Markov processes. Introduction. General modelling concepts. State space diagrams. Stochastic transitional probability matrix. Evaluating limiting state probabilities. Evaluating time dependent state probabilities. Reliability evaluation in repairable systems. Mean time to failure. Application of techniques to complex systems. Conclusions. Problems. *Chapter 10: Frequency and duration techniques.* Introduction. Frequency and duration concepts. Application to multi-state problems. Frequency balance approach. Two stage repair and installation process. Conclusions. Problems. *Chapter 11: Approximate system reliability evaluation.* Introduction. Series systems. Parallel systems. Network reduction techniques. Minimal cut set/failure modes approach. Inclusion of scheduled maintenance. Common mode failures. Conclusions. Problems. *Chapter 12: Systems with non-exponential distributions.* Introduction. Method of stages. Stages in series. Stages in parallel. Series stages in series with two parallel stages. Time dependent and limiting state probabilities. Conclusions. Problem. *Chapter 13: Epilogue.* Appendix 1 - Rules of Boolean algebra. Appendix 2 - The normal distribution function. Appendix 3 - Elementary matrix algebra. Appendix 4 - Differential equations and Laplace transforms. Appendix 5 - Confidence levels and limits. *References. Solutions. Index.*

John M. MULVEY (editor), *Evaluating Mathematical Programming,* Proceedings of a Conference Held at the National Bureau of Standards, Boulder, CO, January 5-6, 1981, Lecture Notes in Economics and Mathematical Systems 199 (Springer-Verlag, Berlin - Heidelberg - New York, 1982) 379 pp.

Abstract. Welcoming Remarks. Opening Address - Darwin Klingman. Part 1: Design and Use of Problem Generators and Hand Selected Test Cases. *Ronald L. Rardin and Benjamin W. Lin:* Test problems for computational experiments - issues and techniques. *Joyne J. Elam and Darwin Klingman:* NETGEN-II: A system for generating structured network-based mathematical programming test problems. *Jerrold H. May and Robert L. Smith:* The definition and generation of geometrically random constraint sets. *Gideon Lidor:* Construction of nonlinear programming test problems with known solution characteristics. *Richard O'Neill:* A comparison of real-world linear programs and their randomly generated analogs. Part 2: Nonlinear Optimization Codes and Empirical Tests. *Ernie Eason:* Evidence of fundamental difficulties in nonlinear optimization code comparisons. *Eric Sandgren:* A statistical review of the Sandgren-Ragsdell comparative study. *Jacques C.P. Bus:* A methodological approach to testing of NLP-software. Part 3: Integer Programming and Combinatorial Optimization. *William R. Stewart, Jr.:* A computational comparison of five heuristic algorithms for the Euclidean traveling salesman problem. *Uwe Suhl:* Implementing an algorithm: performance considerations and a case study. *William J. Riley and Robert L. Sielken, Jr.:* Which options provide the quickest solutions. *Michael Chang and Fred Shepardson:* An integer programming test problem generator. Part 4: Comparative Computational Studies in Mathematical Programming. *Ron S. Dembo:* Introduction. *David M. Himmelblau:* Ramarks on the evaluation of nonlinear programming algorithms. *Robert B. Schnabel:* Comments on evaluating algorithms and codes for mathematical programming. *Jacques C.P. Bus:* Some comments on recent computational testing in mathematical programming. *Ken M. Ragsdell:* Remarks on the comparative experiments of Miele, Sandgern and Schittkowski. Part 5: Testing Methodologies. *Karla L. Hoffman and Richard H.F. Jackson:* In pursuit of a methodology for testing mathematical programming software. *Klaus Schittkowski:* Nonlinear programming methods with linear least squares subproblems. *Kathie L. Hiebert:* An outline for comparison testing of mathematical software - illustrated by comparison testings of software which solves systems of nonlinear equations. *A. Buckley:* A portable package for testing minimization algorithms. Part 6: Approaches to Software Testing from Other Disciplines. *William J. Cody:* Transportable test procedures for elementary function software. *James E. Gentle:* Testing and evaluation of statistical software. *Leon J. Osterweil:* TOOLPACK - An integrated system of tools for mathematical software development. *Lloyd D. Fosdick:* Overview of testing numerical software. *Charles P. Smith:* The application of Halstead's software science difficulty measure to a set of programming projects. Part 7: Special Topics. *Harlan Crowder:* Mathematical programming algorithms in APL. Part 8: Advances in Networks. *Richard S. Barr:* Solution strategies and algorithm behavior in large-scale network codes. *Robert R. Meyer:* Recursive piecewise-linear approximation

methods for nonlinear networks. *Michael Engquist:* computational testing of assignment algorithms. Part 9: On establishing a Group for Testing Mathematical Programs. *John M. Mulvey:* Introduction. Panel Discussion. Part 10: Appendix. Conference program. List of participants. *Klaus Schittkowski:* A model for the performance evaluation in comparative studies. *Angelo Miele:* Remarks on the comparative evaluation of algorithms for mathematical programming problems. *Angelo Miele:* Comments on a testing center. *Salvador Gonzalez:* Systematic approach for comparing the computational speed of unconstrained minimization logarithms. *Ken Ragsdell:* The evaluation of optimization software for engineering design.