathematical programming and discrete event simulation).
- *Meta level representation and reasoning*: requires that the modeling language represent information *about* model components and models, in addition to their mathematical structure in order to enable semantic consistency checking.

Modeling languages incorporate these principles to varying degrees. Examples of modeling languages include spreadsheet-based languages such as IFPS (Gray 1987), algebraic modeling languages such as GAMS (Bischop and Meeraus 1982), AMPL (Fourer et al. 1990), and MODLER (Greenberg 1992),

relational modeling languages such as SQLMP (Choobineh 1991), graphical modeling languages such as NETWORKS (Jones 1991), and Model Graphs (Chari and Sen 1997), typed modeling languages such as ASCEND (Piela et al. 1992), and XML-based languages such as OptML, SNOML, FML, and MathML. A survey of XML-based representations can be found in Valente and Mitra (2007). New developments in algebraic modeling languages include extensions for constraint programming (Fourer and Gay 2002), and extensions for stochastic programming (Valente et al. 2009). The formal analysis of the semantics of typed modeling languages is in Bhargava, Krishnan, and Piela (1997). There is also an active market in commercial modeling languages and systems. A survey of these systems can be found in Sharda and Rampal (1995).

Two developments have had a significant impact on modeling languages. One is the seminal work on Structured Modeling (SM) (Geoffrion 1987). Developments and research directions are described in a survey of structured modeling (Geoffrion 1999a), and an annotated bibliography is given in Geoffrion (1999b). While previous work on modeling languages had sought to provide a computer executable representation of the notation traditionally used by modelers, SM defines a theory that treats models as hierarchical collections of definitional dependencies. This enables structured modeling languages to satisfy all the four design principles discussed above. While several languages have implemented SM, the most completely developed of these is SML (Geoffrion 1992a, b). The other important development is the embedded languages technique, which can be used to define an architecture of considerable generality for modeling environments. This technique is used to specify modeling languages, as well as information about the terms and expressions stated in these languages. The TEFA modeling environment (Bhargava and Kimbrough 1993) has been implemented using this technique.

Operations — The early work on model management focused on model solution. The objective was to transparently bind solution algorithms to model instances. As noted above, modeling languages have realized this objective. Model management research has since focused on operations required to support both pre-solution and post-solution phases of the modeling life cycle. Next, research related to a pre-solution phase, model formulation, and a post-solution phase, model interpretation, are described.

Model Formulation — Model formulation is the task of converting a precise problem description into a mathematical model (Krishnan and Chari 2000). It is a complex task requiring diverse types of knowledge. The appropriateness of a model depends on a variety of factors such as accuracy, tractability, availability of relevant data, and understandability. Model formulation research has primarily focused on the development of theory, tools and techniques to support the formulation of deterministic mathematical programming models. Work by Gassmann and Ireland (1996) has studied the formulation of stochastic mathematical programming models.

Using protocol analysis, detailed studies of the expert modeling process have been conducted and process models have been developed (Krishnan et al. 1992; Raghunathan et al. 1994). Domain-independent and domain-specific model formulation strategies have been implemented in model formulation support systems (Krishnan 1990; Ma et al. 1989; Raghunathan et al. 1994) and a variety of representation and (deductive) reasoning schemes have been investigated. Liang and Konsynski (1993) have also investigated alternative approaches such as analogical reasoning and case-based reasoning to implement model formulation systems. A principled approach to formulating mathematical programming models is in Murphy, Stohr, and Asthana (1992). A survey of this research is given in Bhargava and Krishnan (1993).

Model Interpretation — Model interpretation consists of a variety of techniques to help a modeler comprehend a model. These include parametric analysis, structural analysis, and structure inspection.

Parametric analysis has long been supported in model management systems. Spreadsheets routinely support *what-if* analysis and goal seeking. Modeling languages for mathematical programming implement the theory of sensitivity analysis.

The pioneering work on structural analysis is due to Greenberg on the ANALYZE system (Greenberg 1987). ANALYZE extracts model structures that cause exceptions such as redundancy and infeasibility in linear programming models. The stream of work begun with ANALYZE has been considerably extended. Guieu and Chinneck (1999) described work and a toolkit called Mprobe that analyzes

infeasibility in mixed integer and integer linear programming models. Sharda and Steiger (1996) presented work on applying inductive learning techniques to facilitate model analysis. Kimbrough and Oliver (1994) examined the issue of post-solution analysis for models other than linear programs and have at-tempted to fashion a solution along the lines of ANALYZE. An important feature of their approach is the analysis of the impact on the solution to a model when changes are made to the parameters of a surrogate model.

Piela et al. (1992) described the use of a browser to inspect the structure of a model. Dhar and Jarke (1993) and Raghunathan et al. (1995) examined the usefulness of recording the rationale underlying a model. The documented rationale is used to aid comprehension as well as to correctly and consistently propagate the changes made to the structure of a model. Work on analyzing assumptions associated with models and visualizing the structure as a graph is reported in Basu and Blanning (1998). More recently, model ontology and model schema developed using OWL, a web ontology language based on XML, has been used for model representation and interpretation (Bhrammanee and Wuwongse 2008).

Model Management-II

Model libraries —In contrast to the work reviewed in the previous section, the focus of this stream of research assumes the existence of a library of debugged and validated models. This has led to the study of issues such as the representation of model libraries and operations such as model selection and configuration.

Model Representation — Predominantly, models are abstractly represented as black boxes, i.e., as a set of named inputs and outputs. This is in contrast to the detailed representation of the structure of the model in the previous section. A variety of representations, including virtual relations (Blanning 1982) and predicate logic (Bonczek et al. 1978) have been used to represent models. Additional structure has been imposed on these representations. Mannino, Greenberg, and Hong (1990) proposed the use of categories such as model type, model template, and model instance to organize the collection of models in a library. A model type is a general description of a model class such as linear programming. A model template is a refinement of a model type such as a production planning LP model, and a model instance is an instance of a model template in which the source of values for each parameter has been declared. Model templates have been represented using key-value pairs and filter lists in (Chari 2002), as Web Services Description Language (WSDL) service descriptors (Madhusudan 2007), and as OWL (XML-based) model profiles (Bhrammanee and Wuwongse 2008). Metagraphs (Basu and Blanning 1994a; Basu et al. 1997), a specialized type of graph structure, has been the significant development in this area.

Model Selection — Model selection leverages the existence of previously developed models to create a model for a new problem. In addition to the set of inputs and outputs associated with a model, additional information such as model assumptions need to be represented. Mannino et al. (1990) described model selection operators that match, either exactly or fuzzily, the assumptions associated with a model and those that are part of a problem statement. Work by Banerjee and Basu (1993) adopted the same framework as Mannino et al. (1990) but differed in its use of structuring technique called the Box Structure method (Mills et al. 1986), borrowed from the domain of systems analysis and design to develop its taxonomy of model types. Later, Guenther, Muller, Schmidt, Bhargava, and Krishnan (1997) studied the problem of selecting models and methods from web-based electronic catalogs. Chari (2002), implemented an approach based on matching filter spaces in selecting models. More recently, the work by Guntzer, Muller, Muller, and Schimkat (2007) have used a graph-matching procedure for selecting structured models represented as graphs. The problem of selecting and composing appropriate data mining models from a model library is now gaining attention (Liu and Tuzhilin 2008).

Model Configuration — Model configuration leverages previously developed models by either linking them together (referred to as model composition) or by integrating them (referred to as model integration). Model composition links together independent models such that the output of one model becomes an input to another. Model composition is often used in conjunction with model selection when no one model meets the requirements of a problem.

An example of model composition is the linking together of a demand forecasting model and a production scheduling model.

While the early work only permitted links between variables with the same name, later work of Muhanna (1992) and Krishnan, Piela, and Westerberg (1993) permitted linkages between objects (variables, arrays, instances of types, etc.) as long as certain semantic constraints are met. Muhanna (1992) also proposed methods that determine the order in which a collection of linked models should be solved. Representation methods and algorithms that can determine the set of models that need to be composed in order to obtain a set of outputs from a given set of inputs have been a major focus of model composition research. While the early work was based on virtual relations (Blanning 1982) and predicate logic (Bonczek et al. 1978), later work based on a construct called metagraphs (Basu and Blanning 1994a) has shown considerable promise. In addition to model composition (Basu and Blanning 1994b), the metagraph construct enables the representation of and reasoning with metadata such as assumptions associated with models (Basu, Blanning and Shtub, 1998). Work within the last ten years has focused on automating model composition and execution process, and combining partial solutions from multiple composite models and databases as in Chari (2002), leveraging XML in model composition (Bhrammanee and Wuwongse 2008) and implementing model composition through a sequence of web service invocations as in the WEBOPT project (Valente and Mitra 2007), and in (Madhusudan 2007).

Model integration differs from model composition in allowing modifications to be made to the models being integrated. Model integration involves both schema integration and solver integration (Dolk and Kotteman 1993). Schema integration is the task of merging the internal structure of two or more models to create a new model, while process integration is the task of interweaving associated solution processes in order to solve the integrated model.

Support for conflict resolution is a major focus of research in schema integration. This has involved the development of a variety of typing schemes that seek to integrate data typing (Muhanna 1992), and concepts such as quiddity and dimensions (Bhargava et al. 1991).

Detailed procedures for integrating models specified in the Structured Modeling Language (SML) (Geoffrion 1992a, b) have been proposed (Geoffrion 1989) and extended (Tsai 1998). The method uses to advantage the ability of structured modeling to trace the effects of changes and the formal definition of what constitutes a structured model. An update to structured modeling research is given in Geoffrion (1999a).

The pioneering work on solver integration is the work of Dolk and Kotteman (1993). They used the theory of communicating sequential processes (Hoare 1985) to address the problem of solver integration. A simplified version of the problem was addressed by Muhanna (1992) in the SYMMS system. As software components have emerged as a viable technology for web-based deployment of solvers on the Web, recent work has studied integration of solvers/methods on the Web (Guenther et al. 1997). Technology has made it possible to wrap a solver with a software layer that exposes standard interfaces thereby enabling multiple solvers to be invoked in a standard manner as in the case of Open Solver Interface (OSI) in the COIN-OR repository (Saltzman 2002).

Concluding Remarks

Research in the general area of model management since 2000 has contributed to (1) the extension of modeling languages to represent a variety of models; (2) the development of distributed model management systems using web technologies to support models as services; (3) the automation of model composition process; and (4) the integration of modeling languages and systems with databases. Among the numerous surveys that have been published on the subject, the model management chapter in the book on information systems and decision processes (Stohr and Konsynski 1992), the special issue of *Decision Support Systems* edited by Blanning (1993), and the special issue of the *Annals of Operations Research* edited by Shetty (1992) deserve special mention for their broad coverage of issues and their quality of exposition. A survey of the model management literature may be found in Krishnan and Chari (2000). A survey of model management issues pertaining to data mining models can be found in Liu and Tuzhilin (2008).

See

- ▶ Algebraic Modeling Languages for Optimization
- ▶ Decision Support Systems (DSS)
- ▶ Structured Modeling
- ▶ Verification, Validation, and Testing of Models

References

Banerjee, S., & Basu, A. (1993). Model type selection in an integrated DSS environment. *Decision Support Systems, 9*, 75–89.

Basu, A., & Blanning, R. (1994a). Metagraphs: A tool for modeling decision support systems. *Management Science, 40*, 1579–1600.

Basu, A., & Blanning, R. (1994b). Model integration using metagraphs. *Information Systems Research, 5*, 195–218.

Basu, A., & Blanning, R. (1998). The analysis of assumptions in model bases using metagraphs. *Management Science, 44*, 982–995.

Basu, A., Blanning, R., & Shtub, A. (1997). Metagraphs in hierarchical modeling. *Management Science, 43*, 623–639.

Bhargava, H. K., & Kimbrough, S. O. (1993). Model management: An embedded languages approach. *Decision Support Systems, 10*, 277–300.

Bhargava, H. K., Kimbrough, S., & Krishnan, R. (1991). Unique names violations: A problem for model integration. *ORSA Journal on Computing, 3*, 107–120.

Bhargava, H. K., & Krishnan, R. (1993). Computer aided model construction. *Decision Support Systems, 9*, 91–111.

Bhargava, H. K., & Krishnan, R. (1998). The World Wide Web and its implications for OR/MS. *INFORMS Journal on Computing, 10*, 359–383.

Bhargava, H. K., Krishnan, R., & Piela, P. (1997). On formal semantics and analysis of typed modeling languages. *INFORMS Journal on Computing, 10*, 189–208.

Bhargava, H. K., Power, D. J., & Sun, D. (2007). Progress in web-based decision support technologies. *Decision Support Systems, 43*, 1083–1095.

Bhrammanee, T., & Wuwongse, V. (2008). ODDM: A framework for modelbases. *Decision Support Systems, 44*, 689–709.

Bischop, J., & Meeraus, A. (1982). On the development of a general algebraic modeling system in a strategic planning environment. *Mathematical Programming Study, 20*, 1–29.

Blanning, R. (1982). A relational framework for model management, *DSS-82 Transaction*, 16–28.

Blanning, R. (1993). Decision support systems: *Special issue on model management*. In R. Blanning., C. Holsapple., & A. Whinston (Eds.). Elsevier.

Bonczek, R., Holsapple, C., & Whinston, A. (1978). Mathematical programming within the context of a generalized data base management system. *R.A.I.R.O. Recherche Operationalle, 12*, 117–139.

Bradley, G., & Clemence, R. (1987). A type calculus for executable modeling languages. *IMA Journal on Mathematics in Management, 1*, 277–291.

Chari, K. (2002). Model composition using filter spaces. *Information Systems Research, 13*(1), 15–35.

Chari, K., & Sen, T. K. (1997). An integrated modeling system for structured modeling using model graphs. *INFORMS Journal on Computing, 9*(4), 397–416.

Choobineh, J. (1991). SQLMP: A data sublanguage for the representation and formulation of linear mathematical models. *ORSA Journal on Computing, 3*, 358–375.

Dhar, V., & Jarke, M. (1993). On modeling processes. *Decision Support Systems, 9*, 39–49.

Dolk, D. K., & Kotteman, J. E. (1993). Model integration and a theory of models. *Decision Support Systems, 9*, 51–63.

Fourer, R. (1983). Modeling languages versus matrix generators for linear programming. *ACM Transactions on Mathematical Software, 2*, 143–183.

Fourer, R., & Gay, D. (2002). Extending an algebraic modeling language to support constraint programming. *INFORMS Journal on Computing, 14*(4), 332–344.

Fourer, R., Gay, D., & Kernighan, B. W. (1990). A mathematical programming language. *Management Science, 36*, 519–554.

Gassmann, H. I., & Ireland, A. M. (1996). On the formulation of stochastic linear programs using algebraic modeling languages. *Annals of Operations Research, 64*, 83–112.

Geoffrion, A. M. (1987). An introduction to structured modeling. *Management Science, 33*, 547–588.

Geoffrion, A. M. (1989). Reusing structured models via model integration. In J. F. Nunamaker (Ed.), *Proceedings of Twenty-Second Annual Hawaii International Conference on the System Sciences*, III, (pp. 601–611). Los Alamitos, California: IEEE Press.

Geoffrion, A. M. (1992a). The SML language for structured modeling: Levels 1 and 2. *Operations Research, 40*, 38–57.

Geoffrion, A. M. (1992b). The SML language for structured modeling: Levels 3 and 4. *Operations Research, 40*, 58–75.

Geoffrion, A. M. (1999a). An informal annotated bibliography on structured modeling. *Interactive Transactions OR/MS, 1(2)*, online at http://catt.bus.okstate.edu/ITORMS/.

Geoffrion, A. M. (1999b). Structured modeling: Survey and future research directions. *Interactive Transactions OR/MS, 1(3)*, online at http://catt.bus.okstate.edu/ITORMS/.

Gray, P. (1987). *Guide to IFPS*. New York: McGraw-Hill.

Greenberg, H. J. (1987). ANALYZE: A computer-assisted analysis system for linear programming models. *Operations Research Letters, 6*, 249–255.

Greenberg, H. J. (1992). MODLER: Modeling by object-driven linear elemental relations. *Annals of Operations Research, 38*, 239–280.

Guenther, O., Muller, R., Schmidt, P., Bhargava, H. K., & Krishnan, R. (1997). MMM: A WWW-based method management system for using software modules remotely. *IEEE Internet Computing, 1*(5), 59–68.

Guieu, O., & Chinneck, J. W. (1999). Analyzing infeasible mixed-integer and integer linear programs. *INFORMS Journal on Computing, 11*, 63–77.

Guntzer, U., Muller, R., Muller, S., & Schimkat, R. (2007). Retrieval for decision support resources by structured models. *Decision Support Systems, 43*, 1117–1132.

Hoare, (1985). Communicating sequential processes. Prentice-Hall, Inc. Upper Saddle River, NJ, USA

Jones, C. V. (1990). An introduction to graph based modeling systems, part I: Overview. *ORSA Journal on Computing, 2*, 136–151.
Jones, C. V. (1991). An introduction to graph based modeling systems, part II: Graph grammars and the implementation. *ORSA Journal on Computing, 3*, 180–206.
Kimbrough, S., & Oliver, J. (1994). On automating candle lighting analysis: Insight from search with genetic algorithms and approximate models. In J. F. Nunamaker (Ed.), *Proceedings of the Twenty Seventh Hawaii International Conference on the System Sciences*, III (pp. 536–544). Los Alamitos, California: IEEE Press.
Krishnan, R. (1990). A logic modeling language for model construction. *Decision Support Systems, 6*, 123–152.
Krishnan, R., & Chari, K. (2000). Model management: Survey, future research directions and a bibliography. *Interactive Transactions of ORMS, 3*(1).
Krishnan, R., Li, X., & Steier, D. (1992). Development of a knowledge-based model formulation system. *Communications of the ACM, 35*, 138–146.
Krishnan, R., Piela, P., & Westerberg, A. (1993). Reusing mathematical models in ASCEND. In C. Holsapple & A. Whinston (Eds.), *Advances in decision support systems* (pp. 275–294). Munich, Germany: Springer-Verlag.
Liang, T. P., & Konsynski, B. R. (1993). Modeling by analogy: Use of analogical reasoning in model management systems. *Decision Support Systems, 9*, 113–125.
Liu, B., & Tuzhilin, A. (2008). Managing large collection of data mining models. *Communications of the ACM, 51*(2), 85–89.
Ma, P.-C., Murphy, F., & Stohr, E. (1989). A graphics interface for linear programming. *Communications of the ACM, 32*, 996–1012.
Madhusudan, T. (2007). Web services framework for distributed model management. *Information System Frontiers, 9*, 9–27.
Mannino, M. V., Greenberg, B. S., & Hong, S. N. (1990). Model libraries: Knowledge representation and reasoning. *ORSA Journal on Computing, 2*, 287–301.
Mills, H., Linger, R., & Hevner, A. (1986). *Principles of information systems analysis and design*. Orlando, FL: Academic Press.
Muhanna, W. (1992). On the organization of large shared of model bases. *Annals of Operations Research, 38*, 359–396.
Murphy, F., & Stohr, E. (1986). An intelligent system for formulating linear programs. *Decision Support Systems, 2*, 39–47.
Murphy, F., Stohr, E. A., & Asthana, A. (1992). Representation schemes for mathematical programming models. *Management Science, 38*, 964–991.
Piela, P., McKelvey, R., & Westerberg, A. (1992). An introduction to ASCEND: Its language and interactive environment. In J. F. Nunamaker Jr (Ed.), *Proceedings of the Twenty-Fifth Annual Hawaii International Conference on System Sciences*, Vol. III (pp. 449–461). Los Alamitos, California: IEEE Press.
Raghunathan, S., Krishnan, R., & May, J. (1994). MODFORM: A knowledge tool to support the modeling process. *Information Systems Research, 4*, 331–358.
Raghunathan, S., Krishnan, R., & May, J. (1995). On using belief maintenance systems to assist mathematical modeling. *IEEE Transactions on Systems, Man, and Cybernetics, 25*, 287–303.
Saltzman, M. J. (2002). COIN-OR: An open-source library for optimization. In S. S. Nielsen (Ed.), *Programming languages and systems in computational economics and finance*. Boston: Kluwer Academic Publishers.
Sharda, R., & Rampal, G. (1995). Algebraic Modeling Languages on PCs. *OR/MS Today, 22*(3), 58–63.
Sharda, R., & Steiger, D. (1996). Inductive model analysis systems: Enhancing model analysis in decision support systems. *Information Systems Research, 7*, 328–341.
Shetty, B. (1992). Annals of operations research: *Special issue on model management in operations research*. In B. Shetty (Ed.). Amsterdam: J.C. Baltzer Scientific Publishing.
Sprague, R. H., & Watson, H. J. (1975). Model management in MIS. *Proceedings of Seventeenth National AIDS Conference*, 213–215.
Stohr, E., & Konsynski, B. (1992). *Information systems and decision processes*. Los Altimos, CA: IEEE Press.
Tsai, Y.-C. (1998). Model integration using SML. *Decision Support Systems, 22*, 355–377.
Valente, P., & Mitra, G. (2007). The evolution of web-based optimization: From ASP to e-Services. *Decision Support Systems, 43*, 1096–1116.
Valente, P., Mitra, G., Sadki, M., & Fourer, R. (2009). Extending algebraic modelling languages for stochastic programming. *INFORMS Journal on Computing, 21*, 1, 107–122.6.
Will, H. J. (1975). Model management systems. In E. Grochia & N. Szyperski (Eds.), *Information systems and organization structure* (pp. 468–482). Berlin, Germany: Walter de Gruyter.

Model Testing

Investigating whether inaccuracies or errors exist in a model.

See

- ▶ Validation
- ▶ Verification
- ▶ Verification, Validation, and Testing of Models

Model User's Risk

Probability of accepting the credibility of a model when in fact the model is not sufficiently credible.

Model Validation

- ▶ Validation
- ▶ Verification
- ▶ Verification, Validation, and Testing of Models

Model Verification

- ▶ Validation
- ▶ Verification
- ▶ Verification, Validation, and Testing of Models

Model-based Search Methods

A class of global optimization methods that uses a probability distribution to generate candidate solutions, where in each iteration of the algorithm, the probability distribution is updated according to the performance of the population of candidate solutions. Examples include estimation of distribution algorithms, the cross-entropy method, and model reference adaptive search.

See

- ▶ Cross-Entropy Method

References

Larrañaga, P., & Lozano, J. A. (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Boston: Kluwer Academic.

MODI

Modified Distribution Method. A procedure for organizing the hand computations when solving a transportation problem using the transportation simplex method.

See

- ▶ Transportation Simplex (Primal-Dual) Method

MOIP

Multi-objective integer programming.

See

- ▶ Multiple Criteria Decision Making

MOLP

Multi-objective linear programming.

See

- ▶ Multiobjective Programming

Moment Generating Function

For a random variable X, the moment generating function is given by $M_X(t) = E[e^{tX}]$, assuming the expectation exists. For non-negative continuous random variables, it is basically identical to the Laplace transform for the corresponding probability density function.

Monte Carlo Methods

General term used to refer to the use of random numbers in a particular methodology, e.g., evaluating a high-dimensional deterministic integral or carrying out a randomized algorithm or simulation of a stochastic system, all based on statistical sampling techniques. The term "Monte Carlo" signifies the random or uncertain component that characterizes the

method and was coined in the 1940s by physicists working on the Manhattan nuclear weapons project, an allusion to gambling in Monte Carlo casinos. One of the strengths of the Monte Carlo method is that in many applications its computational burden grows only linearly in the dimension of problems where other methods suffer from an exponential (geometric) growth in computation.

See

- ▶ Las Vegas Algorithm
- ▶ Monte Carlo Simulation
- ▶ Randomized Algorithm
- ▶ Simulation of Stochastic Discrete-Event Systems

References

Fishman, G. S. (1996). *Monte Carlo: Concepts, algorithms, and applications*. New York: Springer.
Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association, 44*(247), 335–341.

Monte Carlo Simulation

Simulation of systems modeled using random variables and/or stochastic processes. The underlying inputs are generally random numbers, sequences of independent and identically distributed random variables uniformly distributed on the unit interval. Sometimes called the Monte Carlo method, where the term "Monte Carlo" signifies the random or uncertain component that characterizes the method and was coined in the 1940s by physicists working on the Manhattan nuclear weapons project, an allusion to gambling in Monte Carlo casinos. Monte Carlo simulation is one of the most widely used tools in operations research and management science (OR/MS) and can be used to provide detailed models of complex systems arising in various OR/MS fields from manufacturing to transportation to computer/communications networks to financial engineering. One of the strengths of the Monte Carlo method is that in many applications its computational burden grows only linearly in the dimension of problems where other methods suffer from an exponential (geometric) growth in computation.

See

- ▶ Simulation of Stochastic Discrete-Event Systems
- ▶ Simulation Optimization
- ▶ Variance Reduction Techniques in Monte Carlo Methods

References

Fishman, G. (2010). *Monte Carlo: Concepts, algorithms, and applications* (4th ed.). New York: Springer.
Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association, 44*(247), 335–341.
Rubinstein, R. Y., & Kroese, D. P. (2007). *Simulation and the Monte Carlo method* (2nd ed.). New York: Wiley-Interscience.

MOR

Military operations research; also used as an abbreviation for the journal *Mathematics of Operations Research*.

See

- ▶ Military Operations Research

Moral Hazard

A term in economics describing a situation in which a decision maker's actions are taken without bearing full risk, responsibility, or consequences for the potential outcomes. For example, having a valuable item with full insurance coverage against theft might make the owner more lax in safeguarding it.

Economist Paul Krugman described moral hazard as: "...any situation in which one person makes the decision about how much risk to take, while someone else bears the cost if things go badly."

References

Krugman, P. (2009). *The return of depression economics and the crisis of 2008*. New York: W.W. Norton.

MORS

Military Operations Research Society.

See

▶ Military Operations Research

MPS

▶ Mathematical-Programming System (MPS)

MRP

▶ Material Requirements Planning

MS

Management Science

MSE

Mean square error.

Multicommodity Network Flows

Bala Shetty
Texas A&M University, College Station, TX, USA

Introduction

The multicommodity minimal cost network flow problem may be described in terms of a distribution problem over a network $[V, E]$, where V is the node set with order n and E is the arc set with order m. The decision variable x^{jk} denotes the flow of commodity k through arc j, and the vector of all flows of commodity k is denoted by $\boldsymbol{x}^k = [x^{1k},\ldots, x^{mk}]$. The unit cost of flow of commodity k through arc j is denoted by c^{jk} and the corresponding vector of costs by $\boldsymbol{c}^k = [c^{1k},\ldots, c^{mk}]$. The total capacity of arc j is denoted by b^j with corresponding vector $\boldsymbol{b} = [b^1,\ldots, b^m]$. Mathematically, the multicommodity minimal cost network flow problem may be defined as follows:

$$\text{Minimize} \quad \sum_k \boldsymbol{c}^k \boldsymbol{x}^k$$

s.t.

$$\boldsymbol{A}\boldsymbol{x}^k = \boldsymbol{r}^k, \quad k = 1,\ldots, K$$

$$\sum_k \boldsymbol{x}^k \le \boldsymbol{b}$$

$$0 \le \boldsymbol{x}^k \le \boldsymbol{u}^k, \quad \text{for all } k,$$

where K denotes the number of commodities, $\boldsymbol{A}$ is a node-arc incidence matrix for $[V, E]$, $\boldsymbol{r}^k$ is the requirements vector for commodity k, and $\boldsymbol{u}^k$ is the vector of upper bounds for decision variable $\boldsymbol{x}^k$.

Multicommodity network flow problems are extensively studied because of their numerous applications and because of the intriguing network structure exhibited by these problems (Ahuja et al. 1993; Ali et al. 1984; Assad 1978; Castro and Nabona 1996; Kennington 1978; McBride 1998). Multicommodity models have been proposed for planning studies involving urban traffic systems (Chen and Meyer 1988; LeBlanc 1973; Potts and Oliver 1972) and communications systems (LeBlanc 1973; Naniwada 1969). Models for solving scheduling and routing problems have been proposed by Bellmore et al. (1971) and by Swoveland (1971). A multicommodity model for assigning students to achieve a desired ethnic composition was suggested by Clark and Surkis (1968). Multicommodity models have also been used for casualty evacuation of war time casualties, grain transportation, and aircraft routing for the USAF. A discussion of these applications can be found in Ali et al. (1984). Additional applications of multicommodity flows are given in Gautier and Granot (1995), and Popken (1994).

Solution Techniques

There are two basic approaches which have been employed to develop specialized techniques for multicommodity network flow problems: decomposition and partitioning. Decomposition approaches may be further characterized as price-directive or resource directive. A price-directive decomposition procedure directs the coordination between a master program and each of several subprograms by the changing the objective functions (prices) of the subprograms. The objective is to obtain a set of prices (dual variables) such that the combined solution for all subproblems yields an optimum for the original problem. A resource-directive decomposition procedure (Held et al. 1974; Kennington and Shalaby 1977), when applied to a multicommodity problem having K commodities, is to distribute the arc capacity among the individual commodities in such a way that solving K sub-programs yields an optimal flow for the coupled problem. At each iteration, an allocation is made and K single commodity flow problems are solved. The sum of capacities allocated to an arc over all commodities is equal to the arc capacity in the original problem. Hence, the combined flow from the solutions of the subproblems provides a feasible flow for the original problem. Optimality is tested and the procedure either terminates or a new arc capacity allocation is developed. Partitioning approaches are specializations of the simplex method where the current basis is partitioned to exploit its special structure. These techniques are specializations of primal, dual, or primal-dual simplex method. The papers of Hartman and Lasdon (1972), and Graves and McBride (1976) are primal techniques, while the work of Grigoriadis and White (1972) is a dual technique. An extensive discussion of these techniques can be found in Ahuja et al. (1993) and Kennington and Helgason (1980).

Several researchers have suggested algorithms for the multicommodity flow problem: Gersht and Shulman (1987), Barnhart (1993), Farvolden and Powell (1990), Farvolden et al. (1993), Liu (1997), and Schneur and Orlin (1998) all present alternative approaches for the multicommodity model. Parallel optimization has also been applied for the solution of multicommodity networks. Pinar and Zenios (1990) present a parallel decomposition algorithm for the multicommodity model using penalty functions. Shetty and Muthukrishnan (1990) develop a parallel projection which can be applied to resource-directive decomposition. Chen and Meyer (1988) decompose a nonlinear multicommodity problem arising in traffic assignment into single commodity network components that are independent by commodity. The difficulty of solving a multicommodity problem explodes when the decision variables are restricted to be integers. Very little work is available in the literature for the integer problem (Evans 1978; Evans and Jarvis 1978; Gendron and Crainic 1997).

Several computational studies involving multicommodity models have been reported in the literature. Ali et al. (1980) present a computational experience using the price-directive decomposition procedure (PPD), the resource directive-decomposition procedure (RDD), and the primal partitioning procedure (PP). They find the primal partitioning and price directive decomposition methods take approximately the same amount of computing time, while the resource directive decomposition runs in approximately one-half the time of the other two methods. Convergence to the optimal solution is guaranteed for PPD and PP, whereas RDD may experience convergence problems. Ali et al. (1984) present a comparison of the primal partitioning algorithm for solving the multicommodity model with a general purpose LP code. On a set of test problems, they find that the primal partitioning technique runs in approximately one-half the time required by the LP code. Farvolden et al. (1993) report very promising computational results for a class of multicommodity network problems using a primal partitioning code (PPLP). On these problems, they find PPLP to be two orders of magnitude faster than MINOS and about 50 times faster than OB1, a state-of-the-art LP solver.

Linear, nonlinear, and integer multicommodity models have numerous important applications in scheduling, routing, transportation, and communications. Real-world multicommodity models tend to be very large and there is a need for faster and more efficient algorithms for solving these models.

Thus, multicommodity models present unlimited opportunities for future research in large-scale optimization.

See

- Large-Scale Systems
- Linear Programming
- Logistics and Supply Chain Management
- Minimum-Cost Network-Flow Problem
- Network Optimization
- Transportation Problem

References

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. New Jersey: Prentice Hall.

Ali, A., Barnett, D., Farhangian, K., Kennington, J., McCarl, B., Patty, B., Shetty, B., & Wong, P. (1984). Multicommodity network flow problems: Applications and computations. *IIE Transactions, 16*, 127–134.

Ali, A., Helgason, R., Kennington, J., & Lall, H. (1980). Computational comparison among three multicommodity network flow algorithms. *Operations Research, 28*, 995–1000.

Assad, A. A. (1978). Multicommodity network flows–A survey. *Networks, 8*, 37–91.

Barnhart, C. (1993). Dual ascent methods for large-scale multicommodity flow problems. *Naval Research Logistics, 40*, 305–324.

Bellmore, M., Bennington, G., & Lubore, S. (1971). A multivehicle tanker scheduling problem. *Transportation Science, 5*, 36–47.

Castro, J., & Nabona, N. (1996). An implementation of linear and nonlinear multicommodity network flows. *European Journal of Operational Research, 92*, 37–53.

Chen, R., & Meyer, R. (1988). Parallel optimization for traffic assignment. *Mathematical Programming, 42*, 327–345.

Clark, S., & Surkis, J. (1968). An operations research approach to racial desegregation of school systems. *Socio-Economic Planning Sciences, 1*, 259–272.

Evans, J. (1978). The simplex method for integral multicommodity networks. *Naval Research Logistics, 25*, 31–38.

Evans, J., & Jarvis, J. (1978). Network topology and integral multicommodity flow problems. *Networks, 8*, 107–120.

Farvolden, J. M., & Powell, W. B. (1990). *A primal partitioning solution for multicommodity network flow problems* (Working Paper 90–04). Canada: Department of Industrial Engineering, University of Toronto.

Farvolden, J. M., Powell, W. B., & Lustig, I. J. (1993). A primal partitioning solution for the arc-chain formulation of a multicommodity network flow problem. *Operations Research, 41*, 669–693.

Gautier, A., & Granot, F. (1995). Forest management: A multicommodity flow formulation and sensitivity analysis. *Management Science, 41*, 1654–1668.

Gendron, B., & Crainic, T. G. (1997). A parallel branch-and-bound algorithm for multicommodity location with balancing requirements? *Computers and Operations Research, 24*, 829–847.

Gersht, A., & Shulman, A. (1987). A new algorithm for the solution of the minimum cost multicommodity flow problem. *Proceedings of the IEEE Conference on Decision and Control, 26*, 748–758.

Graves, G. W., & McBride, R. D. (1976). The factorization approach to large scale linear programming. *Mathematical Programming, 10*, 91–110.

Grigoriadis, M. D., & White, W. W. (1972). A partitioning algorithm for the multicommodity network flow problem. *Mathematical Programming, 3*, 157–177.

Hartman, J. K., & Lasdon, L. S. (1972). A generalized upper bounding algorithm for multicommodity network flow problems. *Networks, 1*, 331–354.

Held, M., Wolfe, P., & Crowder, H. (1974). Validation of subgradient optimization. *Mathematical Programming, 6*, 62–88.

Kennington, J. L. (1978). A survey of linear cost multicommodity network flows. *Operations Research, 26*, 209–236.

Kennington, J. L., & Helgason, R. (1980). *Algorithms for network programming*. New York: Wiley.

Kennington, J., & Shalaby, M. (1977). An effective subgradient procedure for minimal cost multicommodity flow problems. *Management Science, 23*, 994–1004.

LeBlanc, L. J. (1973). *Mathematical programming algorithms for large scale network equilibrium and network design problems*. Unpublished Dissertation, Industrial Engineering and Management Sciences Department, Northwestern University.

Liu, C.-M. (1997). Network dual steepest-edge methods for solving capacitated multicommodity network problems. *Computers and Industrial Engineering, 33*, 697–700.

McBride, R. (1998). Advances in solving the multi-commodity-flow problem. *Interfaces, 28*(2), 32–41.

Naniwada, M. (1969). Multicommodity flows in a communications network. *Electronics and Communications in Japan, 52-A*, 34–41.

Pinar, M. C., & Zenios, S. A. (1990). *Parallel decomposition of multicommodity network flows using smooth penalty functions*. Technical Report 90-12-06, Department of Decision Sciences, Wharton School, University of Pennsylvania, Philadelphia.

Popken, D. A. (1994). An algorithm for the multiattribute, multicommodity flow problem with freight consolidation and inventory costs. *Operations Research, 42*, 274–286.

Potts, R. B., & Oliver, R. M. (1972). *Flows in transportation networks*. New York: Academic.

Schneur, R., & Orlin, J. B. (1998). A scaling algorithm for multicommodity flow problems. *Operations Research, 46*, 231–246.
Shetty, B., & Muthukrishnan, R. (1990). A parallel projection for the multicommodity network model. *Journal of Operational Research, 41*, 837–842.
Swoveland, C. (1971). *Decomposition algorithms for the multi-commodity distribution problem* (Working Paper, No. 184). Los Angeles: Western Management Science Institute, University of California.

Multicommodity Network-Flow Problem

A minimum-cost network flow problem in which more than one commodity simultaneously flows from the supply nodes to the demand nodes. Unlike the single commodity problem, an optimal solution is not guaranteed to have integer flows. The problem takes on the block-angular matrix form that is suitable for solution by Dantzig-Wolfe decomposition. Applications areas include communications, traffic and logistics.

See

- ▶ Dantzig-Wolfe Decomposition Algorithm
- ▶ Minimum-Cost Network-Flow Problem
- ▶ Multicommodity Network Flows
- ▶ Network Optimization

Multidimensional Transportation Problem

Usually a transportation problem with a third index that refers to a product type available at the origins and demanded at the destinations. The variables x_{ijk} represent the amount of the kth product type shipped from the ith origin to the jth destination. The constraint set is a set of linear balance equations, with the usual linear cost objective function. It is also a special form of the multicommodity network-flow problem. Unlike the transportation problem, its optimal solution may not be integer-valued even if the network data are given as integers. The problem can also be defined with more than three indices.

See

- ▶ Multicommodity Network Flows
- ▶ Transportation Problem

Multiobjective Linear-Programming Problem

This problem has the usual set of linear-programming constraints ($\boldsymbol{Ax} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$) but requires the simultaneous optimization of more than one linear objective function, say p of them. It can be written as "Maximize" $\boldsymbol{Cx}$ subject to $\boldsymbol{Ax} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$, where $\boldsymbol{C}$ is a $p \times n$ matrix whose rows are the coefficients defined by the p objectives. Here "Maximize" represents the fact that it is usually impossible to find a solution to $\boldsymbol{Ax} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$, that simultaneously optimizes all the objectives. If there is such an (extreme) point, the problem is thus readily solved. Special multiobjective computational procedures are required to select a solution that is in effect a compromise solution between the extreme point solutions that optimize individual objective functions. The possible compromise solutions are taken from the set of efficient (nondominated) solutions. This problem is also called the vector optimization problem.

See

- ▶ Efficient Solution
- ▶ Multiobjective Programming
- ▶ Pareto-Optimal Solution

Multiobjective Programming

Ralph E. Steuer
University of Georgia, Athens, GA, USA

Introduction

Related to linear, integer, and nonlinear programming, multiobjective programming addresses the extensions to theory and practice of mathematical programming

problems with more than one objective function. Single objective programming must settle on a single objective such as to maximize profit or minimize cost. However, many if not most real-world problems are in an environment of multiple conflicting criteria. A sample of problems modeled with multiple objectives:

Oil Refinery Scheduling

min {cost}
min {imported crude}
min {high sulfur crude}
min {deviations from demand slate}

Production Planning

max {total net revenue}
max {minimum net revenue in any period}
min {backorders}
min {overtime}
min {finished goods inventory}

Forest Management

max {timber production}
max {visitor days of recreation}
max {wildlife habitat}
min {overdeviations from budget}

Emerging as a new topic in the 1970s, multiobjective programming has grown to the extent that numerous books have been written on the subject (e.g., Zeleny 1982; Yu 1985; Steuer 1986; Miettinen 1999; Ehrgott 2005) and applications of multiobjective programming can now be found in virtually all areas of operational research.

Terminology

A multiobjective programming problem is the following:

$$\begin{aligned} &\text{maximize} \quad \{f_1(\boldsymbol{x}) = z_1\} \\ &\vdots \\ &\text{maximize} \quad \{f_k(\boldsymbol{x}) = z_k\} \\ &\text{subject to} \quad \boldsymbol{x} \in S \end{aligned}$$

where k is the number of objectives, the z_i are criterion values, and S is the feasible region in decision space. Let $Z \subset R^k$ be the feasible region in criterion space where $\boldsymbol{z} \in Z$ if and only if there exists an $\boldsymbol{x} \in S$ such that $\boldsymbol{z} = (f_1(\boldsymbol{x}),\ldots,f_k(\boldsymbol{x}))$. Let $K = \{1,\ldots,k\}$. Criterion vector $\overline{z} \in Z$ is nondominated if and only if there does not exist another $\boldsymbol{z} \in Z$ such that $z_i \geq \overline{z}_i$ for all $i \in K$ and $z_i > \overline{z}_i$ for at least one $i \in K$. The set of all nondominated criterion vectors is designated N and is called the nondominated set. A point $\overline{\mathbf{x}} \in S$ is efficient if and only if its criterion vector $\overline{z} = (f_1(\overline{\boldsymbol{x}}),\ldots,f_k(\overline{\boldsymbol{x}})$ is nondominated. The set of all efficient points is designated E and is called the efficient set.

Let $U: R^k \rightarrow R$ be the utility function of the decision maker (DM). A $\boldsymbol{z}^\circ \in Z$ that maximizes U over Z is an optimal criterion vector and any $\boldsymbol{x}^\circ \in S$ such that $(f_1(\boldsymbol{x}^\circ),\ldots, f_k(\boldsymbol{x}^\circ)) = \boldsymbol{z}^\circ$ is an optimal solution of the multiobjective program. The interest in the efficient set E and the nondominated set N stems from the fact that if U is coordinatewise increasing (i.e., more is always better than less of each objective), $\boldsymbol{x}^\circ \in E$ and $\boldsymbol{z}^\circ \in N$. In this way, a multiobjective program can be solved by finding the most preferred criterion vector in N.

One might think that the best way to solve a multiobjective program would be to assess the DM's utility function and then solve

$$\begin{aligned} &\text{maximize} \quad \{U(z_1,\ldots,z_k)\} \\ &\text{subject to} \quad f_i(\boldsymbol{x}) = z_i, \quad i \in K,\ x \in S \end{aligned}$$

because any solution that solves this program is an optimal solution of the multiobjective program. However, multiobjective programs are usually not solved in this way because (1) of the difficulty in assessing an accurate enough U, (2) U would almost certainly be nonlinear, and (3) the DM would not likely see other candidate solutions during the solution process from which to gain an appreciation of the tradeoffs inherent in the problem.

Consequently, multiobjective programming employs mostly interactive procedures that only require implicit, as opposed to explicit, knowledge about the DM's utility function. In interactive procedures, the goal is to search the nondominated set for the DM's most preferred criterion vector. Unfortunately, because of the size of N, finding the best criterion vector in N is not a trivial task. As a result, interactive procedures are carefully crafted and can generally only be expected to conclude with what is called a final solution, a solution that is either optimal or close enough to being optimal to satisfactorily terminate the decision process.

Background Concepts

Along with the basics of conventional mathematical programming, multiobjective programming requires additional concepts not widely employed elsewhere in operations research. The key ones are as follows.

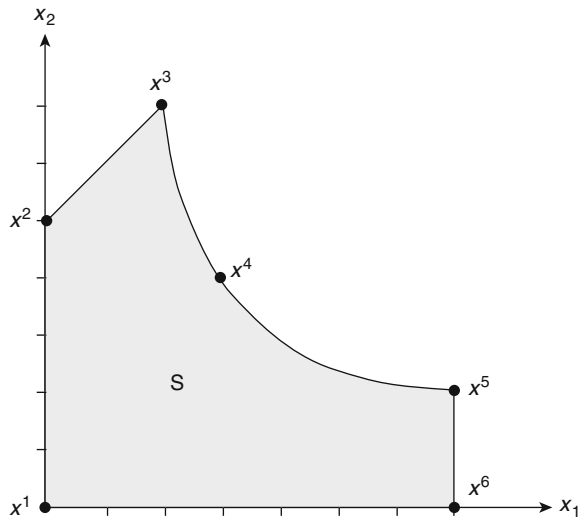
1. *Decision Space vs. Criterion Space*. Whereas single objective programming is typically studied in decision space, multiobjective programming is mostly studied in criterion space. To illustrate, consider

$$\begin{aligned} &\text{maximize} && \{x_1 - 1/2x_2 = z_1\} \\ &\text{maximize} && \{x_2 = z_2\} \\ &\text{subject to} && x \in S \end{aligned}$$

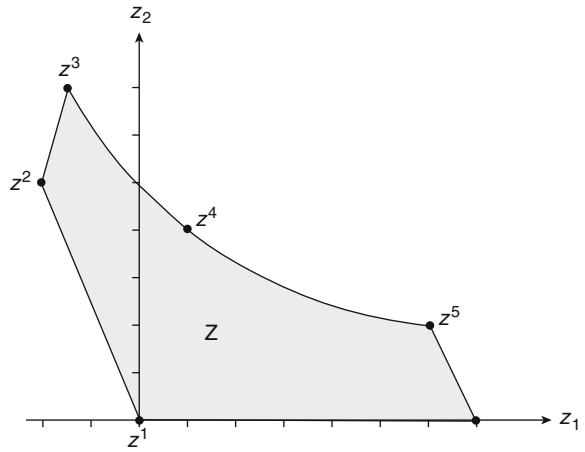
where S in decision space is in Fig. 1, and Z in criterion space is in Fig. 2. For instance z^4, which is the image of $\boldsymbol{x}^4 = (3, 4)$, is obtained by plugging the point (3, 4) into the objective functions to generate $z^4 = (1, 4)$. In Fig. 2, the nondominated set N is the set of boundary criterion vectors z^3 through z^4 to z^5 to z^6, inclusive. In Fig. 1, the efficient set E is the set of inverse images of the criterion vectors in N, namely the set of boundary points $\boldsymbol{x}^3$ through $\boldsymbol{x}^4$ to $\boldsymbol{x}^5$ to $\boldsymbol{x}^6$, inclusive. Note that Z is not necessarily confined to the nonnegative orthant.

2. *Unsupported Nondominated Criterion Vectors*. A $z \in N$ is unsupported if and only if it is possible to dominate it by a convex combination of other nondominated criterion vectors. In Fig. 2, the set of unsupported nondominated criterion vectors is the set of criterion vectors from z^3 through z^4 to z^5, exclusive of z^3 and z^5. The set of supported nondominated criterion vectors is the set that consists of z^3 plus the line segment z^5 to z^6, inclusive. Unsupported nondominated criterion vectors can only occur in problems that possess non-convex feasible regions; hence, they can easily be present in integer and nonlinear multiobjective programs.

3. *Identifying Nondominated Criterion Vectors*. To graphically determine whether a $\overline{z} \in Z$ is non-dominated or not, visualize the nonnegative orthant in R^k translated so that its origin is at $\overline{z}$. Note that, apart from $\overline{z}$, a vector dominates $\overline{z}$ if and only if the vector is in the translated nonnegative orthant. In other words, $\overline{z}$ is nondominated if and only if the translated nonnegative orthant is empty of feasible criterion vectors other than for $\overline{z}$. Visualizing in Fig. 2 the nonnegative orthant translated to z^4, it can be seen that z^4 is nondominated. Visualizing the nonnegative orthant translated to z^2, it can be seen that z^2 is dominated.



Multiobjective Programming, Fig. 1 Representation in decision space



Multiobjective Programming, Fig. 2 Representation in criterion space

4. *Payoff Tables*. Assuming that each objective is bounded over the feasible region, a payoff table is of the form

	z_1	z_2		z_k
z^1	z_1^*	z_{12}		z_{1k}
z^2	z_{21}	z_2^*		z_{2k}
z^k	z_{k1}	z_{k2}		z_k^*

where the rows are criterion vectors resulting from individually maximizing the objectives. For instance, z_{12} is the value of the second objective function at the point that maximizes the first objective. The z_i^* entries along the main diagonal of the payoff table are the maximum criterion values of the different objectives over the nondominated set. The minimum value in the ith column of the payoff table is often used as an estimate of the minimum criterion value of the ith objective over N because the true minimum criterion values over N (called nadir values) are typically difficult to obtain (Isermann and Steuer 1988; Alves and Costa 2009)

5. z^{**} *Reference Criterion Vectors*. A $z^{**} \in R^k$ *reference* criterion vector is a criterion vector that is suspended above the nondominated set. Its components are given by

$$z_i^{**} = z_i^* + \in_i$$

where the $\in_i$ are small computationally significant positive values.

6. *Weighting Vector Space*. Without loss of generality, let

$$\Lambda = \left\{ \lambda \in R^k | \lambda_i \in (0,1), \sum_{i \in k} \lambda_i = 1 \right\}$$

be weighting vector space. In an interactive environment, subsets of Λ called interval defined subsets are of the form

$$\Lambda^{(h)} = \left\{ \lambda \in R^k | \lambda_i \in \left(\ell_i^{(h)}, \mu_i^{(h)} \right), \sum_{i \in k} \lambda_i = 1 \right\}$$

where h is the iteration number and

$$0 \le l_i^{(h)} \le \mu_i^{(h)} \le 1 \quad i \in K$$
$$\mu_i^{(h)} - l_i^{(h)} = \mu_j^{(h)} - l_j^{(h)} \quad \text{for all } i \ne j$$

Sequences of successively smaller interval subsets can be defined by reducing the $\mu_i^{(h)} - l_i^{(h)}$ interval widths at each iteration.

7. *Sampling Programs*. The weighted-sums program

$$\max \left\{ \sum_{i \in K} \lambda_i f_i(x) | x \in S \right\}$$

can be used to sample the nondominated set because, as long as $\lambda \in \Lambda$, the program returns an efficient point. A disadvantage of the weighted-sums program is that it cannot generate unsupported points.

To make downward probes of the nondominated set from a z^{**} as required in many of the interactive procedures of multiobjective programming, the augmented Tchebycheff program is employed

$$\text{minimize} \left\{ \alpha - \rho \sum_{i \in K} z_i \right\}$$
$$\text{subject to}$$
$$\alpha \ge \lambda_i \left(z_i^{**} - z_i \right) \quad i \in K$$
$$f_i(\boldsymbol{x}) = \boldsymbol{z_i} \quad \boldsymbol{i \in K}$$
$$\boldsymbol{x} \in S$$
$$z \in R^k \text{ unrestricted}$$

where $\alpha \in R$, $\lambda \in \Lambda$, and ρ is a small computationally significant positive number. A disadvantage of the augmented Tchebycheff program is that, regardless of the value of ρ, there may still remain unsupported members of the nondominated set that the program is unable to compute (Steuer 1986).

A program that has better mathematical properties, although somewhat more difficult to implement, is the lexicographic Tchebycheff program

$$\text{lex}\ \min\left\{\alpha, -\sum_{i\in K} z_i\right\}$$
$$\text{subject to}$$
$$\alpha \geq \lambda_i\left(z_i^{**} - z_i\right) \quad i \in K$$
$$f_i(\boldsymbol{x}) = z_i \quad i \in K$$
$$\boldsymbol{x} \in S$$
$$z \in R^k \text{ unrestricted}$$

where $\lambda \in \Lambda$. At the first lexicographic level it is solved to minimize α. At the second lexicographic level, subject to only those solutions that minimize α, $-\sum_{i\in K} z_i$ is minimized. Not only does the lexicographic Tchebycheff program always return a nondominated criterion vector, but if $\boldsymbol{z}$ is nondominated, there then exists a $\overline{\lambda} \in \Lambda$ such that $\boldsymbol{z}$ uniquely solves the program (Steuer 1986).

8. *Aspiration Criterion Vectors*. An aspiration criterion vector $\boldsymbol{q} \in R^k$ is a criterion vector specified by a DM to reflect his or her hopes or expectations from a problem. An aspiration criterion vector, when specified, is typically projected onto N by an augmented or lexicographic Tchebycheff program in order to find the nondominated criterion vector closest to the aspiration criterion vector.
9. *T-vertex λ-vector Defined by* $\boldsymbol{q}$ *and* $\boldsymbol{z}^{**}$. The T-vertex (Tchebycheff-vertex) λ-vector defined by $\boldsymbol{q}$ and z^{**} is the $\lambda \in \Lambda$ whose components are given by

$$\lambda_i = \frac{1}{\left(z_i^{**} - q_i\right)}\left[\sum_{i\in K}\frac{1}{\left(z_j^{**} - q_j\right)}\right]^{-1}$$

The T-vertex λ-vector, when installed in an augmented or lexicographic Tchebycheff program, causes the program to probe the nondominated set along a line that goes through both z^{**} and $\boldsymbol{q}$ in the direction

$$-\left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k}\right)$$

Vector-Maximum Algorithms

In the linear case, a multiple objective linear program (MOLP) is sometimes written in vector-maximum form

Multiobjective Programming, Table 1 Average numbers of MOLP efficient extreme points

MOLP size $k \times m \times n$	Efficient extreme points	Approximate times in seconds
$3 \times 50 \times 75$	1,798	2
$3 \times 100 \times 150$	11,897	40
$3 \times 200 \times 300$	128,237	1,600
$4 \times 50 \times 75$	9,921	30
$4 \times 100 \times 150$	682,920	3,500
$5 \times 50 \times 75$	141,444	300

$$\text{“ max ”}, \{\boldsymbol{Cx} = \boldsymbol{z} | \boldsymbol{x} \in \boldsymbol{S}\}$$

where $\boldsymbol{C}$ is the $k \times n$ matrix whose rows are the coefficient vectors of the k objectives. A point is a solution to a vector-maximum problem if and only if it is efficient. Algorithms for characterizing the efficient set E of an MOLP are called vector-maximum algorithms. In the 1970s, considerable effort was spent on the development of vector-maximum codes to compute all efficient extreme points. The thought was that, by reviewing the list of nondominated criterion vectors associated with the efficient extreme points, a DM would be able to identify his or her efficient extreme point of greatest utility in hopes of satisfactorily terminating the decision process.

Unfortunately, MOLPs have many efficient extreme points as indicted in Table 1 (sample size of ten for each problems size). Whereas the number of variables and the number of constraints play a role, the factor most dramatically affecting the number of efficient extreme points is the dimensionality of the criterion cone, the convex cone generated by the gradients of the k objective functions.

With nondominated sets of sizes indicated in Table 1, other approaches have been attempted such as by Klamroth, Tind and Wiecek 2002, but mostly, the figures have led to interactive procedures moving to the forefront of multiobjective programming.

Interactive Procedures

In interactive multiobjective programming, an exploration over the feasible region for the best point in the non-dominated set is conducted. Interactive

procedures are characterized by phases of decision making alternating with phases of computation. A pattern is generally established and kept repeating it until termination. At each iteration, a solution, or a group of solutions, is generated for examination. As a result of the examination, the DM inputs updated preference information to the solution procedure in the form of values of the controlling parameters (preference weights, aspiration criterion vectors, λ-vector interval widths, criterion vector components to be increased/decreased/held fixed, criterion vector lower bounds, etc., depending upon the particular interactive procedure).

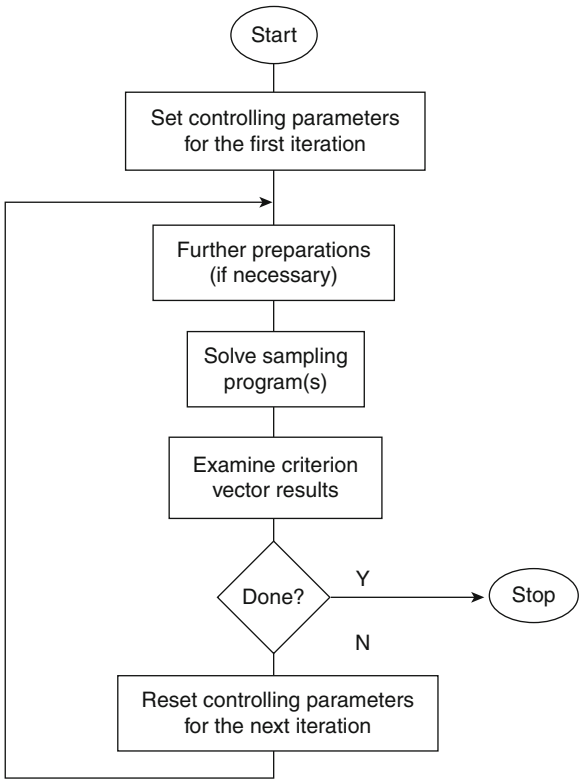
While many interactive procedures have been proposed, virtually all of them more or less follow the same general algorithmic outline. As portrayed in Fig. 3, the general algorithmic outline includes:

- an initial setting of the controlling parameters;
- optimization of one or more mathematical programming problems to probe (i.e., sample) the nondominated set;
- examination of the criterion vector results; and
- a resetting of the controlling parameters for the next iteration in the light of what was learned on the current iteration

With the consensus being that a range of interactive procedures is necessary because the most appropriate one to use is often application or user decision-making style dependent, ten of the most prominent interactive procedures, along with the dates of their original articles, are as follows:

1. ECON: e-Constraint Method, Traditional method
2. STEM: (Benayoun et al. 1971)
3. GDF: Geoffrion-Dyer-Feinberg Procedure (1972)
4. ZW: Zionts-Wallenius Procedure (1976)
5. IGP: Interactive Goal Programming (Spronk 1981)
6. WIERZ: Wierzbicki's Aspiration Criterion Vector Method (1982, 1986)
7. TCH: Tchebycheff Method (Steuer and Choo 1983)
8. RACE: Pareto Race (Korhonen and Laakso 1986; Korhonen and Wallenius 1988)
9. NIMBUS: (Miettinen 1999)
10. MICA: Modified Interactive Chebychev Algorithm (Luque et al. 2010)

Other interactive multiobjective programming procedures include those by Nakayama and Sawaragi (1984), Climaco and Antunes (1987), and Koksalan and Karasakal (2006).



Multiobjective Programming, Fig. 3 General algorithmic outline

Selected Interactive Procedures

The Aspiration Criterion Vector Method (WIERZ) begins by asking the DM to specify an aspiration criterion vector $\boldsymbol{q}^{(1)} < z^{**}$. Using the T-vertex λ-vector defined by $\boldsymbol{q}^{(1)}$ and z^{**}, the augmented Tchebycheff program is solved, thus projecting $\boldsymbol{q}^{(1)}$ onto N in order to produce $z^{(1)}$. In the light of $z^{(1)}$, the DM specifies a new aspiration criterion vector $\boldsymbol{q}^{(2)}$. Using the T-vertex λ-vector defined by $\boldsymbol{q}^{(2)}$ and z^{**}, the augmented Tchebycheff program is solved, thus projecting $\boldsymbol{q}^{(2)}$ onto N in order to produce $z^{(2)}$. In the light of $z^{(2)}$, the DM specifies a third aspiration criterion vector $\boldsymbol{q}^{(3)}$, and so forth. Algorithmically, the steps are as follows:

Step 1. $h = 0$. Construct a payoff table, form a z^{**} reference criterion vector, and specify $\rho > 0$ for use in the augmented Tchebycheff program. The DM specifies aspiration criterion vector $\boldsymbol{q}^{(1)}$.

Step 2. $h = h + 1$. Compute T-vertex λ-vector defined by $\boldsymbol{q}^{(h)}$ and z^{**}.

Step 3. Using the T-vertex λ-vector, solve the augmented Tchebycheff program for $z^{(h)}$.

Step 4. In the light of what the DM has been able to learn about the problem so far, the DM contemplates $z^{(h)}$.

Step 5. If the DM wishes to cease iterating, stop with $(z^{(h)}, x^{(h)})$ as the final solution. Otherwise, continue on to Step 6.

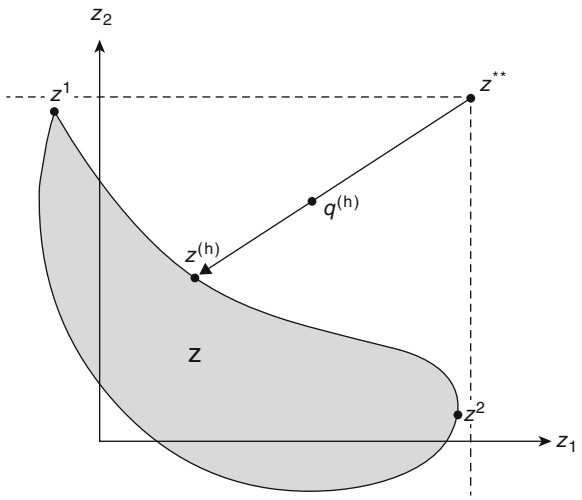
Step 6. The DM specifies another aspiration criterion vector, designated $q^{(h-1)}$. Go to Step 2.

Consider Fig. 4 in which N is the set of boundary criterion vectors z^1 through $z^{(h)}$ to z^2, inclusive. In the figure, it can be seen the way aspiration criterion vector $q^{(h)}$ is projected onto the nondominated set by means of the augmented Tchebycheff program. Note that the direction of the arrow emanating from z^{**} and going through $q^{(h)}$ is given by

$$-\left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_k}\right)$$

where the λ_i are the components of the T-vertex λ-vector defined by $q^{(h)}$ and z^{**}. Thus changing $q^{(h)}$ changes the $z^{(h)}$ generated by the sampling program.

Instead of generating only one solution at each iteration, the Tchebycheff Method (TCH) generates groups of solutions by making multiple probes of each subset in a sequence of progressively smaller subsets of N. Letting P be the number of solutions to be presented to the DM at each iteration, TCH begins by generating P well-spaced λ-vectors from $\Lambda^{(1)} = \Lambda$. Then the lexicographic Tchebycheff program is solved for each of the λ-vectors. From the P resulting nondominated criterion vectors, the DM selects his or her most preferred, designating it $z^{(1)}$. At this point, the interval widths of $\Lambda^{(1)}$ are reduced and centered about the T-vertex λ-vector defined by $z^{(1)}$ and z^{**} to form an interval defined subset $\Lambda^{(2)}$. Then P well-spaced λ-vectors are generated from $\Lambda^{(2)}$ and the lexicographic Tchebycheff program is solved for each of the λ-vectors. From the P resulting non-dominated criterion vectors, the DM selects the most preferred, designating it $z^{(2)}$. Now the interval widths of $\Lambda^{(2)}$ are reduced and centered about the T-vertex λ-vector defined by $z^{(2)}$ and z^{**} to form an interval defined subset $\Lambda^{(3)}$. Then P well-spaced λ-vectors are generated from $\Lambda^{(3)}$ and the lexicographic Tchebycheff program is solved for each of them, and so forth.



Multiobjective Programming, Fig. 4 Projection of $q^{(h)}$ onto the nondominated set

Another procedure that also generates multiple solutions at each iteration, but employs the weighted-sums program, is the Geoffrion-Dyer-Feinberg (GDF) procedure. GDF begins with the specification of an initial feasible criterion vector $z^{(0)}$. Then the DM specifies a λ-vector that is to be reflective of the local marginal tradeoffs at $z^{(0)}$. Using this λ-vector, the weighted-sums program is solved for criterion vector $y^{(1)}$. Then the line through the feasible region in criterion space Z that starts at $z^{(0)}$ and ends at $y^{(1)}$ is divided into segments so as to create P equally spaced criterion vectors. The most preferred of the equally spaced criterion vectors becomes $z^{(1)}$. Then the DM specifies a new λ-vector that is to be reflective of the local marginal tradeoffs at $z^{(1)}$. Using this λ-vector, the weighted-sums program is solved for criterion vector $y^{(2)}$. Then the line segment through Z that starts at $z^{(1)}$ and ends at $y^{(2)}$ is divided into segments so as to create P new equally spaced criterion vectors. The most preferred of the new equally spaced criterion vectors becomes $z^{(2)}$, and so forth.

Features from different procedures can easily be combined. For instance, drawing from STEM, WIERZ and NIMBUS, one could have the following. After forming a z^{**} reference criterion vector, an initial aspiration criterion vector $q^{(1)}$ specified. Then one of the Tchebycheff programs is solved using the T-vertex λ-vector defined by $q^{(1)}$ and z^{**} to produce $z^{(1)}$. The DM then specifies the components of $z^{(1)}$ that are to be

increased, the amounts of each increase, the components that are to be relaxed, and the amounts of each relaxation in order to form a second aspiration criterion vector $\boldsymbol{q}^{(2)}$. Using the T-vertex λ-vector defined by $\boldsymbol{q}^{(2)}$ and z^{**}, one of the Tchebycheff programs is solved to produce $z^{(2)}$. The DM then specifies which components of $z^{(2)}$ are to be increased, the amounts of each increase, the components that are to be relaxed, and the amounts of each relaxation in order to form $\boldsymbol{q}^{(3)}$. Using the T-vertex λ-vector defined by $\boldsymbol{q}^{(3)}$ and z^{**}, one of the Tchebycheff programs is solved to produce $z^{(3)}$, and so forth.

Concluding Remarks

Because the weighted-sums, augmented Tchebycheff, and other variants of these programs that are used to sample the nondominated set are single criterion optimization problems, conventional mathematical programming software can in most cases be employed (Gardiner and Steuer 1994). In this way, interactive procedures can address multiobjective programming problems with as many constraints and variables as in single objective programming. Unfortunately, in multiobjective programming, there are limitations with regard to the number of objectives. Problems with up to about five objectives can generally be accommodated, but above this number, difficulties can arise because of the rate at which the nondominated set grows as the number of objectives increases.

See

- ▶ Decision Analysis
- ▶ Goal Programming
- ▶ Linear Programming
- ▶ Multiple Criteria Decision Making
- ▶ Utility Theory

References

Alves, M. J., & Costa, J. P. (2009). An exact method for computing the nadir values in multiple objective linear programming. *European Journal of Operational Journal, 198*(2), 637–646.

Benayoun, R., de Montgolfier, J., Tergny, J., & Larichev, O. (1971). Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming, 1*(3), 366–375.

Benson, H. P., & Sun, E. (2000). Outcome space partition of the weight set in multiobjective linear programming. *Journal of Optimization Theory and Applications, 105*(1), 17–36.

Climaco, J. C. N., & Antunes, C. H. (1987). TRIMAP–An interactive tricriteria linear programming package. *Foundations Control Engineering, 12*(3), 101–120.

Ehrgott, M. (2005). *Multicriteria optimization*. Berlin: Springer.

Gardiner, L. R., & Steuer, R. E. (1994). Unified interactive multiple objective programming. *European Journal of Operational Journal, 74*(3), 391–406.

Geoffrion, A. M., Dyer, J. S., & Feinberg, A. (1972). An interactive approach for multicriterion optimization, with an application to the operation of an academic department. *Management Science, 19*(4), 357–368.

Isermann, H., & Steuer, R. E. (1988). Computational experience concerning payoff tables and minimum criterion values over the efficient set. *European Journal of Operational Journal, 33*(1), 91–97.

Klamroth, K., Tind, J., & Wiecek, M. (2002). Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research, 36*(3), 413–437.

Koksalan, M., & Karasakal, E. (2006). An interactive approach for multiobjective programming. *Journal of the Operational Research Society, 57*(5), 532–540.

Korhonen, P. J., & Laakso, J. (1986). A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Journal, 24*(3), 277–287.

Korhonen, P. J., & Wallenius, J. (1988). A pareto race. *Naval Research Logistics, 35*(6), 615–623.

Luque, M., Ruiz, F., & Steuer, R. E. (2010). Modified Interactive Chebychev Algorithm (MICA) for convex multiobjective programming. *European Journal of Operational Journal, 204*(3), 557–564.

Miettinen, K. M. (1999). *Nonlinear multiobjective optimization*. Norwell: Kluwer.

Nakayama, H., & Sawaragi, Y. (1984). Satisficing trade off method for multiobjective programming. *Lecture Notes in Economics and Mathematical Systems, 229*, 113–122.

Spronk, J. (1981). *Interactive multiple goal programming*. Boston: Martinus Nijhoff Publishing.

Steuer, R. E. (1986). *Multiple criteria optimization: Theory, computation, and application*. New York: John Wiley.

Steuer, R. E., & Choo, E.-U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming, 26*(1), 326–344.

Wierzbicki, A. P. (1982). A mathematical basis for satisficing decision making. *Mathematical Modelling, 3*, 391–405.

Wierzbicki, A. P. (1986). On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR-Spektrum, 8*, 73–87.

Yu, P. L. (1985). *Multiple-criteria decision making: Concepts techniques and extensions*. New York: Plenum Press.

Zeleny, M. (1982). *Multiple criteria decision making*. New York: McGraw-Hill.

Zionts, S., & Wallenius, J. (1976). An interactive programming method for solving the multiple criteria problem. *Management Science, 22*(6), 652–663.

Multi-armed Bandit Problem

Sequential decision-making problem under uncertainty involving a set of machines (arms) each offering random unknown rewards, in which the decision maker must decide each period which machine (arm) to play (pull), with the objective of maximizing the total reward received. The problem is analogous to playing slot machines in a gambling casino, but has many practical OR/MS applications involving dynamic stochastic resource allocation. One of the basic trade-offs in these types of problems is between exploitation (e.g., playing the machine that has given the best mean reward thus far) versus exploration (playing a machine that has not been tried or one that has been tried infrequently with highly variable rewards).

Multi-attribute Utility Theory

Rakesh K. Sarin
University of California, Los Angeles, CA, USA

Consider a decision problem such as selection of a job, choice of an automobile, or resource allocation in a public program (education, health, criminal justice, etc.). These problems share a common feature—decision alternatives impact multiple attributes. The attractiveness of an alternative therefore depends on how well it scores on each attribute of interest and the relative importance of these attributes. Multi-attribute utility theory (MAUT) is useful in quantifying relative attractiveness of multi-attribute alternatives.

The following notation will be used:

X_i the set of outcomes (scores, consequences) on the ith attribute
x_i a specific outcome in X_i
X $X_1 \times X_2 \times \ldots \times X_n$ (Cartesian product)
u_i a single attribute utility function $u_i : X_i \rightarrow \mathbb{R}$
u the overall utility function, u: $X \rightarrow \mathbb{R}$
$\gtrsim$ "is preferred to"

A decision maker uses the overall utility function, u, to choose among available alternatives. The major emphasis of the work on multi-attribute utility theory has been on questions involving u: on conditions for its decomposition into simple polynomials, on methods for its assessment, and on methods for obtaining sufficient information regarding u so that the evaluation can proceed without its explicit identification with full precision.

The primitive in the theory is the preference relation $\gtrsim$ defined over X. Luce et al. (1965) and Fishburn (1964) provide conditions on a decision maker's preferences that guarantee the existence of a utility function u such that

$$\begin{aligned} &(x_1, \ldots, x_n) \gtrsim (y_1, \ldots, y_n), \\ &\qquad x_i, y_i \in X_i, i = 1, \ldots n \\ &\text{if and only if} \\ &u(x_1, \ldots, x_n) \geq u(y_1, \ldots y_n) \end{aligned} \tag{1}$$

Additional conditions are needed to decompose the multi-attribute utility function u into simple parts. The most common approach for evaluating multi-attribute alternatives is to use an additive representation. For simplicity, assume that there exist the most preferred outcome x_i^* and the least preferred outcome x_i^0 on each attribute $i = 1$ to n. In the additive representation, a real value u is assigned to each outcome $(x_1, \ldots, x_n)$ by

$$u(x_i, \ldots, x_n) = \sum_{i=i}^{n} w_i u_i(x_i) \tag{2}$$

where the $\{u_i\}$ are single attribute utility functions over X_i that are scaled from 0 to 1, i.e., $u_i(x_i^*) = 1, u_i(x_i^0) = 0$ for $i = 1$ to n, and the $\{w_i\}$ are positive scaling constants reflecting relative importance of the attributes with $\sum_{j=1}^{n} w_j = 1$.

If the interest is in simply rank-ordering the available alternatives, then the key condition for the additive form in (2) is mutual preferential independence. The resulting utility function is called an ordinal value function. Attributes X_i and X_j are preferentially independent if the tradeoffs (substitution rates) between X_i and X_j are independent of all other attributes. Mutual preferential independence requires that preference independence holds for all pairs X_i and X_j. Essentially, mutual preferential independence implies that the indifference curves for any pair of attributes are unaffected by the fixed levels of the remaining

attributes. Debreu (1960), Luce and Tukey (1964), and Gorman (1968) provide axiom systems and analysis for the additive form (2).

If, in addition to rank order, one is also interested in the strength of preference between pairs of alternatives, then additional conditions are needed. The resulting utility function is called a measurable value function, and it may be used to order the preference differences between the alternatives.

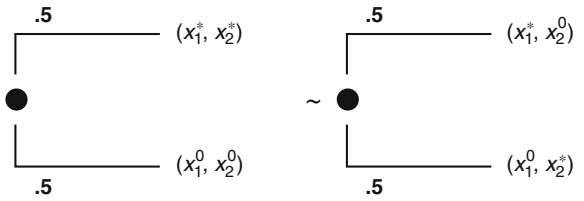
The key condition for an additive measurable value function is difference independence (see Dyer and Sarin 1979). This condition asserts that the preference difference between two alternatives that differ only in terms of one attribute does not depend on the common outcomes on the other $n-1$ attributes.

Finally, perhaps the most researched topic is the case of decisions under risk where the outcome of an alternative is characterized by a probability distribution over X. Denote $\tilde{X}$ as the set of all simple probability distributions over X. Assume that for any $p \in \tilde{X}$ there exists an alternative that can be identified with p, and thus p could be termed as a risky alternative. The outcome of an alternative $p \in \tilde{X}$ might be represented by the lottery which assigns probabilities $p_1, \ldots, p_l, \sum_{j=1}^{l} p_j = 1$, to the outcomes $x^1, \ldots, x^l \in X$, respectively. For the choice among risky alternatives $p, q \in \tilde{X}$, von Neumann and Morgenstern (1947) specified conditions on the decision maker's preference relation $\gtrsim$ over $\tilde{X}$ that imply:

$$\begin{aligned} &p \gtrsim q \\ &\text{if and only if} \\ &\sum_{x \in X} p(x)u(x) \geq \sum_{x \in X} q(x)u(x). \end{aligned} \tag{3}$$

Notice that the same symbol u has been used to denote ordinal value function, measurable value function, and now the von Neumann-Morgenstern utility function. The context, however, makes the interpretation clear.

A majority of the applied work in multi-attribute utility theory deals with the case when the von Neumann-Morgenstern utility function is decomposed into the additive form (2). Fishburn (1965a, b) derived necessary and sufficient conditions for a utility function to be additive. The key condition for additivity is the marginality condition, which states that the preferences for any lottery $p \in X$ should depend only on the marginal probability distributions over X_i and not on their joint probability distribution. Thus, for additivity to hold, the two lotteries below must be indifferent:



Notice that in either lottery, the marginal probability of receiving the most preferred outcome or the least preferred outcome on each attribute is identical. A decision maker may, however, prefer the right-hand side lottery over the left-hand side lottery if the decision maker wishes to avoid a 0.5 chance of the poor outcome (x_1^0, x_2^0) on both attributes.

The assessment of single attribute utility functions $\{u_i\}$ in (2) will require different methods depending on whether the overall utility represents an ordinal value function, a measurable value function, or a von Neumann-Morgenstern utility function. Keeney and Raiffa (1976) discuss methods for assessing multi-attribute ordinal value function and multi-attribute von Neumann-Morgenstern utility function. Dyer and Sarin (1979) and von Winterfeldt and Edwards (1986) discuss assessment of multi-attribute measurable value function.

Besides the additive form (2), a multiplicative form for the overall utility function has also found applications in a wide variety of contexts. In the multiplicative representation, a real value u is assigned to each outcome $(x_1, \ldots, x_n)$ by

$$1 + ku(x_i, \ldots, x_n) = \left[\prod_{i=i}^{n} [1 + kk_i u_i(x_i)]\right]$$

where the $\{u_i\}$ are single attribute utility functions over X_i that are scaled from zero to one, the $\{k_i\}$ are positive scaling constants, k is an additional scaling constant satisfying $k > -1$, and

$$1 + k = \prod_{i=i}^{n} [(1 + kk_i)].$$

If u is a measurable value function, then weak difference independence along with mutual

preference independence provides the desired result. An attribute is weak difference independent of the other attributes if preference difference between pairs of levels of that attribute do not depend on fixed levels of any of the other attributes. Thus, for $x_i, y_i, w_i, z_i \in X_i$, the ordering of preference difference between x_i and y_i, and w_i and z_i, remains unchanged whether one fixes the other attributes at their most preferred levels or at their least preferred levels.

If the overall utility function is used for ranking lotteries as in (3), then a utility independence condition, first introduced by Keeney (1969), is needed to provide the multiplicative representation (4). An attribute is said to be utility independent of the other attributes if the decision maker's preferences for lotteries over this attribute do not depend on the fixed levels of the remaining attributes. Mutual preferential independence and one attribute being utility independent of the others are sufficient to guarantee either the multiplicative form (4) or the additive form (2). The additive form results if in (4) $k = 0$ or $\sum_{j=1}^{n} k_j = 1$. Keeney and Raiffa (1976) discuss methods for calibrating the additive and multiplicative forms for the utility function. In the literature, other independence conditions have been identified that lead to more complex nonadditive decompositions of the utility function. These general conditions are reviewed in Farquhar (1977).

If utilities, importance weights, and probabilities are incompletely specified, then the approaches of Fishburn (1964) and Sarin (1975) can be used to obtain a partial ranking of alternatives.

The key feature of multi-attribute utility theory is to specify verifiable conditions on a decision maker's preferences. If these conditions are satisfied, then the multi-attribute utility function can be decomposed into simple parts. This approach of breaking the complex value problem (objective function) into manageable parts has found significant applications in decision and policy analysis. In broad terms, multi-attribute utility theory facilitates measurement of preferences or values. The axioms of the theory have been found to be useful in suggesting approaches for measurement of values. In physical measurements (e.g., length), the methods for measurement have been known for a long time and the theory of measurement has added little to suggesting new methods. In the measurement of values, however, several new methods have been developed as a direct result of the theory.

See

- ▶ Analytic Hierarchy Process
- ▶ Decision Analysis
- ▶ Multiple Criteria Decision Making
- ▶ Preference Theory
- ▶ Utility Theory

References

Debreu, G. (1960). Topological methods in cardinal utility theory. In *Mathematical methods in the social sciences* (pp. 16–26). Stanford, CA: Stanford University Press.

Dyer, J. S., & Sarin, R. K. (1979). Measurable multi-attribute value functions. *Operations Research, 27*, 810–822.

Edwards, W., & von Winterfeldt, D. (1986). *Decision analysis and behavioral research*. Cambridge: Cambridge University Press.

Farquhar, P. H. (1977). A survey of multiattribute utility theory and applications. *TIMS Studies in Management Science, 6*, 59–89.

Fishburn, P. C. (1964). *Decision and value theory*. New York: Wiley.

Fishburn, P. C. (1965a). Independence in utility theory with whole product sets. *Operations Research, 13*, 28–45.

Fishburn, P. C. (1965b). Utility theory. *Management Science, 14*, 335–378.

Gorman, W. M. (1968). Symposium on aggregation: The structure of utility functions. *Review of Economic Studies, 35*, 367–390.

Keeney, R. L. (1969). *Multidimensional utility functions: Theory, assessment, and applications (Technical Report No. 43)*. Cambridge: Operations Research Center, M.I.T.

Keeney, R. L., & Raiffa, H. (1976). *Decisions with multiple objectives: Preferences and value tradeoffs*. New York: Wiley.

Luce, R. D., & Tukey, J. W. (1964). Simultaneous conjoint measurement: A new type of fundamental measurement. *Journal of Mathematical Psychology, 1*, 1–27.

Luce, R. D., Bush, R. R., & Galantor, E. (1965). *Handbook of mathematical psychology* (Vol. 3). New York: Wiley.

Sarin, R. K. (1975). *Interactive procedures for evaluation of multi-attributed alternatives*. Working paper 232. Los Angeles: Western Management Science Institute, University of California.

von Neumann, J., & Morgenstern, O. (1947). *Theory of games and economic behavior*. Princeton, NJ: Princeton University Press.

Multi-Criteria Decision Making (MCDM)

- ▶ Multiple Criteria Decision Making

Multi-Echelon Inventory Systems

Inventory systems comprised of multiple stages of inventory control decision making, e.g., in a supply chain, there are inventory decisions to be made at the production facility, the distributor, and the retail outlet, among others.

See

► Inventory Modeling

Multi-Echelon Logistics Systems

Logistics systems comprised of several layers of individual logistics problems.

See

► Logistics and Supply Chain Management

Multiple Criteria Decision Making

Ramaswamy Ramesh and Stanley Zionts
University at Buffalo, The State University of New York, Buffalo, NY, USA

Introduction

Multiple Criteria Decision Making (MCDM) refers to making decisions in the presence of multiple, usually conflicting, objectives. Multiple criteria decision problems pervade almost all decision situations ranging from common household decisions to complex strategic and policy level decisions in corporations and governments. Prior to the development of MCDM as a discipline, such problems have been traditionally addressed as single-criterion optimization problems by (i) deriving a composite measure of the objectives and optimizing it, or (ii) by choosing one of the objectives as the main decision objective for optimization and solving the problem by requiring an acceptable level of achievement in each of the other objectives. MCDM as a discipline was founded on two key concepts of human behavior, introduced and explored in detail by Herbert Simon in the 1950s: satisficing and bounded rationality (Simon 1957). The two are intertwined because satisficing involves finding solutions that satisfy constraints rather than optimizing the objectives, while bounded rationality involves setting the constraints and then searching for solutions satisfying the constraints, adjusting the constraints, and then continuing the process until a satisfactory solution is found. The rest of this article overviews important aspects of MCDM, including basic concepts, a taxonomy, modeling techniques, and algorithms.

Basic Concepts

An MCDM problem can be broadly described as follows. Let $D = \{d_1,\ldots, d_n\}$ denote the decision space, comprising the set of possible decision alternatives to a problem. Let $C = \{C_1,\ldots, C_p\}$ denote the objective space, comprising of a set of p mutually conflicting objectives. Without loss of generality, assume all objectives are to be maximized. Let E: $D \rightarrow C$ be a mapping of the decision space on to the objective space, where $E(d_i)$ is the vector $(C_1^i,\ldots, C_p^i)$. Each element of this vector is an assessment, or the value of the corresponding objective provided by the decision alternative d_i. A fundamental concept in MCDM is that of dominance, defined as follows.

***Definition* 1 (*Dominance*).** A decision alternative d_i said to be dominated by another alternative d_j if $C_k^i \leq C_k^j$, $k = 1,\ldots, p$ with at least one strict inequality.

In the above definition, if all the inequalities hold as strict inequalities, then the dominance is said to be strong; otherwise, it is called weak. The following concept is a logical extension of the dominance concept.

***Definition* 2 (Convex *Dominance*).** An alternative d_i is said to be convex dominated by a subset $\hat{D} \subset D$ if it

is dominated by a convex combination of the alternatives in $\hat{D}$

The above definitions lead to a central theme of all MCDM techniques as follows.

***Definition* 3 (*Efficiency*).** An alternative d_j is said to be efficient or nondominated in D if there is no other alternative in D that dominates it, even weakly.

The concept of efficiency can be extended to convex dominance as well. In this case, an efficient alternative is known as convex-efficient or convex-nondominated. The following theorem of Geoffrion (1968) shows how the efficiency of an alternative can be determined. Zionts and Wallenius (1980) introduced a different but equivalent methodology that solves a number of problems including that one.

***Theorem* 1.** Consider any decision alternative d_i and its mapping on the objective space $(C^i_1,\ldots, C^i_p)$. The decision d_i is efficient if only if the following linear program is unbounded:

$$\text{Maximize} \sum_{j=1}^{p} w_j C^i_j$$

$$\text{subject to} \sum_{j=1}^{p} w_j C^k_j \leq 0,\ k = 1,\ldots,n, k \neq i$$

$$w_j \geq 0,\ j = 1,\ldots,p.$$

A Taxonomy of MCDM Methods

The MCDM methods proposed in the literature cover a wide spectrum, and there are several alternative ways of organizing them into a taxonomy. The taxonomy described here is based on Chankong et al. (1984), which is one of the interpretations of the world of MCDM models. At the outset, MCDM methods can be classified into two broad classes: vector optimization methods and utility optimization methods. Vector optimization is primarily concerned with the generation of all efficient decision alternatives. These methods do not require intervention of a decision maker. These methods do generate a subset of nondominated solutions. Some of the well-known vector optimization methods include those of Geoffrion (1968), Villarreal and Karwan (1981), and Yu and Zeleny (1976).

Multiple Criteria Decision Making, Table 1 A taxonomy of MCDM approaches

Decision outcome	Decision space	
	Explicit	Implicit
Deterministic	Deterministic Multiattribute Decision Analysis	Deterministic Multiobjective Mathematical Programming
Stochastic	Stochastic Multiattribute Decision Analysis	Stochastic Multiobjective Mathematical Programming

The utility optimization methods can be broadly organized according to the following dimensions (see, for example, Zionts 1979):

1. Nature of decision space: Explicit or Implicit; and
2. Nature of decision outcomes: Stochastic or Deterministic.

In an explicit decision space, decision alternatives are stated explicitly. A classical example is the home buying problem, where a decision maker/home buyer is faced with a set of possible homes to consider purchasing. For an implicit decision, alternatives are stated using a set of constraints, such as in linear or nonlinear programming where a feasible alternative must satisfy the constraints. An implicit decision situation can be further categorized as continuous or discrete. The decision outcomes are stochastic or deterministic depending on whether the mapping function E: $D \rightarrow C$ is stochastic or deterministic. Table 1 classifies MCDM methods broadly along the two dimensions. There are many approaches in the various segments of this classification. Here, the discussion focuses on the best-known methods.

Methodological Approaches

Deterministic Decision Analysis — Deterministic decision analysis is concerned with finding the most preferred alternative in decision space by constructing a value function representing a decision maker's preference structure, and then using the value function to identify the most preferred solution. A value function $v(C_1, C_2,\ldots, C_p)$ is a scalar-valued

function defined with the property that $v(C_1, C_2,\ldots, C_p) > v(C',\ldots, C'_p)$ if and only if $(C_1, C_2,\ldots, C_p)$ is at least as preferred as $(C'_1,\ldots, C'_p)$ (Keeney and Raiffa 1976). The construction of the value function involves choice decisions made by the decision maker. Generating value functions is simplified if certain conditions hold, in which case it is possible to decompose the above functions into partial value functions $v_k(C_k)$ for each value of k.

The decomposition and certain simplifications of the value function may be carried out if certain underlying assumptions on the decision maker's preference structure hold. One of these is preferential independence, which is stated as follows: Consider a subset of objectives denoted as $\hat{C}$. If the decision maker's preferences in the space $C - \hat{C}$ are the same for any set of arbitrarily fixed levels of the objectives $\hat{C}$, then $\hat{C}$ is said to be preferentially independent of $C - \hat{C}$. The set C is said to be mutually preferentially independent if every subset of C is preferentially independent of its complement with respect to C. When mutual preferential independence holds, an additive value function of the form

$$v(d_i) = \sum_{k=1}^{p} \lambda_k v_k\left(C_k^i\right) \text{ where } \lambda_k \text{ is a scalar constant}$$

is appropriate. There are other nonlinear forms that can be used as well. Of course, an additive value function, if appropriate, is highly desirable. Once the value function has been determined, it can be used to evaluate and rank the alternatives.

Stochastic Decision Analysis — Stochastic decision analysis is similar to the deterministic case, except that the outcomes are stochastic, and utility functions are constructed instead of value functions. The ideas are similar. There is an analogous condition to that described for the discrete case above. It involves utility independence. A subset of objectives $\hat{C}$ is utility independent of its complement if the conditional preference order for lotteries involving changes in $\hat{C}$ does not depend on the levels at which the objectives in $\hat{C}$ are fixed. Since utility independence refers to lotteries and preferential independence refers to deterministic outcomes, utility independence implies preferential independence, but not vice versa. Analogous to mutual preferential independence, the set C is said to be mutually utility independent if every subset of C is utility independent of its complement with respect to C. Keeney and Raiffa (1976) show that if C is mutually utility independent, then a multiplicative utility function is appropriate. This function is of the form

$$u(d_i) = \prod_{k=1}^{p} \mu_k u_k\left(C_k^i\right),$$

where $u(d_i)$ is the overall utility of the decision alternative d_i, $u_k(C_k^i)$ is the utility of its kth objective component, and μ_k is a scalar constant. A more stringent set of assumptions must hold in order that the utility function be additive. In the stochastic case, not only must a utility function be estimated, but probabilities of various outcomes must also be estimated by the decision maker.

Multiobjective Mathematical Programming — Considerable work has been done in the multiobjective mathematical programming area. These include Multiobjective Linear Programming (MOLP) and Multiobjective Integer Programming (MOIP). Goal programming (Lee 1972), the method of Zionts and Wallenius (1976, 1983), the Step Method of Benayoun et al. (1971), and the method of Steuer (1976) are some of the better-known MOLP methods. Goal programming and the method of Zionts and Wallenius are now described in more detail.

Goal programming is an extension of linear programming and was proposed by Charnes and Cooper in 1961. A description of this technique is as follows. Consider the following MOLP problem:

$$\begin{aligned} &\text{Maximize } \boldsymbol{Cx} \\ &\text{subject to } \quad \boldsymbol{Ax} \le \boldsymbol{b} \qquad \text{(MOLP)} \\ &\boldsymbol{x} \ge \boldsymbol{0} \end{aligned}$$

where $\boldsymbol{C} = (c_{kj})$ is a $(p \times n)$ matrix, $\boldsymbol{A}$ is an $(m \times n)$ matrix and $\boldsymbol{x}$ is an $(n \times 1)$ vector. Let $(\alpha_1,\ldots, \alpha_p)$ denote the goals with respect to the desired levels of attainment in the objectives specified by a decision maker. Introduce over and under attainment variables y_k^+ and y_k^- for each objective and add the following constraints, where $\boldsymbol{c}_k$ is the kth row of $\boldsymbol{C}$:

$$\boldsymbol{c}_k\boldsymbol{x} - y_k^+ + y_k^- = \alpha_k, \quad k - 1,\ldots, p.$$

Let w_k denote the penalty for the net deviation from the goal of objective $k = 1,\ldots, p$. Then the goal programming problem is formulated as follows:

$$\text{Minimize} \quad \sum_{j=1}^{p} w_k\left(y_k^+ + y_k^-\right) \qquad \text{(GP)}$$

$$\text{subject to} \quad \sum_{j=1}^{n} c_{kj}x_j - y_k^+ + y_k^- = \alpha_k, \quad k = 1,\ldots,p$$

$$\boldsymbol{Ax} \leq \boldsymbol{b}$$

$$\boldsymbol{x}, \boldsymbol{y} \geq \boldsymbol{0}$$

The above problem minimizes a weighted sum of deviations from the desired goals, where weights are required from the decision maker. The goal programming formulation is an attempt to find a solution that is closest to the decision maker's desired goals, while also responding to his differential emphasis on the nonattainment of the various goals.

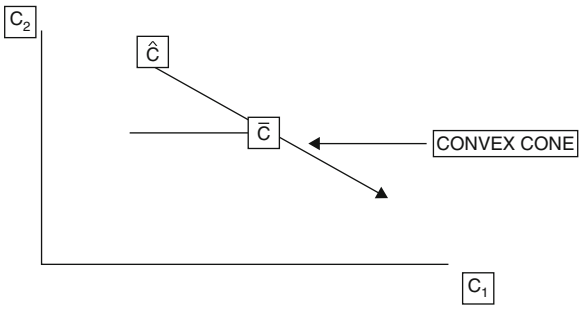
The Zionts and Wallenius method follows an interactive approach using pairwise evaluations of decision alternatives by a decision maker to solve problem MOLP. The method starts by choosing an initial set of weights $\lambda \in R^p$, and maximizing a linear composite objective $\lambda\boldsymbol{Cx}$. This generates a corner point of $\{\boldsymbol{Ax} \leq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ that is efficient. Call this solution $\boldsymbol{x}^0$. Next, the adjacent corner points of $\boldsymbol{x}^0$ that are also efficient (and whose edges leading to them are *also* efficient) are determined. Call this set S^0. The decision maker is asked to choose between $\boldsymbol{x}^0$ and a solution from S^0 until: (i) either he or she prefers $\boldsymbol{x}^0$ to all the points in S^0, or (ii) prefers some solution in S^0 to $\boldsymbol{x}^0$. If $\boldsymbol{x}^0$ is preferred to all the points in S^0, then the method stops with $\boldsymbol{x}^0$ as a "locally" best preferred corner-point solution. Otherwise, if some solution in S^0 is preferred to $\boldsymbol{x}^0$, then it is devoted as $\boldsymbol{x}'$. Linear constraints of the form $\lambda\,(\boldsymbol{Cx}' - \boldsymbol{Cx}'') \leq -\epsilon$ where $\boldsymbol{x}'$ is preferred to $\boldsymbol{x}''$ and ϵ is a small positive quantity are generated from the decision maker's pairwise preferences. A new set of weights that satisfy these constraints are then obtained. If these constraints are in conflict, then some of them are dropped in determining the new weights. Call the new set λ''. Maximizing the composite objective $\lambda''\boldsymbol{Cx}$, a new efficient corner point is generated, and the above steps are repeated until a corner point that is preferred to all its adjacent efficient corner points is obtained.

Compared to MOLP, research on MOIP is rather limited. Some of the earlier works on MOIP have been in the domain of vector optimization. Bitran and Rivera (1982) provided an implicit enumeration algorithm for determining the efficient set of 0-1 MOIP problems. Pasternak and Passy (1973) studied the vector optimization problem for two objectives. Klein and Hannan (1982) extended Pasternak and Passy's work to more than two objectives. Villarreal and Karwan (1981) generalized the classical dynamic programming recursions to a multicriteria framework. Ramesh et al. (1989) followed the utility optimization approach to find the most preferred solution to an MOIP problem.

The method of Ramesh et al. (1989) follows a branch-and-bound search strategy using the Zionts and Wallenius method for bounding. The decision maker's preference structure is assessed using pairwise evaluations and an internal representation of the preference structure is successively built during the course of the branch-and-bound search. This representation is used to deduce the decision maker's preferences wherever possible so that the cognitive load arising out of the pairwise judgments can be minimized. The internal representation is based on the concept of convex cones as described below (Korhonen et al. 1984).

Consider a two-dimensional objective space as shown in Fig. 1. Let $\bar{C}$ and $\hat{C}$ be two points in this space such that $\hat{C}$ is preferred to $\bar{C}$. Assuming a quasiconcave and nondecreasing utility function for the decision maker, it follows that every point falling on the ray $\left\{\hat{C} | \hat{C} = \hat{C} + \mu\,(\bar{C} - \hat{C}), \mu \geq 0\right\}$ is less preferred than $\hat{C}$ and no more preferred than $\bar{C}$. Consequently, every point in this ray and those dominated by it can be eliminated from consideration. This ray is called a convex cone, and is illustrated in Fig. 1. Every pairwise judgment of a decision maker yields a convex cone and the cones are ordered into a tree structured to eliminate search regions efficiently and minimize the need for the decision maker's pairwise evaluations throughout the search procedure.

Other Explicit Decision Space Methods — Several methods have been proposed for finding the most preferred alternative from an explicitly stated decision space without estimating a value function. These techniques are methods of deterministic decision analysis, and there is substantial interest in



Multiple Criteria Decision Making, Fig. 1 Illustration of convex cones

these problems. Three important methods in this category are the Multiple Criteria Decision Making (MCDM) Analytic Hierarchy Process (Saaty 1980), the method of Korhonen et al. (1984), and the AIM method (Lotfi et al. 1992).

The idea of Analytic Hierarchy Process (AHP) is that one can structure a problem hierarchically, and then make judgments regarding the relative importance of various aspects of the problem. As a result of these judgments, a ranking is produced. A simple decision problem would have a hierarchy that consists of three levels, from the top down: 1) the goal; 2) the criteria involved; and 3) the alternatives. The number of levels depends on the nature of the problem involved. In general, consider an n-alternative, p-criteria problem. Then the decision maker is asked to fill in entries in $p + 1$ reciprocal matrices as follows:

1. One $(p \times p)$ matrix relating each criterion to all others; and
2. P $(n \times n)$ matrices, each relating one criterion to all alternatives.

Each reciprocal matrix has all diagonal elements one, and off-diagonal elements reciprocal, that is, $a_{ij} = 1/a_{ji}$. Accordingly, the decision maker need only provide just less than half the entries, more specifically, the $[p(p-1)/2] + p[n(n-1)/2]$ off-diagonal (lower or upper) entries in the matrix. Though the amount can be reduced to as few as $(p-1) + p(n-1)$ entries (having no redundancy), the reduction in information required increases the cognitive load on the decision maker to provide entries, and does not provide the redundancy and cross checking that furnishing the complete input provides.

In filling in the matrices, the decision maker is asked to provide numbers between 1/9 and 9 reflecting the relative importance between the aspects involved. One of the matrices reflects the comparison among criteria and the p other matrices reflect evaluations of alternatives with respect to each criterion. AHP next solves for the right eigenvector, or characteristic vector, of each matrix. An eigenvector of a matrix may be estimated by taking the geometric mean of the elements of each row of the matrix (for a $p \times p$ matrix, the pth root of the product of the p elements of a row), and then normalizing the resulting vector so that the sum of the elements is unity. The consistency of the matrix (as differentiated from a matrix generated at random) may be tested using a calculation on the matrix. By the user furnishing fewer than all $p(p-1)/2$ entries required in the matrix, the test on consistency is compromised. The scaled eigenvectors are then used to score and rank each alternative.

Korhonen et al. (1984) presented an interactive method employing pairwise comparisons for solving the discrete, deterministic MCDM problem. Assuming a quasiconcave and nondecreasing utility function, they introduce the concept of convex cones. Choosing an arbitrary set of positive weights w_i, $i = 1,\ldots, p$, a composite linear utility function is initially generated. Using the composite as a proxy for the true utility function, the decision alternative maximizing the composite is generated. Call this solution d^0. Using the mapping E: $D \rightarrow C$, all adjacent efficient decision alternatives to d^0 (as in the Zionts-Wallenius method) are determined. This is done for the region that consists of all convex combinations of feasible solutions. Call the set of such solutions S^0. The decision maker is asked to choose between d^0 and some solution from S^0. Based on the response, a constraint on the weights is generated, as in the Zionts and Wallenius method for MOLP, and a convex cone is derived. Any solution in the set S^0 dominated by the cone is removed from S^0, and the above step is repeated until either d^0 is preferred to all solutions in S^0 or some solution in S^0 is preferred to d^0. The constraints on the weights and the convex cones generated at each iteration of this step are accumulated. The set of cones is used to deduce the decision maker's preferences wherever possible. This reduces the search space, while also minimizing the number of pairwise comparisons the decision maker has to perform.

Every solution in S^0 that is less preferred than d^0 is dropped from consideration. If d^0 is preferred to all the solutions in S^0, then it is denoted as d. If some solution in S^0 is preferred to d^0, then the preferred solution is denoted as d. If d is the only efficient solution remaining in the decision space, then the procedure stops with d as the most preferred decision. Otherwise, choosing a set of weights consistent with the weight constraints (after dropping any conflicting constraints), a new composite linear utility function is generated. Denoting the decision alternative maximizing this composite function as d, the decision maker chooses between d and d. Denoting the preferred solution as d^0, the above steps are repeated.

Lotfi et al. (1992) develop an eclectic method called the Aspiration-Level Interactive Method (AIM) for MCDM. It involves a philosophy that aspiration levels and feedback regarding the relative feasibility of the aspiration levels provide a powerful tool for decision making. The method is embodied in a computer program called AIM. The method provides the decision maker with various kinds of feedback as he explores the solutions. Several different kinds of objectives may be included: objectives to be maximized; objectives to be minimized; target objectives; any of the above kinds of objectives with thresholds, or levels beyond which the user is indifferent to further gains in the objective; and qualitative objectives. To further explain the idea of thresholds, suppose that in the purchase of a house, the age of the house is an attribute to be minimized. Suppose further that the buyer treats as equivalent, however, any houses ten years or less in age. In this case, there is a threshold of ten years, so that an eight-year-old house is considered to be no better than a ten-year-old house with respect to age.

To begin with, the decision maker has the following basic information:

1. A current goal or aspiration level for each objective, initially set to the median, together with the proportion of alternatives having values of the objective at least as good as that value.
2. Two other aspiration levels, the next better and the next worse than the current goal occurring in the data base.
3. The ideal and nadir solutions to the problem.
4. The proportion of alternatives that simultaneously satisfy aspiration levels given in 1 and 2.
5. A nearest nondominated solution to the current goal. The nearest solution is found by mapping the current goal to a solution on the efficient frontier or in the set of nondominated solutions.

The current goal may be (and should be) changed by the user, component by component, to any desired realizable level of any objective. The intention, however, is to keep the current goal near the efficient frontier and therefore nearly achievable. As the user changes the current goal, all but item(s) 3 above change.

The user can invoke various options to help in decision making. He or she can see which solutions, if any, satisfy his current goal. Second, he or she can obtain a ranking of solutions based on a function resulting from his choice of a current goal. Third, he or she can use a simplified version of a concept called outranking to identify neighbor solutions that are similar to his nearest solution. The decision maker may also review the weights implied by the current goal, see a quartile distribution of the problem by objective, and identify and possibly delete dominated solutions.

Concluding Remarks

The field of multiple criteria decision making has been an active since the 1960s. Many interesting approaches have been developed, explored, and implemented in solving problems. Implementation of MCDM methodologies include multiple criteria decision support systems (MCDSS) and negotiations, which may be regarded as multiple criteria problems involving multiple decision makers. MCDSS integrate the multiple criteria approaches in user-friendly microcomputer systems, such as the VIG/VIMDA system of Korhonen and Laakso (1986), the Expert Choice software that implements AHP, and the AIM package of Lotfi et al. (1992) implemented on the World Wide Web by Wang and Zionts (2005). An objective of most of the MCDSS is to provide inexpensive stand-alone software that is easy to use. A very useful set of computer MCDM method software may be found on the World Wide Web by a search on the word decisionarium; the software is housed at the

Helsinki University of Technology, now part of Aalto University.

Negotiations or multiperson MCDM is a natural extension of MCDM. Many decisions are made by groups, and negotiation theory involves using some of the MCDM concepts to simplify and assist negotiations; see for example, Wang and Zionts (2008).

In addition to the journals devoted to management science and operations research and behavioral science, there are two journals that contain articles more exclusively in this area: *Multi-Criteria Decision Analysis* and *Group Decision and Negotiation*. The paper by Wallenius et al. (2008) explores recent accomplishments and what lies ahead.

See

- ▶ Analytic Hierarchy Process
- ▶ Analytic Network Process
- ▶ Decision Analysis
- ▶ Decision Problem
- ▶ Goal Programming
- ▶ Multi-attribute Utility Theory
- ▶ Multiobjective Programming
- ▶ Utility Theory
- ▶ Value Function

References

Benayoun, R., De Montgolfier, J., Tergny, J., & Larichev, O. (1971). Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming, 1*, 366–375.

Bitran, G. R., & Rivera, J. M. (1982). A combined approach to solving binary multicriteria problems. *Naval Research Logistics, 29*, 181–201.

Chankong, V., Haimes, Y. Y., Thadathil, J., & Zionts, S. (1984). Multiple criteria optimization: A state of the art review. In *Decision making with multiple objectives* (pp. 36–90). Berlin: Springer.

Geoffrion, A. M. (1968). Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications, 22*, 618–630.

Keeney, R. L., & Raiffa, H. (1976). *Decisions with multiple objectives: Preferences and value trade-offs*. New York: John Wiley.

Klein, D., & Hannan, E. (1982). An algorithm for the multiple objective integer linear programming problem. *European Journal of Operational Research, 9*, 378–385.

Korhonen, P., & Laakso, J. (1986). A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research, 24*, 277–287.

Korhonen, P., Wallenius, J., & Zionts, S. (1984). Solving the discrete multiple criteria problem using convex cones. *Management Science, 30*, 1336–1345.

Lee, S. M. (1972). *Goal programming for decision analysis*. Philadelphia: Auerbach Publishers.

Lotfi, V., Stewart, T. J., & Zionts, S. (1992). An aspiration-level interactive model for multiple criteria decision making. *Computers and Operations Research, 19*, 671–681.

Lotfi, V., Yoon, Y. S., & Zionts, S. (1997). Aspiration-based search algorithm (ABSALG) for multiple objective linear programming problems: Theory and comparative tests. *Management Science, 43*, 1047–1059.

Pasternak, H., & Passy, V. (1973). Bicriterion mathematical programs with boolean variables. In *Multiple criteria decision making*. Columbia: University of South Carolina Press.

Ramesh, R., Karwan, M. H., & Zionts, S. (1989). Preference structure representation using convex cones in multicriteria integer programming. *Management Science, 35*, 1092–1105.

Saaty, T. L. (1980). *The analytic hierarchy process*. New York: McGraw-Hill.

Simon, H. (1957). *Administrative behavior*. New York: The Free Press.

Steuer, R. E. (1976). Multiple objective linear programming with interval criterion weights. *Management Science, 23*, 305–316.

Villarreal, B., & Karwan, M. H. (1981). Multicriteria integer programming: A (hybrid) dynamic programming recursive approach. *Mathematical Programming, 21*, 204–223.

Wallenius, J., Dyer, J., Fishburn, P., Steuer, R., Zionts, S., & Deb, K. (2008). Multiple criteria decision making/ multiattribute utility theory: Recent accomplishments and what lies ahead. *Management Science, 54*, 1336–1349.

Wang, J. G., & Zionts, S. (2005). WebAIM: An online aspiration-level interactive method. *Multi-Criteria Decision Analysis, 13*, 51–63.

Wang, J. G., & Zionts, S. (2008). Negotiating wisely: Considerations based on multi-criteria decision making/ multi-attribute utility theory. *European Journal of Operational Research, 188*, 191–205.

Yu, P. L., & Zeleny, M. (1976). Linear multiparametric programming by multicriteria simplex method. *Management Science, 23*, 159–170.

Zionts, S. (1979). MCDM: If not a roman numeral, then what? *Interfaces, 9*, 94–101.

Zionts, S., & Wallenius, J. (1976). An interactive programming method for solving the multiple criteria problem. *Management Science, 22*, 652–663.

Zionts, S., & Wallenius, J. (1980). Identifying efficient vectors: Some theory and computational results. *Operations Research, 28*, 788–793.

Zionts, S., & Wallenius, J. (1983). An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions. *Management Science, 29*, 519–529.

Multiple Optimal Solutions

In an optimization problem, when different feasible solutions yield the same optimal value for the objective function, the problem has multiple optimal solutions. If a linear-programming problem has multiple optimal solutions, then such solutions correspond to extreme point solutions and their convex combinations.

See

▶ Unique Solution

Multiple Pricing

When solving a linear-programming problem using the simplex method, it is computationally efficient to select a small number, say 5, possible candidate vectors from which one would be chosen to enter the basis. The candidate set consists of columns with large (most negative or most positive) reduced costs, and the vector in this set that yields the largest change in the objective function is selected. Succeeding iterations only consider candidate basis vectors from the vectors that remain in the set that have properly signed reduced costs. When all vectors in the set are chosen or none can serve to change the objective function in the proper direction, a new set is determined.

See

▶ Partial Pricing
▶ Simplex Method (Algorithm)

Multiplier Vector

For a given feasible basis $\boldsymbol{B}$ to a linear-programming problem, let the row vector $\boldsymbol{c_B}$ be the ordered set of cost coefficients for the vectors in $\boldsymbol{B}$. The multiplier vector is defined as $\boldsymbol{\pi} = \boldsymbol{c_B}\,\boldsymbol{B}^{-1}$. If $\boldsymbol{B}$ is an optimal basis, then the components of $\boldsymbol{\pi}$ are the dual variables associated with the corresponding primal constraints. The vector $\boldsymbol{\pi}$ is also called the simplex multiplier vector, with the components of $\boldsymbol{\pi}$ being the simplex multipliers.

See

▶ Simplex Method (Algorithm)

Multivariate Quality Control

Francis B. Alt[1] and Scott D. Grimshaw[2]
[1]University of Maryland, College Park, MD, USA
[2]Brigham Young University, Provo, UT, USA

Introduction

A frequent quality control application in the chemical and process industries is the simultaneous monitoring of several correlated quality measurements. For example, González and Sánchez (2010) apply multivariate quality control to manufacturing the window frame for the door of a vehicle, where the five gaps on the window frame are measured at seven locations on the frame. Control charts that simultaneously evaluate all the information available on a process are based on the foundational work of Hotelling (1947) in a military application. While one could create univariate control charts for each measurement, ignoring the correlation between measurements impacts the statistical properties in many ways. Jackson (1956) showed that the use of univariate control charts can be misleading even when the measured characteristics are uncorrelated. Alt (1985) points out that not only is it statistically inefficient to monitor each measurement on its own control chart because the proper out-of-control region is elliptical, the process may exhibit frequent false out-of-control alarms.

Multivariate quality control procedures can be classified into two broad categories: (1) Shewhart procedures designed to quickly detect large out-of-control shifts from the in-control mean vector, and (2) Multivariate EWMA procedures that can be designed to efficiently detect persistent small

and moderate shifts. These are discussed in turn, followed by a discussion of other important methods for multivariate quality control.

Shewhart Charts

At regular time intervals, observe a rational subgroup of size n on p quality characteristics denoted by the vector $\mathbf{x}_i$. When the process is in-control, the quality characteristics will have mean $\boldsymbol{\mu}_0$ and variance-covariance matrix $\boldsymbol{\Sigma}_0$.

The Shewhart χ^2 chart produces an out-of-control signal when

$$\chi^2 = n(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}_0^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)$$

exceeds the upper control limit. The $\bar{\mathbf{x}}$ is the mean of each quality characteristic for the rational subgroup assembled as a $p \times 1$ vector.

The performance of a control chart is judged by its average run length (ARL), which is the average number of time periods taken before an out-of-control signal is given. A control chart is designed to have a large in-control ARL and a small out-of-control ARL. For multivariate Shewhart charts the upper control limit defines the in- and out-of-control ARL. The run length of Shewhart control charts follows a geometric distribution since each time interval is independent and the probability of an out-of-control signal is identical for each time interval. If ARL_0 denotes the in-control ARL, the upper control limit (UCL) is $\chi^2(1/ARL_0; p)$, the $100(1-(1/ARL_0))\%$ percentile of the χ^2 distribution with p degrees of freedom, if the $\bar{\mathbf{x}}$ is multivariate normal. The most frequent choice is $ARL_0 = 200$, so the upper control limit is the 95% percentile of the χ^2_p. When the process is out-of-control with mean $\boldsymbol{\mu}_1$, the multivariate Shewhart statistic has a non-central χ^2 distribution with p degrees of freedom and non-centrality parameter

$$\lambda = \sqrt{n(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)},$$

and the out-of-control ARL, denoted by ARL_1, can be computed $ARL_1 = 1/[1 - F(UCL; p, \lambda)]$, where $F(\cdot; p, \lambda)$ is the cdf of a non-central χ^2.

A frequent obstacle to applying the Shewhart χ^2 control chart is the need for the in-control variance-covariance matrix $\boldsymbol{\Sigma}_0$. The Hotelling T^2 distribution, a generalization of the Student's t distribution, allows the estimated variance-covariance matrix $\mathbf{S}$ to replace $\boldsymbol{\Sigma}_0$. The Shewhart T^2 control chart compares the statistic

$$T^2 = n(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)'\mathbf{S}^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu}_0)$$

to the upper control limit

$$\text{UCL} = \frac{p(n-1)}{n-p} F(1/\text{ARL}_0; p, n-p)$$

which uses a well-known relationship between the Hotelling T^2 distribution and the F distribution.

In many applications, the in-control mean $\boldsymbol{\mu}_0$ and the in-control variance-covariance matrix $\boldsymbol{\Sigma}_0$ are unknown, but are estimated from data collected while the process is believed to be in-control. For this Phase I data of m time periods of rational subgroup size n, Alt (1982) proposed estimating $\boldsymbol{\mu}_0$ by the mean of the m sample mean vectors, denoted by $\bar{\bar{\mathbf{x}}}$, and estimating $\boldsymbol{\Sigma}_0$ by the pooled variance-covariance matrix which is the mean of the m sample variance-covariance matrices, denoted by $\mathbf{S}_p$. Because the in-control parameters are estimated, the upper control limit is inflated to

$$UCL = \frac{p(m-1)(n-1)}{mn-m-p+1} F(1/\text{ARL}_0; p, mn-m-p+1).$$

If any time period in Phase I has an out-of-control signal and an assignable cause is found, this time period is omitted and $\bar{\bar{\mathbf{x}}}$ and $\mathbf{S}_p$ are recomputed. This step is iterated until all $m^* < m$ time periods are considered in-control.

At this time, the monitoring of future time periods begins by using the statistic

$$T_f^2 = n\left(\bar{\mathbf{x}}_f - \bar{\bar{\mathbf{x}}}\right)'\mathbf{S}_\mathbf{p}^{-1}\left(\bar{\mathbf{x}}_f - \bar{\bar{\mathbf{x}}}\right)$$

with

$$\text{UCL} = \frac{p(m^*+1)(n-1)}{m^*n-m^*-p+1} F(1/\text{ARL}_0; p, m^*n-m^*-p+1)$$

where $\bar{\mathbf{x}}_f$ is a vector of sample means based on data for a time period after m^*. It is suggested that $\bar{\bar{\mathbf{x}}}$ and $\mathbf{S}_p$ be

updated fairly often in the beginning, as the number of future subgroups accumulates.

A common follow-up to a large T^2 statistic is to use standardized coefficients of the discriminant function (Rencher 2002, Chap. 5). That is, compute

$$\mathbf{a} = \text{sqrt}[\text{diag}(\mathbf{S})] \cdot \mathbf{S}^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu}_0),$$

where *sqrt* is the elementwise square root of the vector and diag($\mathbf{S}$) creates a diagonal matrix from the diagonal elements of the $\mathbf{S}$ matrix. The absolute values of the coefficients in $\mathbf{a}$ give relative contributions of each quality measurement to T^2. Another approach to interpreting a large T^2 value is a decomposition proposed by Mason et al. (1995, 1997). The T^2 can be written as p independent terms, each of which reflects the contribution of an individual quality characteristic. Runger et al. (1996) use this decomposition to improve diagnostics of an out-of-control signal.

MEWMA Charts

The multivariate exponentially weighted moving average (MEWMA) control charts are well suited to observing a single observation ($n = 1$) at each time period t and combining the information from a window of time to make a decision. The generalization from the univariate EWMA was formulated by Lowry et al. (1992). A weighted average of the observed $\mathbf{x}_t$ is formed by

$$\mathbf{Z}_t = \lambda(\mathbf{x}_t - \boldsymbol{\mu}_0) + (1 - \lambda)\mathbf{Z}_{t-1}$$

where the value λ is chosen in designing the control chart to represent the amount of smoothing ($0 < \lambda < 1$) and $\mathbf{Z}_0 = 0$. Small values of λ pool the data over a wide time interval and produce a control chart that effectively identifies small, persistent changes from the in-control mean, $\boldsymbol{\mu}_0$, or a gradual drift from $\boldsymbol{\mu}_0$. Large values of λ yield a $\mathbf{Z}_t$ with high weight on the current observation so the control chart is sensitive to immediate large shifts from $\boldsymbol{\mu}_0$.

The MEWMA chart signals a process is out-of-control at time t when

$$T_t^2 = \mathbf{Z}_t \boldsymbol{\Sigma}_{\mathbf{Z}}^{-1} \mathbf{Z}_t$$

exceeds an upper control limit. The variance-covariance matrix $\boldsymbol{\Sigma}_{\mathbf{Z}}$ depends on λ and t, and is given by

$$\boldsymbol{\Sigma}_{\mathbf{Z}} = \left(\frac{\lambda\left[1 - (1 - \lambda)^{2t}\right]}{2 - \lambda} \right) \boldsymbol{\Sigma}_0,$$

where $\boldsymbol{\Sigma}_0$ is the in-control variance-covariance matrix of $\mathbf{x}_t$. For a given λ, the upper control limit is chosen to provide an ARL for a specified out-of-control mean $\boldsymbol{\mu}_1$. Tables of the ARL for different p, λ, and upper control limit are given by Prabhu and Runger (1997) for in-control $ARL_0 = 200$.

In the univariate case, the CUSUM (cumulative sum) control charts are quite similar to the EWMA control charts. Although a number of multivariate CUSUM procedures have been proposed, an early suggestion by Woodall and Ncube (1985) was to monitor each of the p quality characteristics simultaneously with individual CUSUM charts. The ARL of this collection of p CUSUM control charts is the minimum of $\{ARL_1, ARL_2, \ldots, ARL_p\}$ if the quality characteristics are independent. If the quality characteristics are correlated, reduce the p dimensional space to the $p' < p$ largest principal components. An improvement to this collection of p CUSUMs is to update the CUSUM at each observation and shrink toward the zero vector as described in Crosier (1988).

Control Charts for Variance-Covariance

While monitoring the mean of p correlated quality characteristics has been well researched, less work has been performed on control charts for the variance-covariance matrix (the generalization from univariate control charts on process variability). The most common approach summarizes the $p(p + 1)/2$ variances and covariances in $\boldsymbol{\Sigma}$ into a scalar by defining the generalized variance $|\boldsymbol{\Sigma}|$, which is the determinant of $\boldsymbol{\Sigma}$. Montgomery and Wadsworth (1972) proposed control limits based on the asymptotic normality of $|\mathbf{S}|$, the determinant of the sample variance-covariance matrix based on the n observations in the rational subgroup. Control limits for the typical Shewhart control charts were proposed by Alt (1985) and are $E(|\mathbf{S}|) \pm 3\sqrt{Var(|\mathbf{S}|)}$ where $E(|\mathbf{S}|) = b_1|\boldsymbol{\Sigma}|$ and $Var(|\mathbf{S}|) = b_2|\boldsymbol{\Sigma}|^2$ with

$$b_1 = \frac{1}{(n-1)^p} \prod_{i=1}^{p} (n-i)$$

and

$$b_2 = \frac{1}{(n-1)^{2p}} \left[\prod_{i=1}^{p} (n-i) \right] \times \left[\prod_{j=1}^{p} (n-j+2) - \prod_{j=1}^{p} (n-j) \right].$$

Profile Monitoring

Many manufacturing processes in the chemical process and semiconductor industries have finite-duration processing periods under controlled conditions which result in the final product. With improved metrology these processes can be monitored during the processing time. In these applications the collection of measurements taken on each quality characteristic during processing when plotted over time creates a profile.

Nomikos and MacGregor (1995a) organized the large amount of profile data as a three-dimensional array whose n rows correspond to the different runs, t columns correspond to the measurements taken over processing time for a given run and the third dimension (depth) is the p different quality characteristics. While this is perhaps the organization of the data in a database, multivariate statistical methods require the expression of $\mathbf{Y}$ as a vector, and an 'unfolded' structure generates a tp vector of each quality characteristic at each processing time. Instead of monitoring this extremely large vector, one approach is to reduce the dimensionality to a set of summary scores $\mathbf{T}$. Nomikos and MacGregor (1995a) use principal components of $\mathbf{Y}$ to form $\mathbf{T}$, and Nomikos and MacGregor (1995b) use partial least squares to obtain linear combinations of $\mathbf{Y}$ which are highly correlated with a product's quality measurements taken after processing. Grimshaw et al. (1998) allow changing inputs that affect the profile and provide a real-time processing control chart statistic.

When there is a hypothesized relationship between the profile and an explanatory variable, the profile can be modeled using the parameters of the relationship. For example, if the relationship is linear the estimated regression coefficients are monitored using a Hotelling T^2 following Kang and Albin (2000). In a Phase II control chart where the profile has been estimated from historical data, Kim et al. (2003) address the linear case. The nonlinear profile case has been modeled by multiple regression and higher-order polynomials in Zou et al. (2007) and Kazemzadeh et al. (2008); nonparametric regression methods are used in Zou et al. (2008); and nonlinear profiles for dose–response applications are in Jensen and Birch (2009). Colosimo et al. (2008) monitor profiles of geometric specifications such as roundness, cylindricity, and flatness.

See

- ▶ Quality Control
- ▶ Total Quality Management

References

Alt, F. B. (1982). Multivariate quality control: State of the art. *ASQC Quality Congress Transactions – Detroit, MI*, pp. 886–893.

Alt, F. B. (1985). Multivariate quality control. In S. Kotz & N. S. Johnson (Eds.), *Encyclopedia of statistical sciences* (Vol. 6, pp. 110–122). New York: John Wiley.

Colosimo, B. M., Semeraro, Q., & Pacella, M. (2008). Statistical process control for geometric specifications: On the monitoring of roundness profiles. *Journal of Quality Technology, 40*(1), 1–18.

Crosier, R. B. (1988). Multivariate generalizations of cumulative sum quality control scheme. *Technometrics, 30*(3), 291–303.

González, I., & Sánchez, I. (2010). Variable selection for multivariate statistical process control. *Journal of Quality Technology, 42*(3), 242–259.

Grimshaw, S. D., Shellman, S. D., & Hurwitz, A. M. (1998). Real-time process monitoring for changing inputs. *Technometrics, 40*(4), 283–296.

Hotelling, H. (1947). Multivariate quality control, Illustrated by the air testing of sample bombsights. In C. Eisenhart, M. W. Hastay & W. A. Willis (Eds.), *Selected techniques of statistical analysis*. New York: McGraw–Hill.

Jackson, J. E. (1956). Quality control methods for two related variables. *Industrial Quality Control, 12*, 2–6.

Jensen, W. A., & Birch, J. B. (2009). Profile monitoring via nonlinear mixed models. *Journal of Quality Technology, 41*(1), 18–34.

Kang, L., & Albin, S. L. (2000). On-line monitoring when the process yields a linear profile. *Journal of Quality Technology, 32*(4), 418–426.

Kazemzadeh, R. B., Noorossana, R., & Amiri, A. (2008). Phase I monitoring of polynomial profiles. *Communications in Statistics: Theory and Methods, 37*(10), 1671–1686.

Kim, K., Mahmoud, M. A., & Woodall, W. H. (2003). On the monitoring of linear profiles. *Journal of Quality Technology, 35*(3), 317–328.

Lowry, C. A., Woodall, W. H., Champ, C. W., & Rigdon, S. E. (1992). A multivariate exponentially weighted moving average control chart. *Technometrics, 34*(1), 46–53.

Mason, R. L., Tracy, N. D., & Young, J. C. (1995). Decomposition of T^2 for multivariate control chart interpretation. *Journal of Quality Technology, 27*(2), 99–108.

Mason, R. L., Tracy, N. D., & Young, J. C. (1997). A practical approach for interpreting multivariate T^2 control chart signals. *Journal of Quality Technology, 29*(4), 399–406.

Montgomery, D. C., & Wadsworth, H. M. (1972). Some Techniques for Multivariate Quality Control Applications. *ASQC Technical Conference Transactions*, Washington, D. C, pp. 427–435.

Nomikos, P., & MacGregor, J. F. (1995a). Multivariate SPC charts for monitoring batch processes. *Technometrics, 37*(1), 41–59.

Nomikos, P., & MacGregor, J. F. (1995b). Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems, 30*, 97–108.

Prabhu, S. S., & Runger, G. C. (1997). Designing a multivariate EWMA control chart. *Journal of Quality Technology, 29*(1), 8–15.

Rencher, A. C. (2002). *Methods of multivariate analysis*. New York: Wiley.

Runger, G. C., Alt, F. B., & Montgomery, D. C. (1996). Contributors to a multivariate statistical process control chart signal. *Communications in Statistics: Theory and Methods, 25*(10), 2203–2213.

Woodall, W. H., & Ncube, M. M. (1985). Multivariate CUSUM quality control procedures. *Technometrics, 27*(3), 285–292.

Zou, C., Tsung, F., & Wang, Z. (2007). Monitoring general linear profiles using multivariate EWMA schemes. *Technometrics, 49*(4), 395–408.

Zou, C., Tsung, F., & Wang, Z. (2008). Monitoring profiles based on nonparametric regression methods. *Technometrics, 50*(4), 512–526.

Music

► Digital Music