B

Backward Chaining

An approach to reasoning in which an inference engine endeavors to find a value for an overall goal by recursively finding values for subgoals. At any point in the recursion, the effort of finding a value for the immediate goal involves examining rule conclusions to identify those rules that could possibly establish a value for that goal. An unknown variable in the premise of one of these candidate rules becomes a new subgoal for recursion purposes.

See

► Expert Systems

Backward Kolmogorov Equations

In a continuous-time Markov chain with state $X(t)$ at time t, define $p_{ij}(t)$ as the probability that $X(t+s) = j$, given that $X(s) = i$, $s, t \geq 0$, and r_{ij} as the transition rate out of state i to state j. Then Kolmogorov's backward equations say that, for all states i, j and times $t \geq 0$, the derivatives $dp_{ij}(t)/dt = \sum_{k \neq i} r_{ik}\, p_{kj}\,(t) - v_i\, p_{ij}(t)$, where v_i is the transition rate out of state i, $v_i = \sum_j r_{ij}$.

See

► Markov Chains
► Markov Processes

Backward-Recurrence Time

Suppose events occur at times $T_1, T_2, \ldots$ such that the interevent times $T_k - T_{k-1}$ are mutually independent, positive random variables with a common cumulative distribution function. Choose an arbitrary time t. The backward recurrence time at t is the elapsed time since the most recent occurrence of an event prior to t.

Balance Equations

(1) In probability modeling, steady-state systems of equations for the state probabilities of a stochastic process found by equating transition rates. For Markov chains, such equations can be derived from the Kolmogorov differential equations or from the fact that the flow rate into a system state or level must equal the rate out of that state or level for steady state to be achieved. (2) In linear programming (usually referring to a production process model), constraints that express the equality of inflows and outflows of material.

See

► Markov Chains

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

Balking

When customers arriving at a queueing system decide not to join the line and instead go away because they anticipate too long a wait.

See

▶ Queueing Theory

Bandit Model

▶ Multi-armed Bandit Problem

Banking

Stavros A. Zenios
University of Cyprus, Nicosia, Cyprus
University of Pennsylvania, Pennsylvania, PA, USA

Introduction

OR/MS techniques find applications in numerous and diverse areas of operation in a banking institution. Applications include the use of data-driven models to measure the operating efficiency of bank branches through data envelopment analysis, the use of image recognition techniques for check processing, the use of artificial neural networks for evaluating loan applications, and the use of facility location theory for opening new branches and placing automatic teller machines (e.g., Harker and Zenios 1999). A primary area of application is that of financial risk control in developing broad asset/liability management strategies. Papers that summarize these areas are Zenios (1993), Jarrow et al. (1994), and Ziemba and Mulvey (1998). This work can be classified into three categories: (1) pricing contingent cashflows, (2) portfolio immunization, and (3) portfolio diversification.

Pricing Contingent Cashflows

The fundamental pricing equation computes the price of a contingent cashflow as the expected net present value of the cashflows, discounted by an appropriate discount rate. In discrete time the pricing equation takes the form

$$P_T = E_S\left\{\sum_{t=0}^{T}\frac{C_{t+1}^S}{1+r_t^S}\right\} \tag{1}$$

where E denotes expectation over the set of scenarios indicated by index s, C_t^S denotes the cashflow received at period t under scenario s, r_t^S is the spot rate for the same period under the scenario s, and T denotes the maturity date. The vector (r_t) is known as the term structure of interest rates. For risk-free cashflows, the appropriate discount rate is the rate implied by the Treasury yield curve. At any given point in time, vector (r_{0t}^S) can be obtained using market data; this is the current term structure scenario. However, the temporal variation of the term structure is stochastic. This stochastic interest rate behavior, together with potential uncertainties in the level of the cashflows (i.e., the scenarios C_t^S) are the primary challenging issues behind the evaluation of (1).

One major strand of research is devoted to the development of stochastic models for the term structure of interest rates. Cox, Ingersoll and Ross (1985) first described the interest rate dynamics via the (continuous) diffusion process

$$dr = \kappa(\mu - r)dt + \sigma\sqrt{r}d\omega \tag{2}$$

Here, μ is the mean and σ the variance of the stochastic interest rate process, and $d\omega$ is the differential of a standard Wiener process. This model exhibits mean reversion with a drift factor $\kappa(\mu - r)$, and guarantees that interest rates remain positive. It is, however, a single factor model: the term structure of interest rates is represented by a single state variable, namely the spot rate, r.

A two-factor model for bond prices was developed by Brennan and Schwartz (1979). They considered two state variables, the spot rate r and a long-term (consol) rate L. The dynamics of these two variables are described by

$$\begin{cases} dr = b_1(r,L,t)dt + a_1(r,L,t)d\omega_1 \\ dL = b_2(r,L,t)dt + a_2(r,L,t)d\omega_2 \end{cases} \tag{3}$$

Here, the drift factors are denoted by the functions $b_1(r, L, t)$ and $b_2(r, L, t)$, and the variance terms are expressed by $a_1(r, L, t)$ and $a_2(r, L, t)$. The elements $d\omega_1$ and $d\omega_2$ are differentials of standard Wiener processes.

Despite the elegance of continuous-time models, since most practical applications deal with discrete time cashflows, there is interest in the development of discrete models. A popular choice of discrete models is based on binomial lattices. Such models typically assume that interest rates can move to one of two possible states, up or down, from period t to $t + 1$. The probability and magnitude of each step are calibrated using the Treasury yield curve and the volatility implied by the prices of traded option instruments. Ho and Lee (1986) and Black, Derman and Toy (1990) proposed some fundamental models. For example, the Black, Derman and Toy model described the spot rates by the process

$$r_t^{\sigma} = r_t^0(\kappa_t)^{\sigma}.$$

Here r_t denotes the spot rate that takes values r_t^{σ} with possible states $\sigma = 0,1, \ldots, t$; r_t^0 is the ground state; and κ_t is the volatility of the spot rate in period t.

Models such as those described above generate the discount rates used in the pricing of riskless cashflows. For risky contingent cashflows (e.g., cashflows with credit, default, lapse, prepayment, and other such risks), the discount rates must be adjusted with a suitable riskpremium. Such premiums can be computed from the observed market prices of actively traded securities with comparable risks through the use of option adjusted analysis (Babbel and Zenios 1992).

Another important modeling issue in evaluating (1) is the forecasting of the cashflow stream (C_t). Statistical analysis and econometric modeling can be used in this context, especially when dealing with the various complex securities that have emerged in the 1980s and 1990s, like callable corporate bonds, mortgage and other assetbacked securities, and a range of insurance products. This kind of modeling was represented for insurance products by Asay, Bouyoucos and Marciano (1993), and for mortgage-backed securities by Kang and Zenios (1992).

Portfolio Immunization

This is a portfolio management strategy for locking in a fixed rate of return during a prespecified horizon. It assumes that all risk in the returns of the securities is systematic, that is, all risks are due to some common underlying factor(s). Portfolio immunization aims at eliminating this systematic risk. In the case of fixed-income securities, systematic risk is primarily due to changes in the term structure. Portfolio immunization traditionally deals with this type of risk.

The actuary F.M. Reddington (1952) was the first to introduce the notion of immunization, and also specified conditions for immunization. Portfolio immunization became a popular strategy in the 1970s at the aftermath of interest rate deregulation in the U.S. and the volatility of the fixed-income markets that followed. Fisher and Weil (1971) defined immunization as follows:

> A portfolio of investments is immunized for a holding period if its value at the end of the holding period, regardless of the course of rates during the holding period, is at least as large as it would have been had the interest rate function remained constant throughout the holding period.

A portfolio of assets used to fund a stream of liabilities can be immunized if the following conditions are met: (1) The present value of the assets is equal to the present value of the liabilities, and (2) the duration of the assets is equal to the duration of the liabilities. The first condition guarantees that the target liabilities are funded if the interest rates remain constant throughout the target period. The second condition guarantees that assets and liabilities have identical sensitivities to parallel shifts of the interest rates. Hence, the target liabilities will be funded even if the term structure experiences parallel shifts. A general overview of portfolio immunization was given in Fabozzi (1991). Linear programming formulations are often used to structure immunized portfolios, as in Zenios (1993).

Briefly, let r_i be the yield of the ith security, and C_{it} be the cashflow of security i at time t. From the fundamental pricing (1), obtain the price of the ith security by

$$P_i = \sum_{t=1}^{T} C_{it}(1 + r_i)^{-t}.$$

The sensitivity of the price — or *dollar duration* — of security i is obtained by differentiating with respect to cashflow yield, $(\partial P_i / \partial r_i)$, to get

$$k_i = -\sum_{t=1}^{T} tC_{it}(1 + r_i)^{-(t+1)}.$$

Given the present value P_L and dollar duration k_L of its liabilities, an immunized portfolio can be structured by solving the linear program

$$\begin{aligned} \text{Maximize} \quad & \sum_i k_i r_i x_i \\ \text{s.t.} \quad & \sum_i P_i x_i = P_L \\ & \sum_i k_i x_i = k_L \\ & x_i \geq 0 \end{aligned}$$

The objective function above maximizes an approximation to the portfolio yield, obtained as the dollar duration-weighted average yield of the individual securities in the portfolio. Several variations exist on the theme of portfolio immunization. One extension is to structure a portfolio that matches not only present value and duration of assets with those of the liabilities, but that also matches convexity, i.e., second partial derivatives $(\partial^2 P_i / \partial r_i^2)$ as well. Another approach is to compute the sensitivity of the prices to more than one factor, than just to parallel shifts of interest rates. The precise form of these factors (i.e., parallel shifts, steepening of the term structure, or term structure inversions) can be obtained using factor analysis of market data. Factor analysis of the term structure was first proposed for the U.S. market by Litterman and Scheinkman (1988). The use of linear programming for factor immunization was proposed by Dahl (1993) and D'Ecclesia and Zenios (1994).

Portfolio Diversification

The principle of diversification — based on the adage "do not put all your eggs in one basket" — remains a universal strategy for portfolio management. It provides a systematic way for dealing with residual risk, assuming that residual risk is accurately represented by a function of the mean and variance in the return of the securities. It also assumes that investors have an (implied) utility function over the mean and variance of portfolio returns, favoring portfolios with higher means and lower variances. The efficient portfolios for an investor are those that achieve the highest expected return for a given level of variance or the smallest possible variance for a given level of return. Such portfolios are called mean-variance efficient portfolios. Mean-variance optimization models were proposed by Markowitz in the 1950s; Ingersoll (1987) gives an advanced textbook treatment.

Minimum variance portfolios, i.e., portfolios with the lowest level of variance for a given target expected return, can be structured using nonlinear quadratic programming. Define

Q as the covariance matrix $\{q_{ij}\}$ between securities i and j,
μ_i as the expected return of security i,
μ_p as the target expected return of the portfolio, and
X_i as the fraction of the portfolio in security i.

Assuming that no short sales are allowed ($x_i \geq 0$ for all i), formulate the problem as

$$\begin{aligned} \text{Minimize} \quad & x^T Q_x \\ \text{s.t.} \quad & \sum_i \mu_i x_i = \mu_p \\ & \sum_i x_i = 1 \\ & x_i \geq 0 \end{aligned}$$

Other constraints, like limits on portfolio turnover, on minimum holdings, or limits of investments in different market segments, etc., can be captured with more complex formulations. These issues have been addressed by Perold (1984). See also the articles in Zenios (1993) and Ziemba and Mulvey (1998).

The major area of investigation in implementing minimum variance models in practice is in the estimation of the covariance matrix. Factor models that relate the returns and variances of individual securities to a set of common factors are widely used in practice (Elton and Gruber 1984).

Mean-variance models have traditionally been used in managing portfolios of equities and for strategic asset allocation. By contrast, fixed-income portfolio management has traditionally been based on the principles of portfolio immunization. In the 1980s,

however, there was a convergence of portfolio management tools towards the ideas of portfolio diversification. More complex fixed income securities (e.g., corporate callable bonds, high-yield bonds, mortgages and other asset-backed securities) have very volatile returns. The notion of duration, as a measure of sensitivity, is extremely restrictive for such instruments. Mulvey and Zenios (1993) advocated the use of diversification models for fixed-income portfolios, indicating how pricing models can be developed to generate scenarios of holding period returns in order to calibrate the models, and illustrating that such models produce better results than traditional portfolio immunization strategies.

Another development deals with the asymmetric returns of fixed-income securities, especially those with embedded options. Mean-variance models are valid assuming a symmetric distribution of return. Furthermore, they penalize both upside and downside deviations from a target return. Development of more practical models for dealing with asymmetric returns and penalizing differentially upside from downside risk include the mean-absolute deviation model of Konno and Yamazaki (1991), the expected utility optimization models of Grauer and Hakkanson (1985), and the dynamic, multiperiod models of Kallberg, White and Ziemba (1982), Mulvey and Vladimirou (1992), and Golub et al. (1995).

See

- ▶ Data Envelopment Analysis
- ▶ Facility Location
- ▶ Financial Engineering
- ▶ Financial Markets
- ▶ Linear Programming
- ▶ Neural Networks
- ▶ Portfolio Theory: Mean-Variance Model
- ▶ Quadratic Programming
- ▶ Utility Theory

References

Asay, M. R., Bouyoucos, P. J., & Marciano, A. M. (1993). An economic approach to valuation of single premium deferred annuities. In S. A. Zenios (Ed.), *Financial optimization* (pp. 100–135). Cambridge: Cambridge University Press.

Babbel, D. F., & Zenios, S. A. (1992). Pitfalls in the analysis of option-adjusted spreads. *Financial Analysts Journal, 48*, 65–69.

Birge, J., & Linetsky, V. (Eds.). (2007). *Handbooks in operations research and management science: Financial engineering*. Maryland Heights, MO: Elsevier Science.

Black, F., Derman, E., and Toy, W. (1990). A one-factor model of interest rates and its application to treasury bond options. *Financial Analysts Journal*, 33–39.

Brennan, M. J., & Schwartz, E. S. (1979). A continuous time approach to the pricing of bonds. *Banking and Finance Journal, 3*, 133–155.

Cornuejols, G., & Tutuncu, R. (2007). *Optimization methods in finance*. Cambridge: Cambridge University Press.

Cox, J. C., Jr., Ingersoll, J. E., & Ross, S. A. (1985). A theory of the term structure of interest rates. *Econometrica, 53*, 385–407.

Dahl, H. (1993). A flexible approach to interest-rate risk management. In S. A. Zenios (Ed.), *Financial optimization* (pp. 189–209). Cambridge: Cambridge University Press.

D'Ecclesia, R., & Zenios, S. A. (1994). Factor analysis and immunization in the Italian bond market. *Journal of Fixed Income, 4*, 51–58.

Elton, E., & Gruber, M. (1984). *Modern Portfolio theory and investment analysis*. New York: Wiley.

Fabozzi, F. J. (Ed.). (1991). *The handbook of fixed-income securities*. Homewood, IL: Business One Erwin.

Fisher, L., & Weil, R. (1971). Coping with the risk of interest-rate fluctuations: Returns to bondholders from naive and optimal strategies. *Journal of Business, 44*, 408–431.

Golub, B., Holmer, M., McKendall, R., Pohlman, L., & Zenios, S. A. (1995). Stochastic programming models for money management. *European Journal of Operational Research, 85*, 282–296.

Grauer, R. R., & Hakansson, N. H. (1985). Returns on levered actively managed long-run portfolios of stocks, bonds and bills. *Financial Analysts Journal, 41*, 24–43.

Harker, P. T., & Zenios, S. A. (1999). *Performance of financial institutions: Efficiency, innovation, regulation*. Cambridge: Cambridge University Press.

Ho, T. S. Y., & Lee, S.-B. (1986). Term structure movements and pricing interest rate contingent claims. *Journal of Finance, 41*, 1011–1029.

Ingersoll, J. E., Jr. (1987). *Theory of financial decision making. Studies in financial economics*. Lanham, MA: Row-man and Littlefield.

Jarrow, R., Maksimovic, M., & Ziemba, W. (Eds.). (1994). *Handbooks in operations research and management science: finance*. Amsterdam: North Holland.

Kallberg, J. G., White, R. W., & Ziemba, W. T. (1982). Short term financial planning under uncertainty. *Management Science, 28*, 670–682.

Kang, P., & Zenios, S. A. (1992). Complete pre-payment models for mortgage backed securities. *Management Science, 38*, 1665–1685.

Konno, H., & Yamazaki, H. (1991). A mean-absolute deviation portfolio optimization model and its applications to the Tokyo stock market. *Management Science, 37*, 519–531.

Litterman, R., & Scheinkman, J. (1988). *Common factors affecting bond returns*. Technical report, Goldman, Sachs & Co., Financial Strategies Group, September.

Markowitz, H. (1952). Portfolio selection. *Journal of Finance, 7*, 77–91.
Mulvey, J. M., & Vladimirou, H. (1992). Stochastic network programming for financial planning problems. *Management Science, 38*, 1643–1664.
Mulvey, J. M., & Zenios, S. A. (1994). Capturing the correlations of fixed-income instruments. *Management Science, 40*, 1329–1342.
Perold, A. F. (1984). Large-scale portfolio optimization. *Management Science, 30*, 1143–1160.
Reddington, F. M. (1952). Review of the principles of life-office valuations. *Journal of Institute Actuaries, 78*, 286–340.
Scott, F.R., Roll, R. (1989). Prepayments on fixed-rate mortgage-backed securities. *Journal of Portfolio Management*, Spring, 73–82.
Zenios, S. A. (Ed.). (1993). *Financial optimization*. Cambridge: Cambridge University Press.
Ziemba, W. T., & Mulvey, J. M. (1998). *Worldwide asset and liability modeling*. Cambridge: Cambridge University Press.

Bar Chart

▶ Gantt Charts
▶ Quality Control

Barrier Functions and their Modifications

Roman A. Polyak
George Mason University, Fairfax, VA, USA

Introduction

In the mid-1950s and the early 1960s, Frisch (1955) and Carroll (1961) proposed the use of Barrier Functions (BFs) for constrained optimization. Since then, the BFs have been extensively studied, with particularly major work in the area due to Fiacco and McCormick (1968) who developed the Sequential Unconstrained Minimization Technique (SUMT). Currently, methods based on barrier functions make up a considerable part of modern optimization theory.

Barrier Functions

Consider the constrained optimization problem

$$x^* \in X^* = \arg\min\{f(x) | x = \in \Omega\} \tag{1}$$

where $\Omega = \{x: g_i(x) \geq 0, i = 1, \ldots, m\}$, $f: \Re^n \to \Re$ is convex, all $g_i: \Re^n \to \Re$ are concave, $m > n$, and X^* is the set of values minimizing $f(x)$ on Ω. Frisch's logarithmic barrier function F: int $\Omega \times \Re_{++} \to \Re$ is defined by formula

$$F(x, \mu) = f(x) - \mu\Sigma \ln g_i(x) \tag{2}$$

and Carroll's hyperbolic barrier function C: int $\Omega \times \Re_{++} \to \Re$ is defined as

$$C(x, \mu) = f(x) + \mu\Sigma \ln g_i^{-1}(x).$$

Assume that X^* is bounded and $\ln t = -\infty$ for $t \leq 0$; then for any $\mu > 0$, there exists a minimum of $F(x, \mu)$ in $\Re^n$, denoted by

$$(x, \mu) = \arg\min\{F(x, \mu) | x \in \Re^n\}. \tag{3}$$

Therefore

$$\begin{aligned} \nabla_x F(x, \mu), \mu) &= \nabla f(x(\mu)) - \Sigma\mu g_i^{-1}(x(\mu))\nabla g_i(x(\mu)) \\ &= \nabla f(x(\mu)) - \sum \lambda_i(\mu)\nabla g_i(x(\mu)) \\ &= \nabla_x L(x(\mu), \lambda(\mu)) = 0 \end{aligned} \tag{4}$$

where $L(x, \lambda) = f(x) - \sum \lambda_i g_i(x)$ is the Lagrangian for the problem (1). Also $g_i(x(\mu)) > 0, i, = 1, \ldots, m$ and

$$\lambda_i(\mu) = \mu g_i^{-1}(x(\mu)) > 0, \quad i = 1, \ldots, m. \tag{5}$$

Hence $x(\mu) \in$ int Ω, $\lambda(\mu) = (\lambda_i(\mu), i, = 1, \ldots, m) \in \Re^m{}_{++}$ and due to (4)

$$L(x(\mu), \lambda(\mu)) = \min\{L(x, \lambda(\mu) | x \in \Re^n\}.$$

Consider the dual problem to (Eq. 1)

$$\lambda^* \in L^* = \text{Arg}\max\{d(\lambda) | \lambda \in \Re_+^m\} \tag{6}$$

where $d(\lambda) = \min\{L(x, \lambda) \mid x \in \Re^n\}$ and L^* is the set of maxima of $d(\lambda)$ on $\Re^m$. The vector $x(\mu)$ is interior primal, the vector $\lambda(\mu)$ is interior dual, and due to (Eq. 5) the primal-dual gap is

$$\begin{aligned} \Delta(\mu) = f(x(\mu)) - d(\lambda(\mu)) &= f(x(\mu)) - L(x(\mu), \lambda(\mu)) \\ &= \Sigma\lambda_i(\mu)g_i(x(\mu)) = m\mu. \end{aligned}$$

Therefore

$$\mu \to 0 \Rightarrow \Delta(\mu) \to 0 \Rightarrow f(x(\mu)) \to f(x^*) \text{ and } d(\lambda(\mu)) \to d(\lambda^*).$$

The primal barrier trajectory $\{x(\mu)\}$ and the primal-dual trajectory $\{x(\mu), \lambda(\mu)\}$ are critical elements in both SUMT (Fiacco and McCormick 1968) and recent developments in Interior Point Methods (IPMs).

Interest in barrier and distance functions was revived after N. Karmarkar (1984) published his polynomial projective scaling method for linear programming (LP) calculations. The connection between Karmarkar's method and the Newton log barrier method for LP calculations was discovered by Gill, Murray, Saunders, Tomlin and Wright (1986). Since then the interest to BFs grew dramatically and IPMs became the main stream in modern optimization. Hundreds of papers and several books have been published recently on the matter (see Nesterov and Nemirovsky 1994; Roos et al. 1997; Wright 1997; Ye 1997).

The main idea of the path-following IPMs (see Gonzaga 1992; Renegar 1988) is to replace in a sense the unconstrained minimization problem (Eq. 3) by one Newton step for solving the system $\Delta_x F(x, \mu) = 0$. The basic path-following IPM consists of performing a Newton step toward the solution $x(\mu)$ of the system

$$\nabla_x F(x, \mu) = 0 \tag{7}$$

followed by the barrier parameter update.

For a given $\mu > 0$ one finds an approxmation x for $x(\mu)$, the so-called "warm" start. The warm start belongs to the area where Newton method for the system (Eq. 7) is well-defined (Smale 1986), that is, from x as a starting point the method

$$\hat{x} = x - (\nabla^2_{xx} F(x, \mu))^{-1} \nabla_x F(x, \mu) \tag{8}$$

converges to $x(\mu)$ quadratically. The step of the path-following method consists in replacing x by $\hat{x}$ and μ by $\hat{\mu} = \mu(1 - \alpha/\sqrt{m})$ where $\alpha > 0$ is independent on $m > n$.

In the late 1980s Nesterov and Nemirovsky (1994) discovered the self-concordant property of the function $F(x, \mu)$ for important classes of constrained optimization problems including LP, QP, and QP with quadratic constraints. A function ϕ: int $\Omega \to \Re$ is self-concordant if it is convex, three times differentiable, and for any $x \in$ int Ω any $h \in \Re^n$ on the interval $I = \{t/x + th \in \text{int } \Omega\}$, the function ϕ: $I \to R$ defined by $\phi(t) = \phi_{x,\, h}(t) = \phi(x + th)$ satisfies the following inequality

$$\varphi'''(0) \le 2(\varphi''(0))^{3/2}.$$

The self-concordant property guarantees that if x is well defined for the system $\Delta_x F(x, \mu) = 0$ then $\hat{x}$ will be well defined for the system $\nabla_x F(x, \hat{\mu}) = 0$. The polynomial complexity of the path-following method for LP follows immediately from the fact that each Newton step shrinks the primal-dual gap by $(1 - \alpha/\sqrt{m})$, where $\alpha > 0$ is independent on m.

The primal-dual algorithms have emerged as the most important and useful class of IPMs (see Wright 1997). On the computational side, the most successful implementation (see Lustig et al. 1992) is based on the Mehrotra predictor-corrector algorithm (Mehrotra 1992). The BFs became the basic tool in the IPM, but the BFs still have their inherent drawbacks: these function, as well as their derivatives, do not exist at the solution; and they grow infinitely large together with the condition number of their Hessians when the approximation approaches the solution and the area where the Newton method is well-defined shrinks to a point.

To eliminate the drawbacks, while still retaining the nice properties of the barrier functions, modified barrier functions (MBFs) were introduced in the early 1980s for both LP and NLP calculations (Polyak 1986, 1992, 1996). The MBFs are particular cases of the Nonlinear Rescaling Principle, which consists of transforming the objective function and/or the constraints into an equivalent problem and using the classical Lagrangian for the equivalent problem in both theoretical analysis and numerical methods (Polyak 1986).

Modified Barrier Functions

Consider the constrained optimization problem

$$g_i(x) \ge 0, i = 1, \ldots, m \tag{9}$$

is equivalent to $\mu \ln(\mu^{-1} g_i(x) + 1) \geq 0$, $i = 1, \ldots, m$. Therefore problem (1) is equivalent to

$$x^* \in X^* = \operatorname{Argmin}\{f(x) / \mu \ln(\mu^{-1} g_i(x) + 1) \geq 0, \quad i = 1, \ldots, m\} \tag{10}$$

where the constraints are transformed by $\psi(t) = \ln(t + 1)$, and rescaled by $\mu = 0$. The classical Lagrangian for the equivalent problem (10)

$$F(x, \lambda, \mu) = f_0(x) - \mu \Sigma \lambda_i \ln(\mu^{-1} g_i(x) + 1),$$

is the logarithmic MBF which corresponds to Frisch's log-barrier function (2). For any $\mu > 0$, the system (9) is equivalent to

$$\mu[(\mu^{-1} g_i(x) + 1)^{-1} - 1] \leq 0, i = 1, \ldots, m$$

where the constraints transformation is given by $v(t) = (t + 1)^{-1} - 1$. The classical Lagrangian for the equivalent problem is the hyperbolic MBF

$$C(x, \lambda, \mu) = f_0(x) + \mu \Sigma \lambda_i [(\mu^{-1} g_i(x) + 1)^{-1} - 1],$$

which corresponds to Carroll's hyperbolic barrier function (3).

The MBF's properties make them fundamentally different from the BFs. The MBFs, as well as their derivatives, exist at the solution, and for any Karush-Kuhn-Tucker pair (x^*, λ^*) and any $\mu > 0$, the following critical properties hold:

P1. $F(x^*, \lambda^*, \mu) = C(x^*, \lambda^*, k) = f_0(x^*)$;

P2. $\nabla_x F(x^*, \lambda^*, \mu) = \nabla_x C(x^*, \lambda^*, \mu) = \nabla_x L(x^*, \lambda^*) = 0$;

P3. $\nabla_{xx} F(x^*, \lambda^*, \mu) = \nabla_{xx} L(x^*, \lambda^*) + \mu^{-1} \nabla g^T(x^*) \Lambda^* \nabla g(x^*)$,

$\nabla_{xx} C(x^*, \lambda^*, \mu) = \nabla_{xx} L(x^*, \lambda^*) + 2\mu^{-1} \nabla g^T(x^*) \Lambda^* \nabla g(x^*)$.

where $\wedge = \operatorname{diag}(\lambda_i)$ and $\Delta g(x) = J[g(x)]$ is the Jacobian of the vector-function $g(x)^T = (g_i(x), i = 1, \ldots, m)$.

The MBF's properties resemble that of augmented Lagrangians (Bertsekas 1982; Golshtein and Tretyakov 1974; Hestenes 1969; Mangasarian 1975; Polyak and Tretyakov 1973; Powell 1969; Rockafellar 1973). One can consider the MBFs as interior augmented Lagrangians. At the same time, MBFs have some distinctive features, which make them different from both quadratic augmented Lagrangian (Rockafellar 1973) and nonquadratic augmented Lagrangian (Bertsekas 1982). The MBFs' properties lead to the following multipliers method.

Let $\mu > 0$, $\lambda^0 = e = (1, \ldots, 1) \in \Re^m$ and $x^0 \in \Omega_\mu = \{x | g_i(x) \geq -\mu, i = 1, \ldots, m\}$. The logarithmic MBF method consists of generating two sequences $\{x^s\}$ and $\{\lambda^s\}$:

$$x^{s+1} \in \arg\min\{F(x, \lambda^s, \mu) | x \in \Re^n\} \tag{11}$$

and

$$\lambda^{s+1} = \operatorname{diag}[\mu^{-1} g_i(s^{s+1}) + 1]^{-1} \lambda^s. \tag{12}$$

There is a fundamental difference between the logarithmic MBF method and SUMT or other IPM that is based on BFs. The MBF method converges to the primal-dual solution with *any* fixed $\mu > 0$ for any convex programming which has bounded optimal primal and dual solutions (Jensen and Polyak 1994). Moreover, for LP calculations, M. Powell proved that for any fixed barrier parameter, the MBF method produces such primal sequences that the objective function tends to its optimal value and constraints violations tend to zero with R-linear rate (Powell 1995).

If the second order optimality conditions hold then the primal-dual sequence converges with Q-linear rate:

$$\max\{\| x^{s+1} - x^* \|, \| u^{s+1} - u^* \|\} \leq c\mu \| u^s - u^* \| \tag{13}$$

where $c > 0$ is the condition number of the constrained optimization problem, which depends on the input data and the size of the problem, but it is independent on $\mu > 0$ (Polyak 1992).

The numerical realization of the MBF method leads to the Newton MBF. The Newton method is used to find an approximation for x^s, followed by the Lagrange multiplier update. Due to the convergence of the MBF method under the fixed barrier parameter $\mu > 0$, both the condition number of the MBF Hessian and the area where the Newton method is well defined remain stable. These properties contribute to both numerical stability and complexity, and they lead to the discovery of the "hot" start phenomenon in constrained optimization. It means that from some

point on for large classes of nondegenerate-constrained optimization problems including LP, QP and QP with quadratic constraints, the approximation for the primal minimizer will remain in the Newton area after each Lagrange multipliers update (Polyak 1992; Melman and Polyak 1996).

Due to (13) from the "hot" start on it takes only (ln ln ε^{-1}) Newton steps to improve the primal-dual approximation by a given factor $0 < q < 1$ as soon as $\mu \leq qc^{-1}$. The neighborhood of (x^*, λ^*) where the "hot" start occurs can be characterized by the condition number $c > 0$. Using the IPM with the Shifted Barrier Function (SBF) $S(x, \mu) = f(x) - \mu \sum \ln(\mu^{-1} g_i(x) + 1)$, which is self-concordant for the same classes of problem as $F(x, \mu)$, it takes $0(\sqrt{m} \ln c)$ to reach the "hot" start.

Combining the IPM based on SBF with the Newton MBF method, it is possible to improve substantially the complexity bounds for nondegenerate LP, QP and QP with quadratic constraints. In particular, for nondegenerate QP the total number of Newton step sufficient to obtain an approximation for (x^*, λ^*) with accuracy $\varepsilon = 2^{-L}$ is

$$N = 0(\sqrt{m} \ln c) + 0((L - \ln c) \ln m),$$

where L is the input length, $c > 0$ is the condition number of QP and $n < m$ (Melman and Polyak 1996).

The MBF method has an interesting dual interpretation. Assuming that the dual function $d(\lambda)$ is differentiable,

$$\Delta d(\lambda) = -g(x(\lambda))$$

where $x(\lambda) = \arg\min\{L(x, \lambda) | x \in \Re^n\}$ and $g(x(\lambda)) = (g_i(x(\lambda)), i = 1, \ldots, m)$, that is,

$$\nabla d(\lambda^{s+1}) = -g(x^{s+1}). \tag{14}$$

From the formula (12) for the Lagrange multipliers update

$$\begin{aligned} g_i(x^{s+1}) &= \mu \psi'^{-1}(\lambda_i^{s+1}/2\lambda_i^s) \\ &= \mu \psi^{*\prime}(\lambda_i^{s+1}/2\lambda_i^s), i = 1, \ldots, m \end{aligned} \tag{15}$$

where $\psi^*(s) = \inf\{st - \psi(t)\} = 1 - s + \ln s$ is the Legendre transformation of $\psi(t) = \ln(t + 1)$. Using (15), rewrite (14) as

$$\nabla d(\lambda^{s+1}) + \mu \Sigma \psi^{*\prime}(\lambda_i^{s+1}/\lambda_i^s) e_i = 0$$

where $e_i = (0, \ldots, 1, \ldots, 0)$. Hence,

$$\begin{aligned} \lambda^{s+1} &= \arg\max\{d(\lambda) + \mu \Sigma \lambda_i^s \psi^*(\lambda_i/\lambda_i^s)/\lambda \in \Re_+^m\} \\ &= \arg\max\{d(\lambda) - \mu D(\lambda, \lambda^s)/\lambda \in \Re_+^m\} \end{aligned} \tag{16}$$

where $D(\lambda, \lambda^s) = \sum \lambda_i^s \varphi(\lambda_i/\lambda_i^s)$ – a ϕ -divergence entropy-like distance with the kernel $\phi = -\psi^*$. Note that (16) is an IPM for the dual problem (see Teboulle 1993; Polyak and Teboulle 1997).

The formula (12) is in fact a method for solving the dual problem (6). It can be rewritten as

$$\lambda_i^{s+1}(1 - \mu^{-1} \nabla_{\lambda_i} d(\lambda_i^{s+1})) = \lambda_i^s, i = 1, \ldots, m. \tag{17}$$

Such a method is a well-known multiplicative image reconstruction algorithm for positron emission tomography (Eggermont 1990). On the other hand, it is nothing but the implicit Euler method for numerical solution of the following system of ordinary differential equations

$$\frac{d\lambda_i}{dt} = \mu^{-1} \lambda_i \frac{\partial d(\lambda)}{d\lambda_i}, \lambda(0) = \lambda_0, \ i = 1, \ldots, m$$

and $\lim_{t \to \infty} \lambda(t) = \lambda^*$, which is the solution of following nonlinear complementarity problem

$$\begin{aligned} \nabla d(\lambda) \leq 0, \ \lambda \leq 0 \\ \lambda^T \nabla d(\lambda) = 0. \end{aligned}$$

See

- Classical Optimization
- Computational Complexity
- Interior-Point Methods for Conic-Linear Optimization
- Nonlinear Programming

References

Bertsekas, D. (1982). *Constrained optimization and lagrange multipliers methods*. New York: Academic.

Carroll, C. (1961). The created response surface technique for optimizing nonlinear restrained systems. *Operations Research, 9*, 169–184.

Eggermont, P. (1990). Multiplicative iterative algorithm for convex programming. *Linear Algebral and its Applications, 130*, 25–42.
Fiacco, A. V., & McCormick, G. P. (1968). *Nonlinear programming: Sequential unconstrained minimization techniques*. New York: Wiley.
Frisch, K. (1955). *The logarithmic potential method of convex programming*. Technical Memorandum, May 13, University Institute of Economics, Oslo.
Gill, P., Murray, W., Saunders, M., Tomlin, J., & Wright, M. (1986). On projected barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Mathematical Programming, 36*, 183–209.
Golshtein, E. G., & Tretyakov, N. V. (1974). Modified Lagrangean functions. *Economics and Mathematical Methods, 10*, 568–591 (Russian).
Gonzaga, C. (1992). Path following methods for linear programming. *SIAM Review, 34*, 167–224.
Hestenes, M. (1969). Multiplier and gradient methods. *Journal of Optimization Theory and Applications, 4*, 303–320.
Huard, P. (1967). Resolution of mathematical programming with nonlinear constraints by the method of centers. In J. Abadie (Ed.), *Nonlinear programming*. Amsterdam: North-Holland.
Jensen, D., & Polyak, R. (1994). The convergence of the modified barrier method for convex programming. *IBM Journal of Research and Development, 38*, 307–321.
Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica, 4*, 373–395.
Lustig, I., Marsten, R., & Shanno, D. (1992). On implementing Mehrotra's predictor-corrector interior point method for linear programming. *Siam Journal of Optimization, 2*, 435–449.
Mangasarian, O. (1975). Unconstrained Lagrangians in nonlinear programming. *Siam Journal of Control, 13*, 772–791.
Mehrotra, S. (1992). On the implementation of primal-dual interior point method. *Siam Journal of Optimization, 2*, 575–601.
Melman, A., & Polyak, R. (1996). The Newton modified barrier method for QP. *Annals of Operations Research, 62*, 465–519.
Nesterov, Y., & Nemirovsky, A. (1994). *Interior point polynomial algorithms in convex programming*. Philadelphia: SIAM Studies in Applied Mathematics.
Polyak, B. T., Tretyakov, N. V. (1973). The method of penalty bounds for constrained extremum problems. Zh. Vych. Mat. iMat. Fiz, 13, 34–46, *USSR Computational Methods Mathematics and Physics 13*, 42–58.
Polyak, R. (1986). *Controlled processes in extremal and equilibrium problems*. Moscow (Russian): VINITY.
Polyak, R. (1992). Modified Barrier functions (theory and methods). *Mathematical Programming, 54*, 177–222.
Polyak, R. (1996). *Modified Barrier functions in linear programming*. Research Report, Department of Operations Research, George Mason University, pp. 1–56.
Polyak, R. (1997). Modified interior distance functions. *Contemporary Mathematics, 209*, 183–209.
Polyak, R., & Teboulle, M. (1997). Nonlinear rescaling and proximal-like methods in convex optimization. *Mathematical Programming, 76*, 265–284.
Powell, M. (1969). A method for nonlinear constraints in minimization problems. In R. Fletcher (Ed.), *Optimization*. New York: Academic.
Powell, M. (1995). Some convergence properties of the modified log barrier method for linear programming. *SIAM Journal of Optimization, 5*, 695–739.
Renegar, J. (1988). A polynomial-time algorithm, based on Newton's method for linear programming. *Mathematical Programming, 40*, 59–93.
Rockafellar, R. T. (1973). The multiplier method of hestenes and powell applied to convex programming. *Journal of Optimization Theory and Applications, 12*, 555–562.
Roos, C., Terlaky, T., & Vial, J.-P. (1997). *Theory and algorithms for linear optimization: an interior point approach*. New York: Wiley.
Smale, S. (1986). A Newton method estimates from data at one point. In R. Ewing (Ed.), *The merging of disciplines in pure, applied and computational mathematics*. New York/Berlin: Springer.
Teboulle, M. (1993). Entropic proximal mappings with application to nonlinear programming. *Mathematics of Operations Research, 17*, 670–690.
Wright, S. (1997). *Primal-dual-interior point methods*. Philadelphia: SIAM.
Ye, Y. (1997). *Interior point algorithms: Theory and analysis*. New York: Wiley.

Basic Feasible Solution

A nonnegative basic solution to a set of $(m \times n)$ linear equations $\boldsymbol{Ax} = \boldsymbol{b}$, where $m \leq n$. The major importance of basic feasible solutions is that, for a linear-programming problem, they correspond to extreme points of the convex set of solutions. The simplex algorithm moves through a sequence of adjacent extreme points (basic feasible solutions).

See

- ▶ Adjacent (Neighboring) Extreme Points
- ▶ Basic Solution
- ▶ Linear Programming

Basic Solution

For a set of $(m \times n)$ linear equations $\boldsymbol{Ax} = \boldsymbol{b}$ $(m \leq n)$, with rank m, a basic solution is a solution obtained by setting $(n - m)$ variables equal to zero and solving for the remaining m variables, provided that the column vectors associated with the m variables form a linearly independent set of vectors. The m variables are called basic variables, and the remaining $n - m$

variables that were set equal to zero are called nonbasic variables. The vectors associated with the basic variables form an $(m \times m)$ basis matrix $\boldsymbol{B}$.

See

► Linear Programming

Basic Variables

The set of variables corresponding to the columns of a basis matrix in a linear system $\boldsymbol{Ax} = \boldsymbol{b}$.

See

► Basic Solution
► Basis
► Linear Programming

Basis

A nonsingular square matrix $\boldsymbol{B}$ obtained by selecting linearly independent columns of a full row rank matrix $\boldsymbol{A}$. The matrix $\boldsymbol{B}$ is then a basis matrix for the system $\boldsymbol{Ax} = \boldsymbol{b}$. The components of $\boldsymbol{x}$ associated with $\boldsymbol{B}$ are called the basic variables, and the remaining components are called the nonbasic variables. The term basis also refers to the set of indices of the basic variables.

See

► Basic Variables
► Linear Programming

Basis Inverse

The inverse of a basis matrix.

See

► Basis
► Linear Programming

Basis Vector

A column of a basis matrix.

See

► Basis
► Linear Programming

Batch Shops

► Production Management

Battle Modeling

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

Introduction

The ideal battle model completely, accurately, quickly, and easily predicts the results of any postulated battle from the initial conditions. Several factors prevent the existence of an ideal battle model.

One factor is computational complexity. For example, medical planners could use such a battle model to determine the size of treatment facilities, the breakdown of physician skills needed, and the medical supply inventory requirements. It is reasonable to suppose a battle model would track individuals and their separate wounds for engagements of a dozen participants on a side; however, maintaining that level of detail for engagements of tens of thousands of people would be prohibitively expensive in time and hardware requirements. Thus the requirement for complete predictions competes with the requirements for generality and speed of computation.

The second factor preventing the existence of an ideal battle model is the fact that not enough is known about battle dynamics to model it accurately. Where components can be modeled accurately (e.g., firing disciplines for weapons and probabilities of kills given hits), it is not known how the components fit together (e.g., when do soldiers fire their weapons and

how do conditions modify their ideal performance). Further, it is not known when, where, and why battles are joined or when and how they stop. The ignorance is not absolute, but is relative to the desired accuracy for the battle models.

A third factor also proceeds from ignorance. It is not known which initial conditions are significant for determining battle results. In general, those battle models that deliver massive details about the model results require extremely large quantities of input data. Thus, perceived accuracy of results is a competitor of ease and rapidity of use.

Battle Model Classification

Although the ideal battle model cannot be built, many individual battle models can be built, each conceived to fulfill a particular set of objectives. These models of combat may be classified by their position along several dimensions; however, they all have one feature in common, and that is the object that is modeled is some aspect of combat. These dimensions are listed below with illustrative examples of positions along the dimension.

DOMAIN	Land; air; naval; space; combinations.
SPAN (size of conflict)	Platoon battle; division combat; theater-level combat; global combat.
SCOPE (type of conflict)	Politico-military; special operations; low intensity conflict; urban warfare; conventional warfare; theater-level nuclear, chemical, and biological conflict; strategic nuclear conflict.
SCORING (adjudication topics and methodology)	Measures of merit: attrition, movement, tons of bombs dropped, supplies delivered, victory; methodologies: weapon weights (simple or complex, as in anti-potential potential, which uses eigenvalues to value weapon by the value of the weapons it can kill), process simulations.
RANDOMNESS	Deterministic or stochastic calculations.
COMBAT ACTIVITIES AND FORCE COMPOSITION (military assets and mission areas)	Small-arms; armor; aircraft; artillery; engineer; logistics; signal; command and control; intelligence; surface navy; submarine; electronic warfare; space assets; missiles.
LEVEL OF RESOLUTION OR DETAIL (smallest item modeled as a separate entity)	Bullet; soldier; tank; platoon; company; battalion; brigade; division; corps.
ENVIRONMENT	One-dimensional terrain (pistonmodel); two-dimensional terrain (including ocean or air), latitude-longitude or hexagonal grid-based; three-dimensional terrain; weather; day-night; smoke.
PURPOSE (design purpose or users' purpose)	Training; weapon system employment; force composition decisions; operations plans testing.
LEVEL OF TRAINING (training audience)	Individual skills; platoon leaders' skills; division staff skills; commanders' skills; combinations.
MODEL TREATMENT OF TIME	Linear code with no time representation or algorithmically computed time (generally analytic combat models); time-stepped simulations; event-driven simulations; expected value models; stochastic simulations.
HUMAN INTERACTION	Data preparation and output interpretation; interruptible with modification and restart; computer-assisted human participation on one or more sides; continuous human participation on all sides.
SIDEDNESS	One-sided (e.g., strategic nuclear strike damage effects); two-sided; multi-sided; hard-coded identical properties for each side, hard-coded different properties for each side (e.g., U.S. vs Soviet-style tactics), or data-driven properties for each side.
COMPUTER INVOLVEMENT	None; moderate; complete.
SIZE COMPUTER REQUIRED	PC; mini-computer; mainframe; supercomputer; peripheral equipment required; large run-times, small run-times.
EXTERNAL INTERACTIONS (interfaces with parts of the real world)	None; distributed processing; interfaces with weapon simulators; interfaces with real equipment; sand tables, scripting.

Battle modeling started the first time someone scratched a battle plan in the dirt and tried to conceive of the consequences. Sand tables, with miniature troops and landscaping, added discipline to the modeling process; however, the modeling remained essentially qualitative. Sand table models were used as war games, in which opposing players took turns moving the pieces and used rules to

adjudicate the results of the moves. Modern war games include sand table games and computer adjudicated games.

Attrition Laws

Lanchester (1916) introduced the concept of a quantitative model of attrition. (Osipov in Russia and Fiske in the U.S. introduced similar concepts at virtually the same time; however, most Western works refer to Lanchester's Laws and Lanchestrian attrition.) Lanchester showed that one could express the value of concentration of forces precisely, using mathematics, and thus evaluate what forces would be needed for victory before a battle. Engel (1954) provided what many took to be proof that Lanchester's square law was correct.

Lanchester's simple concepts have been elaborated to the extent that Taylor (1983) required two volumes to discuss the many uses and implications of Lanchester theory. The computational power of computers has permitted this elaboration. First, heterogeneous Lanchester equations could be solved without undue manual labor. Once, heterogeneous equations were admitted, the coefficients could be represented as functions of other factors, such as weather, firing discipline, and distance to the target. Bonder and Farrel (in Taylor 1983) introduced rigorous thinking into this area by observing direct fire activities and creating a mathematical model of those activities.

Dupuy (1985) argued that there are many important factors in combat that were not being included in the physics-based combat models. Morale, training, and leadership are at least as important as force sizes according to Dupuy. He proposed a model based on quantified judgments of these and other "soft" factors. His Quantified Judgment Model (QJM) stirred considerable controversy. Regardless of the merits of the QJM itself, the quantified judgments of soft factors is currently receiving more favorable reviews. The difference in public opinion at home during the Vietnam and Gulf wars and the impact on troop morale and the outcomes of the wars provides some justification for increased emphasis on soft factors.

Computers also made the computation of stochastic processes possible. The differential equations of Lanchester attrition were viewed as approximations to a random process model of the actual killing process that should be correct for large numbers. Stochastic duels addressed the results for small numbers. Ancker and Gafarian have made significant contributions in this area (Ancker 1994).

Helmbold has made contributions to both the theoretical and the practical aspects of battle modeling. His empirical studies of attrition (1961, 1964), breakpoints (1971), and movement (1990) injected the element of reality into the sometimes rarefied atmosphere of theoretical battle modeling. Hartley (1991) continued in this vein with results indicating that the best description of attrition (using a homogeneous approximation) is not the Lanchester square or linear law, but an intermediate form between the linear law and a logarithmic law. Speight (1995, 1997) and Speight and Rowland (1999) have continued the process, introducing duels (mini-battles) and simulations of combat exercises (trials) and showing the impact of firing on dead targets on the formulation of attrition equations.

With computer battle models also came a proliferation of structural types of models. Battle models involving anti-submarine warfare have a peculiar requirement of finding the enemy before the battle can be prosecuted. Search theory must be implemented in such models, just as it is used in actual battles or exercises (Shudde 1971). In some types of war, the proper allocation of resources or mix of strategies provides an easily defined variable (e.g., strategic nuclear targeting or allocation of combat air forces to mission types). Because game theory deals with optimal strategies considering both sides' options, it provided an obvious technique for addressing the problem and providing prescriptive models (Bracken et al. 1974; HQ USAF/SAMA 1974).

Dimension, Data, and Output

In earlier times, land warfare models were one-dimensional: the forward edge of the battle area (FEBA) advanced or retreated. More sophisticated versions allowed one-dimensional structures for each sector (piston-models). More powerful computers now permit two-dimensional representations of the battlefield, using either x, y (or latitude, longitude) coordinates or (rectangular or hexagonal) grid structures. Some models are now three-dimensional, having terrain elevation and playing the effects of

flying aircraft at different altitudes. [See, for example, the Research Evaluation and Systems Analysis (RESA) model (Naval Ocean Systems 1992), which plays aircraft at different altitudes and submarines at different depths].

Most large models have extremely large input and output data sets and require sophisticated database management systems to keep track of the data. These large output data sets also stress the human ability to understand the results. Sophisticated graphics are necessary adjuncts to most large models today. The graphics are required to define realistic scenarios and to understand the process and results of the model.

Advances in computer power have resulted in the capability for human interfaces that are qualitatively different from past capabilities. Such interfaces include real-time depictions of a battlefield from a human perspective and auditory and tactile interfaces. The first full-scale example of this kind of interface, called virtual reality, in a battle model was SIMNET (HQ US Army Armor School 1987). SIMNET is a network of tank and other vehicle simulators, each participating in a shared virtual battlefield. Work is proceeding to tie virtual reality battle models to other, more conventional battle models. The success of connecting simulators has motivated recent work in connecting interactive training models. The connection of these battle models permits distributed processing and cost sharing among users.

The history of battle modeling has not been a smooth process of constant improvements. It has been beset with controversies in many areas. Some of the controversies have involved the standard resource allocation question: where do you spend the money? One of the first of these concerned documentation. Early (1960–1970s) computer models were usually undocumented and, because of frequent modifications, had virtually indecipherable code. The need for proper documentation was obvious but the need for better (or at least more complex) models appeared overriding. While the readability of the documentation of today's models may be variable, most models are documented.

Verification, Validation, and Accreditation

One controversy probably began with the first model that produced a result someone did not like: is the model right? During the 1960s and early 1970s, it was said there were two kinds of generals: those for whom computer printout was the gospel and those who would believe nothing produced by a computer. The problem in dealing with the first type was in conveying that there were caveats. All results had to be retyped manually to disguise their origin for the second type of general. Today's generals (and politicians) grew up with computers. They want to understand to what extent the results are believable. They require verification, validation, and accreditation. Although progress is being made, no one knows how to completely verify, validate, or accredit the general battle model.

Other Controversies

There have also been technical controversies in battle modeling. Notable controversies have included the proper interpretation (and thus use) of the differences between the Lanchester linear and square laws, the connection between attrition and advance rates (if any), the value of force ratios, the connection between deterministic Lanchester formulations and stochastic attrition formulations and which should be used. There is a precept that states that a force ratio of 3-1, attacker-defender is required for a successful attack. Numerous studies have criticized this precept, yet it is still heard.

There are disagreements about the proper level of detail in deterministic models, despite agreement on the principle that what is appropriate depends on the uses to be made of a model. High resolution models of large span require tremendous quantities of data and run slowly. One camp advocates small, fast "roughly right" models as better than high resolution models. Another camp protests that such models will miss the critical points that differentiate the issues in question. The stochastic process camp protests that both the large, high resolution and the small, low resolution models are not grounded in the reality of stochastic battles, and cannot thus be even roughly right.

There have also been disagreements about the proper uses of models. At one time prescriptive battle models were popular (finding optimal strategies, where the definition of optimal varied with the model). Lately they have been out of favor. Complaints about the misuse of models have ranged from the use of models designed for other purposes and failing to understand the resulting mismatch of assumptions to charges of advocacy modeling. Advocacy modeling, in the

pejorative sense, entails fiddling with input parameters until a combination is found that gives the desired result. Most large models have sufficient numbers of parameters with sufficiently tenuous connections to physical factors that plausible values can be found that generate almost any result.

One controversy involves the discovery that very simple deterministic battle models can exhibit chaos (Dewar et al. 1991). The question of the impact of chaos on the more complex models that are actually used is obvious. Most issues are settled by point estimates. For example, suppose the impact of weapon X is being investigated. Model runs with 25% X, 50% X, 75% X, and 100% X are executed. The runs with 75% X and 100% X are found to have superior results. It is assumed that such results are valid for values between 75% and 100%. If the results are chaos driven, such an assumption is unwarranted. The question has not been finally answered; however, investigations with one of the currently used complex models indicates that any uncertainty due to chaotic behavior in that model is no larger than a few percent. Because this is within the uncertainty that was already present in the model, the impact of possible chaotic behavior was claimed to be minimal (Herndon 1993).

Concluding Remarks

Despite all controversy, battle modeling remains the only method of answering some questions and is widely used. Battle models are used to inform decisions on weapons' procurement issues (balancing costs against effectiveness), to test strategies and tactics, and to train personnel. Battle training models provide inexpensive tools for training commanders because the large numbers of combat personnel maneuver in the computer rather than on the ground. As military funding is reduced, this supplement to traditional training methods has become indispensable. New models continue to be created as the requirements for greater scope arise. The insertion of information technology into combat has necessitated new models that can discriminate among the effects of different Command, Control, Communications and Intelligence (C^3 I) systems, such as the Joint Warfare System (JWARS) for analysis and the Joint Simulation System (JSIMS) for training.

See

- ▶ Cost Analysis
- ▶ Cost-Effectiveness Analysis
- ▶ Documentation
- ▶ Game Theory
- ▶ Gaming
- ▶ Lanchester's Equations
- ▶ Military Operations Research
- ▶ Model Accreditation
- ▶ Operations Research Office and Research Analysis Corporation
- ▶ RAND Corporation
- ▶ Search Theory
- ▶ Validation
- ▶ Verification

References

Ancker, C. J., Jr. (1994). *An axiom set (laws) for a theory of combat* (Technical Report). Los Angeles: Systems Engineering, University of Southern California.

Bracken, J., Falk, J. E., & Miercort, F. A. (1974). *A strategic weapons exchange allocation model, Serial T-325. School of Engineering and Applied Science*. Washington, DC: The George Washington University.

Dewar, J. A., Gillogly, J. J., & Junessa, M. L. (1991). *Non-monotonicity, chaos and combat models, R-3995-RC*. Santa Monica, CA: RAND.

Dupuy, T. N. (1985). *Numbers, predictions & war*. Fairfax: Hero Books.

Engel, J. H. (1954). A verification of Lanchester's law. *Operations Research, 2*, 163–171.

Hartley, D. S., III. (1991). *Predicting combat effects, K/DSRD-412*. Oak Ridge, TN: Martin Marietta Energy Systems.

Helmbold, R. L. (1961). *Historical data and Lanchester's theory of combat, AD 480 975, CORG-SP-128*. Fort Belvoir, VA: Combat Operations Research Group.

Helmbold, R. L. (1964). *Historical data and Lanchester's theory of combat, Part II, AD 480 109, CORG-SP-190*. Fort Belvoir, VA: Combat Operations Research Group.

Helmbold, R. L. (1971). *Decision in nattle: Breakpoint hypotheses and engagement termination data, AD 729 769*. Alexandria, VA: Defense Technical Information Center.

Helmbold, R. L. (1990). *Rates of advance in historical land combat operations, CAA-RP-90-1*. Bethesda, MD: Concepts Analysis Agency.

Herndon, S. K. (1993). *TRADOC analysis command research on VIC variability*, (Technical Document TRAC-TD-0293). Kansas: TRADOC Analysis Command, Fort Leavenworth.

HQ USAF/SAMA (1974). *A computer program for measuring the effectiveness of tactical fighter forces* (Documentation and Users Manual for TAC CONTENDER) SABER GRAND (CHARLIE).

Hq, U. S. A., & School, A. (1987). *M-1 SIMNET operator's guide*. Kentucky: Fort Knox.

Lanchester, F. W. (1916). Mathematics in warfare. In *Aircraft in warfare: The dawn of the fourth arm, constable and company*, London. (Reprinted in *The World of Mathematics*, by J. R. Newman, Ed., 1956, New York: Simon and Schuster).
Naval Ocean Systems Center (1992). *RESA Users Guide Version 5.5*, Vols. 1–8.
Shudde, R. H. (1971). Contact and attack problems. In P. W. Zehna (Ed.), *Selected methods and models in military operations research* (pp. 125–146). Alexandria, VA: Military Operations Research Society.
Speight, L. R. (1995). Modelling the mobile land battle: The Lanchester frame of reference and some key issues at the tactical level. *Military Operations Research, 1*(3), 53–56.
Speight, L. R. (1997). Modelling the mobile land battle: Lanchester's equations, mini-battle formation and the acquisition of targets. *Military Operations Research, 3*(5), 35–62.
Speight, L. R., & Rowland, D. (1999). Modelling the mobile land battle: Combat degradation and criteria for defeat. *Military Operations Research, 4*(3), 45–62.
Speight, L. R., & Rowland, D. (2010). Modelling the rural infantry battle: Group morale and the chances of attack success. *Military Operations Research, 15*(1), 31–52.
Taylor, J. G. (1980). *Force-on-force attrition modeling*. Linthicum, MD: Military Applications Society, INFORMS.
Taylor, J. G. (1983). *Lanchester models of warfare* (Vol. I and II). Linthicum, MD: Military Applications Society, INFORMS.
Taylor, B., & Lane, A. (2004). Development of a novel family of military campaign simulation models. *Journal of the Operational Research Society, 55*, 333–339.

Bayes Rule

When a decision maker receives data bearing on an uncertain event, the probability of the event can be updated by computing the conditional probability of the uncertain hypothesis given the new evidence. The derivation of the revised or a posteriori probability can be easily derived from fundamental principles and its discovery has been attributed to the Reverend Thomas Bayes (1763). The result is therefore known as Bayes rule or theorem:

$$\Pr\{H_1|E\} = \frac{\Pr\{E|H_1\}\Pr\{H_1\}}{\sum_i \Pr\{E|H_i\}\Pr\{H_i\}}$$

In this equation, H_1 refers to the specific, uncertain hypothesis entertained by the decision maker, the $\{H_i\}$ are the complete set of possible hypotheses, and E refers to the new evidence or information received.

See

► Bayesian Decision Theory, Subjective Probability, and Utility

Bayesian Decision Theory, Subjective Probability, and Utility

Kathryn Blackmond Laskey
George Mason University, Fairfax, VA, USA

Introduction

In every field of human endeavor, individuals and organizations make decisions under conditions of uncertainty and ignorance. The consequences of a decision and their value to the decision maker often depend on events or quantities which are unknown to the decision maker at the time the choice must be made. Such problems of decision under uncertainty form the subject matter of Bayesian decision theory. Bayesian decision theory has been applied to problems in a broad variety of fields, including engineering, economics, business, public policy, and artificial intelligence.

A decision-theoretic model for a problem of decision under uncertainty contains the following basic elements:

- A set of options from which the decision maker may choose;
- A set of consequences that may occur as a result of the decision;
- A probability distribution that quantifies the decision maker's beliefs about the consequences that may occur if each of the options is chosen; and
- A utility function that quantifies the decision maker's preferences among different consequences.

Subjective Probability

Decision theory applies the probability calculus to quantify a decision maker's beliefs about uncertain events or quantities, and to update beliefs upon receipt of additional information. De Finetti (1974) showed that any decision maker who acts on degrees of beliefs not

conforming to the probability calculus can be exploited by a series of gambles guaranteed to result in a net loss. Such a bet is called a dutch book. The Dutch Book Theorem and other related derivations of probability from axioms of rationality have been used to justify probability as a calculus of rational degrees of belief (De Groot 1970; Pratt et al. 1965).

Bayes Rule

When a decision maker receives information bearing on an uncertain hypothesis, degrees of belief are updated by computing the conditional probability of the uncertain hypothesis given the new evidence. The equation expressing how beliefs change with new evidence has been attributed to the Reverend Thomas Bayes (1763) and is known as Bayes Rule. The odds-likelihood form of Bayes Rule is:

$$\frac{\Pr\{H_1|E\}}{\Pr\{H_2|E\}} = \frac{\Pr\{E|H_1\}\Pr\{H_1\}}{\Pr\{E|H_2\}\Pr\{H_2\}}$$

In this equation, H_1 and H_2 refer to two uncertain hypotheses entertained by the decision maker and E refers to the new evidence or information received by the decision maker. Bayes rule quantifies how evidence is used to obtain the relative posterior probabilities $\Pr\{H_i|E\}$ of the hypotheses given the evidence. The ratio of posterior probabilities is determined by two factors. One is the ratio of prior probabilities $\Pr\{H_i\}$: all other things being equal, the stronger the prior belief in H_1 relative to H_2, the stronger the posterior belief in H_1 relative to H_2. The other is the likelihood ratio, or ratio of the probabilities $\Pr\{E|H_i\}$ of the evidence given each of the hypotheses. Again, all other things being equal, the better H_1 accounts for the evidence relative to H_2, the stronger the posterior belief in H_1 relative to H_2.

Other Interpretations of The Probability Calculus

There has been considerable debate about how to interpret the concept of probability. The term Bayesian, after Bayes Rule, is used to refer to the subjective interpretation. A subjective probability distribution represents an individual's degrees of belief about the likelihood of uncertain outcomes. Alternative interpretations of probability include the classical, the logical, and the frequentist approaches (Fine 1973). Much of standard statistical theory is based on the frequentist approach. Frequentists argue that probability models are appropriate only for repeatable phenomena exhibiting inherent randomness. For such phenomena, it is argued, there exist objectively correct probabilities intrinsic to the process producing the uncertain outcomes. Subjectivists apply probability theory to any outcomes about which a decision maker is uncertain. For subjectivists, no objectively correct probabilities need exist. Different decision makers are free to have different opinions about the probability of an outcome.

The only constraint subjective theory places on a probability distribution is that it be coherent, that is, that degrees of belief conform to the probability calculus. Within this constraint, decision makers are free to choose any probability distribution to model their uncertainty about a problem. Its inherent subjectivity has been a persistent criticism of the subjectivist approach. This is often of little practical consequence for problems that can be said to exhibit inherent randomness. The subjectivist draws inferences about the posterior distribution of the unknown parameter, while the frequentist draws inferences about the distribution of the data given different values of the unknown parameter. Nevertheless, it can be shown that when there are sufficient data to draw accurate inferences, the subjectivist and the frequentist will usually agree on the implications of the results. Thus, the major difference of practical import between the subjectivist and the frequentist is their attitudes toward problems for which there are too little data to estimate parameters accurately or for which the assumption of intrinsic objective frequencies is problematic. The frequentist maintains that probability models are in-appropriate for such problems; the subjectivist argues that probabilities are appropriate and that it is legitimate for rational people to disagree until there are sufficient data to bring them to agreement.

Utility Theory

Decision theory quantifies preferences by a utility function. It is assumed that the decision maker can

assign a numerical utility to each possible consequence of each option being entertained. Consequences with higher utilities are preferred to consequences with lower utilities. When there is uncertainty, the decision maker selects the option for which the expected value of the utility function is the largest. For some problems it is customary to deal with losses, or negative utilities. Smaller losses are preferred to larger losses.

The concept of utility appears to have been first introduced by Daniel Bernoulli (1738) in his solution to a puzzle known as the St. Petersburg Paradox. Bernoulli considered the problem of what price to pay for the opportunity to play the following gamble. A fair coin (probability 0.5 of landing heads) is tossed repeatedly until the first head appears. If the first head appears on the *n*th toss, the decision maker receives a prize of 2^n units of currency. The decision maker's expected monetary prize is

$$2(0.5) + 2^2(0.5)^2 + 2^3(0.5)^3 + \ldots,$$

which is infinite. A decision maker who maximized expected monetary value should be prepared to pay an arbitrarily large sum of money for the opportunity to play this gamble. As Bernoulli noted, most people would be willing to pay only a modest amount. Bernoulli suggested that the resolution to this apparent paradox was that a prize's worth to a decision maker was a nonlinear function of the monetary value of the prize. For example, replacing 2^n with $\log 2^n$ in the above equation yields a finite expected monetary prize.

Von Neumann and Morgenstern (1944) were the first to present a formal axiomatic development of utility theory. They defined the utility of a consequence in terms of a comparison between two options, one sure and one uncertain. The sure option is the consequence itself; the uncertain option is a lottery between two standard reference prizes, one worth more and one worth less than the consequence in question. If the reference prizes are assigned utility one and zero, then the utility of the consequence in question is defined as the probability at which the decision maker is indifferent between the two lotteries. Several similar axiom systems can be shown to lead to the maximization of expected utility as a principle of rational decision making (De Groot 1970; Pratt et al. 1965).

Concluding Remarks

It has been observed that people systematically violate the axioms of expected utility theory in their everyday behavior. Some of these violations can be reversed by informing people of the implications of their stated preferences. In other cases, many people resist changes to their original judgments. Even when the decision maker regards expected utility theory as a norm of rational behavior, it cannot be assumed that unaided judgments will be consistent with the theory. The field of decision analysis applies theories and methods from decision theory and the psychology of human information processing to construct decision theoretic models for practical decision problems (Clemen 1996).

Interest has been growing in decision theoretic formulations of statistical problems. For example, to formulate an hypothesis testing problem, one defines a prior probability for the null and alternative hypotheses. One also defines losses associated with accepting a false alternative hypothesis and rejecting a true null hypothesis. The optimal decision rule is to accept or reject the hypothesis according to which decision yields the lower posterior expected loss given the observed sample. Similarly, decisions of whether to gather information and how large a sample to draw can be formulated as decision problems that consider both the cost of gathering information and the benefit of obtaining the information. Some problems that are quite complex when viewed from a frequentist perspective become straightforward when viewed from a Bayesian perspective. Examples include hierarchical models and problems of missing data (Gelman et al. 1995).

An area of application is the field of intelligent systems (Haddawy 1999). Utility theory is being applied to planning and control of reasoning in expert systems. Diagnostic expert systems based on probability theory have achieved performance comparable to human decision makers (e.g., the Pathfinder system for diagnosing lymph node pathology, Heckerman 1991). Perhaps the most important and challenging aspect of decision analysis is the creative process of model formulation. Decision theory takes options, consequences, and their interrelationships as given. Automated decision model generation is an open research area of great importance to application of decision theory to the field of intelligent systems (Haddawy 1994).

See

- ▶ Decision Analysis
- ▶ Decision Problem
- ▶ Decision Trees
- ▶ Expert Systems
- ▶ Utility Theory

References

Bayes, T. R. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London, 53*, 370–418 (Reprinted with biographical note by G. Barnard, 1958, in *Biometrika*, *45*, 293–315).
Bernoulli, D. (1738). Specimen Theoriae Novae de Mensura Sortis. *Commentarii Academiae Scientarium Imperalis Petropolitanae* 175–192 (Translated in L. Sommer, 1984, *Econometrica*, *22*, 23–26).
Clemen, R. (1996). *Making hard decisions*. Belmont, CA: Duxbury Press.
de Finetti, B. (1974). *Theory of probability: A critical introductory treatment*. New York: Wiley.
De Groot, M. H. (1970). *Optimal statistical decisions*. New York: McGraw Hill.
Fine, T. L. (1973). *Theories of probability*. New York: Academic.
Gelman, A., Carlin, J., Stern, H., & Rubin, D. (1995). *Bayesian data analysis*. London: Chapman and Hall.
Haddawy, P. (1994). Generating Bayesian networks from probabilistic knowledge bases. In *Proceedings of tenth conference on uncertainty in artificial intelligence* (pp. 262–299). San Mateo, CA: Morgan Kaufmann.
Haddawy, P. (1999). An overview of some recent developments in Bayesian problem solving. *AI Magazine, 20*(2), 11–19.
Heckerman, D. (1991). *Probabilistic similarity networks*. Ph.D. dissertation, Program in Medical Information Sciences. Stanford University, CA.
Pratt, J. W., Raiffa, H., & Schlaifer, R. (1965). *The foundations of decision under uncertainty: An elementary exposition*. New York: McGraw Hill.
von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior*. New Jersey: Princeton University Press.

Beale Tableau

A modification of the simplex tableau arranged in an equation form such that the basic variables and the objective function value are expressed explicitly as functions of the nonbasic variables. This tableau is often used when solving integer-programming problems.

See

- ▶ Linear Programming
- ▶ Tucker Tableau

Bellman Equation

- ▶ Bellman Optimality Equation

Bellman Optimality Equation

Dynamic programming equation that the optimal value (or cost-to-go) function must satisfy, according to the principle of optimality. One simple form is the following finite-action, finite-state, finite-horizon version for a minimization problem:

$$f_n(i) = \min_a \{c_n(i,a) + \Sigma_j p_{ij}(a) f_{n+1}(j)\},$$

where $f_n(i)$ represents the optimal cost-to-go function in state i for stage (period) n, $c_n(i,a)$ is the one-period cost in stage n for state i and action a, and $p_{ij}(a)$ is the probability of transitioning from state i to state j when action a is taken.

See

- ▶ Approximate Dynamic Programming
- ▶ Dynamic Programming
- ▶ Markov Decision Processes

Benders Decomposition Method

A procedure for solving integer-programming problems that have a few integer variables. These so-called complicating variables, when given specific values, enables the resulting problem to be readily solved as a linear-programming problem.

See

- ▶ Integer and Combinatorial Optimization
- ▶ Linear Programming

Best-Fit Decreasing Algorithm

▶ Bin-Packing

Bidding Models

▶ Auction and Bidding Models

Big M Method

A method to drive artificial variables out of the basis in the simplex algorithm, by imposing a sufficiently large, finite penalty M for using these variables.

See

▶ Artificial Variables
▶ Linear Programming
▶ Phase I Procedure
▶ Phase II Procedure
▶ Simplex Method (Algorithm)

Bilevel Linear Programming

Bilevel linear programming (BLP) is a hierarchical, decentralized, multilevel mathematical programming problem in which the objective functions and constraints are linear. It can be stated in terms of upper and lower problems as follows:

$$\underset{x}{\text{Maximize}} f_1(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{c}_1\boldsymbol{x} + \boldsymbol{d}_1\boldsymbol{y}$$

where $\boldsymbol{y}$ solves:

$$\underset{y}{\text{Maximize}} f_2(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{c}_2\boldsymbol{x} + \boldsymbol{d}_2\boldsymbol{y}$$

subject to

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} \leq \boldsymbol{b}$$
$$\boldsymbol{x}, \boldsymbol{y} \geq 0$$

where $\boldsymbol{c_1}, \boldsymbol{c_2}, \boldsymbol{d_1}, \boldsymbol{d_2}$, and $\boldsymbol{b}$ are constant vectors, $\boldsymbol{A}$ and $\boldsymbol{B}$ are constant matrices; $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of the decision variables of the upper and lower problems, respectively; f_1 and f_2 are the objective functions of the upper and lower problems, respectively.

See

▶ Linear Programming

References

Bard, J. F. (1984). Optimality conditions for the bilevel programming problem. *Naval Research Logistics Quarterly, 31*, 13–16.
Ben-Ayed, O. (1993). Bilevel linear programming. *Computers & Operations Research, 20*, 485–501.
Bialas, W. F., & Karwan, M. H. (1984). Two-level linear programming. *Management Science, 30*, 1004–1020.
Colson, B., Marcotte, P., & Savard, G. (2005). Bilevel programming: A survey. *4OR, 3*, 87–107.

Binary Variable

A variable that is restricted to be equal to 0 or 1. Binary variables are often used to handle logical, nonlinear conditions associated with a problem whose constraining conditions are linear.

See

▶ Integer and Combinatorial Optimization
▶ Integer-Programming Problem

Bin-Packing

Nastaran Coleman[1] and Pearl Wang[2]
[1]Federal Aviation Administration, Washington, DC, USA
[2]George Mason University, Fairfax, VA, USA

Introduction

The bin-packing problem is concerned with the determination of the minimum number of bins that are

needed to pack a given set of input data items. The problem has numerous applications in operations research, computer science, and engineering, where the items and bins to be packed can be of varying shapes or multi-dimensional in size. These applications include industrial manufacturing, container loading, stock cutting, vehicle routing, television commercial scheduling, job scheduling on multiple processors, file backup creation in removable media, integrated circuit manufacturing and fault detection, and location testing in linear circuits. Since the bin-packing problem is known to be NP-hard (Garey and Johnson 1979), it is of interest to find efficient heuristics that obtain near-optimal solutions to the problem.

Problem Definition

The classical one-dimensional bin-packing problem (1DBPP) is defined as follows: Given a positive bin capacity C and a list of items $L = (p_1, p_2, \ldots, p_n)$, where p_i has size $s(p_i)$ satisfying $0 \leq s(p_i) < C$, determine the smallest integer m such that there is a partition of $L = B_1 \cup B_2 \cup \ldots \cup B_m$ where the sum of the sizes of the items $p_i \in B_j$ do not exceed the capacity C. Each set B_j is usually viewed as the contents of a bin of capacity C. In much of the literature, C is taken to be 1.

Several versions of two-dimensional bin-packing problems have also been studied. For example, if L is a set of rectangles p_i having heights h_i and widths w_i, one type of bin packing problem requires that the rectangles of L be packed into a single two-dimensional bin of width C and infinite height. The goal is to determine a minimum height packing of the pieces into this bin. These problems are referred to as strip-packing problems.

For an alternative form of the two-dimensional packing problem, the rectangles of L are to be packed into a minimum number of rectangular bins. A common version of the problem concerns packing a list of squares into m unit squares with the objective being to minimize m. When the rectangles to be packed are not square, restrictions might be made on the types of allowable placements of the rectangles within the bins. Depending on the application, rotations of the items may not be permitted; packings may also require that the items are placed parallel to the sides of the bins.

The items being packed in two-dimensional problems do not need to be rectangular in shape. Circular and polygonal shapes may also be packed into circular or rectangular bins.

Three-dimensional bin-packing problems have goals that are similar to their lower dimensional counterparts. For example, given the set L of rectangular prisms having widths w_i, height h_i, and depth d_j, a common problem is to pack the items into a minimum number of bins of width W, height H, and depth D. In the case of container packing, the pieces are not rotated and must be placed parallel to the sides of the bins.

Cutting stock problems are variants of bin packing problems because the amount of wasted space within stock sheets is to be minimized while the pieces are being cut from stock sheets. Similarly, if just a single bin of fixed size is to be packed and each item is characterized by both a volume and a value, the problem of maximizing the total value of a subset of items that can fit into the bin by volume is known as the knapsack problem.

Approximation algorithms for bin-packing problems were among the earliest algorithms studied in the literature. In the 1970s, it was shown that near-optimal solutions could be guaranteed for some frequently used one-dimensional packing techniques. Since then, many heuristics have been proposed for obtaining approximate solutions to both the one and two-dimensional problems for sequential and parallel models of computation. Three-dimensional problems were initially studied to a lesser degree, but recent work now appears regularly in the literature. The performance of a given heuristic (i.e., the computational time and resources needed to find a packing), as well as the quality of the packing that is constructed by the heuristic are important considerations that have been analyzed by many researchers.

Surveys of many classical bin-packing algorithms can be found in Coffman et al. (1996). A bibliography of cutting and packing research was presented by Sweeney and Paternoster (1992), while a more recent typology that characterizes cutting and packing problems is described in Wäscher et al. (2007). Recent probabilistic analyses of approaches for solving one-dimensional bin-packing problems are discussed in Coffman et al. (2000). Two-dimensional packing problems are surveyed by Lodi et al. (2002)

and meta-heuristic algorithms for strip packing problems are reviewed in (Hopper and Turton 2001). Recent work that addresses three-dimensional packing problems includes (Martello et al. 2000), (Faroe et al. 2003), and (Parreño et al. 2008). Heuristic approaches for solving irregular and polygonal packing problems are presented by Jakobs (1996) and Burke et al. (2010).

Algorithms for solving the problem on various parallel models of computation can be found in Anderson et al. (1989), Fenrich et al. (1989), Berkey (1990), and Coleman and Wang (1992). The EURO Special Interest Group on Cutting and Packing maintains a website for research activities related to cutting and packing.

Characterizations of Bin-Packing Algorithms

Many types of bin-packing algorithms have been proposed and analyzed for both sequential and parallel systems. Sequential heuristics can be classified as either on-line or off-line algorithms. On-line algorithms assign data items to bins in the same order as originally input, without utilizing any global knowledge of the data list. For example, the Next-Fit packing and Sum of Squares heuristics are on-line algorithms that perform one-dimensional packing. Off-line algorithms preprocess the data, usually by sorting. Well-known examples are the First-Fit Decreasing and Best-Fit Decreasing algorithms. Alternatively, other methods may preprocess the input data by partitioning the items by size into subintervals, and then pack the data using those sub-intervals. These techniques are described in more detail below.

Approximation algorithms for solving the one-dimensional bin-packing problem on various models of parallel computation have been reported. It has been shown that several frequently used sequential bin packing strategies such as First-Fit Decreasing are P-Complete. Thus, it is unlikely that these heuristics can be parallelized into efficient algorithms for the theoretical Parallel Random Access Machine (PRAM) model of computation. However, other well-known sequential strategies such as Harmonic packing can be parallelized efficiently. In the previous decade, experimental studies of similar heuristics were performed on Single-Instruction, Multiple-Data (SIMD) and Multiple-Instruction, Multiple-Data (MIMD) parallel computers.

Theoretical Studies

Performance metrics have been formulated as a means to compare these different packing algorithms when executed on random data. Theoretical analyses typically include worst-case and average-case packing performance of the heuristics. The asymptotic worst-case performance can be defined as the limiting ratio of an algorithm's worst instant packing to its optimal packing. For example, if $A(L)$ and $OPT(L)$ are the number of bins packed by an algorithm A and the optimal number of bins needed for a list L, respectively, then the asymptotic performance ratio can be defined as

$$R_A^{\infty} = \inf\{r \geq 1 : \text{ for some } N > 0, A(L)/OPT(L) \leq r \text{ for all } L \text{ with } OPT(L) \geq N\}$$

Two measures of average-case packing performance that have been studied are the expected values $E(R_N)$ and $E(U)$ where R_N is the ratio of the average number of bins packed by the algorithm to the average size of all data items and U is the difference between these quantities. Further, an algorithm is often said to exhibit perfect packing if $E(R) = 1$, where $E(R)$ is the limiting distribution of $E(R_N)$, or when $E(U) = O(\sqrt{N})$.

These metrics are studied analytically as well as by simulation. The input data are usually assumed to come from a uniform distribution $U[a, b]$. Coffman et al. (2000) introduced the perfect packing theorem and show that the optimal expected wasted space for a random list is either $o(n)$, $o(n^{0.5})$ or $o(1)$. These researchers have also shown that the average case can differ substantially between discrete and continuous uniform distributions.

An alternative measure of packing performance is to determine the expected waste of the packing. If $L_n(F)$ denotes a list of n items drawn according to a probability distribution F and $P_n^A(F)$ denotes a packing resulting from the application of algorithm A, then the expected waste is defined as $EW_n^A(F) = E\left[W\left(P_n^A(F)\right)\right]$ where expectation is taken over the random variable $L_n(F)$.

Theoretical studies of bin packing problems are often aimed at determining whether asymptotic approximation schemes can be constructed. In this

case, researchers seek to determine if for every $\varepsilon > 0$, there is a polynomial time algorithm A_ε having an asymptotic approximation ratio of $1 + \varepsilon$.

Some One-Dimensional Packing Heuristics

The Next-Fit algorithm packs one-dimensional items into one-dimensional bins in the simplest fashion. The data items are processed one at a time, beginning with p_1, which is put into bin B_1. If item p_i is to be packed and B_j is the highest indexed nonempty bin, then p_i is placed into bin B_j if it fits into B_j; that is, $p_i + size(B_j) \leq C$. Otherwise, a new bin B_{j+1} is started and p_i is placed into it. In this manner, each successive piece is packed into the most recently used bin, and previously packed bins are not considered. Next-Fit is a fast on-line algorithm whose time complexity is $O(n)$. Its worst-case performance ratio is bounded by 2, and its average performance by 3/2. Variants of Next-Fit have been proposed and include Next-Fit-Decreasing, Next-1-Fit, and Next-K-fit. The basic approach is also used to obtain level-oriented heuristics for solving two-dimensional bin packing problems.

The Harmonic packing algorithm begins by partitioning the unit interval into the set of intervals $I_k = (1/(k + 1), 1/k]$, $1 \leq k < m$ and $I_m = (0, 1/m]$. The bins are divided into m categories and an I_k-bin packs at most kI_k data. The packing of each I_k piece into an I_k-bin is done using the Next-Fit Algorithm. At any given time, an active list of all unfilled I_k-bins is kept. The Harmonic algorithm has a worst-case performance bound of 1.69; some modified versions of the approach have been shown to have lower performance bounds.

The Sum-of-Squares (SS) algorithm is an online method for packing items with integral sizes into bins of capacity C. It has time complexity $O(nC)$. If the amount of unpacked space in a bin is called its gap, g, and $N(g)$ is the number of bins in a current packing with gap g, then this algorithm puts an item p_i into a bin such that after placing the item, the value of $\sum_{g=1}^{C-1} N(g)^2$ is minimized.

Theoretical analysis of this algorithm demonstrates that for any perfectly packable distribution F, that $EW_n^{SS}(F) = O(\sqrt{N})$ and if F is a discrete uniform distribution $U(j, C)$ where $j < C - 1$, then $EW_n^{SS}(F) = O(1)$. For all lists L, it is further demonstrated that $SS(L) < 3OPT(L)$. Csirik et al. (2006) survey other online algorithms including randomized variants of sum-of-squares. Bender et al. (2007) propose two variants of the sum-of-squares algorithm and Seiden (2002) presents a survey as well as an online algorithm based on the Harmonic approach.

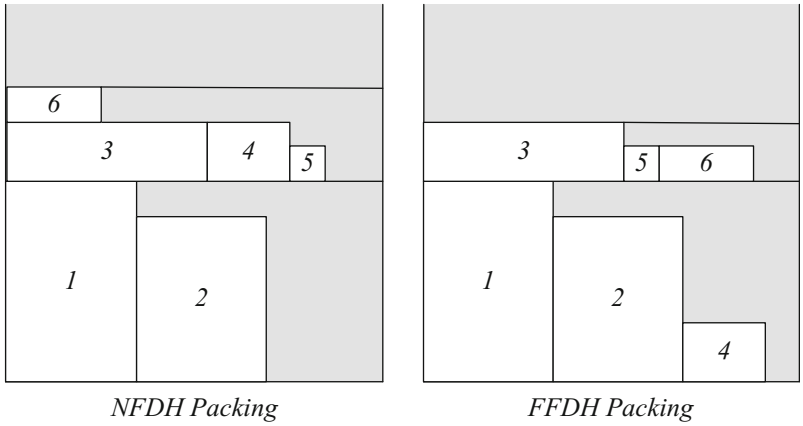
The First-Fit (FF) heuristic packs each successive data item p_i into the lowest indexed bin B_j into which it fits. When this is not possible, a new bin is created. Thus, it is necessary to maintain a list of all partially filled bins. For the worst-case, average case, and lower bound performance of First-Fit, it has been shown that the numberof bins used by this algorithm is 17/10 $OPT(L) \pm 2$, where OPT is the number of bins used by the optimal solution. Xia and Tan (2010) decreased the upper bound for the asymptotic performance ratio to 17/10 OPT + 7/10 for First-Fit and for the absolute performance ratio– to 12/7 OPT. The time complexity of First-Fit is $O(n \log n)$.

If the items are initially sorted in non-increasing order before packing proceeds, the heuristic is referred to as First-Fit Decreasing, and the performance bound decreases to 11/9 OPT + 6/9. Other algorithms that are based on this approach include Best-Fit (where the "best" bin is chosen if there is more than one possibility), Best-Fit Decreasing, Worst-Fit, Almost Worst-Fit, Revised First-Fit, and Modified First-Fit Decreasing bounded by 71/60 OPT + 1. When the data items are drawn from a uniform distribution, then $E(A(L)) - n/2 = O(n)$ for the First-Fit Decreasing and Best-Fit Decreasing algorithms. Asymptotic polynomial-time approximation schemes show that it is possible to find a solution for any $0 < \varepsilon \leq 1/2$ in polynomial time using at most $(1 + 2\varepsilon)OPT + 1$ bins.

Some Multi-Dimensional Packing Heuristics

Two-Dimensional Packing

The Two-Dimensional Bin-Packing Problem requires packing a finite set of small rectangles into the minimum number of rectangular bins without overlapping. The problem is strongly NP-hard, and has several industrial applications. Other variants of two-dimensional bin-packing problems occur in real-world applications, especially in the manufacturing industries. Additional constraints may include orientation where items can be rotated by 90°



Bin-Packing, Fig. 1 Level-oriented packings

or have to stay fixed. For example, rotation is not allowed when the items are articles to be paged in newspapers.

Researchers have applied one and two-phase algorithms that make use of upper and lower bounds on the number of bins needed to pack the input rectangles. These approaches are often integrated into greedy heuristics and tabu searches. One-phase algorithms directly pack the items into the finite bins. Two-phase algorithms start by packing the items into a single strip, i.e., a bin having width W and infinite height. In the second phase, the strip solution is used to construct a packing into finite bins. Lodi et al. (2002) survey advances obtained for the two-dimensional bin and strip packing packing problems, with emphasis on exact algorithms whose goal is to find an optimal solution, as well as effective heuristic and metaheuristic approaches.

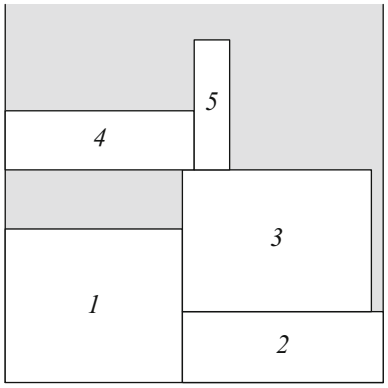
Level-oriented packing heuristics pack rectangles into a single two-dimensional bin (or strip) that has infinite height. In these approaches, the rectangles to be packed are first ordered by non-increasing height. The packing is constructed as a sequence of levels, whose heights are defined by the heights of the first rectangles placed in the respective levels. The Next-Fit or First-Fit approaches can be used to define and fill these levels of the bin. The asymptotic performance bounds of the Next-Fit Decreasing Height (NFDH) and First-Fit Decreasing Height (FFDH) heuristics are 2 and 1.7, respectively. Figure 1 illustrates these packing heuristics.

Similar approaches in which the heights of the levels are preset by a parameter yield a variety of shelf heuristics, where these levels can be packed in a similar fashion. Next-Fit Shelf and First-Fit Shelf are examples of these heuristics. Their corresponding execution times are $O(n)$ and $O(n^2)$. If the parameter that dictates the shelf heights is defined by r, then these methods have asymptotic performance bounds of $2/r$ and $1.7/r$, respectively.

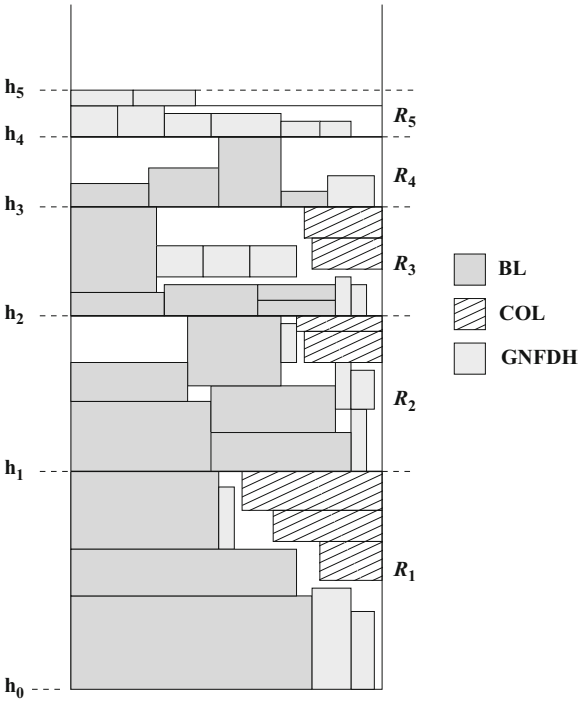
Bottom-Left (BL) packing approaches pack rectangles into an infinite height bin by successively placing each item into the bottom-most, left-most position in which it fits without overlapping any rectangles that have already been packed. If the items are preordered by non-increasing width, then the worst case bound of this heuristic indicates that the height of the packing does not exceed twice the height of an optimal packing. The algorithm can be implemented in $O(n^2)$ time and a sample packing is shown in Fig. 2.

Alternative methods may divide the set of items being packed into sublists that are used to obtain a split packing. In this case, the infinite height bin is also divided into subregions where one-dimensional heuristics are used to pack the rectangles. Classical techniques include Split-Fit, Mixed Fit, and Up-Down (see Fig. 3) which require $O(n \log n)$ time. Performance ratios of 2, 1.33, and 1.25, respectively, have been proven for these approaches. Other similar methods appear in the literature. Coffman and Shor (1993) discuss asymptotic average-case analysis for two-dimensional bin-packing.

One particular heuristic that uses a split packing approach addresses the problem of packing squares into a two-dimensional strip of unit width. The squares whose widths are greater than 1/2 are first

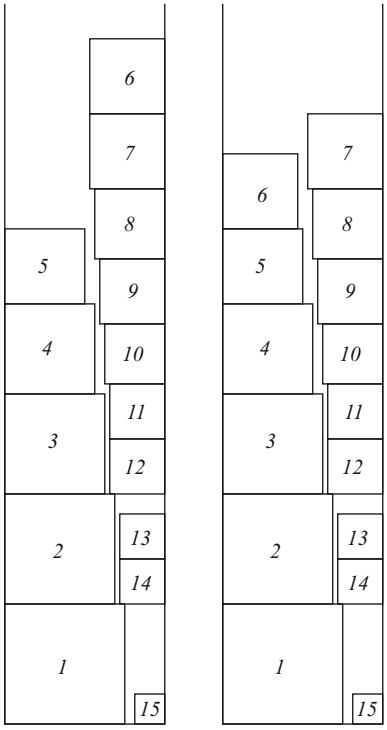


Bin-Packing, Fig. 2 BL packing



Bin-Packing, Fig. 3 Up-down packing

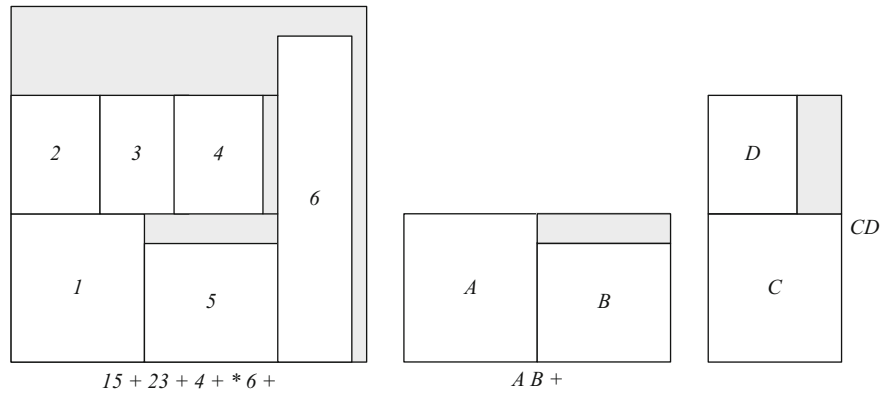
stacked along the left edge of the strip in order of decreasing width. Starting at the height, $H_{1/2} = \sum_{w_i > 1/2} h_i$, where the sum of the sizes of packed squares exceeds 1/2, the remaining squares are stacked along the right edge of the strip in order of decreasing width. This stack slides downward until



Bin-Packing, Fig. 4 Packing squares

it either rests on the bottom of the strip, or a square in the right stack comes in contact with a square in the left stack, whichever occurs first. Finally, all the squares lying entirely above $H_{1/2}$ are repacked into two stacks, one against the left edge of the strip and the other against the right edge. This is done in decreasing order of size, placing each successive square on the shorter of the two stacks already created. A sample packing is shown in Fig. 4. It can be shown for this algorithm, that $E(A(L)) = E(OPT(L)) + O(1)$.

When multi-dimensional objects are to be packed into a minimum number of multidimensional bins, the vector packing approach can be used. This technique is a direct generalization of the one-dimensional problem. For example, if rectangles are to be packed into square bins, then the only types of packing that are permitted are those where the rectangles are diagonally placed corner-to-corner across the bins. In general, if a vector packing algorithm is such that no two nonempty bins can be combined into a single bin, then the ratio of the number of bins packed to the optimal solution does not exceed $d + 1$, where d is the number of dimensions. Extensions of the First-Fit and First-Fit Decreasing



Bin-Packing, Fig. 5 Postfix GA encoding

heuristics to this multi-dimensional case have yielded approaches whose asymptotic worst case ratio is $d + 7/10$ and $d + 1/3$, respectively.

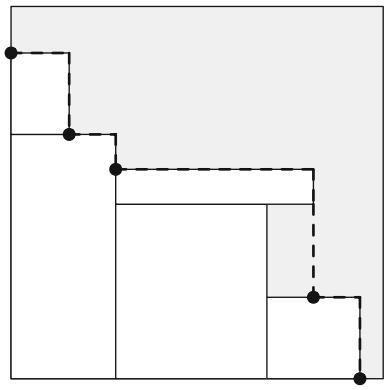
Metaheuristic algorithms have been used extensively in recent years to solve two-dimensional bin-packing problems. In short, metaheuristic methods are general frameworks that try to improve the direction of the search for the best solution, thus finding a better solution at every iteration. There are no guarantees of finding an optimal solution, but many metaheuristics implement some form of stochastic or linear optimization. Genetic algorithms, simulated annealing, and tabu search are examples of metaheuristic algorithms.

Hopper and Turton (2001) review several approaches developed to solve two-dimensional packing problems with metaheuristic algorithms. Genetic algorithms (GAs) were first used in the mid-1980s to solve strip and bin-packing problems. Many employ a two-step approach referred to as a hybrid genetic algorithm. Encoded solutions corresponding to physical layouts are manipulated by GAs that evaluate the solutions using decoding algorithms. In some cases, these decoded layouts correspond to non-overlapping packings that are obtained using a bottom-left packing heuristic. Other researchers have used a sliding principle that gives priority to the downward shifting of the rectangle being packed for the decoding routine.

Other genetic algorithms incorporate layouts directly into the encoding technique. For example, postfix strings corresponding to packing layouts can be manipulated by GAs. In the example shown in Fig. 5, the A B + and CD* substrings correspond to placements of two rectangles that are horizontally or vertically adjacent, respectively.

It is also possible for genetic algorithms to operate without encodings. An initial layout can be modified by rotating, translating, and/or relocating an item (or subset of items) in the layout. These operators correspond to hill-climbing and the mutation and recombination features of GAs. Hopper and Turton (2001) compare some meta-heuristic algorithms to two-dimensional random search and heuristic packing routines. The comparison is made in terms of the solution quality and the computation time for a number of packing instances of different sizes.

Simulated annealing, tabu search and exact algorithms have also been used to compute solutions to two-dimensional bin and strip packing problems. See (Lodi et al. 2002) for a survey of some of these approaches. A simulated annealing approach was first applied to a pallet loading problem (i.e., a three-dimensional packing problem that has been reduced to its two-dimensional footprint). Simulated annealing is a hill-climbing approach where solutions that are worse may be accepted as dictated by a cooling schedule which is determined by a given probability function. For the pallet loading problem, the number of feasible solutions for a box is equated with multiples of the item length. Neighborhoods are defined by moving each item in a solution to another position (with some restrictions). As a result, the simulated annealing heuristic would allow both legal and illegal packings as it attempted to improve the solution quality. The objective function must then minimize any overlaps that occur in the packing layout.



Bin-Packing, Fig. 6 Defining corner points

The tabu search strategy utilises a search scheme and a candidate neighborhood that is consturcted from a feasible solution: a heuristic recombines a subset of items currently packed into k different bins along with one item packed into a bin that is likely to be emptied. The value of k is also updated during the search to escape from local optima. A mechanism (i.e., the use of memory) must be built into the tabu search to prevent the heuristic from returning to recently examined packings.

Lower bounds are used to guide search strategies in exact algorithms whose goal is to find optimal solutions. For example, in one branching scheme, each node in the search tree represents a subset of packed rectangles which define a set of corner points for the bottom-left placement of unpacked items (see Fig. 6). The use of bounds to traverse a search tree corresponds to the selection of branches to investigate or ignore.

The average performance of exact algorithms and metaheuristics are typically evaluated through extensive computational experiments using benchmark data sets as described in (Parreño et al. 2008). Other more recent two-dimensional and three-dimensional examples include Bekrar and Kacem (2009) and Puchinger et al. (2010).

Three-Dimensional Packing

Algorithms for obtaining heuristic solutions to three-dimensional packing problems in which boxes are to be packed into a minimum number of identical three-dimensional bins have been characterized as either local-search or construction heuristics (Faroe et al. 2003). Analgous to the two-dimensional case, local-search methods iteratively seek better packings of the boxes by examining neighborhoods of solutions, while constructive heuristics add boxes to a packing using strategies such as First-Fit or Best-Fit. Examples of recent heuristics that have employed these methods include guided local search (GLS), a two-level tabu search (TS^2PACK), and a greedy randomized adaptive search procedure (GRASP) that is combined with a variable neighborhood descent (VND) structure.

The GLS strategy has roots in constraint-satisfaction applications and uses memory (typical of tabu search methods) to guide the search of the solution space by augmenting the objective function with penalties for previously visited solutions. It begins with an upper bound calculated from an initial greedy solution and then iteratively removes one bin from the feasible solution. Translation of boxes within one bin or between bins defines the neighborhood of the local search algorithm. To speed up the search process, some boxes are temporarily fixed in position. As before, the objective function additionally reflects the total volume of an overlap between boxes.

The TS^2PACK heuristic uses a first-level tabu search that addresses the optimality of the packing problem and a second-level tabu search that finds feasible solutions for the items assigned to the bins. An initial solution is computed using a Next-Fit Decreasing packing based on box volumes and the extreme points that are identified for a given box – these indicate positions where an additional item can be accommodated with respect to the given box. Then the TS^2PACK heuristic iteratively discards the bin with the worst fitness function value (defined as the weighted sum of the volume used by the items in the bin and the number of items). Each discarded item is packed into one of the remaining bins which yields the maximum fitness function value (i.e., minimizes the height of the new packing with bin size constraints relaxed). If this packing is not feasible due to bin size violations, a second heuristic is employed to optimize the packing with respect to the bin size constraints. This heuristic is a tabu search that uses interval graphs to represent the layout. By manipulating the graphs, alternate layouts can be generated that correspond to moving boxes by locating them in different positions.

The packing performance of these and other heuristics is compared against the GRASP/VND

approach summarized in (Parreño et al. 2008). Large sets of test cases are studied that include both two and three-dimensional problem instances. The results indicate that this method obtains comparable or better solutions to the other algorithms.

The GRASP/VND heuristic is an iterative method that combines a randomized constructive phase and an improvement phase. The constructive phase iteratively fills one bin with boxes by considering the maximal spaces created by placing boxes near corners of the bin. Boxes to be packed are selected based on best-volume or best-fit criteria. This is repeated until all boxes are packed into bins.

Attempts may be made to improve the packing by moving boxes in the bins that have first been sorted by volume. Four improvement moves were proposed: move the last k percent of boxes, move a percentage of boxes in every bin that has below average occupancy, move different parts of the bins to be emptied, or combine subsets of boxes in complementary bins and refill both with the remaining boxes.

The application of improvements (i.e., the movements of the chosen boxes) was dictated by several strategies. One of these applied the VND strategy to explore the solution neighborhood defined by the four possible moves. If the GRASP/VND heuristic appeared to be stuck at a local solution, diversification iterations are applied in the constructive process which require packing the most frequently remaining boxes first.

Recent Theoretical Studies of Multi-Dimensional Packing

Several theoretical analyses have been performed for multi-dimensional bin-packing heuristics that provide performance guarantees for packing quality as well as for algorithm execution time. One example is the recent work related to polynomial time approximation schemes (APTAS) for the three-dimensional strip packing problem. It has been shown that APTAS's exist for one-dimensional bin-packing and two-dimensional strip packing problems, but an APTAS will only exist for two-dimensional bin-packing problems if P = NP. These results are reviewed by Bansal et al. (2007) who also develop two approximation schemes: one for packing three-dimensional strips with arbitrarily sized boxes and a second for packing boxes with square bases.

Their first algorithm initially applies a Harmonic transformation (i.e., using intervals similar to those defined in the 1DBPP Harmonic heuristic) to the box widths, then it creates slabs of items to form two-dimensional strip packing instances. The two-dimensional strip is then cut into slices to produce new items that are placed on top of each other in the height dimension of a three-dimensional strip. The authors prove that this algorithm has an asymptotic approximation ratio that is arbitrarily close to the Harmonic number $T_\infty \approx 1.69$. The second algorithm A packs of set I of three-dimensional boxes with square bases so that the height of the packing does not exceed $(1 + 12\varepsilon)OPT(I) + O(K)$ where $K = \varepsilon^{-O(2^{1/\varepsilon})}$.

An APTAS for packing d-dimensional cubes into a minimum number of unit cubes has been developed by Correa and Kenyon (2004) who also present a scheme for packing rectangles into at most OPT square bins whose sides have length $1 + \varepsilon$ and OPT denotes the minimum number of unit bins required to pack the rectangles.

Parallel Algorithms

Many parallel algorithms have been proposed and studied for solving the cutting stock and knapsack variants of the bin-packing problem. Heuristics have also been proposed that obtain approximate solutions to the one-dimensional bin-packing problem on various models of parallel computation.

For the shared-memory Exclusive-Read Exclusive-Write PRAM model of computation, a heuristic based on First-Fit Decreasing has been proposed which runs in $O(\log n)$ time on $n \log n$ processors (Anderson et al. 1989). This approach divides the data items into two groups. Items in the first group are partitioned into sublists that are packed into "runs" of bins. The bins are then filled using items in the second group. The algorithm relies on parallel prefix, merging, and parenthesis matching operations, and has a worst-case performance bound of 11/9.

Practical one-dimensional bin-packing algorithms (including parallelizations of the Harmonic algorithm) have also been proposed and implemented on parallel architectures such as systolic arrays, SIMD arrays, and MIMD hypercubes. Quantitative studies

and theoretical analyses have been performed on some of these approaches. The Systolic packing algorithm, for example, has a worst-case performance bound of 1.5 and executes in $O(n)$ time. Similar results were reported in Berkey (1990).

Coleman and Wang (1992) formulated an online heuristic for massively parallel systems that used interval partitioning. The average case behavior of the heuristic could be predicted when the input have a symmetric distribution. The method is asymptotically optimal, yields perfect packings, and achieves the best possible average case behavior with high probability.

See

- ▶ Combinatorics
- ▶ Computational Complexity
- ▶ Cutting Stock Problems
- ▶ Heuristics
- ▶ Knapsack Problem
- ▶ Metaheuristics
- ▶ Parallel Computing

References

Anderson, R. J., Mayr, E. W., & Warmuth, M. K. (1989). Parallel approximation algorithms for bin packing. *Information and Computation, 82*(3), 262–271.

Bansal, N., Han, X., Iwama, K., Sviridenko, M., & Zhang, G. (2007) Harmonic algorithm for 3-dimensional strip packing problem. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans (pp. 1197–2007).

Bekrar, A., & Kacem, I. (2009). An exact method for the 2D Guillotine strip packing problem. *Advances in Operations Research*. doi:10.1155/2009/732010. Article ID 732010.

Bender, M., Bradley, B., Jagannathan, G., & Pillaipakkamnatt, K. (2007) Sum-of-Squares Heuristics for Bin Packing and Memory Allocation. *ACM Journal of Experimental Algorithmics*, *12*, 1–19. Article No. 2.3.

Berkey, J. O. (1990). *The design and analysis of parallel algorithms for the one-dimensional bin packing problem*. Ph. D. Dissertation, School of Information and Technology, George Mason University, Fairfax, VA.

Burke, E. K., Hellier, R. S. R., Kendall, G., & Whitwell, G. (2010). Irregular packing using the line and arc no-fit polygon. *Operations Research, 58*(4), 948–970.

Coffman, E. G., Jr., Courcoubetis, C., Garey, M. R., Johnson, D. S., Shor, P. W., Weber, R. R., et al. (2000). Bin packing with discrete item sizes, Part I: Perfect packing theorems and the average case behavior of optimal packing. *SIAM Journal of Discrete Mathematics, 13*(3), 384–402.

Coffman, E. G., Jr., Garey, M. R., & Johnson, D. S. (1996). Approximation algorithms for bin-packing a survey. In D. S. Hochbaum (Ed.), *Approximation algorithms for bin packing for NP-hard problems* (pp. 46–93). Boston: PWS Publishing Company.

Coffman, E. G., Jr., & Shor, P. W. (1993). Packing in two dimensions: Asymptotic average-case analysis of algorithms. *Algorithmica, 9*, 253–277.

Coleman, N. S., & Wang, P. Y. (1992). An asymptotically optimal parallel bin-packing algorithm. In *Proceedings of the fourth symposium on the frontiers of massively parallel computation*, IEEE Society, (pp. 515–516).

Correa, J., & Kenyon, C. (2004) Approximation schemes for multidimensional packing, In *Proceedings of the fifteenth ACM-SIAM symposium on discrete algorithms*, SIAM, (pp. 186–105).

Csirik, J., Johnson, D. S., Kenyon, C., Orlin, J. B., Shor, P. W., & Weber, R. R. (2006). On the sum-of-squares algorithm for bin packing. *Journal of the ACM, 53*(1), 1–65.

Faroe, O., Pisinger, D., & Zachariasen, M. (2003). Guided local search for the three-dimensional bin-packing problem. *INFORMS Journal on Computing, 15*(3), 267–283.

Fenrich, R., Miller, R., & Stout, Q. F. (1989). Hypercube algorithms for some NP-hard packing problems. In *Proceedings of the 4th conference and hypercubes, concurrent computers, and applications* (pp. 769–776). Los Altos, CA: Golden Gate Enterprises.

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

Hopper, E., & Turton, B. C. H. (2001). A review of the application of meta-heuristic algorithms to 2D strip packing problems. *Artificial Inteligence Review, 16*(4), 257–300.

Jakobs, S. (1996). On genetic algorithms for the packing of polygons. *European Journal of Operational Research, 88*, 165–181.

Lodi, A., Martello, S., & Monaci, M. (2002). Two-dimensional packing problems: A survey. *European Journal of Operational Research, 141*, 241–252.

Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research, 48*(2), 256–267.

Parreño, F., Alvarez-Valdes, R., Oliveira, J. R., & Tamarit, J. M. (2008). A hybrid GRASP/VND algorithm for two-and three-dimensional bin packing. *Annals of Operations Research, 131*, 203–213.

Puchinger, J., Raidl, G. R., & Pferschy, U. (2010). The multidimensional knapsack problem: Structure and algorithms. *INFORMS Journal on Computing, 22*(2), 250–265.

Seiden, S. (2002). On the online bin packing problem. *Journal of the ACM, 49*(5), 640–671.

Sweeney, P., & Paternoster, E. (1992). Cutting and packing problems: A categorized, application-orientated research bibliography. *Journal of the Operational Research Society, 43*(7), 691–706.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research, 183*(3), 1109–1130.
Xia, B., & Tan, Z. (2010). Tighter bounds of the First Fit algorithm for the bin-packing problem. *Discrete Applied Mathematics, 158*(15), 1668–1675.

Bipartite Graph

A graph or network whose nodes can be partitioned into two subsets such that its edges connect a node in each partition.

See

- ▶ Assignment Problem
- ▶ Graph Theory
- ▶ Network Optimization
- ▶ Transportation Problem

Birth-Death Process

A stochastic counting process that satisfies the following is called a birth-death process: (1) changes from state n (sometimes written more generally as state E_n) may only be to states $n+1$ or $n-1$ (i.e., changes can only be ± 1 unit); (2) the probability of a birth (death) occurring in the "small" interval of time, $(t, t+dt)$, given that the process was in state n at the start of the interval, is $\lambda_n dt + o(dt)[\mu_n dt + o(dt)]$, where $o(dt)$ is a function going to 0 faster than dt. Such processes are in fact Markov chains in continuous time. The system size of an M/M/1 queueing system is an example of a birth-death process where $\lambda_n = \lambda(n = 0, 1, 2, \ldots)$ and $\mu_n = \mu(n = 1, 2, \ldots)$. Markov chains; Markov processes.

Bland's Anticycling Rules

A set of pivot rules, the application of which to linear-programming (degenerate) problems, prevents cycling in the simplex algorithm. Their basic principle is that whenever there is more than one eligible candidate in selection of the variable entering the basis, or the variable leaving the basis, the candidate with the smallest index is chosen.

See

- ▶ Anticycling Rules
- ▶ Cycling
- ▶ Degeneracy

References

Bland, R. (1977). New finite pivoting rules for the simplex method. *Mathematics of Operations Research, 2*(2), 103–107.
Dantzig, G. B., & Thapa, M. N. (2003). *Linear programming 2: Theory and extensions*. New York: Springer.

Blending Problem

The linear-programming problem of blending raw materials, for example, crude oils, meats, to produce one or more final products, for example, fuels, sausages, so that the total cost of production is minimized. The problem is subject to restrictions on material availability, blending requirements, quality restrictions, etc.

See

- ▶ Activity-Analysis Problem
- ▶ Stigler's Diet Problem

Block Pivoting

The process of entering several nonbasic variables simultaneously into the basis in the simplex algorithm.

See

- ▶ Simplex Method (Algorithm)

Block-Angular System

A linear system of equations for which its matrix of coefficients A can be decomposed into k separate blocks of coefficients A_i, where each A_i represents the coefficients of a different set of equations. This structure typically represents a system consisting of k subsystems whose activities are almost autonomous, except for a few top-level system constraints whose variables couple the k blocks of the subsystems. Such systems can also have a few variables external to the blocks that couple the blocks.

See

- ▶ Dantzig-Wolfe Decomposition Algorithm
- ▶ Large-Scale Systems
- ▶ Weakly-Coupled Systems

Block-Triangular Matrix

A matrix which is lower (upper) triangular except for a number of blocks along the diagonal.

See

- ▶ Triangular Matrix

Bonferroni Inequality

Result in basic probability that provides a general lower bound on the intersection of events $E_1,\ldots,E_n$:

$$P(\bigcap_{i=1}^{n} E_i) \geq 1 - \sum_{i=1}^{n} P(E_i^c).$$

Note that the events need not be independent (nor mutually exclusive).

Applied in stochastic simulation output analysis to make statements about the overall confidence level of multiple performance measures (simultaneous confidence intervals). For example, for three output performance measures each with 99% confidence levels, the overall confidence level would be at least 97%.

Bootstrapping

In forecasting, the term bootstrapping refers to models that have been developed by regressing an individual's (or group's) forecasts against the inputs that the individual used to make the forecasts.

See

- ▶ Forecasting
- ▶ Regression Analysis

Bootstrapping: Resampling Methodology

Linda Weiser Friedman[1] and Hershey H. Friedman[2]
[1]Baruch College, City University of New York, New York, NY, USA
[2]City University of New York, Brooklyn, NY, USA

Introduction

Researchers typically encounter many situations in which parametric statistical techniques are less than ideal. The t-statistic, for example, assumes that the data were sampled from a normal distribution. Of course, much real-world data follow distributions that are far from normal, and may in fact be quite skewed. Suppose a researcher is investigating data that is known to follow an exponential distribution. Clearly, it would take an extremely large sample and a great deal of manipulation (e.g., averages of averages), for the central limit theorem to apply. In many cases, there is no parametric test for the measurement of interest because the sampling distribution of that measurement may be unknown and thus there would be no tractable analytic formulas for estimating such measures, for example, the difference between two medians (Mooney and Duval 1993, p. 8).

There are a number of nonparametric statistical techniques that do not rely on distributional assumptions and often may be used in place of the more traditional parametric tests. Many nonparametric techniques, however, work only with the median as a measure of central tendency (e.g., Mann-Whitney-Wilcoxon). This may present a problem for researchers who are more interested in the mean as the measure of interest.

The bootstrap statistic (Efron 1981, 1982; Mooney and Duval 1993) is a nonparametric, computer-intensive resampling technique, which makes no distributional assumptions and may be used for estimation and hypothesis testing. The bootstrap, jackknife, and other related resampling methods are beginning to generate interest among management scientists. Indeed, these tools can be very useful for the type of data that is frequently encountered by management scientists.

The Bootstrap Method

With traditional parametric inference, a sample is taken and a statistic, often the sample mean, is computed. This statistic is assumed to follow a known distribution (normal, t-distribution, F-distribution, χ^2-distribution, etc.), which then allows the researcher to perform hypothesis tests and/or estimate confidence intervals. With bootstrapping, which was developed mainly to determine the standard error for other types of estimates (Efron and Tibshirani 1991), the sample itself is used to construct a sampling distribution by selecting from it many resamples, or pseudo-samples.

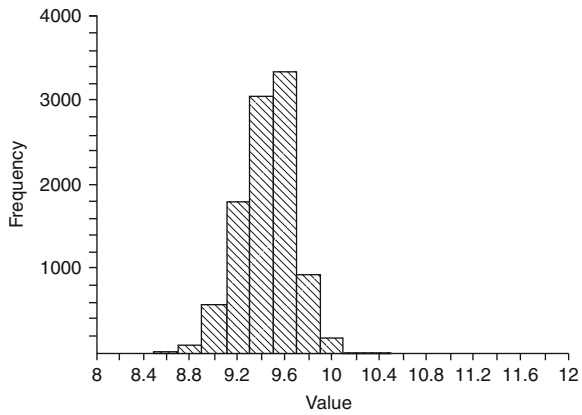
Resampling from the sample is done with replacement. Thus, it is like sampling from an infinite population with a composition that exactly matches that of the sample that was originally drawn. After resampling a great number of times one may construct a sampling distribution for a statistic of interest, such as the mean, median, or any percentile. This distribution, which is entirely based on the original sample and not on any theoretical distribution, may then be used to test hypotheses about measures of interest and to construct confidence intervals.

To illustrate the method, two illustrative examples are presented. The first is a hypothesis test for a sample from a single population; the second, for samples from two presumably different populations.

Bootstrapping: Resampling Methodology, Table 1 Sample data and statistics, Example I

Life	Frequency
8.0	3
9.0	5
10.0	6
11.0	2

$\bar{x} = 9.438$, $s = 0.964$, $n = 16$



Bootstrapping: Resampling Methodology, Fig. 1 Histogram of mean lifetimes, 10,000 resamples, Example I

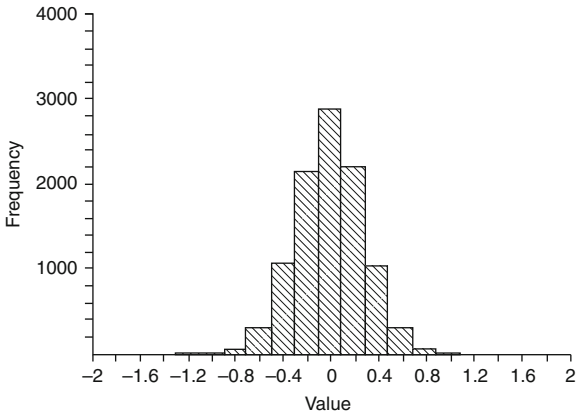
Example I A company claims that the average life of a part that it manufactures is at least 10 hours. A sample of 16 parts is taken in order to test this claim. The sampled data is summarized in Table 1.

A parametric analysis using the t-statistic would have to assume that the underlying population is normally distributed since the sample is too small to rely on the central limit theorem. Moreover, this type of data is usually not normally distributed, or even symmetrical.

Using the bootstrap method, 10,000 resamples, each of size $n = 16$, were taken from the original data. Figure 1 is a histogram of the 10,000 resampled means. One can see that the means seem to be hovering about the values 9.3 to 9.7 hours, and very few are actually above 10.0. Table 2 confirms that only a small fraction of the means were above 10.0. As a matter of fact, the 95 percent one-sided confidence interval is bounded by the value of 9.8125 hours. This means that only 5 per cent of the resamples had mean values above 9.8125. Clearly, the claim that the average life of these parts is at least 10 hours should be rejected.

Bootstrapping: Resampling Methodology, Table 2 Frequency distribution of mean lifetimes, Example I (note that each category covers all values within 0.1 of its center)

Center value	Frequency	Percent	Cum percent
8.6	7	0.1	0.1
8.8	92	0.9	1.0
9.0	583	5.8	6.8
9.2	1798	18.0	24.8
9.4	3057	30.6	55.4
9.6	3342	33.4	88.8
9.8	920	9.2	98.0
10.0	184	1.8	99.8
10.2	16	0.2	100.0
10.4	1	0.0	100.0



Bootstrapping: Resampling Methodology, Fig. 2 Histogram of mean lifetimes, 10,000 resamples, Example II

Bootstrapping: Resampling Methodology, Table 3 Sample data and statistics, Example II

Group 1	Group 2
13.8	12.6
13.3	12.4
13.7	12.9
13.6	13.3
15.2	14.2
14.4	13.0
13.6	13.4
13.3	12.9
13.6	13.5
13.8	13.6
$\bar{x}_1 = 13.83$	$\bar{x}_2 = 13.18$
$s_1 = 0.57$	$s_2 = 0.53$
$n n_1 = 10$	$n_2 = 10$

Bootstrapping: Resampling Methodology, Table 4 Frequency distribution of mean lifetimes, Example II (note that each category covers all values within 0.1 of its center)

Center value	Frequency	Percent	Cum percent
21.2	1	0.0	0.0
21.0	4	0.0	0.1
20.8	42	0.4	0.5
20.6	273	2.7	3.2
20.4	1065	10.6	13.9
20.2	2164	21.6	35.5
0.0	2876	28.8	64.3
0.2	2189	21.9	86.1
0.4	1026	10.3	96.4
0.6	305	3.0	99.4
0.8	50	0.5	99.9
1.0	5	0.1	100.0

Example II A similar type of analysis can be done for a two-sample test. Consider the data in Table 3, representing the life (in weeks) of similar parts from two different manufacturers or two different production processes. As in Example I, a parametric test would require an assumption of normally distributed lifetimes, which again may be unrealistic.

With the bootstrapping approach, first combine the two groups of data into one (i.e., under the assumption that H_0 is true). Then, this combined group is resampled to produce two groups of data items, and the mean difference of the two groups $\bar{x}_1 - \bar{x}_2$ is recorded. This resampling is done many times, and the resulting mean differences are compared with the observed mean difference in the original set of data.

In the above example, the observed mean difference is 0.65 weeks (13.83–13.18). The question is, what is the likelihood that this difference occurred by chance? Since this is a two-tailed test, consider resamples for mean differences greater than 0.65 or less than -0.65. Figure 2 contains the histogram of the mean differences of 10,000 re-samples, in which each resample produced two groups of size $n = 10$ each. Examination of this histogram and of Table 4 shows that almost all of the mean differences fall between–0.5 to 0.5. Actually, only 1.85% of the resampled mean differences were either greater than 0.65 or below–0.65. At a significance level of 0.05, reject the hypothesis that the two population means are the same.

Concluding Remarks

Bootstrapping is clearly a technique that is very useful to researchers. It should, however, be pointed out that this technique is totally dependent on the integrity of the original sample of data. If the sampled data is indeed a good representation of the underlying distribution, inferences based on resampling this data will be valid. On the other hand, if the original sample, say, over represents high values of the output distribution, then the resamples and inferences based on them cannot be trusted. If the sample is biased, the resampling technique may reflect and possibly magnify these biases.

Some areas in operations research and management science that have made use of bootstrapping and other resampling techniques include: quality control (Jeske 1997; Seppala 1995), analysis of simulation output (Friedman and Friedman 1995; Kim et al. 1993), neural networks (LeBaron 1998; Shimshoni 1998), performance evaluation (Cho 1997), and production (Jochen 1997).

Mooney and Duval (1993) describe how the bootstrap procedure may be used with SAS and RATS. Resampling Stats (Simon 1995), a simple computer package for bootstrapping, is user-friendly, relatively inexpensive, and comes with numerous examples. Fan and Jacoby (1995) describe a SAS/IML program for performing the bootstrap resampling technique in regression analysis. Bootstrapping can also be done with spreadsheets (Willemain 1994).

See

▶ Regression Analysis

References

Cho, K. (1997). Performance assessment through bootstrap. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*, 1185–1198.

Efron, B. (1981). Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrika, 68*, 589–599.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. Philadelphia: SIAM.

Efron, B., & Tibshirani, R. (1991). Statistical data analysis in the computer age. *Science, 253*, 390–395.

Fan, X., & Jacoby, W. G. (1995). Bootsreg: An SAS matrix language program for bootstrapping linear regression models. *Educational and Psychological Measurement, 55*, 764–768.

Friedman, L. W., & Friedman, H. H. (1995). Analyzing simulation output using the bootstrap method. *Simulation, 64*, 95–100.

Jeske, D. R. (1997). Alternative prediction intervals for pareto proportions. *Journal of Quality Technology, 29*, 317–326.

Jochen, V. A. (1997). Using the bootstrap method to obtain probabilistic reserves estimates from production data. *Petroleum Engineer International, 70*, 55 +.

Kim, Y. B., Willemain, T. R., Haddock, J., & Runger, G. C. (1993). The threshold bootstrap: A new approach to simulation output analysis. *Proceedings of the 1993 Winter Simulation Conference*, 498–502.

LeBaron, B. (1998). A bootstrap evaluation of the effect of data splitting on financial time series. *IEEE Transactions on Neural Networks, 9*, 213–220.

Mooney, C. Z., & Duval, R. D. (1993). *Bootstrapping: A nonparametric approach to statistical inference*. Newbury Park, CA: Sage Publications.

Seppala, T. (1995). Statistical process control via the subgroup Bootstrap. *Journal of Quality Technology, 27*, 139–153.

Shimshoni, Y. (1998). Classification of seismic signals by integrating ensembles of neural networks. *IEEE Transactions on Signal Processing, 46*, 1194–1120.

Simon, J. L. (1995). *Resampling stats user's guide*. Arlington, VA: Resampling Stats.

Willemain, T. R. (1994). Bootstrap on a shoestring: Resampling using spreadsheets. *The American Statistician, 48*, 40–42.

Bounded Rationality

The concept that a decision maker lacks both the knowledge and computational skill required to make choices in a manner compatible with economic notions of rational behavior.

See

▶ Choice Theory
▶ Decision Analysis
▶ Multiple Criteria Decision Making
▶ Satisficing

References

Simon, H. A. (1955). A behavioral model of rational choice. *Quatrely Journal of Economics, 69*, 99–118.

Simon, H. A. (1957). *Models of man: Social and rational*. New York: John Wiley & Sons.

Bounded Variable

A variable x_j in a linear-programming problem that is required to satisfy a constraint of the form $0 \leq x_j \leq b$, $-b \leq x_j \leq 0$, or $b_1 \leq x_j \leq b_2$, where b is some positive constant and $b_1 \leq b_2$.

See

► Linear Programming

Branch

To move and analyze a new computational path (i.e., branch) based on the results obtained from a previous path.

See

► Branch and Bound

Branch and Bound

A method for solving an optimization problem, by successively partitioning (branching) the set of feasible points to smaller subsets, and solving the problem over each subset. The resulting problems are called subproblems or nodes in the enumeration tree. The idea in branch and bound is that the optimal solution to the problem is the best among the optimal solutions to the subproblems. To reduce the number of subproblems solved, best-case bounds are computed by solving relaxed problems defined at the nodes. If the best-case bound on a solution to a subproblem is worse than the best available solution, the subproblem is eliminated from consideration (fathomed). Branch and bound techniques are frequently used to solve integer-programming problems, as well as in global optimization.

See

► Global Optimization
► Integer and Combinatorial Optimization
► Integer-Programming Problem

Brownian Motion

A one-dimensional Brownian motion $\{B(t),\ 0 \leq t\}$ is a continuous-time, Markovian, real-valued stochastic process having continuous sample paths; its distribution is Gaussian with mean function $E[B(t)] = \mu t$ and covariance function $\mathrm{Cov}[B(s), B(t)] = \sigma^2 \min(s, t)$. An n-dimensional Brownian motion is a stochastic process on $\mathbb{R}^n$ whose n components are independent one-dimensional Brownian motions. Named after Scottish botanist Robert Brown. Also known as the Wiener process, named after mathematician Norbert Wiener.

See

► Markov Processes

BTRAN

The procedure for computing the dual variables in a simplex iteration, when the LU factors of the basis matrix are given in product form. The name BTRAN (backward transformation) derives from the fact that the eta file is scanned backwards in the solution process.

See

► Eta File

Buffer

The queue or the waiting room in a queueing system. Most often used for networks, especially tandem networks or series queues.

See

► Queueing Theory

Bulk Queues

Arrivals to a queueing system may consist of more than one customer at a time, and/or service might process more than one customer simultaneously.

See

► Queueing Theory

Bullwhip Effect

► Supply Chain Management

Burke's Theorem

The steady-state departure process of a stable M/M/*c* queueing system is a Poisson process with the same rate as the arrival process, irrespective of the service rate.

See

► Queueing Theory

Business Intelligence

Paul Gray
Claremont Graduate University, Claremont, CA, USA

Introduction

Business Intelligence (BI) systems are sophisticated analytical tools that present complex organizational and competitive information in a way that allows decision makers to decide quickly and appropriately. While the term Business Intelligence is relatively new (it was introduced in 1989, popularized in the 1990s), computer-based BI systems existed, in one guise or another, decades prior to that. BI-type functionality was available previously to varying degrees in Financial Planning Systems (4GLs), Executive Information Systems (EIS), Decision Support Systems (DSS), Data Mining, and On Line Analytic Programming (OLAP). With each new iteration, capabilities increased as enterprises grew ever-more sophisticated in their computational and analytical needs and as computer hardware and software matured. This article explores the capabilities of state-of-the-art BI, their benefits to adopters, and the role of Analytics in BI.

BI describes data-driven decision support systems (Power 2005) for managers. In its initial form, it involved business analysts who refined (mostly internal) business data to create input for management. Such systems have been marketed commercially since the 1960's, if not earlier. BI is now closely linked to Analytics, the use of quantitative methods for solving organizational problems. BI is broader than Analytics because it involves soft methodologies and information systems, as well as Operations Research (OR).

Objective and definition of BI: The objective of BI is to improve the timeliness and quality of the input to the decision process.

To achieve this objective, BI systems combine:

Data gathering		
Data storage	with	Analysis
Knowledge management		

to evaluate complex organizational and competitive information and present the results to planners and decision makers.

The first three operations are inputs, typically performed by people with information systems and data analysis skills. The skills of Analytics are brought to the table by people trained in OR, statistics, and other quantitative disciplines.

Implicit in its definition is the idea that BI systems provide actionable information and knowledge at the right time, at the right place, and in the right form.

Problems to which BI is Applied: BI aims to convert data available to the organization into

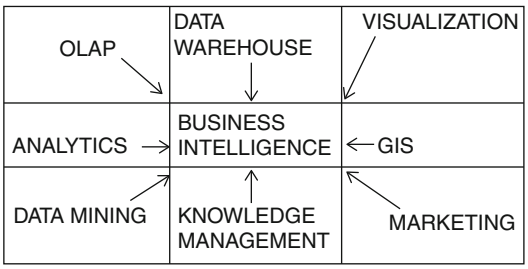
information and, through analysis, into knowledge. Among the many tasks that BI performs are:

- Examine the opportunities for
  - proposed products,
  - mergers and acquisitions,
  - acquiring new customers, and
  - locating sites for new branches.
- Create forecasts based on historical data, current performance, and estimated future performance. Futures methodologies such as Delphi and cross-impact analysis are discussed in Glenn and Gordon (2008).
- Monitor key performance indicators (KPI) both for the organization and its competitors.
- Do "what if" analysis to examine the impacts of changes and of alternative scenarios.
- Ad hoc access to data to answer specific, non-routine questions.

These examples cover both regular, repetitive scheduled reporting (e.g., monthly reports on sales by region, department, or strategic business unit), and special investigations aimed to solve specific problems. Forecasting and many of the specific problem studies involve OR modeling that uses the organization's data warehousing capabilities for the underlying information. For example, specific studies undertaken in response to a crisis or an opportunity such as a contract proposal.

BI vs. Competitive Intelligence: Business Intelligence uses technologies, processes, and applications to analyze mostly internal, structured data, and business processes, while Competitive Intelligence (discussed below) gathers, analyzes, and disseminates information from both external and internal sources to provide a framework for assessing the organization's position relative to its industry and non-industry competitors and its vulnerability to disruptive technologies.

Previous Systems: Present-day BI systems reflect a series of iterations to obtain their present functionality. These included (1) 3rd Generation financial planning languages that allowed writing relations in words rather than symbols. (For example, rather than saying S = M*MS, one could write Sales = market*market share.) (2) Executive information systems that can create PowerPoint charts to brief management on the current state of the business. (3) On-Line Analytic Processing in which data warehouses that store data in the form of 2-dimensional relational data bases are used to create multidimensional data cubes (see below). Although each of these elements is more sophisticated than the one before, they were individual systems, while the hallmark of current BIs is the integration of such systems.



Business Intelligence, Fig. 1 Software Components of Inputs to Business Intelligence

BI Input Software

BI is deeply tied to the ability to store data bases and to compute at the organizational or departmental level. Key elements include data warehouses and data marts. As shown in Fig. 1, many software capabilities are involved.

The software components used in BI include:

- A *Data Warehouse* is a collection of data bases that contain both current and historical information about the organization. The warehouse is separate from operational systems that support on-line transaction systems. It contains "a single version of the truth" and is intended to support understanding of the organizational data over time. It is particularly important for BI.

To create the single version of the truth, data goes through a process known as ETL (extract, transform, load). The ETL applies procedures that extract data from selected sources, transforms it into the format of the data warehouse that is consonant with the warehouse's rules, and then stores the data into the warehouse or mart. ETL is important for BI because it standardize the data and eliminates redundancies and inaccuracies.

Data warehouses come in two sizes:

- A data warehouse, which support an entire organization or one of its major portions.
- A data mart that is a smaller version of a data warehouse but has all features of a warehouse. It can

Business Intelligence, Table 1 Characteristics of the Data Warehouse

Characteristics	Description
Subject oriented	Data are organized by how users refer to it.
Integrated	Inconsistencies are removed in both nomenclature and conflicting information, i.e, combining of all related data around a common identifier/key
Non-volatile	Read-only data are not updated by users
Time Series	Data are time series, not current value. A typical data warehouse has 5 to 10 years of data.
Summarized	Operational data are aggregated into decision usable form where appropriate
Larger	Much more data is retained than in transaction systems because it offers time series.
Non-normalized	Data can be redundant for ease of retrieval and use.
Metadata	Data about the data are available to users and to warehouse personnel.
Input	Include both operational data and external data

Source: Gray and Watson (1998)

be dependent or independent. If dependent, it contains a subset of the data warehouse needed by specific groups, such as Analytics. But, multiple independent data marts cannot substitute for a data warehouse because data integrity is not maintained among them.

Over time, a number of specialized data warehouses have evolved. They include operational data stores, real time warehouses, prototype warehouses, and exploration warehouses discussed below.

The characteristics of the data warehouse are listed in Table 1. These characteristics assume that the data warehouse is physically separated from operational systems and that its databases are not used for on-line transaction processing (Inmon 1992).

- *OLAP* (**o**n-**l**ine **a**nalytic **p**rocessing) is used to analyze multidimensional data for a BI. It is used for such tasks as sales analysis, budgeting, forecasting, and financial reporting where it is necessary to manipulate and consolidate data from multiple sources. Data are specially configured for OLAP into data cubes to allow complex questions to be answered more quickly than for relational data bases. OLAP has subdivided into relational, multidimensional, hybrid, and other forms, which are typically referred to as ROLAP, MOLAP, and HOLAP.
- *Analytics* refers to the use of quantitative and statistical methods together with extensive computing and modeling to make sense of the data. It is the area of BI that attracts operations researchers. Mathematics is the base for Analytics. The objective is to obtain realistic and, if possible, optimal alternatives for decision making about the future. Analytics is discussed in more detail below.
- *Data Mining* [also referred to as knowledge data discovery (KDD)] is a form of predictive analytics discussed below. It is a set of analytical techniques to obtain new insights from the data in the data warehouse that an analyst or a manager had not thought to ask. It is used to find answers that reports and queries do not reveal effectively. KDD seeks to find patterns in data and to infer rules. Data mining differs from conventional hypothesis testing in that it looks at data for the relationships it contains to form hypotheses that can be tested. KDD techniques include neural networks, expert systems, fuzzy logic, intelligent agents, multidimensional analysis, data visualization, and decision trees. Data mining is used in wide range of topics, e.g., to identify where people are likely to take vacations, detect fraud, analyze loan quality, and the reported (but apocryphal) association that men who buy diapers on Friday night also buy beer.
- *Knowledge Management*. Knowledge can be tacit and explicit. Tacit knowledge is what is in one's head but cannot usually be expressed, although there are techniques for obtaining some tacit knowledge. Explicit knowledge is about what can be written down, stored, and retrieved. Knowledge management is about knowing what the organization knows and finding new knowledge that is needed when the organization does not know. It focuses on creating, sharing, and applying knowledge. In BI, the explicit information in the data warehouse and in reports is merged with the tacit knowledge in the heads of analysts and professionals.
- *Geographic Information Systems*. These systems link data bases to geographic maps of physical locations. They are used to analyze spatial phenomena. For example, they allow overlaying of customer, distribution center, retailer, and other information about a firm's and its competitor's products.

- *Marketing*. Analytics are used to understand the implications of existing and proposed policies in the marketplace. For example, data from aggregators and from the firm are used to create forecasts of market size and market size.
- *Visualization*. Visualization refers to methods to present information on-screen in a form comprehensible to non-technical managers. It does not replace Analytics; it focuses the analytic results. By exploiting visuals, it provides an overview of complex data sets and allows for identifying relationships and trends in data and in analytical results.

BI Outputs: Dashboards and Reports

Dashboards. In BI, a dashboard is way of communicating results in a form that is easily understood by managers. A dashboard is a visual screen that shows the key performance indicators. The data, drawn from internal information systems and analyses, not only summarize the current status, but also provide historical data, warning levels, next steps, and notices. It includes financial and non-financial measures.

The idea of a dashboard has been in use since the 1960's. At that time, summarized data for managers was displayed on color slides at regular management meetings. For example, the experience at AT&T from color slides was that if the dashboard slides presented the current data in the same format at each meeting, managers would rapidly find and be sensitive to changes that required action.

Introducing the computer provided an instant display device, improved visualization, and provided data on the desktop tailored to each user. For example, the VP for manufacturing and the VP for human resources can see results specifically oriented to their issues. Furthermore, the displays allow drill down; that is they start with a broad view and then let the user see greater and greater detail..

The three main applications are:

- *monitoring* information at a glance. Usually involves key performance indicators (KPI) in graphical, symbol, or symbolic form.
- *analysis* of exceptions to find root causes of problems. Summarized multi-dimensional data and drill down in "slice and dice" fashion are used.
- *Identify actions* to solve problems based on access to detailed operational data, queries, and reports.

Reports include:

- regular, repetitively scheduled documents (e.g., monthly sales by region, department, or strategic business unit),
- exception reports which are produced whenever parameters are outside pre-specified bounds,
- documents presenting the results of special investigations (often in response to requests from BI users), and
- custom data cubes based on specific requests from analysts.

Forecasting and many of the specific studies involve OR modeling that uses the organization's data warehousing capabilities for the underlying information. For example, specific studies are undertaken in response to a crisis or an opportunity such as a contract proposal.

BI Architecture

Figure 2 (based on Skriletz 2002) is typical of the architecture for a large installation that centers on the use of Web technology for distribution. As shown, the input data come from a variety of systems into the data warehouse. The specific data needed for BI is downloaded to a data mart used by planners and executives.

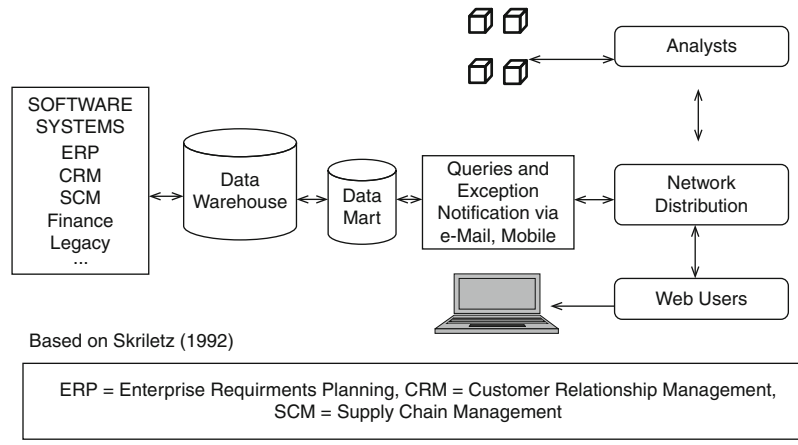
As shown, the specific applications used for BI include the organizational focus and the audience (Skriletz 2002).

The left side of Table 2 shows the business focus of the technologies, while the right side shows the levels of people in the organization who are the consumers of the intelligence. At the bottom of the hierarchy is transaction processing based on application-specific data in the warehouse or in ERP or in sales systems. The next level involves processing the data so that it is useful to first level managers. Here, Analytics and pattern analysis are performed and data are presented in visual form. At the top level, predictions, compilations of competitive analyses, and summary presentations for executives are created.

Tools. Many of the tools for BI are used for other applications as well. They include:

- simple querying and reporting,
- on-line analytic processing (OLAP),

Business Intelligence, Fig. 2 BI Architecture



Business Intelligence, Table 2 Business Intelligence Aplications

Organizational focus				Audience
Strategy	Predictive and Prescriptive Analytics	Competitive Intelligence	Intelligence Presentation and Visualization	**Top executives**
Operations Analysis	Data Analytics			**Operations Directors**
Operations Monitoring		Heuristic Pattern Analysis		**Operations Supervisors**
Transaction Processing	Platform for BI - Manage Through Metadata Application-Specific BI (e.g., SAS, IBM, Oracle, SAP)			**Operations staff**

- statistical analyses and data mining,
- forecasting, and
- geographic information systems and visualization.

In addition, the extraction, translation, and loading (ETL) tools of data warehousing are important for BI because they help standardize the data so it can be analyzed with accuracy and provide a single truth. When operational data is used, as from an Operational Data Store (ODS), the objective is to get dynamic data that reflects the current situation.

The key dissemination method for business technology is internet technology, whether it be an intranet within the firm or an extranet connected to suppliers and/or clients. The idea is to reach everyone who needs specific data, rather than a few at corporate headquarters.

Business Analytics

In the 20th century, most information systems were used to standardize routine business processes to minimize cost and time. Fairly sophisticated decision support systems and data warehouses were in use, but these systems rarely directly affected the ways decisions were made (Drucker 1999). BI was mostly the province of the information systems groups in organizations. It centered on providing inputs for data-based decision making. It was only after the turn of the century that it was generally realized that applying Analytics would improve to data-based decision making. This realization was reinforced by leading vendors, such as IBM, Microsoft, Oracle, and SAP, acquiring major BI software firms and investing in expanding BI software capabilities. It became clear that the analytic skills and the methods of OR analysts are needed to exploit information technology capabilities.

The definition of business Analytics is still in flux. Davenport and Harris (2007) defined it as "the extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact-based management to drive decisions and actions." In this definition, Analytics is a subset of BI, that is, technologies and processes that use data to understand and analyze performance. A broader

definition comes from IBM who uses it to refer to both software applications and analytic solutions (Lustig et al. 2010). In their view, software includes BI, performance management, prediction, optimization, enterprise information management, content, and collaboration. Analytic solutions involve finance, risk, fraud, customer relations, human capital, and supply chain. Underlying Analytics is the idea that data and information are strategic assets.

Analytics can be divided into three categories (Lustig et al. 2010):

- Descriptive analytics
- Predictive analytics
- Prescriptive analytics

All three categories start with the underlying idea that data and information are strategic assets.

Descriptive analytics is the classic form of BI. It starts by examining, consolidating, and classifying data. Data sources include information from departments (marketing, sales, operations, accounting), enterprise systems (ERP, CRM, and Supply Chain Management SCM), as well as spreadsheets, other databases, and external data from 3rd parties. The outputs are ad hoc and exception reports, dashboards, KPI, statistical analyses, drill down, and answers to ad hoc queries about business performance. These outputs allow for the managing and monitoring of business processes. Descriptive analytics are often inputs to predictive and prescriptive analytics.

Predictive analytics combines the data within a wide variety of mathematical procedures to create models that explain and/or predict performance. It is based on inherent relations between the data inputs and outcomes. Predictive analytics uses data on what happened in the past to detect patterns and relations to make forecasts. Its methods include, among others (Lustig, et al. 2010):

Data mining	Correlations among data
Forecasting	Extrapolations of trends into the future
Monte Carlo simulation	What may happen if changes occur
Root-cause analysis	Evaluation of why things happened
Pattern recognition	Alerts when unusual situations occur
Predictive modeling	Forecasts by Delphi or other methods

Prescriptive analytics refers to mathematical techniques that provide understanding of alternative courses of actions when there are competing objectives, requirements, and constraints. It involves both static and stochastic optimization. The former leads to determining the best outcome, while the latter considers the effects of data uncertainty to improve decisions. Given the increase in computer speed and memory, improved algorithm performance, and in data quality, prescriptive analytics can be run in near-real-time so they can affect operational as well as strategic decisions.

Integrating Analytics and BI

Where traditional BI depends principally on aggregating, evaluating and manipulating the information in the data warehouse, Analytics adds modeling and optimization. Irrespective of which type of Analytics (descriptive, predictive or prescriptive) is used, the results need to be communicated to the user community.

This communication capability involves a series of steps (Shapiro 2010):

- Develop models to optimize decisions for key performance indicators.
- Select the right modeling system. It may be customized or off the shelf.
- Define the database needed for the optimization model. It may be customized or off-the-shelf
- Create the decision database. This may require new ETL routines and descriptive models
- Link the database and the outputs from the optimization model to the organization's reporting tools to be able to communicate results to users.
- For strategic and tactical decisions, reuse criteria for alerts and redo modeling studies at regular intervals. For operational decisions, exercise the operational models in real-time with current data.

Competitive Intelligence

The notion of competitive intelligence (CI) as spy vs. spy, fed by such examples as Japan and China allegedly stealing U.S. industrial secrets, is far from the real situation. That does not mean that companies do not try to find out as much as possible about their current and potential competitors. The people involved, however, claim that they do so in a legal and ethical manner. CI is defined by the Society for

Competitive Intelligence Professionals (SCIP) as the process of monitoring the competitive environment. To do so, analysts systematically gather, analyze, and manage information that can affect a company's plans, decisions, and operations.

The competitive intelligence cycle includes:

- Determine the intelligence needs of decision makers
- Collect information to meet these needs
- Analyze the data and recommend actions
- Present results to the decision makers
- Use the response to the findings to refine collection.

The focus is on determining both the current activities and the likely intentions of other firms and of governments. It also includes looking for the possible appearance of disruptive technologies and finding out about how competitors are responding to your actions.

The collected raw data (facts, statistics) are organized and then analyzed to find patterns, trends, and relationships. The tools used include:

- Simulations of alternative scenarios to test what if conditions
- Data mining of information about both competitors and the firm
- Assessing competitor technologies by tracking (and extrapolating from) patent filings.
- Attending trade shows and conferences
- Scanning publicly available data: public records, the Internet, press releases, and mass media.
- Talking with customers, suppliers, partners, industry experts

Much of the data gathering work is terribly dull and routine. To be effective, it has to be someone's (or some group's) responsibility.

For many organizations, the only basis for evaluating their competitors is by applying the SWOT technique: **S**trengths, **W**eaknesses, **O**pportunities and **T**hreats. SWOT, as taught in business schools, is often done qualitatively based on individuals intuitively assessing a particular competitor. The technique can and should be done using Analytics.

True competitive analysis goes far beyond SWOT. Table 3 shows the results of a survey of the use and effectiveness of CI analysis techniques.

Since this table was compiled, an important new source of competitive data has come to the forefront. That is the analysis of social media data. People do put things on social media (e.g., Facebook, Twitter) that they would not put in writing in e-mail or other forms.

Business Intelligence, Table 3 CI Analysis Tools

Tool	Percent Using	Tool	Effectiveness Percentage
Competitor profiles	88.9	SWOT analysis	63.1
Financial analysis	72.1	Competitor profiles	52.4
SWOT analysis	55.2	Financial analysis	45.5
Scenarios	53.8	Win/loss analysis	31.4
Win/loss analysis	40.4	Gaming	21.9
Gaming	27.5	Scenarios	19.2
Conjoint analysis	25.5	Conjoint analysis	15.8
Simulation	25.0	Simulation	15.4

Source: Powell and Allgaier (1998)
Note: The two columns of table on the *left* shows the percentage of respondents using the technique. The two columns of the table on the *right*, which list the same techniques, shows the percentage of those who believe the technique is effective.

Some companies that practice competitive analysis realize that just as they gather data about competitors, competitors are likely to gather data about them. They therefore try to protect their own information by becoming secretive about their plans. They control their press releases, approve speeches by their executives, provide security training for their employees, and more to avoid leaks about their intentions.

References

Davenport, T. H., & Harris, J. G. (2007). *Competing on analytics: The new science of winning*. Boston: Harvard Business School Press.

Davenport, T. H., Harris, J. G., & Morison, R. (2010). *Analytics at work: Smarter decisions, better results*. Boston: Harvard Business School Press.

Drucker, P. (1999). Beyond the information revolution. *The Atlantic, 284*(4), 47–54.

Eckerson, W. W. (2006). *Performance dashboards: Measuring, monitoring, and managing your business*. New York: Wiley.

Glenn, J. C., & Gordon, T. J. (2008). *Futures research methodology, version 3.0*. The millennium project, Washington, DC.

Gray, P., & Watson, H. (1998). *Decision support in the data warehouse*. Prentice Hall, NJ.
Howson, C. (2008). *Successful business intelligence: Secrets to making BI a killer app*. New York: McGraw Hill.
Lustig, I., Dietrich, B., Johnson, C., & Dzekian, C. (2010, November-December). The analytic journey. *Analytics*.
Powell, T., & Allgaier, C. (1998). Enhancing sales and marketing effectiveness through competitive intelligence. *Competitive Intelligence Review, 9*(4), 29–41.
Power, D. J., (2005). *Decision support systems: Frequently asked questions*. New York: iUniverse.
Sabherwal, R., & Becerra-Fernandez, I. (2011). *Business intelligence: Practices, technologies, and management*. New York: Wiley.
Siegel, J. (2010 September-October). Business intelligence & modeling systems synergy. *Analytics*.
Skriletz, R. (2002 April). New directions for business intelligence. *Information Management*.

Busy Period

A time interval that starts when all the servers of a queueing system become busy and ends when at least one server becomes free. May also refer to a time interval that starts when a previously completely idle system begins serving any customer and ends when the system becomes idle again. The two definitions (sometimes distinguished as full and partial busy periods, respectively) coincide for a single-server queue.

See

► Queueing Theory