THE NATIONAL ACADEMIES PRESS

Multi-Objective Optimization for Bridge Management Systems

AUTHORS

BUY THIS BOOK

FIND RELATED TITLES

NCHRP REPORT 590

Multi-Objective Optimization for Bridge Management Systems

Vandana Patidar

Samuel Labi

Kumares C. Sinha
School of Civil Engineering
Purdue University
West Lafayette, IN

Paul Thompson
Castle Rock, CO

Subject Areas
Bridges, Other Structures, and Hydraulics and Hydrology

TRANSPORTATION RESEARCH BOARD

WASHINGTON, D.C.
2007
www.TRB.org

NATIONAL COOPERATIVE HIGHWAY RESEARCH PROGRAM

Systematic, well-designed research provides the most effective approach to the solution of many problems facing highway administrators and engineers. Often, highway problems are of local interest and can best be studied by highway departments individually or in cooperation with their state universities and others. However, the accelerating growth of highway transportation develops increasingly complex problems of wide interest to highway authorities. These problems are best studied through a coordinated program of cooperative research.

In recognition of these needs, the highway administrators of the American Association of State Highway and Transportation Officials initiated in 1962 an objective national highway research program employing modern scientific techniques. This program is supported on a continuing basis by funds from participating member states of the Association and it receives the full cooperation and support of the Federal Highway Administration, United States Department of Transportation.

The Transportation Research Board of the National Academies was requested by the Association to administer the research program because of the Board's recognized objectivity and understanding of modern research practices. The Board is uniquely suited for this purpose as it maintains an extensive committee structure from which authorities on any highway transportation subject may be drawn; it possesses avenues of communications and cooperation with federal, state and local governmental agencies, universities, and industry; its relationship to the National Research Council is an insurance of objectivity; it maintains a full-time research correlation staff of specialists in highway transportation matters to bring the findings of research directly to those who are in a position to use them.

The program is developed on the basis of research needs identified by chief administrators of the highway and transportation departments and by committees of AASHTO. Each year, specific areas of research needs to be included in the program are proposed to the National Research Council and the Board by the American Association of State Highway and Transportation Officials. Research projects to fulfill these needs are defined by the Board, and qualified research agencies are selected from those that have submitted proposals. Administration and surveillance of research contracts are the responsibilities of the National Research Council and the Transportation Research Board.

The needs for highway research are many, and the National Cooperative Highway Research Program can make significant contributions to the solution of highway transportation problems of mutual concern to many responsible groups. The program, however, is intended to complement rather than to substitute for or duplicate other highway research programs.

THE NATIONAL ACADEMIES
Advisers to the Nation on Science, Engineering, and Medicine

The **National Academy of Sciences** is a private, nonprofit, self-perpetuating society of distinguished scholars engaged in scientific and engineering research, dedicated to the furtherance of science and technology and to their use for the general welfare. On the authority of the charter granted to it by the Congress in 1863, the Academy has a mandate that requires it to advise the federal government on scientific and technical matters. Dr. Ralph J. Cicerone is president of the National Academy of Sciences.

The **National Academy of Engineering** was established in 1964, under the charter of the National Academy of Sciences, as a parallel organization of outstanding engineers. It is autonomous in its administration and in the selection of its members, sharing with the National Academy of Sciences the responsibility for advising the federal government. The National Academy of Engineering also sponsors engineering programs aimed at meeting national needs, encourages education and research, and recognizes the superior achievements of engineers. Dr. Charles M. Vest is president of the National Academy of Engineering.

The **Institute of Medicine** was established in 1970 by the National Academy of Sciences to secure the services of eminent members of appropriate professions in the examination of policy matters pertaining to the health of the public. The Institute acts under the responsibility given to the National Academy of Sciences by its congressional charter to be an adviser to the federal government and, on its own initiative, to identify issues of medical care, research, and education. Dr. Harvey V. Fineberg is president of the Institute of Medicine.

The **National Research Council** was organized by the National Academy of Sciences in 1916 to associate the broad community of science and technology with the Academy's purposes of furthering knowledge and advising the federal government. Functioning in accordance with general policies determined by the Academy, the Council has become the principal operating agency of both the National Academy of Sciences and the National Academy of Engineering in providing services to the government, the public, and the scientific and engineering communities. The Council is administered jointly by both the Academies and the Institute of Medicine. Dr. Ralph J. Cicerone and Dr. Charles M. Vest are chair and vice chair, respectively, of the National Research Council.

The **Transportation Research Board** is a division of the National Research Council, which serves the National Academy of Sciences and the National Academy of Engineering. The Board's mission is to promote innovation and progress in transportation through research. In an objective and interdisciplinary setting, the Board facilitates the sharing of information on transportation practice and policy by researchers and practitioners; stimulates research and offers research management services that promote technical excellence; provides expert advice on transportation policy and programs; and disseminates research results broadly and encourages their implementation. The Board's varied activities annually engage more than 5,000 engineers, scientists, and other transportation researchers and practitioners from the public and private sectors and academia, all of whom contribute their expertise in the public interest. The program is supported by state transportation departments, federal agencies including the component administrations of the U.S. Department of Transportation, and other organizations and individuals interested in the development of transportation. **www.TRB.org**

www.national-academies.org

COOPERATIVE RESEARCH PROGRAMS

FOREWORD

By David B. Beal
   Staff Officer
   Transportation Research Board

This report describes the development of methodologies for network- and project-level optimization of multiple, user-specified performance criteria. Bridge management software modules to implement the methodologies were also developed. The report details the development of methodologies. The software modules, user's manual, and demonstration database are provided on an accompanying CD-ROM. The material in this report will be of immediate interest to bridge managers and planners.

Currently available bridge management system (BMS) tools compute an optimal solution based on the objective of least long-term cost. Bridge managers are finding that their constituents require bridge conditions to be substantially better than a least long-term cost solution would provide. Research was needed to develop a multi-objective optimization model.

To address this need, two distinct BMS optimization models were developed: a network-level model and a bridge-level model. The network-level model provides a decision-making tool that optimizes bridge actions for multiple performance criteria. These performance criteria could be cost, condition, risk, highway bridge replacement and rehabilitation (HBRR) program eligibility, bridge health index, or others. The bridge-level model evaluates the effect of bridge action alternatives on life-cycle cost and other performance criteria for the purpose of selecting projects that are consistent with the network goals.

Both models use the AASHTO BridgeWare database supplemented with additional data as needed. Commonly Recognized (CoRe) Element data are used for condition assessments. The bridge-level model considers recommendations from the network-level model. In addition, the network-level model can consider projects selected within the bridge-level model. These models also can operate independently. Both models explicitly consider the inherent uncertainties of estimated costs and outcomes. The models are implemented in graphical design software that will help bridge managers visualize the life cycle of individual bridges and bridge inventories.

This research was performed by Purdue University, in West Lafayette, Indiana, and Paul D. Thompson, Consultant. The report fully documents the research leading to the recommended models.

CONTENTS

S U M M A R Y

Multi-Objective Optimization for Bridge Management Systems

Key functions of bridge management include the establishment of optimal investment funding levels and performance goals for an inventory of bridges, as well as identification of the appropriate combinations of treatment scope and timing for each individual bridge over its life cycle. This report is expected to aid bridge management decision making and thus to enhance the cost-effectiveness of agency spending. Past experience suggests that bridge investment decisions made only on the basis of lowest cost yield unsatisfactory results. Therefore, bridge agencies have expressed a need to enhance current decision-making methodologies to include other performance criteria, such as bridge condition, safety, traffic flow disruption, and vulnerability. That way, more balanced, rational, defensible, and cost-effective decisions can be made, and better investigation of trade-offs between performance criteria can be carried out.

In responding to these critical bridge management issues, the present study developed network- and bridge-level methodologies that involve multiple performance criteria and also involve selection of investment choices based on optimization. An accompanying software package was developed to facilitate implementation of the methodologies. The concepts of value and utility theory were duly incorporated to account for inherent risk and uncertainty associated with the decision parameters.

The first task accomplished in the study was to develop a basis upon which the alternative bridge actions could be evaluated. This was done by establishing a comprehensive, yet minimal and operational, set of goals and performance criteria as follows:

- **Preservation of bridge condition:** National Bridge Inventory (NBI) condition ratings, health index, and sufficiency rating.
- **Traffic safety enhancement:** Geometric and inventory/operating rating.
- **Protection from extreme events:** Vulnerability ratings for scour, fatigue/fracture, earthquake, collision, overload, and other human-made hazards.
- **Agency cost minimization:** Initial cost, life-cycle agency cost.
- **User cost minimization:** Life-cycle user cost.

For decision-making problems that involve several performance criteria, there exists a class of solution techniques that involve the construction of a preference order (of the performance measures) from the perspectives of the decision maker. This class of solution techniques is predicated on utility theory, which is based on the premise that the decision maker's preference structure can be represented by a utility function. Construction of the utility function generally involved three major steps:

- **Weighting:** This assigned relative weights to each of the multiple criteria on the basis of their relative importance.

- **Scaling:** This provided a common scale of measurement for performance criteria that have different units. This involved developing a single-criterion utility function for each performance criterion.
- **Amalgamation:** This combined the single-criterion utility functions using the relative weights into a single overall performance measure. The structure of the overall objective function was influenced by the mathematical assumptions made with regard to the decision maker's preference structure. These assumptions were tested using data obtained from the survey of bridge experts (that is, the NCHRP 12-67 panel). The key step in the amalgamation process, therefore, was the identification of the appropriate functional form for combining the individual utility functions of the various performance criteria into a single quintessential utility value.

In this approach, the underlying assumptions and the appropriate assessment methods depend on whether the decision-making problem is one of certainty (where the consequence of each alternative, in terms of multiple criteria, is known with exactitude) or of uncertainty (a.k.a. "risk," where the exact consequences of each of the alternatives are not known). For the risk and certainty scenarios, the task of scaling involved the development of utility and value functions, respectively.

After the tasks of weighting, scaling, and amalgamation had been carried out to yield the objective function (a single utility function that represented all the utility functions or values of the multiple performance criteria), the next step was optimization, where the researchers sought to identify, from a number of bridge investment alternatives, that which yields the maximum value of the objective function. The study developed default parameters for the value and utility functions for the various performance criteria (and thus, for the objective function) using data obtained from a questionnaire administered to the NCHRP 12-67 panel.

The study approach described above was applied to both bridge level and network level. The developed network-level model helps the bridge engineer to select the optimal mix of candidate projects (from a networkwide candidate list) that yields the maximum value of the objective function subject to multiple constraints. The optimization problem was formulated as a multi-choice, multi-dimensional knapsack problem (MCMDKP). After a careful investigation of alternative heuristic and exact solution approaches for solving the MCMDKP, the study determined that the incremental utility-cost (IUC) ratio, Lagrangian, and pivot and complement approaches were satisfactory. The study then proceeded to carry out further investigation of these heuristics through a series of computational experiments, using real field data, for their suitability in realistic inventories typically encountered in statewide bridge management. The tests were carried out on networks of sizes 100; 1,000; 9,265; 12,000; and 50,000; and the datasets were from the Florida DOT's Project Level Analysis Tool, which contains inventory, deterioration, and cost data files. The heuristics were evaluated on the basis of the following theoretical precision and practicality criteria: computational speed, accuracy, simplicity, and robustness. Accuracy was based on comparing heuristic solutions to true optimal solutions derived using CPLEX–Concert technology, a state-of-the-art commercial optimization software package.

Overall, the computational experiments provided valuable information regarding the appropriateness of the heuristics for the decision-making (i.e., optimization) problem for bridge management. For example, for the 12,000-bridge network (the average network size at state agencies), an average computational time of 666 seconds (11 minutes) and an average accuracy of 99.62% were obtained using the IUC heuristic. This was superior to the performance of the two other promising heuristics. The experiments were repeated for various network sizes and constraint formulations and yielded results that were unequivocal: of the

three heuristics, IUC consistently turned out to be the most superior in terms of computational speed, accuracy, robustness, and simplicity. The IUC heuristic also provided the quickest computation of the optimal solution in the case of small changes in the input parameters without having to redo the entire optimization for each budget level. The results also suggest that IUC is the most robust of all the heuristics. The study also determined that the underlying algorithm of the IUC heuristic is relatively easy to comprehend and has a close conceptual interpretation to that of classic incremental benefit-cost (IBC) heuristic (and thus is more familiar and acceptable to the bridge management community). On the basis of these findings, it was concluded that the IUC heuristic is most suited for addressing the bridge optimization and decision-making problem under consideration. As such, this heuristic was selected for incorporation in the application tool (software product) that was developed as part of this research.

The bridge-level methodology includes a life-cycle cost framework, preservation and functionality models, candidate definitions, and their evaluation. The methodology also includes optimization at the bridge level in a bid to maximize utility of bridge actions in the long term by selecting from an array of scoping and timing alternatives. The methodology also includes a recursive approach that is consistent with input data available from existing bridge management systems.

For each bridge in the inventory, the bridge-level model evaluates three general scoping approaches: do-nothing; maintenance, repair, rehabilitation, and improvement (MRR&I); and total replacement. The MRR&I approach is generated through an investigation of individual possible actions defined for each condition state of each bridge element—this approach is consistent with the Pontis methodology. Unlike Pontis, however, the model separates fixed and variable costs of treatments and duly considers actions whose life-cycle benefit exceeds their initial variable costs. This tends to produce more comprehensive interventions. The bridge-level model of the software produced in this study can therefore help bridge engineers and planners to review and fine-tune their selection of actions in custom candidates.

Another innovation in the bridge-level model is the classification of actions into a simpler typology suitable for describing the scope of work on a bridge that has many elements. For example, if a bridge has several elements constructed of concrete material (all of which, at a given time, need different types of maintenance and repair work), these elements can be grouped as maintenance, repair, and rehabilitation (MRR) concrete elements for manipulation by the engineer. This allows for more effective recognition of scale economies. More significantly, total system replacement actions, such as deck replacement and coating system replacement, provide a structure for more accurate estimation of costs and benefits than is currently possible in Pontis.

In the bridge-level model, functional deficiencies of bridge elements are evaluated using level-of-service standards as provided in Pontis, but have an updated user cost model. In the software product, the parameters and functional form of major inputs, such as truck height and weight histograms and accident risk models, can easily be customized by the agency to fit local conditions.

The optimization algorithm selected to provide decisions at the bridge level is a recursive search heuristic that allows for several interventions over a period of up to 30 years. The first intervention is the subject of decision making, while subsequent interventions are considered to be consequences of the first. Customized first interventions input by the engineer are evaluated within the same utility function framework as are automatically generated first and consequent interventions. This provides feedback to the engineer on the forecast consequences of any envisioned maintenance approach.

The study implemented the new analytical framework in the form of a software package called Multi-Objective Optimization System (MOOS). MOOS can be deployed immediately

as an add-on to Pontis by accessing the Pontis database, or can be integrated more closely through future AASHTOWare work. MOOS employs a digital dashboard display for analysis at the network and bridge levels, thereby presenting a greater view of analysis inputs and outcomes and offering more engineer control than the current Pontis software does. Developed as a set of Microsoft Excel workbook files, MOOS fully uses the presentation, graphics, and customization facilities that are available in Excel. There are two modules in MOOS: network level and bridge level.

The study developed a use-case model to describe how the research product will be useful in the duties of key bridge management personnel. This provided the necessary framework for defining requirements for the analytical techniques and software tools. Also, the use-case model will serve as an implementation framework for agencies wanting to put the software tool to work for improving the decision-making processes in their bridge programs.

CHAPTER 1

Decision-Making Context

1.1 Introduction

The software module resulting from the present research is intended to be used by various management personnel at transportation agencies to support bridge project scoping and timing and asset management decisions involving prioritizing, scheduling, and budgeting of bridge work. The software is divided into two primary models, network level and bridge level, which are intended to address two distinct but often overlapping levels of management functions. In subsequent sections of this chapter, these management functions are described in the form of *use-case models.* A use-case is a specific class of activity, performed by a human decision maker with assistance from the intended software module. In this report, the use-cases are documented in a generic manner so that they can account for the variability in management styles and structures that exist across the entire market spectrum of bridge owners and other intended users of the research product.

One benefit of a use-case analysis is that essential assumptions about procedures, terminology, and preferences are made explicit so they can be reviewed and refined. This promotes clear understanding and well-designed functionality that fits the needs of intended users.

Another benefit is the ability of use-cases to anchor much of the analytical and software design and development. In this report, major requirements (including inputs and outputs) of the analytical models and software are justified against the background of the identified use-cases. Use-cases generate information that helps set priorities for project and product resources such as user interface, algorithm execution time, and software development time. For example, in evaluating alternative analytical methods for bridge- or network-level optimization, the ability to efficiently provide the exact information (neither more nor less) identified in the use-case analysis is a major consideration. Ultimately, use-case analysis helps provide the outline and structure for the user interface, reports, documentation, and test plan for the optimization software module.

Use-case analysis as a design methodology is described in numerous standard texts, including Jacobson et al. (1995) and Schneider and Winters (2001). The information in this report uses the graphical conventions of the Unified Modeling Language (Booch et al. 1998; Fowler 2000) that have been tailored as needed to clearly communicate the most important decisions associated with bridge management optimization. In the context of decision making, two perspectives are elaborated: (1) characterization of users and (2) characterization of decision-making activities to be addressed by the research product. The results of the use-case analysis were developed into a set of functional requirements for the bridge management system optimization software modules. So the problem-oriented perspective in this chapter leads to the solution-oriented perspective in the final product.

1.2 Characterization of Users

The software module is intended to serve a planning process for maintenance and improvement of existing bridges. This is an important statement for defining the population of intended users. The subsequent sections describe how the research product will be useful in the duties of key bridge management personnel, including the bridge maintenance planner and the bridge program manager.

1.2.1 Bridge Maintenance Planner

Every transportation agency is likely to have staff responsible for defining the scope, timing, programmatic cost, and justification of future work to be performed on each existing bridge, applying a specialized level of training and responsibility. We define the conceptual role of bridge maintenance planner to encompass this set of responsibilities.

Many types of people may perform this job function: inspectors and designers who plan maintenance work; consultants who perform this function under contract; or staff with titles

such as bridge management engineer, project engineer, or bridge maintenance engineer responsible for bridge maintenance planning. In the present report, the title bridge maintenance planner is used to include persons in any such roles, and use-cases have been defined to fit the range of responsibilities and capabilities associated with such roles.

The bridge maintenance planner is usually trained and registered as a bridge engineer, though in local governments it is possible that the person may have some other type of engineering training. It is reasonable to expect the person to be familiar with commonly accepted concepts, terminology, and jargon used in bridge inspection, bridge maintenance, and engineering economics.

1.2.2 Bridge Program Manager

For network-level analysis, we define the conceptual role of *program manager.* The program manager may not necessarily be a senior-level staff person with management responsibility, but may be an analyst or staff member who gathers information and possibly makes decisions on behalf of a manager. Such decisions may touch on several domains: priority setting, capital budgeting, operational resource budgeting, programming, or liaison with elected officials involved in budgeting. Some transportation agencies may not have a staff person with the specific job title of "bridge program manager." However, certain staff may have the stated duties of a program manager as part of other job responsibilities. Such personnel may have titles such as program planner, bridge management engineer, maintenance engineer, maintenance director or manager, chief bridge engineer, bridge maintenance engineer, district engineer, city or county engineer, or budget analyst.

It is important to note that the role of program manager is often not dedicated solely to bridges. Rather, the network-level planning of bridge preservation and improvements in many transportation agencies is carried out in conjunction with planning for other assets, such as pavements or traffic infrastructure. This is especially true in the district offices of decentralized highway agencies and in local government agencies.

Furthermore, it is important to recognize that program managers may have relatively little or no background in engineering. The job of a program manager is highly interdisciplinary in nature, requiring an understanding of economics, planning, computing, basic systems analysis, and more than a passing familiarity with engineering and policy analysis. Often, the person playing this role serves as an interface between the political world and the engineering world and is placed at the center of an agencywide asset management process.

A discussion of the needs of program management of facilities other than bridges is beyond the scope of the present report. However, for purposes of facilitating acceptance of the research product, it is necessary to design the network optimization model to accommodate (at a later time) the management of any type of asset, not just bridges, so it can be made relevant to program managers who develop programs for other highway facilities as well. Specifically, the optimization model will (1) take due cognizance of multiple objectives that are also considered in other infrastructure management systems and (2) use a conceptual methodology that can later be expanded to include decision making regarding pavements and other facilities.

Recent work in the area of asset management provides a background (Cambridge Systematics et al. 2002) as well as methodologies (Li and Sinha 2004) that have similarities to the present study. Also, some recent research provides concepts of asset performance measurement (Poister 1997; OECD 2001; Hyman 2004) that are well suited to abstracting engineering issues into a form that can easily be understood and used by non-engineers. We assume therefore that the program manager is not necessarily an engineer, but is familiar with transportation performance measures and their application to planning and budgeting.

It is important, as part of the use-case analysis, to note that both the bridge maintenance planner and the program manager are assumed to be computer literate and able to use standard office software tools, such as word processors and spreadsheets, as well as more specialized software already developed for their job functions, such as accounting systems and load rating software. They are assumed to have at their desks a computer capable of running these types of software effectively and to have in-house information technology support to help them solve common computer problems and gain access to the requisite enterprise databases. An agency may have multiple bridge maintenance planners and program managers whose responsibilities typically are divided by geographical jurisdiction, by functional class, or by facility ownership. It is common in many agencies for one person to play both roles. At most agencies, however, the number of bridge maintenance planners generally exceeds that of program managers.

1.3 Bridge-Level Use-Cases

Bridge maintenance planning is a process of deciding the scope, timing, costs, and benefits of future work on a specific bridge. Its purpose is to start a process of obtaining the necessary funding, permissions, and resources so that the work can proceed. In the current state of the practice, most bridge management systems are very effective at storage and retrieval of all the raw data needed for maintenance planning. However, they are not as effective at providing the bridge maintenance planner with a comprehensive picture of the current economic health of a bridge and its possible futures. For example, it is not possible in most current bridge management systems to see a

side-by-side performance comparison of alternative candidates, to plot the effect of intervention timing (such as the identification of what timing yields the best performance or lowest life-cycle cost), to compare plots of forecast performance over time for alternative candidates, or to see the marginal cost and/or benefit of possible changes to the scope of an intervention.

The left side of Figure 1 breaks down the business process of bridge maintenance planning into its components. The planner gathers a set of relevant information, manipulates it, and records any decision made from the data analysis. The decision maker typically seeks information about the effect of current decisions on future performance of the bridge. Much of the data manipulation is concerned with predictive models intended to provide this feedback. The decision maker defines a project, forecasts the future outcome, and then adjusts the project and re-estimates the impact. When the result is as good as possible, or at least good enough, the process stops and the result are recorded. The "result" is the identification of one or more candidate projects for the bridge, for promotion into the programming and budgeting process.

Terminology note: In this report we generally avoid the use of the word "project," except when used as a generic term for a combination of work items. This is because the word is applied inconsistently in common use and has administrative meanings that differ from one agency to another. The static-view class diagram presented later in this chapter gives some precise definitions of certain terms that we rely upon in the design of the system.

1.3.1 Prepare Background Inputs

In the sequence of bridge-level analysis, a logical first step is to gather relevant data on the bridge, policies affecting the bridge, costs, and typical patterns of deterioration or performance. Housekeeping tasks such as data gathering are not central to the job, so bridge maintenance planners typically seek to minimize or delegate this step whenever possible. It is expected that an automated decision-support system will take care of this process with minimal effort from the planner.
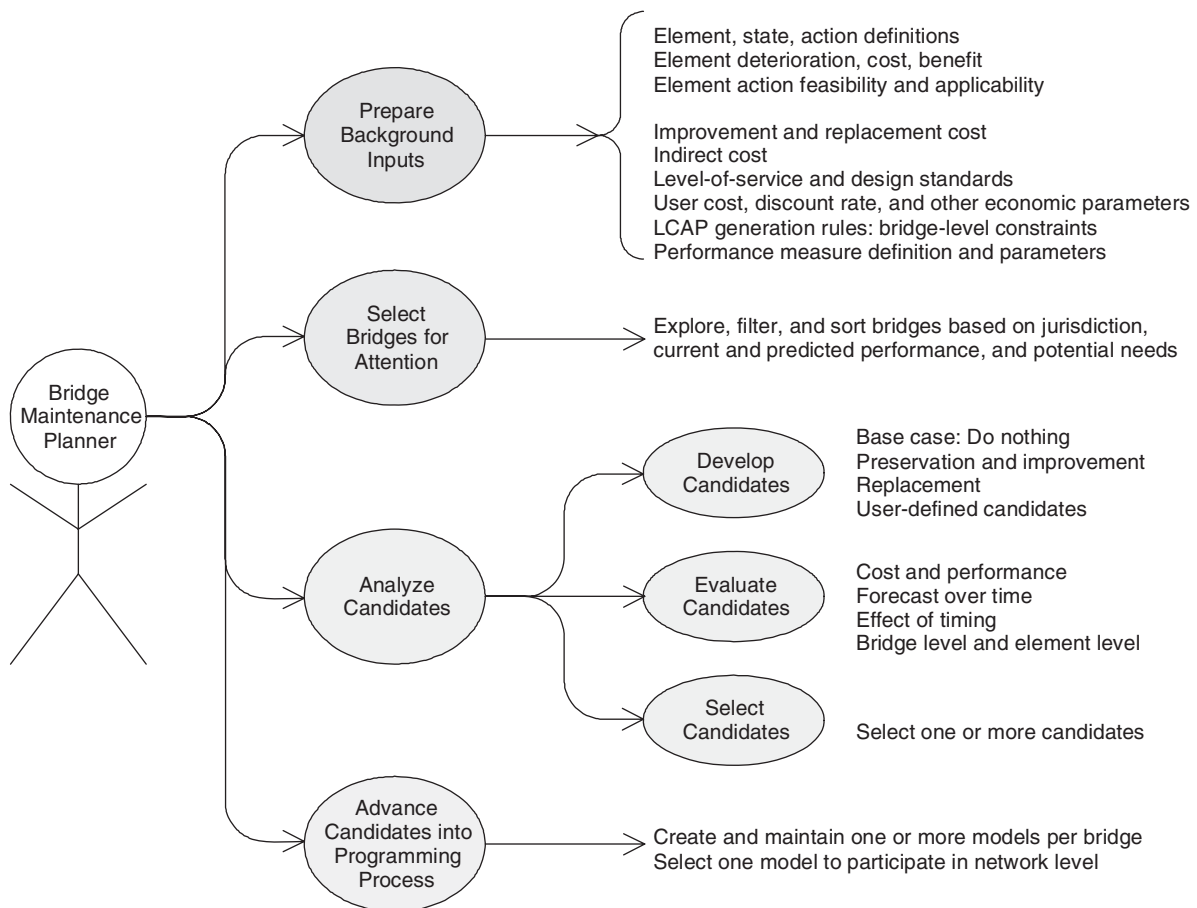
LCAP = life-cycle activity profile.

Figure 1. Use-cases at bridge level.

The right side of Figure 1 shows a list of needed data items for supporting the desired predictive capability. Each agency has its own preferences for economic analysis and performance measures, so each will have unique data requirements. The diagram shows the types of data that are most commonly needed. Most of these can be found in existing bridge management systems of most state highway agencies (AASHTO 1992).

1.3.2 Select Bridges for Attention

It is fairly common for a bridge maintenance planner to have responsibility for 500 to 2,000 bridge structures. In any given year, only a small fraction—perhaps less than 20%—of these structures typically need any planning attention. Less than 10% of the inventory actually receives work in any given year. However, approximately 50% of all bridges are inspected in a given year, and the new information always has at least the potential to generate a need for planning attention.

It is generally not possible for the planner to re-analyze every bridge every year. Therefore, it is necessary to have a means to monitor and analyze changes in bridge condition and to bring such information to the planner's attention. This process may be termed "screening." At well-defined, regular points in time—for example, at the start of the annual programming cycle—the planner would like to have an updated view of planning activities that need to be accomplished, preferably in a prioritized order of urgency. The inspector has the first line of responsibility to flag critical safety issues, but there is also a need to flag life-cycle economic issues such as preventive maintenance opportunities.

The concept of economic urgency is different from the concept of priority, but derives from it. Economic priority is usually an indication of life-cycle benefit as a multiple of initial cost. Urgency, on the other hand, is a measure of the loss in life-cycle cost or performance benefit that is predicted if action is postponed for one decision cycle. Therefore, the concept of economic priority addresses the long-term question of how best to use a fixed set of resources to greatest benefit, while economic urgency addresses the short-term need to seize the best opportunities. Both concepts are important for screening, but economic urgency is handled in a special way because it has a near-term effect on the personal workflow of the bridge maintenance planner.

As a part of the screening process, the planner will explore a list of bridges in order to make comparisons. During a given planning exercise, the planner may focus on a specific program goal and may seek the candidate projects to address that goal. At different times, the focus may be on different subsets of the bridge inventory. This implies a need for a convenient exploratory tool for sorting and filtering bridges based on current and predicted performance and predicted maintenance needs. The user should be able to analyze each bridge at any time between the arrival of new inspection data and the making of planning workflow decisions, so any recommendations can be on the basis of the most current data.

1.3.3 Analyze Candidates (Treatment Types and Timings)

If a bridge is identified as having a deficiency or preventive maintenance opportunity, the next step for the bridge maintenance planner is to investigate possible candidate treatments. There may be more than one possible scope approach, depending on available maintenance capabilities, life-cycle considerations, and competing needs. Based on accepted warrants, decision rules, and standard operating procedures, certain approaches will obviously deserve consideration: for example, replacing the structure, performing all needed maintenance and improvement activities, or doing nothing (i.e., waiting until more urgent needs are met). Often, these obvious treatments are easily eliminated from further consideration, as in the case of replacement of a relatively new bridge. At other times, there are needs that are not obvious from the data but result from the planner's personal knowledge of the bridge structure.

Aside from questions of scope, there are also questions of timing. Scheduling of work may often be based on economic priority, but could also depend on coordination with other projects (not necessarily bridge projects), traffic flow considerations, resource availability, project readiness, and political concerns.

An important observation about scope and timing decisions is that there is not an infinite number of possibilities. Typically, only one or two scoping approaches at any point in time will make practical sense. The construction season, budgeting cycle, and letting cycle usually dictate that there are only five timing possibilities in a 5-year program horizon. As the program horizon becomes longer, even annual distinctions in timing may have less and less value: for example, there may be no practical reason to distinguish between an alternative that calls for intervention at year 8 and another that calls for a similar intervention in year 9. Decision makers are relatively insensitive to changes in timing for events far in the future.

To express and manage this spectrum of possible futures, we define the concept of a candidate. A candidate is hereby defined as a life-cycle activity profile for one bridge, consisting of a sequence of agency activities—including do-nothing—in each of a sequence of future time periods. Development of alternative candidates and selection of the best one is a cardinal aspect of decision making by the bridge maintenance planner. The planner decides which of the alternative candidates are worthy of consideration and, over time, narrows the list to just one or a small number that the planner then submits to the next level of bridge management (i.e., programming and budgeting at the network level).

As a general proposition, we assume that the first do-something agency action in a life-cycle activity profile is the most important to the decision maker. There is a rest period, usually a matter of policy (e.g., 10 years), after the first action when no further actions will be planned. Subsequent actions may appear in a published program but are considered outcomes of the decision process and are not the focus of decision making. So, for example, the bridge maintenance planner may spend time investigating the work to be done in year 2 of a program on a given bridge, but may not want to devote time to deciding what subsequently needs to be done in year 12. With automated decision support, it is preferred that predictive models be used to generate a year 12 action in an automated way, sensitive to the choices made by the planner in year 2.

Even for the first action in a life-cycle activity profile, the decision maker does not assign equal importance to each program year but rather cares most about the first program year, somewhat less about the second year, and so forth. The bridge maintenance planner is typically willing to devote relatively more resources—time, data collection, screen layout space, computer paper, and so on—to the early years of a program and very little to the later years of a program.

When a bridge first attracts the planner's attention because of a change in condition, the priority of the need is usually low and the timeframe relatively far in the future. From year to year, the priority of these needs will increase because of deterioration and traffic growth, making the bridge increasingly deserving of the planner's attention. For a bridge that is 2–5 years away from imminent work, there may be several candidate alternatives that the planner may want to develop. Furthermore, the planner typically seeks to generate and save the reports that show the details as well as impacts of each alternative candidate. As the year of intervention draws nearer, the planner may seek to further refine and eliminate relatively inferior candidates until only one alternative is left.

From this discussion, we model a continuous process of candidate development for each bridge. It starts with small needs far in the future that are generated by automated means and receive little attention. These are evaluated entirely based on economic and other considerations that can be quantified with data from the bridge management system. As time goes by and a bridge moves closer to implementation, the probability of manual candidate definition and decision making increases, and multiple candidates may be generated manually. Their justification may start to include factors outside of the bridge management system, such as project interrelationships and politics. The planner will want the decision-support tool to evaluate the economic and quantitative consequences of these candidates to help with defining them and may still want to consider reference candidates generated by an automated process. Finally, when a bridge is very close to implementation, the process is almost entirely manual and the number of

candidates is reduced. Economic consequences become relatively less important compared with exogenous factors as the agency increases its public commitment to the work.

1.3.4 Advance Candidates into Programming Process

As a part of the process of reducing the number of candidates, there is an important interface between the bridge maintenance planning process and the program management process. Part of what eliminates candidates from consideration is the competition among all bridges for limited resources. Typically, the bridge maintenance planner, from personal experience, may have an insight into what types of work are likely to be funded, but does not have any definite limits because the candidates are not yet well defined and future budgets are uncertain.

When a bridge planner feels intuitively that a specific bridge needs work soon, but is doubtful about its funding, that planner may want to tailor the scope and timing of the work to maximize its competitiveness. The planner needs a rough sense—perhaps a probabilistic sense—of the funding and competition, how the bridge in question competes with other bridges, and how likely the funding may be. The operative question is what combination of scope and timing maximizes the probability of the bridge being funded. This does not suggest the need for an automated optimization for maximization of funding probability, because the inputs are too uncertain. However, it does suggest that the scoping and timing decision participates in the larger program management optimization because human decisions in the two frameworks affect each other.

1.4 Candidates as an Interface Between Bridge and Network Levels

For each bridge, the bridge maintenance planner will be expected to provide to the program manager at least one candidate, specifying the choice of scope and timing of the first intervention in a life-cycle activity profile. This should result from an evaluation by the planner of all relevant engineering tradeoffs and the selection of a recommended strategy for the bridge. Typically, the program manager does not modify the scope of the selection, but merely regulates the timing. Changing of the timing recommended for each bridge to satisfy network optimization goals may compromise the optimality of the bridge-level model because the optimal scope of work in year 1, for example, might be quite different from the optimal scope if work is delayed to year 6. When a bridge has a large quantity of needs, it may be unsatisfactory to limit consideration to a binary choice between doing everything and doing nothing; an in-between alternative could be wise in a funding-constrained

environment. From our observation of best practice, the bridge maintenance planner, acting in a defensive posture, often responds to such possibility by offering alternatives that anticipate adjustments that may have to be made by the program manager.

So in the desired rational, optimized network-level process, each bridge has multiple alternative candidates to be manipulated by the program manager. Each candidate has a different scope and timing of its first intervention, but to the program manager the differences are expressed as differences in initial cost and in performance measures, such as life-cycle cost. The intrinsic importance of a bridge may affect its priority apart from performance measures, so bridge attributes such as traffic volume are also important at the network level.

1.5 Network-Level Objectives and Constraints

In the course of developing network-level bridge programs, program managers typically face a variety of objectives and constraints. Examples of objectives are to maximize cost-effectiveness, to minimize vulnerability to damage, to maximize average condition, and to optimize a utility index that combines various objectives. Constraints include a budgetary ceiling that cannot be exceeded or a minimum level of average bridge health. However, most objectives and constraints are not as clear-cut as these. Indeed, any given performance measure could exist in either the objective function or the constraint, or both, depending on what the program manager seeks. For example, an agency might set a target level of bridge deck condition as a commitment to the public. In any given timeframe, it is uncertain whether this objective is achievable. Nevertheless, there is an objective to maximize deck condition, even if the target levels are not achieved.

The overlap between objectives and constraints is key to practical multi-objective optimization of an asset management program. The quantitative level of constraints is uncertain, and the achievability of a combined set of constraints is also uncertain. In the face of uncertainty, it is necessary to set priorities: certain constraints absolutely must be met, while others may be met if possible or may be relaxed if necessary. In fact, there is a continuous scale of constraint rigidity that may be difficult to quantify for purposes of optimization.

For example, it may be a policy decision to target an average health index of 95 and an average accident rate of 50 crashes per million daily vehicles, within a budget of $100 million per year over a 5-year period. In a scenario where such multiple constraints are not achievable, it may be a policy decision to postpone or modify the average health index target so that the constraints are likely to be satisfied. In this example, the characteristic feature of the program is an implicit trade-off between safety and condition as a part of establishing performance targets. Both the accident rate and the health index are actually in the objective function, and not in the constraint set, of this implicit mathematical program. A decision is made about the relative weighting of the two objectives, and then, once a satisfying ("good enough") solution is found, the performance measure targets that are found to be achievable are expressed as constraints.

The most important performance measures are moved to the constraint set first and then followed by less important criteria. For example, if a program has condition targets for deck elements and for substructure elements, usually the deck condition targets are more important, and the program manager wants to be certain that the presence of substructure targets cannot unduly change the shape of the overall program or compromise the ability to meet the deck targets.

Prudent program managers typically leave some slack in the system and render some performance measures unbounded to provide room for uncertainty and programmatic risk. One example of this is the common practice of *over-programming,* where a budget constraint is set at an artificially high level, often by 30% or more, to ensure that the agency is ready to respond to unanticipated changes in either the funding level or the readiness of projects. Over-programming is simply a tactic of using professional judgment to manage risk. In a practical multi-objective optimization framework, this tactic or some other risk management strategy may be necessary.

One implication of this flexibility in the definition of objectives is that the network-level model may use different performance objectives from the bridge-level model. In fact, an agency might have different performance criteria for different bridges or groups of bridges; for example, bridges on lifeline routes would have more emphasis on risk. This is considered an important and realistic feature of the decision-making context.

Summing up, it is recognized that for a program management decision-support tool, the analytical methods must provide ample opportunity for the program manager to interact with, and participate in, the optimization. In particular, the manager must be allowed to take part in the movement of performance measures from objectives to constraints and must be helped to understand the amount of slack in the solution space and to gauge the ability to allow for programmatic risk. This flexibility is more important than exactness in a mathematical solution method. Program managers who lack mathematical inclination may prefer that such analyses be carried out in the form of user-friendly graphical interfaces with all detailed mathematics relegated to the background.

1.6 Use-Cases at Network Level

Program management is a process of reconciling competing objectives of resource use and performance by selecting and

scheduling actions. For most purposes, program management is understood as a process of making choices of project scope and timing across an entire asset inventory or subset. However, for more senior managers and most elected officials, it is more often understood as a sort of economic supply curve—a representation of how much performance can be purchased at various levels of investment. Program managers typically lack adequate time resources or expertise to evaluate engineering trade-offs, but this does not mean such trade-offs are unimportant. It merely means that the trade-offs should al-

ready have been considered by maintenance planners and the results communicated to program managers in an efficient and consistent manner.

Program managers typically are responsible for their decisions and choose what criteria they will consider. Their decisions are more often governed by economic than engineering considerations. Figure 2 shows a use-case diagram that reflects this philosophy. A program manager builds one or more programs, each of which has a set of objectives and constraints. Programs may represent different subsets of the inventory,
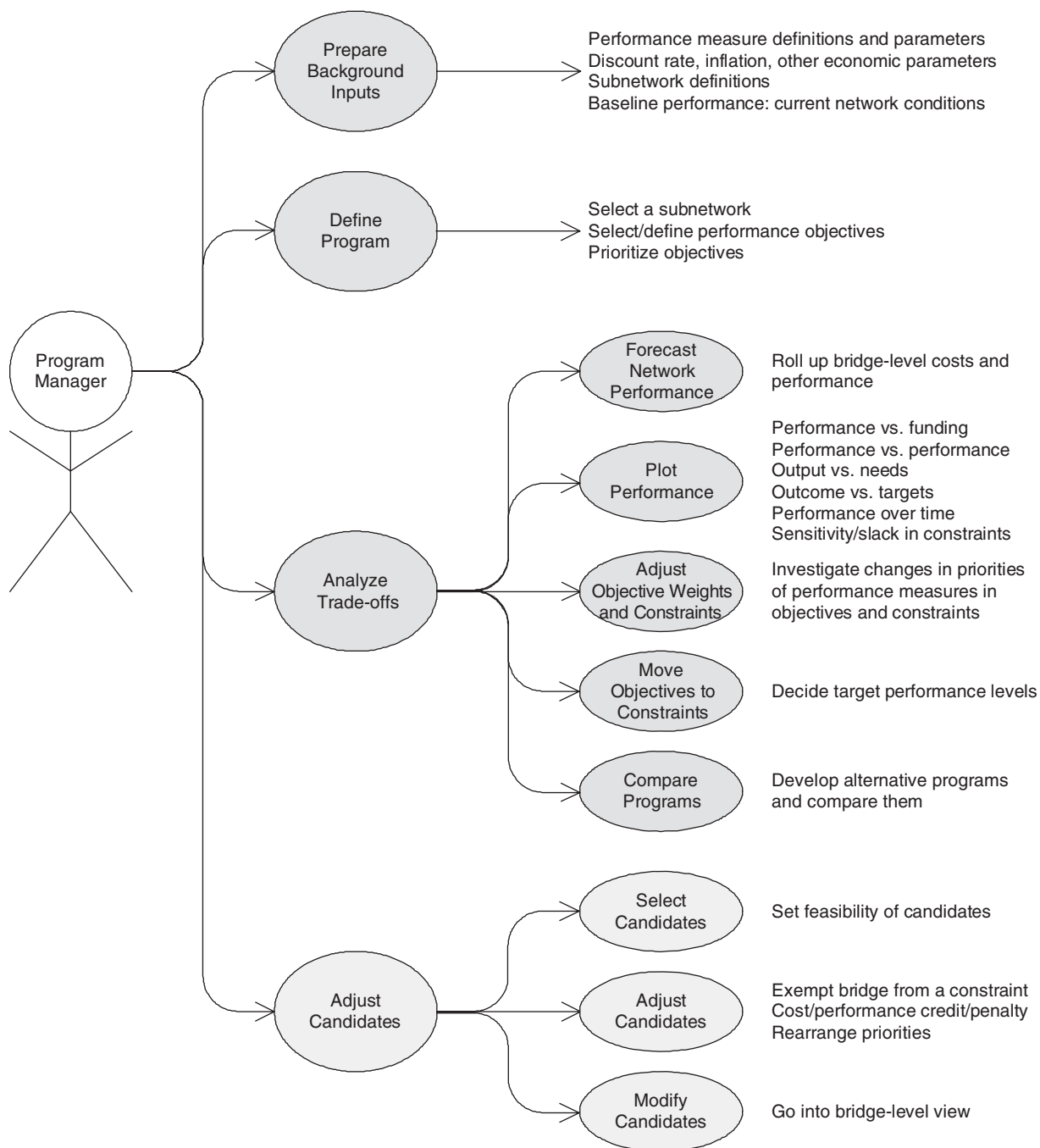
Figure 2. Use-cases at network level.

may focus on different goals, or may merely be experimental alternatives to each other. A program may be built over several days or over an entire year, so it is necessary to keep track of the status of its development by saving and retrieving any partially or fully developed programs for subsequent review and possible modification.

1.6.1 Prepare Background Inputs

In the course of their duties, program managers typically collect some background information that provides policy and economic context of bridge preservation actions. The collection of this information is assumed to be done exogenously and, as with maintenance planning, is not the focus of the effort. It should therefore be carried out with minimal intervention from the program manager.

1.6.2 Define Program

Development of a program begins with the selection of a subnetwork and identification of the performance measures of interest. The list of available candidates from the bridge-level model is evaluated and structured according to the performance trade-offs. It is worth devoting significant computational effort—which may be entirely automated—to this stage of the process because the results of such effort make all subsequent stages much easier for the manager. An automated process can arrange the candidates in priority order according to each separate performance criterion and according to a combined objective function and can build an analytical data structure that describes how the selection of candidates jointly affects all the performance measures of interest. The inputs needed for this process are the costs and performance measures already calculated at the bridge level.

1.6.3 Analyze Trade-offs

After the initial preparation, the program manager takes control and manipulates the objectives and constraints as described above. The program manager views a number of graphical presentations of trade-offs and sensitivity analysis to acquire an understanding of what goals are achievable with available inputs. Adjustments to such inputs yield immediate feedback on forecast outputs and outcomes. The right side of Figure 2 suggests the types of adjustments and feedback that would be expected.

All of the indicated presentations are envisioned as screen displays that can be printed as reports. Most of them are self-explanatory from the preceding discussion and diagram. A report of "Output vs. Needs" compares the work that is programmed in the model (respecting the applied constraints) against an unconstrained scenario where all cost-effective work

is done on every bridge. A report of "Outcome vs. Targets" compares the predicted performance of the program against desired performance targets.

1.6.4 Adjust Candidates

In addition to adjusting and viewing network-level performance measures, the program manager typically seeks to view and adjust individual candidates. The non-engineer can still perform useful work at this level if the maintenance planner has provided a good set of alternatives. All such adjustments involve selecting or deselecting candidates or making economic adjustments to reflect non-economic factors. For example, the manager might apply a penalty to a candidate that involves significant traffic disruption. A program manager who is also a bridge maintenance planner may start the analysis from the network level—investigating the options for the bridge from a network-level optimization standpoint and then proceeding to the bridge level to determine the appropriate courses of action. This way, it is possible to switch back and forth between bridge and network levels to fine-tune a program.

1.7 Characterizing the Problem Domain

Against the background of use-cases, it is possible to establish an outline of the required capabilities of the software module. The main issues are still problem oriented because they describe the aspects of the decision-support problem that the software is intended to model. Most important is a thorough understanding of how the objects manipulated by the software relate to the real-world problems of bridge management. The world of operations research offers a wide range of algorithms for multi-objective optimization. For bridge management, implementation of these algorithms requires a wide range of data requirements, time, and other resources. Each type of algorithm emphasizes different aspects of the problem, so naturally an appropriate choice of algorithm should focus on the problem attributes most important to bridge managers, thereby minimizing the consumption of resources.

Documentation and analysis of the problem domain takes the form of a domain model, which is a diagram and narrative that give a comprehensive view of the problem structure. The model has both dynamic and static components: the dynamic components describe the workflow of the real-world activities modeled by the software, while the static components show how concepts are permanently related to each other.

Effective use of any tool for bridge management optimization involves provision of step-by-step instructions on how to use the tool, including data preparation and interpretation of results and outputs. The dynamic view of the domain model, shown in the form of an activity diagram (see Figure 3),

Start

Bridge
Inspector

Conduct
Inspections

Previous
Bridge Model

Bridge
Maintenance
Planner

Previous
Program

Program
Manager

When a bridge
inspection is
completed, the
new data
become
available for
project
planning.

Select
Bridge

Generate
Standard
Candidates

The bridge-level model is
fully automated for
interventions far in the
future, but can be manual
for interventions in the
near term.

no   yes

Near-term
Intervention?

The optimal candidate is the life-cycle
activity profile that maximizes the
objective. Candidates are defined by
repeatedly manipulating the scope
and timing of the first intervention, and
then further interventions later in the
bridge's life are forecast analytically.

Adjust
Scope
Items

Adjust
Intervention
Timing

Selector
Rule Out
Candidates

Inspection, Bridge Maintenance
Planning, and Program
Management are three distinct
processes, conducted by different
people, that occur through out the
year. This diagram follows one
bridge through the process.

Evaluate and
Compare
Candidates

At some point in
maintenance planning, the
set of candidates defined
thus far on each bridge is
made available to the
Program Manager.

no   Satisfied?

yes

Programs are defined by repeatedly
manipulating individual bridges and
candidates, and by changing
objectives and constraints. It is often a
process of negotiation.

Select
Objects

Adjust
Candidates

Adjust
Objectives

Move
Objective to
Constraint

Adjust
Constraints

Next
Bridge

Next
Bridge Model

Evaluate
Results

An optimization model
maximizes the chosen
objectives while satisfying the
constraints, and reports the
resulting cost and
performance.

no   Satisfied?

yes

The process repeats year after year

Done

Program status changes constantly, but the analytical process keeps
the planning information up-to-date.

Figure 3. Activity diagram.

identifies the step-by-step workflows for the bridge management business processes. In order to provide support for the indicated workflows, the research product has been developed to use computer data structures and algorithms that are consistent with such workflows.

The static view of the problem domain, shown in the form of a class diagram (see Figure 4), is a map of the real-world objects and concepts for which computational effort is expended and data is manipulated. The class diagram divides the problem into small modules, each having its own data store and analytical functionality. Later in this report, the features of the class diagram will translate into a structure of database tables, worksheets, and executable code modules.

In any large optimization problem such as that of the present study, it is usually necessary to adopt a "divide-and-conquer" approach to reduce the computational effort by addressing a simpler set of problems that are feasible to solve. For example, the optimization process could be done for both bridge and network level simultaneously, but for practical and tractability purposes, the problem has been divided into the two levels. The activity diagram and class diagram expose the seams where the problem can most readily be divided.

1.8 Dynamic View of the Problem Domain

As shown in Figure 3, and consistent with the original research problem statement, there are two primary business processes being modeled: bridge-level planning (center) and program planning (right). Activities associated with both models occur all year round, but the diagram also shows the traced path of project development decision support for one bridge.

1.8.1 Bridge-Level Planning

Bridge inspections are typically completed over a 2-year cycle. After completion of each bridge inspection, data are stored in a bridge management system database and subsequently used for many purposes, including bridge maintenance planning.

At any given point in time, the bridge maintenance planner is able to view the entire bridge inventory, including some bridges with recent inspections and some bridges with inspections up to 2 years old. The planner typically seeks to focus on one bridge at a time and may consequently select for analysis either a recently inspected bridge or a bridge that has an urgent or high-priority need.

In any design or decision-support activity, it is often very difficult, impractical, or unnecessary to start with a "blank slate" and build up a vision for the future of a bridge. In the dynamic view of the problem, the bridge is an ongoing concern, with a wealth of information already available from previous plan-

ning activities. Therefore, after selecting a bridge, the planner should be presented with the most relevant information possible about the current status of the maintenance plans. This indicates a natural expectation that any information entered about a bridge in the project planning process should be saved and retrieved when analysis for the bridge is revisited at a later stage that could range from 1 day to 2 years later. (The system has been designed to work just as well with bridges inspected at intervals longer or shorter than 2 years.)

This set of features expresses a minimum requirement for the software, but does not completely describe what the bridge maintenance planner would like to see when analytical data for a bridge are "brought up" on the planner's screen. The missing key concept is one of relevance: in most cases, it is useful to know what had been planned 2 years earlier, but much may have changed in the intervening time, and some of the changes may have rendered the earlier plans irrelevant. In the context of current policies and a recent inspection, the planner will typically seek a new look at revised future needs. However, the planner will typically want not to have to draw this picture from scratch, so the system needs a way to generate one or more reasonable candidates for future life-cycle activity profiles.

The need to generate one or more reasonable candidates for future life-cycle activity profiles implies that that the decision-support tool has to be capable of handling multiple versions of the future of a bridge: one or more recapitulations of past planning activities, and one or more new versions reflecting the latest available data. The planner may select and/or update previous plans, select a new plan, or design a new candidate that reconciles the best features of the old plan and the new inputs. Therefore, the major decision-making activity in bridge-level planning is the design, evaluation, and selection of alternative life-cycle activity profiles.

As shown in Figure 3, the decision-making activities tend to "loop" a large number of times. Beginning with earlier plans and newly generated life-cycle activity profiles, the planner makes adjustments to the most controllable variables defining each candidate: the scope of the work and the timing of the first planned intervention. The planner evaluates the results and makes more adjustments in an effort to improve the performance of the candidates.

Because the status of the work can be saved and retrieved at any time, and because relevant information may become available at any time of the year, this loop is asynchronous with other bridges and other business processes. It is not necessary to reach a final optimal solution as a precondition for another business process. Instead, other business processes such as program management simply use the most recent available status of the analysis at the time it is needed. Although the maintenance planner may make the current status of a bridge's candidates available to the program planning process at any time, the planner will usually do so as soon as the planner

Figure 4. Class diagram.
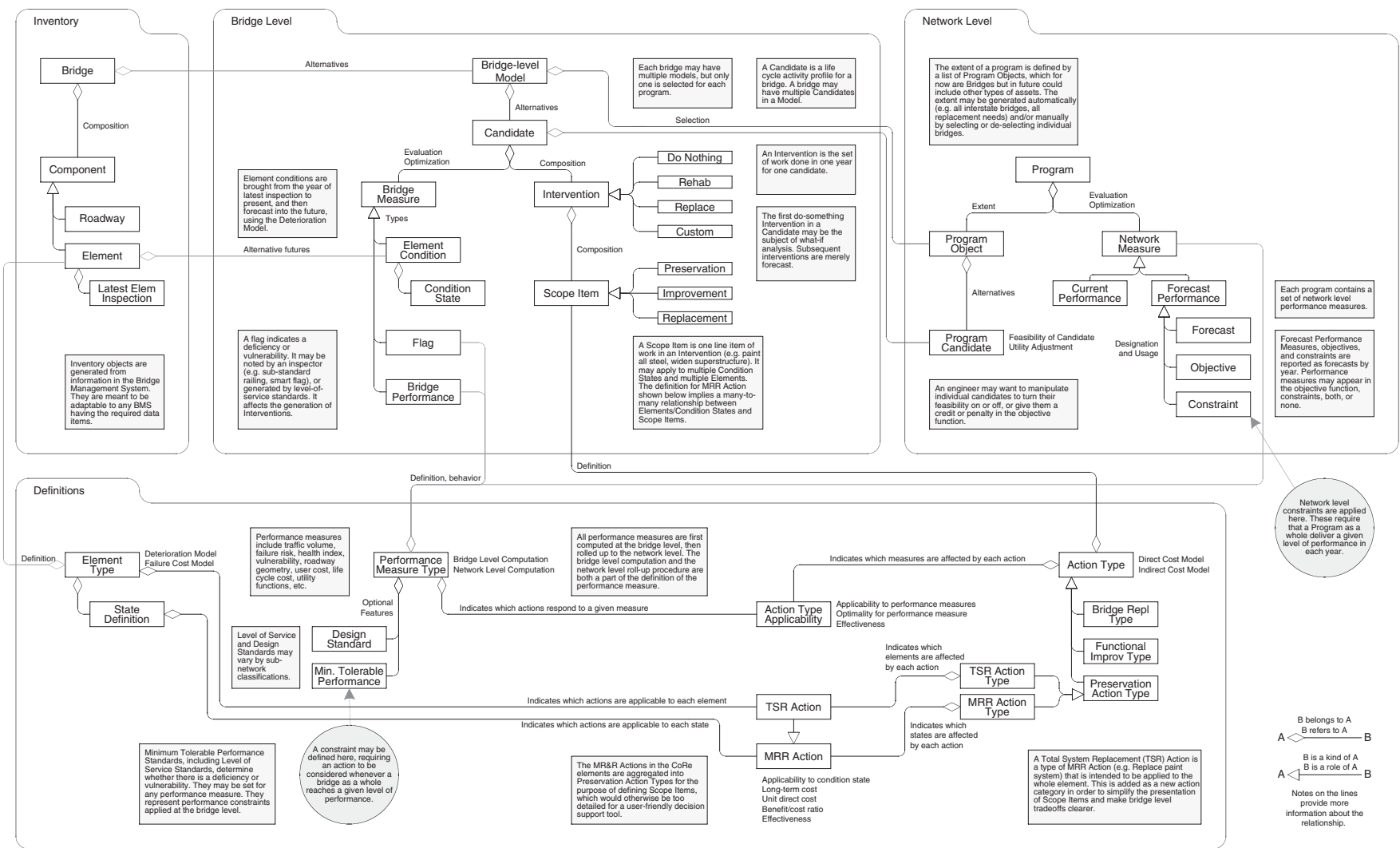
MRR = maintenance, repair, rehabilitation.
TSR = total system replacement.

believes that the developed candidates are superior to those previously submitted to program planning.

The refinement and evaluation loop can be recognized as a sort of optimization process. This is an open optimization loop in that the bridge planner can adjust candidates, create new ones, or eliminate candidates at any time, and the results may be used in other processes at any time, before a unique optimal solution is found. The planner is allowed (but not required) to participate in this loop. For bridges in relatively good condition when it is clear that no near-term work is needed, the planner may be content to allow an automated process to generate, refine, evaluate, and select candidates. The planner can inspect the automated results at any time and decide whether to make any manual adjustments. The automated process is described in detail, with examples, in Section 2.4 (Bridge-Level Optimization).

One implication of this decision-support model is that the evaluation capability in the system must be considerably more robust than what is typically found in other bridge management systems such as Pontis. For example, the bridge planner may want to evaluate the complete recoating of all steel on a bridge, even if not all of the elements have portions in condition states where painting is regarded as justifiable. Each candidate must be evaluated according to all relevant criteria in a multi-objective framework, including criteria that are non-economic or cannot be calculated within the assumptions and data available to the bridge management system. Methods for this evaluation are explained in Section 2.2 (Techniques for Multi-Criteria Decision Making).

Another implication of this decision-support model is that reasonable and speedy procedures are needed for generating standardized candidates and selecting from among them. Most bridges typically do not need immediate maintenance work. Therefore, these automated procedures will be relied upon in most cases to accomplish bridge-level optimization. In theory, there is a large number of variations in scope and timing of candidates on a bridge, given the number of elements that each have a unique set of deterioration models, feasible actions, and cost models. If the program horizon is long enough, each bridge may be subject to multiple interventions, and the need for each future intervention will depend on what happens in earlier interventions. This produces a tree-structured combinatorial explosion of possibilities. It is therefore necessary to give due attention to two distinct algorithms:

- A candidate generation algorithm that limits the number of distinct life-cycle activity profiles while not omitting any that might be selected by the optimization (i.e., that does a good job of narrowing down the possibilities) and
- An algorithm that efficiently evaluates the alternatives, including both system-generated and user-generated candidates, to send to the program management process only

those candidates that could be selected reasonably or feasibly in the network-level model.

As the second algorithm makes clear, the interface with the program management process is a driving force in structuring the bridge-level optimization. Most of what will happen in program management is a manipulation of the timing of work to fit programmatic constraints while maximizing the transportation system performance achieved. Therefore, bridge-level optimization actually needs to deliver a set of solutions, each conditional on a possible outcome of program planning.

1.8.2 Program Planning

Program planning typically starts with a list of bridges and their candidates provided by the bridge-level planning process. Program planning activities are a part of budgeting, programming, and policy-making activities that happen all year round, so the process is asynchronous with bridge-level planning. Each time a program is analyzed, it makes use of whatever candidates are made available to it when they are needed. Like bridge-level planning, program planning is a dynamic process that does not begin with a blank slate but starts from a previously developed program. It is possible, for certain purposes (e.g., when the software is first implemented), to generate an entirely new program using automated means. However, a more common scenario is when the program manager starts with an existing program. Development of a program is therefore an ongoing process that pauses only briefly at breakpoints where official budgets or program documents are prepared and released to the public.

The term "program" could have various meanings depending on the perspectives and needs of the user: it could be a simple list of bridges with general plans to do some work; a specific list of projects needing to proceed into the project development stage; a set of performance goals and objectives; or a funding request, authorization, or appropriation. Usually, a program is a means of reconciling the needs of a variety of stakeholders by manipulating a schedule of bridge work. It is the product of a negotiation. Therefore, a decision-support system for program planning should be considered a tool to support negotiation.

To design a decision-support framework, we regard every aspect of the negotiation as being open to consideration: the selection of bridges, the scope and timing of work, the objectives to be achieved, and the requisite level of funding. Some of the trade-offs are clear: more funding yields superior performance, and performance objectives are then set based on both public expectations and funding availability. These variables are uncertain. In particular, expectations about funding availability change constantly with ongoing discussions about federal and state authorizations and appropriations.

Therefore, the decision-making process is again modeled as a loop of continuous refinement and evaluation. In each pass through the loop, changes may be made to the selection of bridges to be considered, the specific timing and scope of work on each bridge, the objectives to be achieved (especially the relative weights of multiple objectives), and commitments to specific levels of performance and spending.

A key concept in this decision-support framework is that no constraint is established with absolute certainty because all are subject to negotiation. The same variables that define constraints may also define objectives. For example, there is a goal to improve condition as much as possible, to not only achieve some target condition level as a commitment to the public but also try to exceed that. A performance measure may appear in the objective function, the constraints, or both. It may start as an objective and be the subject of a what-if analysis to determine the performance achievable under even a pessimistic scenario. After the decision makers are comfortable about the achievability of the objective, a specified level of that objective may be considered as a constraint.

The decision-support framework for program planning imposes some rather severe computational issues. Each bridge may receive work in any program period and may have more than one scope alternative. There may be as many as almost 50,000 bridges in a state's inventory, as in Texas. The number of combinations grows exponentially with the number of bridges. It is easily possible to create mathematical programs whose solution times range from milliseconds to centuries depending on the problem size.

In spite of the computational issues, the algorithm should respond immediately to common types of changes to the inputs. Common inputs are changes in national budget levels and performance constraints, as well as changes in candidate scope, feasibility, cost, and benefit for an individual bridge. We would want the fastest performance for these changes. Our speed requirements are somewhat relaxed for changes involving redefinition of custom performance criteria used in either the objective function or the constraint, or both. We would consider these changes to be less common. Least common would be changes to the procedures used for candidate definition at the bridge level, where a large number of bridges might have to be recomputed.

1.9 Static View of the Problem Domain

If the dynamic view brings out the "verbs" of the decision-support problem, then there is another perspective that brings out the "nouns." This perspective is the static view. The static view is basically an outline of the data we will need about all the things and concepts that the model will manipulate. However, the static view is also a broader (but less detailed) concept than a database design because it also includes data that are not stored permanently and that may be generated "on-the-fly" or viewed only in reports. The static view focuses on the relationships among objects, the division of responsibility among different parts of the system, and the level of detail of key topics of analysis, as described in the class diagram in Figure 4. Major portions of this class diagram will form the outline of the database design in a later chapter of this report.

1.9.1 Inventory Domain

It is assumed that the decision-support software will be separate from, but linked to, an agency's bridge management system. Therefore, the domain model abstracts the objects that it needs from the bridge management system, and this abstract then acts as an outline for the bridge management system interface. The interface will be a combination of software and data storage that acts as an adapter between the bridge management system and the decision-support software. This part of the domain model does not exactly match any existing bridge management system, since it is organized in the manner needed for the models to be developed here; however, it is meant to be compatible with all existing bridge management systems. The class diagram indicates that each bridge is made of components, and among such components are roadways (both on and under the bridge) and elements.

For each element, there is a record of condition data from the latest inspection. The product is explicitly developed to fit the *AASHTO Guide for Commonly Recognized (CoRe) Structural Elements* (hereafter referred to as the "CoRe Element Guide"), but must be adaptable to agency customizations and to inspection regimes that do not reference the CoRe Element Guide. Certain parts of the bridge-level model depend on having a condition state inspection of elements and a Markovian deterioration model. Agencies using other approaches would have to replace the state-based parts of the model with their own custom deterioration forecasting methods. The network-level model does not have a deterioration forecasting component and therefore does not depend on the CoRe Element Guide.

1.9.2 Bridge-Level Domain

The bridge-level domain describes alternative futures for a bridge, including a selection of feasible candidates, with their scope and timing of work, and a forecast of resulting performance and condition of the bridge and its elements. Candidates are the organizing concept for the bridge-level model. Each candidate is a life-cycle activity profile, a time series of program periods when actions (i.e., interventions) may take place on one bridge. Each intervention is the collection of work to be undertaken in one program period or year, consisting of a list of scope items describing the particular actions envisioned. The

optimization loop, described above in the dynamic view, may generate and evaluate a large number of candidates, computing for each of them the impacts in terms of a predefined utility function and performance measures. These performance measures include forecasts of future element conditions, flags that indicate deficiencies or vulnerabilities, and measures derived from this information (such as a health index, life-cycle cost, or vulnerability index).

The use of utility functions, along with appropriate weights, allows combining important performance measures into a single objective function that can be used for comparing candidates. Within each program period or year, the scope of work yielding the best utility function value is automatically selected to participate in program planning. The bridge maintenance planner may select additional candidates if desired. Among all the program periods, the one with the best-performing candidate is considered to give the optimal timing for the bridge.

1.9.3 Network-Level Domain

Although the network-level domain of the model was named to conform to the research problem statement, its functionality is actually more parallel to what is called the program level in Pontis and other bridge management systems. Its organizing concept is a program, which is a collection of selected candidates and the network performance that would be forecast if the selections were implemented. More specifically, a program is the selection of candidates that would result on a subset of the bridge inventory if a defined set of objectives and constraints were applied. As discussed above, the performance measures that define a program may serve as objectives, constraints, or both. Since the program comes out of a process of negotiation, the relative weights given to various performance measures could vary.

It is important to note that a program, as defined in the present study, does not actually contain candidates, but merely refers to them. So a candidate may be used by multiple programs and does not have to be defined at the same time that the program is defined. This will be very important for data

management efficiency and for efficient distribution of the computational workload.

1.9.4 Definitions Domain

Supporting the main analytical parts of the domain model are definition objects that organize elements, performance measures, and actions. Some of these objects may be derived from existing tables in a bridge management system, which means that they can import the data they need rather than requiring a user to enter the data manually. For example, if a bridge management system has a table of definitions of elements and condition states, the software will be able to use that information. The organization of performance measures and action types differs substantially from the existing bridge management system in order to serve the needs of a multi-objective decision-support tool. Performance measures are expanded to include vulnerability, for example, and it is assumed that agencies will customize these measures. Thus, there must be a place for agencies to define the computations for these measures.

It is necessary to broaden the concepts of level-of-service standards and design standards to take into account that deficiencies of any performance measure can create a need for action. This permits the application of performance constraints, including condition constraints, at the bridge level. In other types of infrastructure management systems, this use of level-of-service standards would be referred to as "minimum tolerable conditions," "trigger values," or "thresholds." In the definition of action types, it is necessary to relate each action to each performance measure in order to define the concepts of applicability, optimality, and effectiveness. An action type is "applicable" to a performance measure if it is capable of improving that performance measure (whether cost-effectively or not). The degree of improvement is called the "effectiveness." An action type is "optimal" for a performance measure if it is the most appropriate response to a deficiency in that measure. The concept of "most appropriate" may be defined by minimizing life-cycle cost (as in the Pontis network optimization) or by agency policy as a result of research or from experience.

CHAPTER 2

Research Approach

2.1 Goals and Performance Measures

The foundation of any decision analysis is a clear statement of goals and performance measures. To describe the consequences of alternative bridge actions and enable trade-offs between competing goals, it is necessary to identify a set of goals and a set of performance measures for each goal. For purposes of this study, the consequences of bridge actions are evaluated on the basis of the following general goals:

- Preservation of bridge condition
- Traffic safety enhancement
- Protection from extreme events
- Agency cost minimization
- User cost minimization

A set of performance measures for each goal clarifies the meaning of each goal and is required to measure the consequences of alternative bridge actions. Performance measures are also sometimes referred to as *attributes* or *criteria*. Some desirable properties for the set of performance measures for each goal are the following (Keeney and Raiffa 1976):

- **Completeness:** A set of performance measures is complete if it is adequate in indicating the degree to which the goal is met.
- **Operational:** Since the idea of decision analysis is to help the decision maker choose the best course of action, the performance measures must be useful and meaningful to understand the implications of the alternatives and to make the problem more tractable.
- **Non-redundancy:** The performance measures should be defined to avoid double-counting of consequences.
- **Minimal:** It is desirable to keep the set as small as possible to reduce dimensionality.

Individual performance measures should be *unambiguous, comprehensive, direct, operational,* and *understandable* (Keeney and Gregory 2005). A performance measure is *unambiguous* when there is a clear relationship between the consequences that might or will occur and the level of performance measure used to describe those consequences. A performance measure is *comprehensive* when its levels cover the full range of possible consequences and any implicit judgments appropriate for the decision problem. A performance measure is *direct* when its levels directly describe the consequences of the fundamental goal of interest. A performance measure is *operational* when information about it can be easily gathered. Finally, a performance measure is *understandable* when anyone interested in the analysis can understand it.

Table 1 shows the set of performance measures for each goal. Although non-redundancy is a desirable property for each set of performance measures, it is difficult to achieve given the type of performance measures that are typically used for bridge candidates.

If the decision makers choose to, they can define their own performance measures. However, if they do so, they will need to provide any pertinent data.

The following sections describe the performance measures suggested here.

2.2 Techniques for Multi-Criteria Decision Making

2.2.1 Introduction

Implicit in any decision-making process is the need to construct, either directly or indirectly, the preference order, so that alternative candidates can be ranked and the best candidate can be selected. For some decision-making problems, this may easily be accomplished. For example, in case of a decision based on a cost-minimization rule (where the lowest-cost candidate is chosen), the preference order is adequately represented by the natural order of real numbers (representing costs). Hence, in such a case, the preference order need not be constructed explicitly.

Table 1. Performance measures.

GOAL	PERFORMANCE MEASURES
1. Preservation of Bridge Condition	(a) Condition Ratings (NBI 58-60, 62) (b) Health Index (c) Sufficiency Rating
2. Traffic Safety Enhancement	(a) Geometric Rating/Functional Obsolescence (b) Inventory Rating *or* Operating Rating
3. Protection from Extreme Events	(a) Scour Vulnerability Rating (b) Fatigue/Fracture Criticality Rating (c) Earthquake Vulnerability Rating (d) Other Disaster Vulnerability Rating (Collision, Overload, Human-Made)
4. Agency Cost Minimization	(a) Initial Cost (b) Life-Cycle Agency Cost
5. User Cost Minimization	(a) Life-Cycle User Cost

However, complex decision-making problems typically involve multiple conflicting criteria. It is often true that no alternative is better than all other alternatives in terms of all of these criteria. For example, one usually cannot maximize service levels and at the same time minimize costs. In the context of a bridge management decision-making process, some of the criteria other than cost are health index and vulnerability ratings.

Inherent in any multi-criteria decision making is the value trade-off—that is, the decision maker is faced with a problem of trading off the achievement of one criterion against another criterion. In such a case, an important aspect of the decision-making process is to be able to capture these value trade-offs effectively. Hence, it is desirable to explore the decision maker's preference structure in some direct fashion and to attempt to construct some sort of preference order directly.

An important class of decision-making techniques that attempt to construct the preference order by directly eliciting the decision maker's preference is predicated on what is known as *utility theory*. This, in turn, is based on the premise that the decision maker's preference structure can be represented by a real-valued function called a *utility function*. Once such a function is constructed, the selection of the alternative candidates can be done using an optimization method. Broadly speaking, this technique involves three steps:

1. **Weighting:** This assigns relative weights to the multiple criteria (described in Section 2.2.4).
2. **Scaling:** Because the performance criteria can be of different units, scaling provides a common scale of measurement and translates the decision maker's preferences for each performance criterion on a 0–100 scale. This involves developing single-criterion *utility* functions (described in Sections 2.2.2 and 2.2.3).
3. **Amalgamation:** Amalgamation is combining the single-criterion utility functions using the relative weights into

one measure based on mathematical assumptions about the decision maker's preference structure. This involves deriving the functional forms of multi-criteria utility functions.

The underlying assumptions and the assessment methods for each of the three procedures depend on either of two scenarios: certainty scenario and risk scenario. The certainty scenario refers to the case where the consequence of each alternative in terms of multiple criteria is known with certainty. For example, performance measures like the health index or geometric rating that any alternative candidate would yield are assumed to be known with certainty. The risk scenario refers to the case where the trade-off issue remains (as in the certainty scenario), but difficulties are compounded because it is not clear what will be the exact consequences of each of the alternatives. Each possible consequence of any alternative is associated with some probability.

The subsequent sections of this chapter describe in detail how to deal with these issues of developing the preference structure and the above three procedures under the certainty and risk scenarios. To be mathematically precise, the scaling procedure is referred to as the development of *utility* functions under the risk scenario and *value* functions under the certainty scenario. Utility and value functions are similar in terms of what they represent, but differ in capturing the risk attitudes of decision makers. Hence, the assessment methods for value and utility functions also differ. A utility function is more general because it incorporates decision maker's risk attitudes.

2.2.2 Decision Making Under Certainty Scenario

This section describes methods to capture, construct, and quantify the decision maker's preferences under this scenario.

2.2.2.1 Value Function

Utility theory assumes that the decision makers can choose among the alternatives available to them in such a manner that the satisfaction derived from their choice is as large as possible. This, of course, implies that the decision makers are aware of the alternatives and capable of evaluating them.

It is assumed that all information pertaining to the various levels of the criteria can be captured by the decision maker's *value function,* which represents a scalar index of preference or value for the available alternatives. In effect, the decision maker's value function is a formal, mathematical representation of his or her preference structure. A simple example of value function for health index is shown in Figure 5.

Rendering decision analysis operational for multi-criteria problems may entail assessing the decision maker's multivariate value function:

$$v(z) = v(z_1, z_2, \ldots, z_p) \tag{2-1}$$

where z represents the consequence set of an alternative in terms of p criteria: $z_1, z_2, \ldots, z_p$.

This function has the following property that makes it useful for addressing the issue of trade-offs among multiple criteria (Keeney and Raiffa 1976):

$$v(z') > v(z'') \tag{2-2}$$

if and only if z' is preferred to z''.

An example of a multivariate value function would be a function in three-dimensional space that assigns a scalar value to every possible combination of health index and geometric rating. Such multivariate functions would capture the decision maker's preferences precisely but are not useful from a prac-

tical standpoint. And the difficulty gets compounded as the number of dimensions is increased.

Assessing such multivariate value functions can be a difficult and cumbersome task, especially because of the multi-dimensionality of the problem. An obvious and effective way to alleviate this difficulty is to reduce the dimensionality if possible. Hence, decision theorists often decompose the multivariate value function into single-criterion value functions. The next section deals with the decomposed functional form and the underlying assumptions.

2.2.2.2 Functional Form

One of the main theorems in value theory states the following: Given the criteria $Z_1, Z_2, \ldots, Z_p$, the following additive value function exists if and only if the criteria are *mutually preferentially independent* (Keeney and Raiffa 1976):

$$v(z_1, z_2, \ldots, z_p) = \sum_{i=1}^{p} v_i(z_i) \tag{2-3}$$

where v_i is a single-criterion value function over the criterion Z_i.

Concept of Mutual Preferential Independence. Consider two mutually exclusive and collectively exhaustive subsets of the set $Z \equiv \{Z_1, Z_2, \ldots, Z_p\}$: X and Y. The set of criteria X is *preferentially independent* of the complementary set Y if and only if the conditional preference structure in the x space given y' does not depend on y'. In other words, the preference structure among the criteria in set X does not depend on the levels of the criteria in Y. Symbolically, if (x_1, y_0) is preferred to (x_2, y_0), then (x_1, y) is preferred to (x_2, y) for all y.
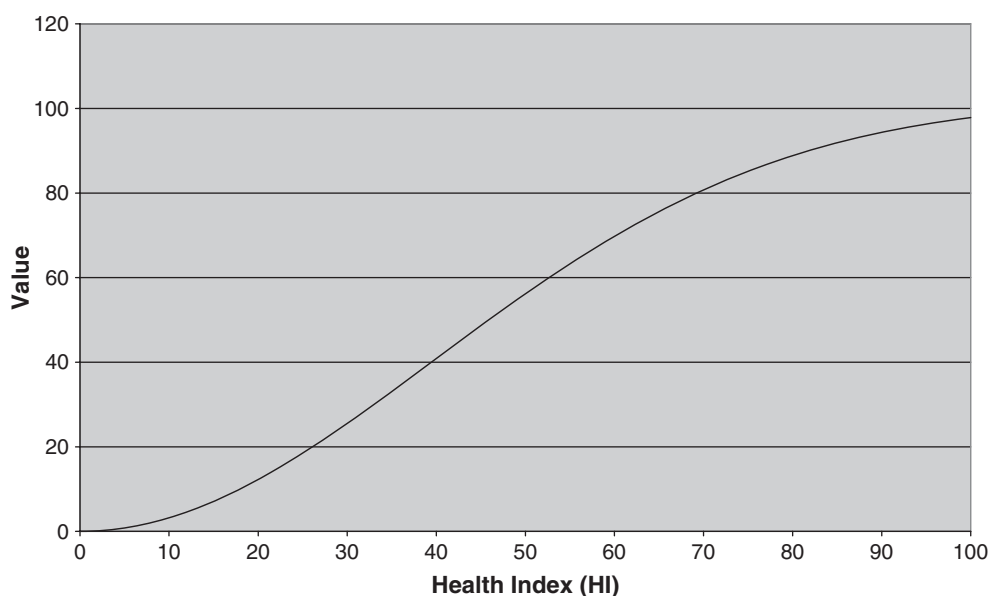
Figure 5. Example of a value function.

The set of criteria Z are mutually preferentially independent if every subset X of these criteria are preferentially independent of subset X's complementary set of criteria.

Example.   In the context of bridge decision making, say there are three criteria to evaluate an alternative: health index (HI), cost, and geometric rating (GR). If these three criteria are mutually preferentially independent, the decomposed value function is:

$$v(\text{alternative}) = v(\text{HI}) + v(\text{cost}) + v(\text{GR}) \qquad (2\text{-}4)$$

In the above value function, the relative scaling of the three criteria is implicit in the single-criterion value functions. To simplify the assessment procedures of the single-criterion value functions as defined above, a common technique is to represent the relative scaling explicitly as follows:

$$v(\text{alternative}) = w_1 * v(\text{HI}) + w_2 * v(\text{cost}) + w_3 * v(\text{GR}) \qquad (2\text{-}5)$$

where w_1, w_2, and w_3 are referred to as *relative weights.*

2.2.2.3 Relative Weights

As discussed in the previous section, relative weights are a means of capturing the relative importance of multiple criteria. In other words, these weights represent trade-offs between various criteria. Section 2.2.4 describes in detail various procedures to develop the relative weights. For the case of certainty and the functional form as discussed above, either of the following two methods can be used:

- Direct weighting or
- Analytic hierarchy process.

2.2.3 Decision Making Under Risk Scenario

This section describes methods to capture, construct, and quantify the decision maker's preferences under this scenario.

2.2.3.1 Multi-Attribute Utility Theory

The power of the concept of utility and the grounds of multi-attribute utility theory are based on the following: If an appropriate utility is assigned to each possible consequence and the expected utility of each alternative is calculated, then the best course of action is the alternative with the highest *expected utility* (Keeney and Raiffa 1976). Utility of an alternative is a random variable, and the expected utility refers to the *first moment* or *mean* of the random variable. A typical application of multi-attribute theory involves the following steps (Goicoechea et al. 1982):

1. Assumptions about the decision maker's preferences are postulated.
2. An appropriate functional form is derived based on the assumptions.
3. Appropriateness of the assumption is verified with the decision maker.
4. Preference orders (i.e., utility functions) are constructed for each criterion.
5. Single-criterion utility functions are synthesized using the derived functional form and assessed relative weights.
6. The preference order for alternatives is constructed based on the expected utilities.

2.2.3.2 Utility Function

A utility function captures a decision maker's preferences regarding levels of selected attributes, as well as his or her individual attitude toward risk for each attribute. The utility function is similar to a value function, but it also captures the risk preferences of the decision maker for each attribute. The expected values of the utility function are then used to evaluate the alternatives. The alternative with maximum expected utility value is most preferred.

Although multi-attribute utility functions capture a decision maker's preferences regarding levels of attributes and attitude toward risk for several attributes simultaneously, assessing such multi-attribute functions can be an extremely difficult task. Hence, decision theorists often decompose the multivariate utility function into single-criterion utility functions by using mathematical theorems in utility theory. This then simplifies the problem to one where weighting and synthesizing single-attribute utility functions are used to derive a multi-attribute utility function. The next section describes the decomposed functional form and the underlying assumptions.

2.2.3.3 Functional Form

A very important theorem in utility theory states the following: Given the criteria $Z_1, Z_2, \ldots\ldots, Z_p$, the following multiplicative utility function exists if and only if the criteria are *mutually utility independent* (Keeney and Raiffa 1976):

$$ku(z_1, z_2, \ldots\ldots, z_p) + 1 = \prod_{i=1}^{p} \left[kk_i u_i(z_i) + 1 \right] \qquad (2\text{-}6)$$

where u_i is a single-criterion utility function over the criterion Z_i, and k and k_i are scaling constants.

Concept of Mutual Utility Independence.   Consider two mutually exclusive and collectively exhaustive subsets of the set $Z \equiv \{Z_1, Z_2, \ldots\ldots, Z_p\}$: X and Y. The set of criteria X is *utility independent* of set Y if and only if the conditional preference order for *lotteries* involving only changes in the levels of attri-

butes in X does not depend on the levels at which the attributes in Y are held fixed. Symbolically, if $<x_1,y_0>$ is preferred to $<x_2,y_0>$ then $<x_1,y>$ is preferred to $<x_2,y>$ for all y. The symbol "$<>$" represents a lottery that captures the risk preference of the decision maker in the presence of uncertainty.

The set of criteria Z are mutually utility independent if every subset X of these criteria is utility independent of its complementary set of criteria.

Another important theorem states the existence of additive utility function: Given the criteria $Z_1, Z_2, \ldots, Z_p$, the following additive utility function exists if and only if the additive independence condition holds among the criteria (Keeney and Raiffa 1976):

$$u(z_1, z_2, \ldots, z_p) = \sum_{i=1}^{p} k_i u_i(z_i) \qquad (2\text{-}7)$$

where u_i is a single-criterion utility function over the criterion Z_i.

This means that preferences over lotteries on $Z_1, Z_2, \ldots, Z_p$ depend only on their marginal probability distributions and not on their joint probability distribution.

The applicability of the multiplicative or additive functional forms for multi-attribute utility function depends on the underlying assumptions as stated in the theorems above. The appropriateness of the underlying assumptions can be checked through eliciting information from the decision maker. An appropriate functional form is selected on the basis of the results of a questionnaire (details of which are provided in Section 2.2.5 and Appendix C).

2.2.4 Methods for Developing Relative Weights

2.2.4.1 Direct Weighting

In the direct weighting method, the decision maker assigns numerical values to weights in a direct manner. Examples of direct methods include the following:

- **Ranking:** Rank all the criteria in the order of decreasing importance.
- **Categorization:** Assign criteria to different categories of importance, each carrying a different weight.
- **Point allocation:** Allocate 100 points among criteria in proportion to their importance.

Ranking is not the best choice for bridge decision making because it will yield only an ordinal scale of importance, as opposed to a cardinal scale of importance—cardinality is important because these weights will be used in a multivariate value function. Categorization is also not the best choice because it is useful only when there are many criteria. Thus, among the three approaches, point allocation is best suited for the bridge decision-making problem.

Although the direct weighting method is simple, it may not capture the decision maker's preferences of relative weights as effectively as other, more rigorous methods. However, this method is useful because actual program priorities and resource allocations are the result of a process of negotiation. In many agencies, a major goal of the program manager is to develop a program that elected officials will approve. The program manager does not know the implicit weights of the politicians who must approve the program, and these weights are unknowable because they may be negotiated in a venue where the program manager has no involvement. Therefore, the program manager in such a situation may want to develop several program alternatives to try to find a set from which the politicians will be willing to choose, each with a different set of weights. The direct weighting method is useful in such a scenario because it is simple and it gives a starting point for a human process that will actually set the weights.

2.2.4.2 Analytic Hierarchy Process

The analytic hierarchy process aims to arrive at the relative weights for multiple criteria in a realistic manner while allowing for differences in opinion and conflicts that exist in the real world. The analytic hierarchy process can handle quantitative, qualitative, tangible, and intangible criteria. The process is based on three principles: decomposition, comparative judgments, and synthesis of priorities. It constructs a hierarchy and uses pairwise comparisons at each level to estimate the relative weights.

Let $z(i)$, $i = 1, 2, \ldots, p$ be the set of criteria at a given level, and let quantified judgments on a pair of criteria $z(i)$, $z(j)$ be represented by the following matrix:

$$A = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1p} \\ 1/a_{12} & 1 & \cdots & a_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ 1/a_{1p} & 1/a_{2p} & \cdots & 1 \end{pmatrix} \qquad (2\text{-}8)$$

If the measurements were exact, the weights would be given by the following:

$$w_i/w_j = a_{ij} \ (\text{for } i, j = 1, 2, \ldots, p) \qquad (2\text{-}9)$$

But the measurements are not exact, and the matrix may not be consistent. So, to allow for deviations, the above formulation reduces to the following:

$$w_i = \frac{1}{p} \sum_{j=1}^{p} a_{ij} w_j \ (\text{for } i = 1, 2, \ldots, p) \qquad (2\text{-}10)$$

24

For the existence of a unique solution, the above formulation can further be reduced to the following (Saaty 1980):

$$A'w' = \lambda_{max} w' \tag{2-11}$$

where

A' = reciprocal matrix that is a perturbation of A,
w' = eigenvector of A', and
λ_{max} = largest eigenvalue of the matrix A'.

So, based on this theorem, the analytic hierarchy process boils down to (1) constructing a pairwise comparison matrix and (2) estimating the value of the eigenvector that reflects the relative weights. Numerical methods can be used to compute the eigenvector of the matrix, which in turn yields the relative weights. The theoretical proof of this theorem is outside the scope of this report, but the following example illustrates a simple method that can be used to estimate an approximate value of the eigenvector.

In order to construct this example matrix, a relative scale is defined, as shown in Table 2. Then, using this scale, all criteria in the set are compared pairwise, as shown in Table 3. The values given by a decision maker can be interpreted as follows:

- 7 indicates that decision maker believes deck condition is strongly more important than superstructure condition rating.
- 5 indicates that decision maker believes deck condition is moderately more important than substructure condition rating.
- ⅓ indicates that decision maker believes superstructure condition is slightly less important than substructure condition rating.

These values are then numerically processed to arrive at the relative weights in the following way: The lower triangle of the matrix is filled with corresponding reciprocal values. Then each entry in column j is divided by the sum of entries in column j. This yields a new matrix:

$$A_{norm} = \begin{pmatrix} 0.7447 & 0.6364 & 0.7895 \\ 0.1064 & 0.0909 & 0.0526 \\ 0.1489 & 0.2727 & 0.1579 \end{pmatrix} \tag{2-12}$$

To find an estimate of the relative weight w_i, compute the average of entries in row i of A_{norm}. This exercise combines the scores by normalizing and averaging to yield the relative weights as:

- Deck Condition: 0.724,
- Superstructure Condition: 0.083, and
- Substructure Condition: 0.193.

It is also possible to compute the eigenvector directly using standard mathematical software. The eigenvector, if reported normalized, can be read directly to get the relative weights. Each element of the vector corresponds to the relative weight of a criterion.

The analytic hierarchy process is well suited to the bridge decision-making problem because of a natural hierarchy of criteria in the decision-making process. It effectively captures the decision maker's preferences for relative weights by pairwise comparisons. It is also an appropriate method when using the relative weights in an additive multivariate value function or utility function because of the inherent structure of analytic hierarchy process. However, one disadvantage of the analytic

Table 2. Relative scale for example pairwise comparison matrix.

IF:	THEN ratio of X/Y should be:
Criterion X is extremely more important than Criterion Y	9
Criterion X is strongly more important than Criterion Y	7
Criterion X is moderately more important than Criterion Y	5
Criterion X is slightly more important than Criterion Y	3
Criterion X is equally important to Criterion Y	1
Criterion X is slightly less important than Criterion Y	1/3
Criterion X is moderately less important than Criterion Y	1/5
Criterion X is strongly less important than Criterion Y	1/7
Criterion X is extremely less important than Criterion Y	1/9

Table 3. Pairwise comparison matrix (A).

Criterion	Deck Condition	Superstructure Condition	Substructure Condition
Deck Condition	1	7	5
Superstructure Condition	--	1	1/3
Substructure Condition	--	--	1

hierarchy process is that the number of pairwise comparisons can become very large when there are many criteria. However, this is not a problem with bridge decision making.

2.2.4.3 Observer-Derived Weights

Observer-derived weights estimate relative weights of multiple goals by analyzing unaided subjective evaluations of alternatives using regression analysis (Hobbs and Meier 2000). For each of the given alternatives, the decision maker is asked to assign scores of benefits under individual goals and a total score on a scale of 0 to 100. A functional relationship is then established using the total score as a response variable and the scores assigned under individual goals as explanatory variables through regression analysis. The calibrated coefficients of the model thus become the relative weights of the multiple goals.

One advantage of this method is that it is based on the simple regression methodology. This is a "policy capturing" method used by psychologists and pollsters to yield weights that best predict unaided opinions. However, researchers have mentioned that the purpose of multi-criteria decision analysis is to improve, not simulate, such holistic judgments. It has also been noted that people tend to ignore all but a handful of attributes when ranking multi-attribute alternatives. So, the method may not be particularly effective when there are many criteria. This is not an issue with the analytic hierarchy process technique because the decision makers look at only one pair (two criteria) at a time. Hence, we do not recommend using the observer-derived weights method.

2.2.4.4 Gamble Method

The gamble method chooses a weight for one goal at a time by asking the decision maker to compare a "sure thing" and a "gamble." The first step is to determine which goal is most important to move from its worst to best possible level. Then, consider two situations: First, the most important goal is set at its best level, and other goals are at their least desirable levels. Second, the chance of all goals at their most desirable levels is set to p, and chance of $(1-p)$ for all goals at their worst values. If the two situations are equally desirable, the weight for the most important goal will be precisely p. The same approach is repeated to derive the weights for remaining goals with decreasing relative importance. This method is particularly helpful in the uncertainty scenario because this captures the relative risk attitudes of the decision maker toward multiple criteria. This method is also helpful in testing some underlying assumptions of the functional forms of the multi-attribute utility function based on standard theorems in utility theory. However, one of the disadvantages of this method is that it is relatively difficult to understand.

2.2.5 Methods for Developing Single-Criterion Value Functions

As discussed earlier, a value function represents a scalar index of preference or value for the available alternatives. This section describes two methods to assess and construct the single-criterion value function.

2.2.5.1 Midvalue Splitting Technique

The midvalue splitting technique solicits information from the decision maker about his or her indifference between changes in levels of the criterion. This method is explained using an example for deck condition (DC) rating. The steps of the method are presented below in the form of a dialogue between the analyst and decision maker:

STEP 0: Set $V(\text{DC}=0)=0$ and $V(\text{DC}=9)=100$
STEP 1: Find X_{50} for which $V(X_{50})=50$
Find X_{50} such that you are equally delighted with
  – an improvement of DC from 0 to X_{50}
  – an improvement of DC from X_{50} to 9
$X_{50}=6$
STEP 2: Find X_{25} for which $V(X_{25})=25$
Find X_{25} such that you are equally delighted with
  – an improvement of DC from 0 to X_{25}
  – an improvement of DC from X_{25} to X_{50}
$X_{25}=4$
STEP 3: Find X_{75} for which $V(X_{75})=75$
Find X_{75} such that you are equally delighted with
  – an improvement of DC from X_{50} to X_{75}
  – an improvement of DC from X_{75} to 9
$X_{75}=8$
STEP 4: Consistency Check
Are you equally delighted with
  – an improvement of DC from X_{25} to X_{50}?
  – an improvement of DC from X_{50} to X_{75}?

If the answer to the last question is yes, the values are consistent. If not, the decision maker is asked to revise the previous three responses (Steps 1–3). On the basis of these values, the value function of deck condition rating for the decision maker can be constructed as shown in Figure 6.

As shown in this example, the midvalue splitting technique is simple to implement and can be used to assess the value functions of various bridge performance criteria.

2.2.5.2 Direct Rating Method

Direct rating is a relatively simple method to assess the decision maker's preferences for a performance criterion. The method is particularly useful for developing value functions of
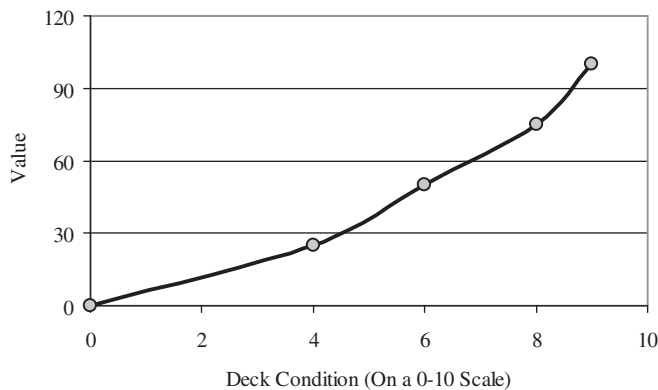
Figure 6. A decision maker's value function for deck condition.

the performance criteria that have only few possible discrete levels. In such a case, it becomes feasible to ask the decision maker to directly assign the values for each of the possible levels of the performance criterion. This method could be used for the vulnerability ratings.

2.2.6 Developing Single-Criterion Utility Functions

As discussed earlier, a utility function captures the decision maker's preferences regarding both levels of attributes and risk for the attribute. There are methods in the literature to assess and construct utility functions based on eliciting information from the decision maker about preferences over lotteries. However, these methods tend to be rather cumbersome and difficult.

By definition, a utility function is a value function, but a value function is not necessarily a utility function. A utility function is more general because it incorporates the decision maker's risk attitudes. Using this concept, assessing utility functions can be greatly simplified because they can be developed *over* the value functions. This proves to be effective because developing value functions tends to be much easier than developing utility functions. And because we have already considered developing value functions for the certainty case, we can use those value functions to develop utility functions.

The following theorem gives the mathematical result we need to develop utility functions: Given a set of criteria Z_1, Z_2, ... , Z_p, that are mutually preferentially independent (in the certainty scenario), the utility function u must have one of the following three forms (Keeney and Raiffa 1976):

- $u(z) \sim -\text{EXP}(-cv(z)), c > 0$
- $u(z) \sim v(z)$
- $u(z) \sim \text{EXP}(cv(z)), c > 0$

where $v(z)$ is the value function of a criterion (for example, $v(\text{HI})$ and $u(\text{HI})$ represent the value and utility functions respectively for HI), and c is a constant. This theorem is useful in the following two ways:

- It provides for a simple procedure to obtain a multi-attribute utility function given that the value function has already been assessed.
- The analyst can independently assess both a multiplicative (or additive) utility function and a value function and use one function to check against the other.

2.2.7 Group Decision Making

The methods described in the previous sections are used to assess the relative weights, value, and utility functions for *individual* decision makers. Aggregation techniques are then used to synthesize the local priorities of the criteria with respect to each decision to arrive at global priorities.

There might be inconsistencies among the individual decision makers in terms of assigning relative weights to multiple criteria. A decision maker may also be concerned about the effects of his or her actions on other individuals, and he or she may want to incorporate others' preferences into his or her own value assessments. The Delphi technique (Dalkey and Helmer 1963) is a group decision-making tool that is widely used to come to a consensus and a holistic decision through aggregation of judgments from individual experts. The results from a questionnaire are analyzed, and the summary statistics are provided. The decision makers can then review their responses and change them if they like. The responses for the relative weights can be synthesized by averaging values across individual decision makers. To synthesize the value function assessments of individual decision makers, regression can be used to arrive at a global function that best represents the individual preference orders.

2.2.8 Summary

Section 2.2 described the multi-criteria decision-making methodologies that can be applied to the bridge management decision-support problem. Implicit in any decision-making process is the need to construct, either directly or indirectly, the preference order for the alternatives to enable ranking and selection. Many complex decision-making problems involve multiple conflicting criteria that make it necessary to assess the trade-offs.

Two different scenarios were considered: the certainty scenario and the risk scenario. The certainty scenario refers to the case where the consequences of each alternative in terms of multiple criteria are known with certainty. On the other

hand, if there is uncertainty (risk scenario) in the problem, the trade-off issue remains, but difficulties are compounded because it is not clear what will be the exact consequences of each of the alternatives.

Section 2.2 included a discussion of analytical tools to address the multi-dimensional nature of the problem. Different functional forms and underlying assumptions were discussed for both certainty and risk scenarios based on the mathematical concepts from *value* and *utility* theory. Methods to assess the relative weights, single criterion value, and utility functions, along with their merits and limitations, were also presented. The concepts were illustrated using examples of bridge performance measures.

The theoretical concepts from value and utility theory were applied in a practical manner to simplify the process of assessing the relative weights, value, and utility functions. A combination of the methods presented in Section 2.2 was used in designing a questionnaire from which relative weights and value functions were developed. This questionnaire was administered during the panel meeting. The purpose of the questionnaire was to develop a set of default weights, value, and utility functions that can be incorporated as default parameters in the final software product. However, the users of the software will be able to change the default relative weights and value functions.

2.3 Network-Level Optimization— Formulation

2.3.1 Introduction

The network-level model consists of optimization methods for selecting a subset of candidate projects from a network-wide candidate project list to yield maximum network benefits subject to multiple constraints. Each network benefit is measured with multiple criteria, and the constraints can be budgetary limitations or performance constraints. The model helps to investigate the impact of various funding levels on network performance and can be used to estimate funding needed to achieve user-specified condition targets and acceptable risk levels. Because of the need for consistency of the analyses at the network level and bridge level, it is necessary for bridges and candidates to keep their identity in the network level.

The problem we are dealing with here is a special case of integer programming problems known as the knapsack problem, which is a problem of combinatorial optimization. It is one of the most well-known integer programming problems and has received wide attention from the operations research community during the last four decades. Although recent advances have made possible the solution of medium-size instances, solving this computationally hard problem remains a very interesting challenge (Freville 2004). One of the most

common applications of the knapsack problem is the capital budgeting problem (Lorie and Savage 1955). Other applications include cutting stock, investment policy for the tourism sector, allocation of databases and processors in a distributed data processing, delivery of groceries in multi-compartment vehicles, multi-commodity network optimization, and daily management of a remote-sensing satellite (Freville 2004). More specifically, the network optimization problem is a multi-choice, multi-dimensional knapsack problem. The following section defines the problem.

2.3.2 Problem Definition

This section starts with defining a simple knapsack problem and goes further to describe the multi-choice and multi-dimensional aspects of more complex knapsack problems.

2.3.2.1 The 0-1 Knapsack Problem

The knapsack problem can be explained using a simple analogy: Consider a shopper with a shopping cart at a grocery shop. The shopper intends to purchase as many distinct items that can be purchased with the available funds. Each item in the store has some associated volume, cost, and utility (i.e., degree of satisfaction) to the shopper. The shopper wishes to fill the shopping cart with as many items as possible to maximize his or her overall satisfaction with the items purchased (this is the objective). However, the shopping cart is not very large and therefore can hold only a certain volume of items (a "size" constraint), and the total cost of items purchased cannot exceed the available funds or budget (another "size" constraint). Then the knapsack problem is to determine which items the shopper should select. This is a simple form of the knapsack problem. It has many variations and generalizations in the literature.

2.3.2.2 Multi-Choice Knapsack Problem (MCKP)

In a more generalized form of the knapsack problem, the consumer has a set of *n* classes, where each class contains a number of items. The consumer needs to pick exactly one item from each class. The consumer faces a "multi-choice" problem because there is a set of choices for each class. For example, the consumer needs to select one car from a set of cars, one computer from a set of computers, and one mobile phone from a set of phones to maximize the reward gained with the constraint that the basket cannot hold more than a certain weight. In the context of bridge management, each class represents a specific bridge in the network. The choices for each bridge include the possible candidate projects, including the do-nothing alternative. The reward is measured in terms of multiple criteria,

such as health index and vulnerability rating. The "size constraint" of the knapsack corresponds to the network budget constraint for the program period.

2.3.2.3 Multi-Dimensional Knapsack Problem (MDKP)

In another variation of the knapsack problem, the consumer seeks to select from a set of distinct items subject to more than one size constraint, and each item has a known weight, volume, and width. For example, the basket cannot hold more than a certain weight, more than a certain volume, and more than a certain width. This gives the multi-dimensionality aspect to the problem. In the context of bridge management, a scenario with multiple "size" constraints could be one having a budget constraint, a networkwide condition constraint (i.e., a minimum condition target), a networkwide risk constraint (i.e., maximum risk levels tolerable), and so forth.

2.3.2.4 Multi-Choice, Multi-Dimensional Knapsack Problem (MCMDKP)

In a further generalization of the knapsack problem, both the multi-choice (more than one item or activity in each class) and the multi-dimensional (more than one size constraint) aspects are present. The bridge network optimization problem being addressed in the present study falls in this category. The multi-choice aspect of the problem is that exactly one candidate project must be selected for each bridge, where the do-nothing alternative is included in the set of candidates. The multi-dimensional aspect of the problem is that there are multiple constraints, such as budget and performance target constraints. The multi-objective aspect of the problem refers to the use of more than one criterion in the objective function.

2.3.3 Handling Multiple Criteria and Constraints

As stated in the previous section, multiple criteria (i.e., performance measures) are included in the objective function of the optimization problem. These are denoted as variables to be maximized (or minimized). The solution will attempt to make the objectives as large (or small) as possible, but it is not guaranteed to reach any specific value as it would in a constraint. In the problem, it is possible to have multiple criteria—such as life-cycle cost, condition, and risk—that are all maximized or minimized at the same time. These criteria are converted to utilities on the basis of utility theory. This involves combining the criteria in a way that adequately reflects their relative importance, their various units of measure, and ways in which their intrinsic value varies along the measurement scale. Therefore, the objective function of the optimization problem is to maximize the networkwide utility of the selected candidates.

The optimization process attempts to maximize networkwide utility subject to a set of constraints. Multiple constraints are inherent in the problem structure and formulation and give the multi-dimensionality to the knapsack problem. The solution methods (as discussed in subsequent sections) are designed to incorporate these multiple constraints. From a decision-making standpoint, the multiple constraints could be

- A budgetary limitation (i.e., maximum initial cost),
- A minimum threshold for the average health index of the network, or
- A maximum threshold for the average vulnerability rating of the network.

These constraints become the various constraint inputs provided by the decision maker.

2.3.4 Problem Formulation

Depending on the decision maker's concern at any time, the optimization problem can take different formulations. These formulations are classified on the basis of the number and type of size constraints in the knapsack problem, as summarized in Table 4.

We first define some variables and then construct the formulations to illustrate the process of multi-objective network optimization. Let U_{jk} equal the utility associated with project j for bridge k. This utility, in general, is a function of various performance indicators, for example:

$$U_{jk} = f\left(UC_{jk}, H_{jk}, V_{jk}\right) \tag{2-13}$$

Table 4. Classifications of knapsack formulation based on number and type of size constraints.

Classification	Type of Size Constraints	Number of Size Constraints
Formulation 1	Only Budget Constraint	One-Dimensional (MCKP)
Formulation 2	Only Nonbudget Constraint	One-Dimensional (MCKP)
Formulation 3	Budget and Multiple Nonbudget Constraints	Multi-Dimensional (MCMDKP)
Formulation 4	Multiple Nonbudget Constraints	Multi-Dimensional (MCMDKP)

where

UC_{jk} = user cost associated with project j for bridge k,
H_{jk} = health index for bridge k at the end of the program period if project j is implemented, and
V_{jk} = vulnerability rating for bridge k at the end of the program period if project j is implemented.

Depending on the formulation of the problem, the utility can be defined as a function of only a subset of the performance measures, which can be chosen by the decision maker.

Let:

AC_{jk} = agency cost associated with project j for bridge k;
n = number of bridges in the network;
L_k = set of candidate projects considered for bridge k;
X_{jk} = 1 if project j is selected for bridge k, and 0 otherwise; and
B = budget available for the program period.

2.3.4.1 Formulation 1 (Only Budget Constraint, MCKP)

The problem is classified as Formulation 1 when the decision maker's concern is to determine the best possible candidate projects to be implemented to maximize the networkwide reward in terms of various performance measures and subject to a budget constraint. The network-level optimization problem can be formulated as follows:

$$\max \sum_{k=1}^{n} \sum_{j \in L_k} U_{jk} X_{jk}$$

$$s.t. \quad \sum_{k=1}^{n} \sum_{j \in L_k} AC_{jk} X_{jk} \leq B$$

$$\sum_{j \in L_k} X_{jk} = 1, \quad k = 1, 2, \ldots . n$$

$$X_{jk} \in \{0,1\} \quad k = 1, 2, \ldots , n, \ j \in L_k \qquad (2\text{-}14)$$

This problem is a multi-choice knapsack problem (MCKP). The objective function is to maximize the networkwide reward, which is a function of multiple performance indicators. The first constraint is the budget constraint, which simply states that total agency cost of all the selected candidate projects must be less than the budget available. This is called the size constraint of the knapsack problem. The second constraint specifies that exactly one candidate must be selected for any bridge, where do-nothing is included as one possible alternative. This is called the choice constraint of the knapsack problem.

2.3.4.2 Formulation 2 (Only Non-Budget Constraint, MCKP)

The problem is classified as Formulation 2 when the budget constraint is replaced by a condition constraint (which means that there is still only one constraint in total). It is pos-

sible to transform the Formulation 2 problem to be very similar to Formulation 1. For example, the decision maker might want to minimize the agency costs to attain given condition targets or risk levels, while simultaneously optimizing for multiple objectives. The non-budget constraint could be, for example, that the average health index of the network be at least H_{min}, or that the average vulnerability rating of the network be no more than V_{max}. The problem can then be formulated as the following MCKP:

$$\max \sum_{k=1}^{n} \sum_{j \in L_k} U_{jk} X_{jk}$$

$$s.t. \quad \left(\frac{1}{n}\right) \sum_{k=1}^{n} \sum_{j \in L_k} H_{jk} X_{jk} \geq H_{min}$$

$$\sum_{j \in L_k} X_{jk} = 1, \quad k = 1, 2, \ldots . n$$

$$X_{jk} \in \{0,1\} \quad k = 1, 2, \ldots , n, \ j \in L_k \qquad (2\text{-}15)$$

where H_{min} is the performance target specified by the decision maker.

The first constraint is the size constraint of the knapsack problem. The second constraint is the choice constraint of the knapsack problem.

2.3.4.3 Formulation 3 (Budget and Multiple Non-Budget Constraints, MCMDKP)

The problem is classified as Formulation 3 when the decision maker's concern is to determine the best possible projects to be implemented to maximize networkwide reward in terms of various performance measures and subject to a budget constraint and one or more condition targets and/or risk-level constraints. The problem can be formulated as follows:

$$\max \sum_{k=1}^{n} \sum_{j \in L_k} U_{jk} X_{jk}$$

$$s.t. \quad \sum_{k=1}^{n} \sum_{j \in L_k} AC_{jk} X_{jk} \leq B$$

$$\left(\frac{1}{n}\right) \sum_{k=1}^{n} \sum_{j \in L_k} H_{jk} X_{jk} \geq H_{min}$$

$$\sum_{j \in L_k} X_{jk} = 1, \quad k = 1, 2, \ldots . n$$

$$X_{jk} \in \{0,1\} \quad k = 1, 2, \ldots , n, \ j \in L_k \qquad (2\text{-}16)$$

The problem is an MCMDKP. In the above example, there exist two size constraints for the knapsack problem.

2.3.4.4 Formulation 4 (Multiple Non-Budget Constraints, MCMDKP)

The problem is classified as Formulation 4 when the decision maker seeks to determine minimum possible costs to achieve

condition and risk targets, while maximizing the network-wide rewards in terms of multiple performance criteria. The formulation is as follows:

$$\max \sum_{k=1}^{n} \sum_{j \in L_k} U_{jk} X_{jk}$$

$$s.t. \quad \left(\frac{1}{n}\right) \sum_{k=1}^{n} \sum_{j \in L_k} H_{jk} X_{jk} \geq H_{min}$$

$$\left(\frac{1}{n}\right) \sum_{k=1}^{n} \sum_{j \in L_k} V_{jk} X_{jk} \leq V_{max}$$

$$\sum_{j \in L_k} X_{jk} = 1, \quad k = 1, 2, \ldots. n$$

$$X_{jk} \in \{0, 1\} \quad k = 1, 2, \ldots, n, \; j \in L_k \qquad (2\text{-}17)$$

2.4 Bridge-Level Optimization— Approach

2.4.1 Life-Cycle Cost Framework

A bridge-level optimization is conducted by generating and comparing candidates that are alternative life-cycle activity profiles and resulting performance predictions for a bridge. Each life-cycle activity profile is modeled as an infinite time series of cash flows representing various types of annual agency and user costs. Agency costs are concentrated in discrete interventions, each of which represents all the work done on the bridge in a given year.

The analytical framework evaluates each candidate for a range of performance measures. Many of the performance measures are economic, developed from a life-cycle cost analysis. Figure 7 shows a typical example pattern of life-cycle costs— that is, a "life-cycle activity profile"—as modeled in the system. The diagram shows the situation faced by an analysis conducted in 2007 to develop a program to start in 2008. The program is assumed to be finalized at the end of 2007, with the following

year being the first year of that program. For the example candidate described in the diagram, the first intervention occurs in 2011, followed by a period of inaction until 2023. The economic concepts mentioned in the diagram will be carefully defined in the remaining sections of this chapter, but can be summarized in the form of life-cycle cost components, as shown in Table 5. Also important in the diagram are several time intervals and milestones that will be significant in the discussion that follows:

- **First waiting period.** In the given example, the period spanning the start of 2008 to the start of 2011 is the first waiting period, the time interval when needs build up on the structure before a first intervention can be programmed. During this period, elements on the bridge are modeled to deteriorate according to Markovian transition probabilities; those that reach their worst-condition state and are allowed to stay there incur a risk of unprogrammed or emergency work. If any functional deficiencies exist on the bridge, user costs of accident risk and/or delays may be experienced.
- **First intervention.** At the start of 2011, the agency implements an intervention that may address some or all of the needs that have built up. The element condition is improved, and functional deficiencies may be corrected.
- **First rest period.** Following the first intervention is a period mandated by agency policy, when no action may be taken on the bridge. Deterioration of bridge elements continues, and failure risk costs may be incurred. If there are any functional deficiencies that were not remedied by the first intervention, then user costs may be experienced.
- **Consequent waiting periods, interventions, and rest periods.** If the program horizon is long enough, additional cycles may be modeled. For every consequent intervention that is modeled, a corresponding waiting period before, and rest period after, is also modeled. The final rest period of the analysis may extend beyond the end of the program horizon; if so, it extends the life-cycle cost analysis period
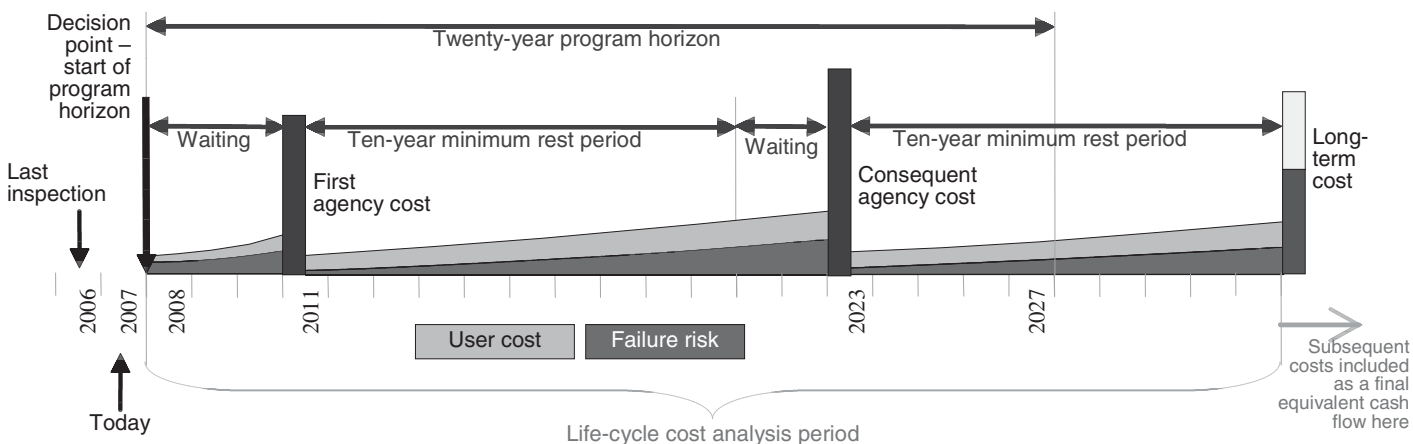
Figure 7. Example pattern of life-cycle costs.

Table 5. Life-cycle cost components.

Agency Life-Cycle Costs		User Life-Cycle Costs	
Intervention cost	Cost of programmed work, including the direct costs of preservation and functional improvements, plus the indirect cost of mobilization and maintenance of traffic	Accident cost	Expected value of user costs due to excess accident risk because of narrow bridge roadway
Failure risk cost	Possibility of "economic failure," needing emergency or unprogrammed repairs if deteriorated conditions are not remedied	Delay cost	Expected value of user costs due to height or weight restrictions
Long-term cost	Total life-cycle costs beyond the end of the model, based on ending conditions	Long-term cost	Remaining user costs beyond the end of the model if functional deficiencies are not remedied
Total agency life-cycle cost	Sum of the above, all discounted	Total user life-cycle cost	Sum of the above, all discounted

accordingly. No interventions may be programmed beyond the program horizon, so a waiting period may not extend beyond that point under the conventions we are using here.

- **Long term.** The end of a life-cycle activity profile, as modeled here, occurs when the following year brings either an intervention or the start or extension of a waiting period. We do not model the possibility of interventions beyond this point, however, but instead use less detailed models to collapse all subsequent life-cycle costs into a final cash flow at the end of the analysis period. This is similar to a salvage value analysis, except the structure is modeled as an ongoing concern in perpetuity.

Because all of the costs occur at various times in the future, they are processed in a standard engineering procedure called *net present value analysis.* Each cost item is discounted (i.e., reduced in value) by an amount that depends on how far in the future it occurs. Naturally if a cost needs to be incurred, we prefer to put it off as long as possible, because then it matters less to us. The discount factor represents how much less it matters for each year that we can delay the cost.

Discounting makes the analysis relatively insensitive to costs that occur far in the future. The effect is enhanced by the final rest period, where agency policy mandates that a fixed period, usually 10 years, must elapse before any further intervention costs can be incurred. This further reduces the sensitivity of the model to long-term costs.

The essential decision to be optimized in the bridge-level analysis is the scope and timing of the first intervention. When multiple candidates are defined for a bridge, they differ in terms of the first intervention. Timing of the first intervention determines the length of the first waiting period, which may vary from zero to the full length of the program horizon. Consequent interventions are forecast for programming and for life-cycle cost analysis, but are not the subject of decision making by the bridge maintenance planner.

2.4.2 Discounting and Present Value

Net present value analysis is used to compare costs occurring at different times in the bridge life cycle. Cash flows that occur in the future are discounted to a lower value when compared with cash flows that occur today to reflect the fact that cash received today is more valuable (i.e., less risky) than cash received in the future. There is no standard value for discount rate; what is most important is to select a reasonable rate consistent with agency policy and use it consistently across all asset types managed by the agency.

The discount rate, α, is based on the forecast real interest rate (i.e., the interest rate with inflation removed). It is calculated as follows:

$$\alpha = \frac{1}{1+int} \tag{2-18}$$

where *int* is the real interest rate.

Although it is not, in principle, required that inflation be removed from a life-cycle cost analysis, it is normally done for simplification. Inflation is less predictable than other economic inputs to the analysis, and it does not have a material effect on the results unless different cost factors are modeled to inflate at different rates. Including inflated unit costs at every point of input of economic data would complicate the models considerably, so it is not recommended. Certain conventions in the life-cycle cost analysis govern the length of discounting:

- Intervention costs and long-term costs occur at the beginning of the implementation year.
- User costs and failure risk costs occur at the beginning of every year.
- All costs are discounted to the beginning of the first year of the program.

Table 6 shows an example of a discounted cash flow analysis.

Table 6. Example of a discounted cash flow analysis.

	Base year	2007
	Discount rate	0.95

Program year	Initial cost	User cost	Failure risk cost	Long-term cost	Total cost	Discount	Discoun-ted sum
2007		30789	1245		32034	1.000	32034
2008		30978	1386		32364	0.950	30746
2009		31167	1425		32592	0.903	29414
2010		31358	1556		32914	0.857	28220
2011	109425				109425	0.815	89127
2012					0	0.774	0
2013					0	0.735	0
2014					0	0.698	0
2015					0	0.663	0
2016					0	0.630	0
2017				134711	134711	0.599	80656
					Net present value		290198

2.4.3 Optimization Framework

The decision to be made by the bridge maintenance planner has two dimensions: scope and timing. The planner must decide how long the first waiting period should be and what kind of intervention to undertake. Naturally, if the waiting period is extended, deterioration will continue and the scope and cost of the first intervention will likely increase. So, scope and timing are interrelated. If the program horizon is long enough, the decision becomes a multi-stage process. After the first intervention and its rest period, the need may arise for another intervention. The scope and timing of the second intervention depend on the decisions made about the first intervention. So, we end up with a tree-like structure of decisions, as shown in Figure 8.

Each path through the tree is a candidate and its life-cycle activity profile. The topmost path is the base case, against which all other paths are compared. Each path is evaluated for life-

Typical Interventions
- ○ Do nothing
- ◐ Rehabilitate
- ● Replace
- ⬤ Residual

Typical Candidates

LCAP -
Life Cycle Activity Profile
●→●→●

Base case LCAP: do nothing each year
○--→○--→○

This simple example is for a 15-year program horizon with decision points at 5-year intervals with a minimum of 10 years between projects. The actual model will have 1-year intervals, a potentially longer horizon, and potentially more different types of projects, including user-defined projects.

Figure 8. Tree-like structure of decisions.

cycle costs and other performance measures, summarized as a utility function. The best path is the one with highest utility—for example, the bold-faced path involving interventions in year 0 and year 15.

As noted in the diagram, this is a simplified picture that becomes much larger when we consider 1-year decision intervals and the possibility of user-defined candidates. Therefore, a need was realized for an organized approach to generate and evaluate alternatives that reduce the complexity and execution time.

2.4.3.1 Recursive Approach

The familiar tool of recursion gives us a way to divide the problem into smaller parts that are easier to analyze and require less computation. This method is familiar because a variation of it is used in the Pontis network optimization, where the assumption of a steady state is exploited in order to say that each stage of the decision is just like every other stage. So, instead of a long time series of computations, the model is reduced to a single stage plus an extra equation representing the steady-state assumption.

For the bridge-level problem, we cannot reduce the problem size so far, because a steady state does not exist: conditions do not stay the same from year to year. Yet, we can exploit structural similarities between one stage and the next. Setting aside rest periods, where the only allowed intervention is do-nothing, the decision points during a waiting period share the following common features:

- They all start with the same structure of forecast initial conditions, either at the start of the program horizon or at the end of the previous rest period.
- They all have the same set of alternative types of interventions to be considered: do-nothing, rehabilitation, or replacement. (For the first intervention, custom candidates are also considered.)
- The choice of intervention approach in every year is based on the same utility function.
- The structure of future consequent interventions is the same, differing only in how soon the detailed simulation of interventions is replaced by a less detailed long-term cost model.

With these stipulations, we can apply a recursive approach, as described in Figure 9. The optimization works in either or both of two modes:

- **Optimizing,** where all program years are investigated choosing the one with highest utility, or
- **Worst tolerable performance,** where an intervention is considered only when one or more performance measures fail to meet performance thresholds.

If both optimization methods are chosen, the algorithm stops as soon as a performance threshold is reached, but it may select a candidate in an earlier year if the latter has a higher utility.

A key aspect of the diagram is that whenever the algorithm identifies a feasible intervention, it simulates a full rest period of deterioration and then calls another copy of the same algorithm to analyze the situation at the end of the rest period. This recursion continues until finally the rest period meets or exceeds the end of the program horizon. At that point, the algorithm simply computes the long-term residual cost and does not call itself again.

The algorithm has two primary loops, one through program years (ProgYears) and one through candidates (rehabilitation and replacement). In a 10-year program horizon with a 10-year rest period, there would be up to $10 \times 2 = 20$ passes through the center of the loop, which includes scoping, forecasting to the end of the rest period, and calculation of utility. There would be no recursive calls to the algorithm in this short program horizon.

If the program horizon is 20 years with a 10-year rest period, then the identification of each feasible intervention is followed by one recursive call to the algorithm to analyze what happens 10 years or more after the intervention. In the second round, the start year ranges from 11 to 20, depending on the year of the first intervention, leaving 10 years to 1 year, respectively, to analyze up to the end of the program horizon. The second round therefore adds an average of $5 \times 2 = 10$ inner loops for each outer loop in the first round. The total number of passes in this case is $20 \times 10 = 200$. So if each pass takes 0.5 milliseconds, the total execution time would be 0.1 second.

The actual number of passes through the inner loop is much smaller than 200 for several reasons:

- If the procedure is run in worst tolerable preference mode without optimizing, usually only one program year is fully analyzed. This cuts an entire loop out of the process for each round through the algorithm. This would be the recommended procedure for very long program periods, such as 30 years, or for very short rest periods.
- For several reasons, an intervention might not be feasible (i.e., it might have negative utility relative to the do-nothing intervention). For example, replacement of a bridge that is already new and in perfect condition would yield negative life-cycle benefits and probably negative utility and therefore is not considered. The algorithm skips over the recursion and the utility computation if the intervention is not feasible.
- The maintenance planner may explicitly select or block a candidate for consideration as the first intervention.

Since the optimization is instantaneous in most practical applications, it is run automatically when a bridge is loaded in the bridge-level decision-support tool.
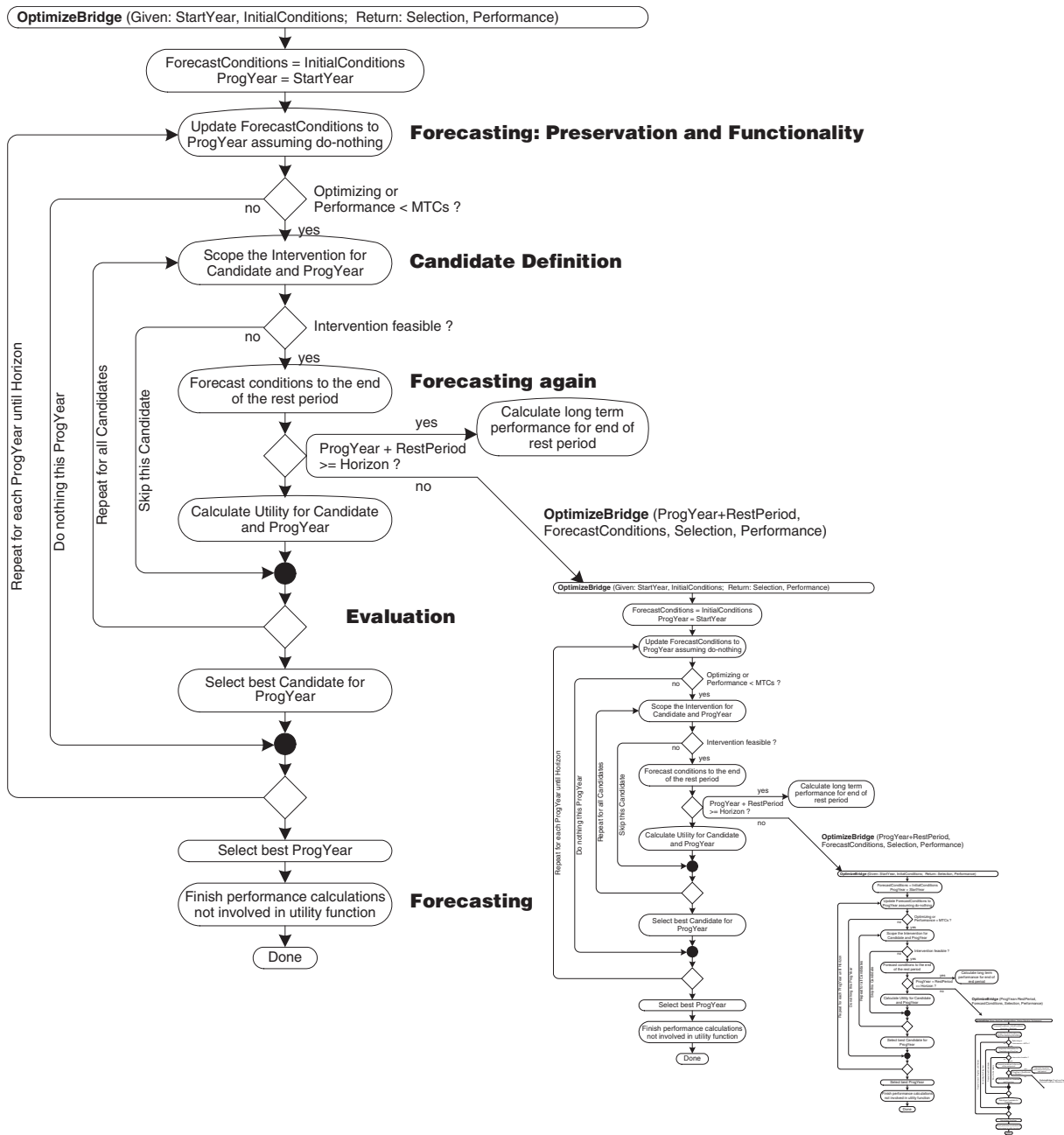
MTC = minimum tolerable conditions.

Figure 9. Recursive optimization approach.

2.4.3.2 Submodels

The three primary phases that feed into the optimization framework—forecasting, candidate definition, and evaluation—are divided into submodels as follows:

Forecasting
– Preservation
   Deterioration
   Action effectiveness
   Cost estimation

   Economic failure
   Long-term cost
– Functionality
   User costs and traffic growth
   Accident risk
   Vertical clearance
   Load capacity
   Truck detour cost
Candidate definition
– Do-nothing

– Auto maintenance, repair, rehabilitation, and improvement (MRR&I) and custom candidates

Action classification

Scoping

Scale feasibility

Indirect costs

– Replacement

Evaluation

2.4.3.3 Differences from the Pontis Approach

The bridge-level model addresses some of the same issues as the Pontis program simulation and project planning capabilities and uses the same basic inputs: AASHTO CoRe Elements, Markovian deterioration and action effectiveness models, MR&R action types, the concepts of element failure and long-term costs, level-of-service and design standards for functional improvements, and user costs. However, the modeling requirements are significantly different from Pontis, leading to a different analytical approach. Among the major sources of differences are the following:

- The need to optimize both the scope and timing of interventions—thus, the need for a two-dimensional solution space of alternatives to be presented and manipulated.
- The need to accommodate multiple objectives at the same time.
- The need for both automated and manual generation of candidates, which are fully evaluated, compared, and optimized within the analytical framework.
- The ability for the engineer to interact with candidates and make bridge-level decisions.
- The need for a robust and fast trade-off analysis at the network level.
- The requirement that the model be operable even by agencies that do not have a Pontis license.

Together, these requirements point to a simpler analytical framework, organized in a manner different from Pontis. Some of the most significant differences between the bridge-level model approach and Pontis 4 are the following:

- With a bridge-level model, we address only one bridge at a time. We do not attempt to duplicate the extensive project development functionality provided in Pontis 4, but instead use a framework that can eventually (perhaps in a future AASHTOWare project) feed into the existing Pontis 4.0 project development system.
- We maintain an array of separate candidates representing different combinations of scoping and timing alternatives. Some of these candidates are generated by automated models, and some by the user. Candidates are never a combination of model-generated and user-generated; therefore, it is always clear which outputs are influenced by user inputs and which are not.
- All types of actions, even those involving multiple elements or those created by a user, are evaluated for life-cycle cost and performance measures and are fully represented in the optimization model.
- Since the maintenance planner has the opportunity to review and modify each bridge individually, we don't use as many automated simulation rules to generate candidates. Instead, we provide a robust means of defining new types of actions that are easy for the engineer to understand and use. Where there is complexity in the analytical framework, there is complexity in the predictive models and not in the decision-making models.

In addition, it is important to remember that the proposed framework has a clear separation between bridge level and network level. The bridge-level analytical models are almost always run one bridge at a time, rarely in large batches as in the Pontis program simulation. So, computational speed is not as much of an issue here as it is in Pontis. The analytical choices favor simple, flexible presentation and manipulation by the maintenance planner, rather than having automated processes that anticipate every possible variation in real-world projects.

C H A P T E R   3

Findings and Applications

3.1 Developing Multi-Criteria Parameters

3.1.1 Introduction

Chapter 2 presented the multi-criteria methodologies that can be applied to the bridge management decision-support problem. A combination of these methods was used in designing a questionnaire from which relative weights and utility/value functions were developed. The research effort aimed at applying the theoretical concepts from value and utility theory in a practical manner to simplify the process of assessing the relative weights, values, and utility functions. This questionnaire was administered to the NCHRP Project 12-67 panel of bridge experts. The purpose of the questionnaire was twofold: (1) to select and recommend the most appropriate multi-criteria methods for the bridge problem based on the panel response in terms of ease of understanding and based on the analysis of questionnaire results and (2) to develop a set of default weights, values, and utility functions that could be used by the agencies that do not wish to go through the assessing process. The present chapter presents the findings and applications of the various theoretical aspects of the multi-criteria decision-making framework. Specifically, this chapter provides details on the expert questionnaire, analysis of the questionnaire results, and final recommendations for the multi-criteria methodologies for the bridge decision-support problem.

3.1.2 Assessing Relative Weights

The relative weights were developed using two alternative approaches: the direct questioning approach and the analytic hierarchy process approach. The weights were aggregated across questionnaire participants using the average values of their responses. The weights were developed across all levels of the hierarchy of performance criteria. On the second day of the panel meeting, participants were given a chance to review their responses from the previous day's questionnaire. This is

a standard Delphi technique used to encourage questionnaire participants to arrive at a consensus. Using the revised (day 2) responses, the relative weights of the bridge performance measures were recomputed, as presented in Figures 10 through 13. For each performance measure, the reported standard deviation reflects the level of agreement among the participants regarding the relative weight of that performance measure. The smaller the standard deviation, the higher is the level of agreement. The smaller standard deviation values for day 2 show that there was a considerably higher level of agreement among the panel members on that day compared with on day 1.

Tables 7 and 8 present the recommended default relative weights for overall goals and individual performance measures, respectively. For the set of performance measures representing condition preservation, the use of the analytic hierarchy process approach was favored by the panel, and analytic hierarchy process weights were therefore adopted for those performance measures. For all other sets of performance measures, the direct questioning approach weights (obtained on day 2) were selected because of smaller variances and a higher level of agreement among participants. Because the higher level of agreement (i.e., smaller variances) in the results and the simplicity of its methodology, the direct questioning approach is generally recommended for developing the relative weights of performance measures.

3.1.3 Development of Value Functions

Value functions for performance measures were developed using two methods: the direct rating method and the midvalue splitting technique. The direct rating method is appropriate for vulnerability ratings because these ratings have very few possible levels. The midvalue splitting technique is appropriate for all other performance measures because they have many possible levels. The aggregation of data across the questionnaire participants was carried out using statistical regression. For each performance measure, nonlinear regression was carried

Day-1

Day-1

Day-2

Day-2
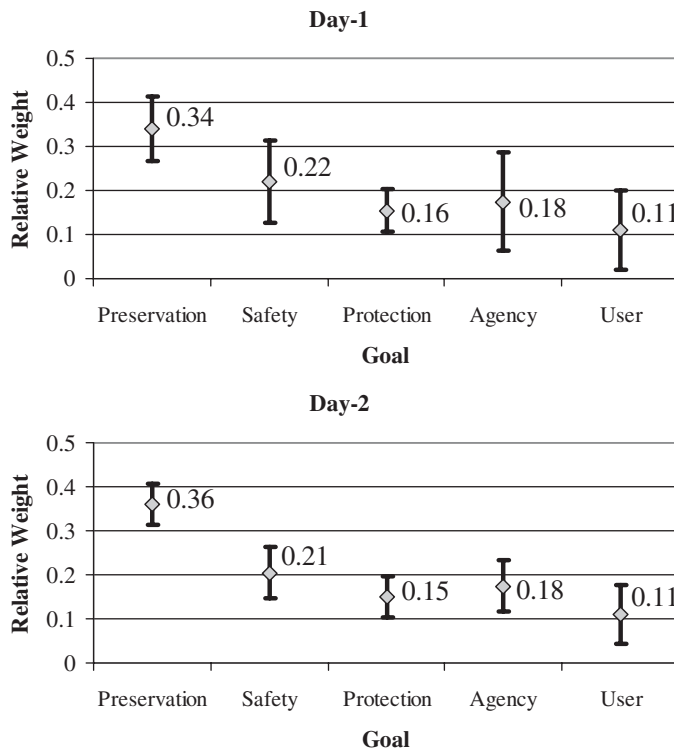
Figure 10. Relative weights across overall goals: Direct questioning approach.
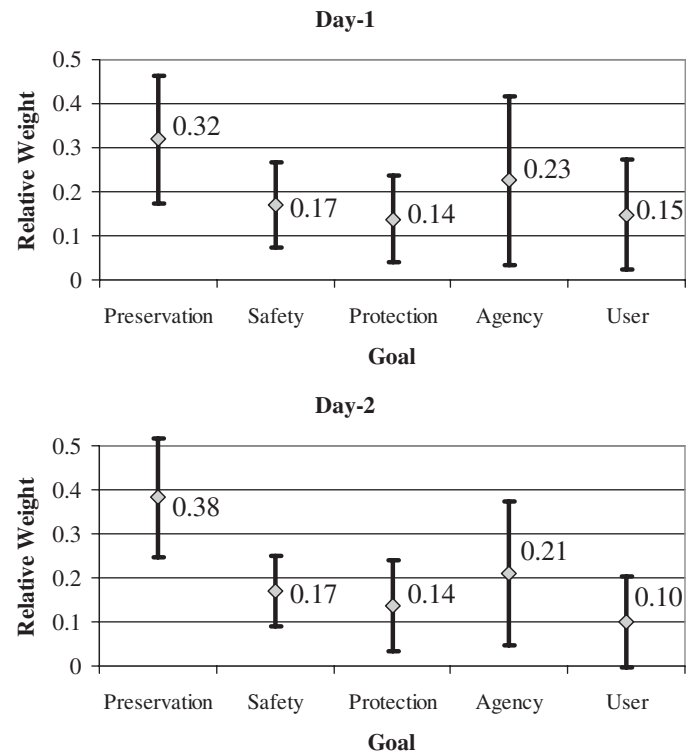
Figure 11. Relative weights across overall goals: Analytic hierarchy process.

out using one of the alternative mathematical (i.e., functional) forms: logistic and concave-shaped forms to find the best fit curve. Figures 14 through 17 present the plots of observed data, best-fitting curves and equations, and R^2 values for the performance measures.

3.1.4 Multi-Criteria Utility Functions

In the presence of uncertainty, the gamble method is used to derive the functional form of a multi-criteria utility function. This method establishes the scaling constants for a given set of performance measures. The scaling constants are aggregated by simple averaging and then summed up for each set of performance measures. A hypothesis test is then carried out to ascertain whether the sum is statistically equal to 1. A sum of 1 implies that an *additive utility function* is appropriate for the given multi-criteria problem (i.e., the weighted values or utilities of the performance measures can be added together to obtain the overall performance). However, if the sum is not equal to 1, then a *multiplicative utility function* is more appropriate (Keeney and Raiffa 1976). For purposes of the statistical test, the null and alternate hypotheses, H_0 and H_1, respectively, are formulated as follows:

- H_0: The sum of average constants = 1 (that is, an *additive* utility function is appropriate).

- H_1: The sum of average constants $\neq 1$ (that is, a *multiplicative* utility function is appropriate).

Table 9 presents the t-statistics, along with the results of hypothesis tests for the overall goals and performance measures.

With the exception of condition ratings, all sets of performance measures had their sum greater than 1, thus rejecting the null hypothesis and suggesting that a multiplicative functional form is more appropriate for their utility functions. However, the method becomes difficult for the overall goals and condition ratings because the method consists of comparing groups of performance measures. Also, because the measures in the condition ratings set (NBI ratings, health index, and sufficiency rating) overlap to a much larger extent, it can be argued that a multiplicative functional form would be more appropriate.

Thus, the multi-criteria utility function, in the presence of uncertainty, is of multiplicative form for all sets of performance measures. This function is given as follows (Keeney and Raiffa 1976):

$$ku(z_1, z_2, \ldots, z_p) + 1 = \prod_{i=1}^{p} [kk_i u(z_i) + 1] \qquad (3\text{-}1)$$

for p criteria $Z_1, Z_2, \ldots, Z_p$. The value of k is determined by evaluating the above equation for a candidate that is best in

DAY 1

DAY 2

Relative Weights - Condition Preservation

Relative Weights - Condition Preservation

0.28   0.470   0.26

0.325   0.445   0.23

Relative Weights - NBI Ratings

Relative Weights - NBI Ratings

0.349   0.348   0.303

0.330   0.340   0.330

Relative Weights - Traffic Safety

Relative Weights - Traffic Safety

0.595   0.405

0.570   0.430

Relative Weights - Protection from Extreme Events

Relative Weights - Protection from Extreme Events

0.345   0.29   0.18   0.185

0.385   0.265   0.205   0.145

Figure 12. Relative weights across individual performance measures: Direct questioning approach.

DAY 1

DAY 2

Relative Weights - Condition Preservation

Relative Weights - Condition Preservation

Relative Weights - NBI Ratings

Relative Weights - NBI Ratings

Relative Weights - Traffic Safety

Relative Weights - Traffic Safety

Relative Weights - Protection from Extreme Events

Relative Weights - Protection from Extreme Events
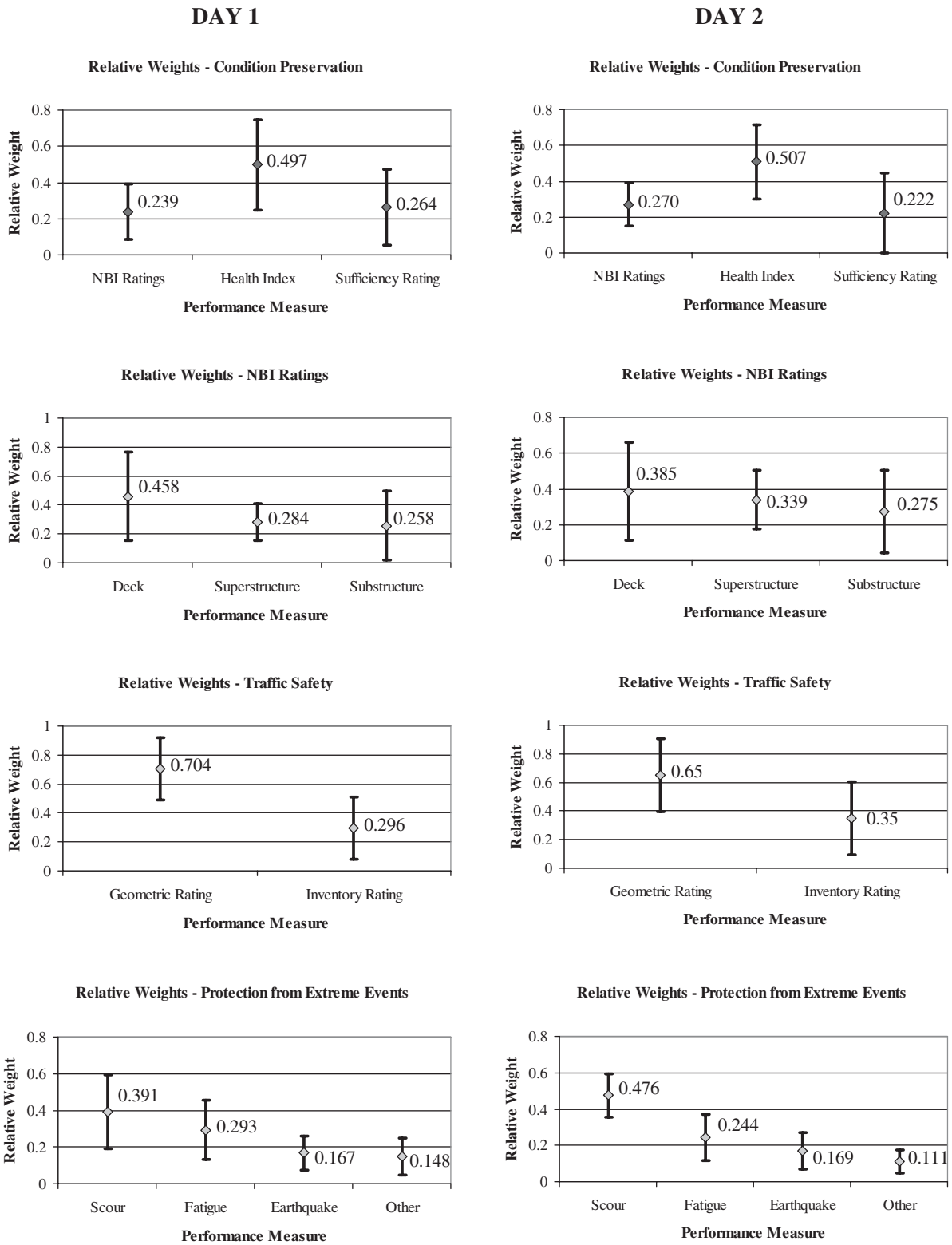
Figure 13. Relative weights across individual performance measures: Analytic hierarchy process.

Table 7. Recommended relative weights: Overall goals.

Overall Goal	Relative Weight
Bridge Preservation	0.360
Safety	0.205
Protection from Extreme Events	0.150
Agency Cost	0.175
User Cost	0.110
TOTAL	**1.000**

terms of all criteria—that is, by solving the following equation numerically:

$$k + 1 = \prod_{i=1}^{P} [kk_i + 1] \qquad (3\text{-}2)$$

Table 10 presents the scaling constants k_i established by the gamble method for individual performance measures.

3.1.5 Developing Single-Criterion Utility Functions

The certainty equivalent method is used to derive the utility functions for individual performance measures (i.e., the single-criterion utility functions). For a given performance measure, the utility function not only captures the decision maker's preferences for various levels of that criterion but also describes the decision maker's risk attitudes toward that criterion.

In developing utility functions, the *value* of the certainty equivalent for each performance criterion and each participant (i.e., questionnaire respondent) is first calculated using the previously derived value functions. The values are then averaged across participants. This is compared with the expected value of the gamble (which is 50 in our case because there is a 50% chance of best and worst levels). The mathematical forms of individual utility functions are then derived as follows (Keeney and Raiffa 1976):

- For average value of certainty equivalents > 50, $u(x) = e^{cv(x)}$, $c > 0$

- For average value of certainty equivalents = 50, $u(x) = v(x)$
- For average value of certainty equivalents < 50, $u(x) = -e^{-cv(x)}$, $c > 0$

where $u(x)$ is utility function and $v(x)$ is value function for criterion x.

A hypothesis test was carried out for each performance measure to test if the average of values of certainty equivalents is statistically different from 50. The results and conclusions for each performance measure are shown in Table 11. As can be seen in the table, the utility functions for health index, sufficiency rating, geometric rating, inventory rating, operating rating, fatigue—steel, and earthquake vulnerability are the *same* as their respective value functions. The rest of the performance measures in the table are further analyzed to evaluate the constant c. To do this, the utility of certainty equivalent is equated to the expected utility of the gamble using the functional form given above (Keeney and Raiffa 1976). The equation is then numerically solved to compute c. Results are shown in Table 12.

The utility function is then scaled from the lowest utility (0) to the highest utility (100). The utility functions and value functions are shown in Figures 18 and 19.

The utility functions differ from the value functions in terms of curvature. Mathematically, the effect of the specific functional form is to convert the concave function into an S-shaped curve. However, as seen from Figures 18 and 19, the curvature of S-function is small and the function is practically linear. Hence, the utility functions for these performance measures can be assumed to be linear. The final results of the utility functions of performance measures are shown in Table 13.

3.2 Network-Level Optimization—Solution Methods

3.2.1 Introduction

This section describes the approach used for solving the optimization problem. The MCMDKP is considered as hard in

Table 8. Recommended relative weights: Individual performance measures.

Overall Goal	Performance Measure	Relative Weight	Total
Condition Preservation	NBI Ratings	0.271	1.000
	Health Index	0.507	
	Sufficiency Rating	0.222	
NBI Ratings	Deck	0.330	1.000
	Superstructure	0.340	
	Substructure	0.330	
Traffic Safety	Geometric Rating	0.570	1.000
	Inventory Rating	0.430	
Protection from Extreme Events	Scour	0.385	1.000
	Fatigue	0.265	
	Earthquake	0.205	
	Other	0.145	

Figure 14. Value functions: Deck conditioning rating, superstructure condition rating, substructure condition rating, and culvert condition rating.

Figure 15. Value functions: Health index, sufficiency rating, geometric rating, and inventory rating.

$R^2 = 0.83$

$V(OR) = 134.13 \times (1 - e^{-0.014x})$

$R^2 = 0.95$

$V(SVR) = 121.76 \times (1 - e^{-0.43(x-1)})$

$R^2 = 0.93$

$V(FCVR) = 137.03 \times (1 - e^{-0.33(x-1)})$

$R^2 = 0.93$

$V(FSVR) = 125.35 \times (1 - e^{-0.40(x-1)})$
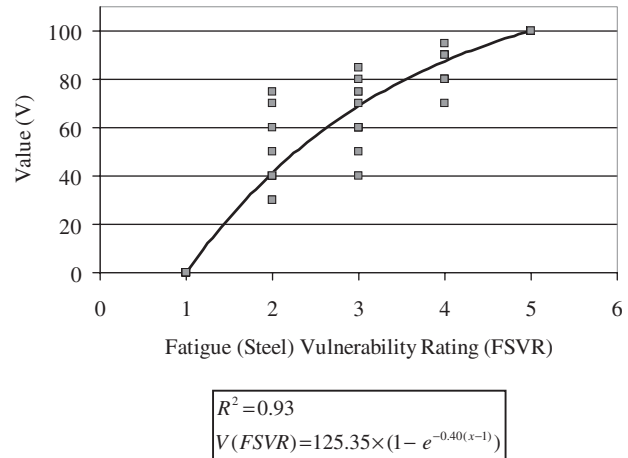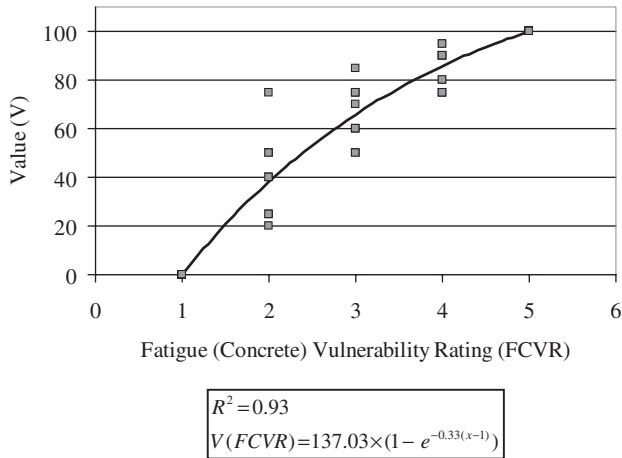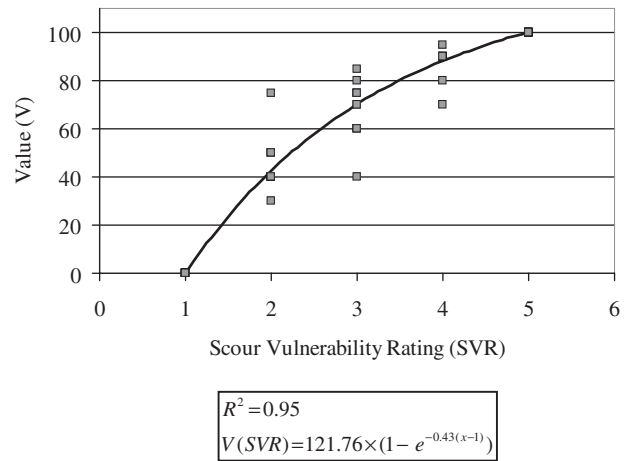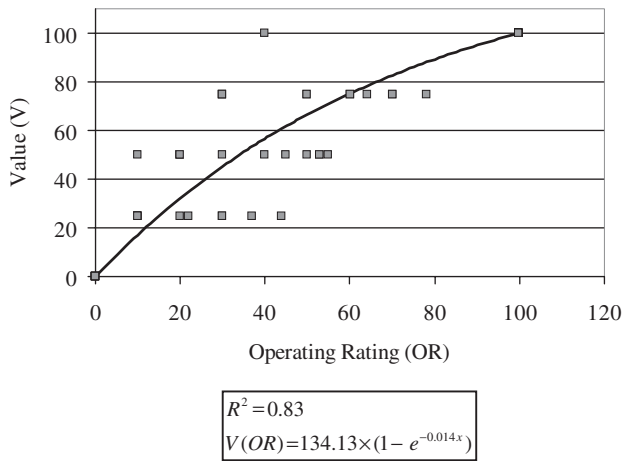
Figure 16. Value functions: Operating rating, scour vulnerability rating, fatigue (concrete) vulnerability rating, and fatigue (steel) vulnerability rating.

$R^2 = 0.95$

$V(EVR) = 130.57 \times (1 - e^{-0.36(x-1)})$

$R^2 = 0.94$

$V(OVR) = 129.5 \times (1 - e^{-0.37(x-1)})$
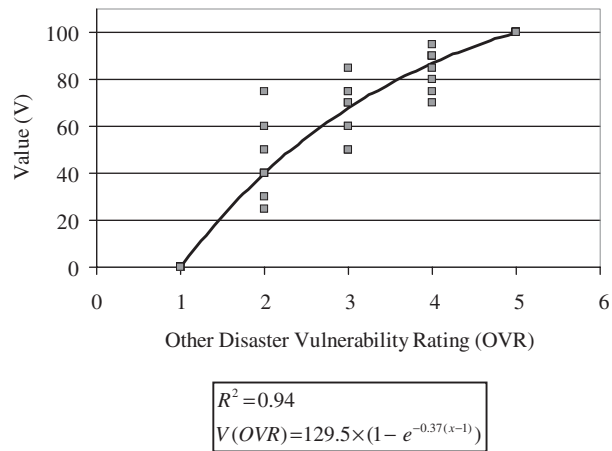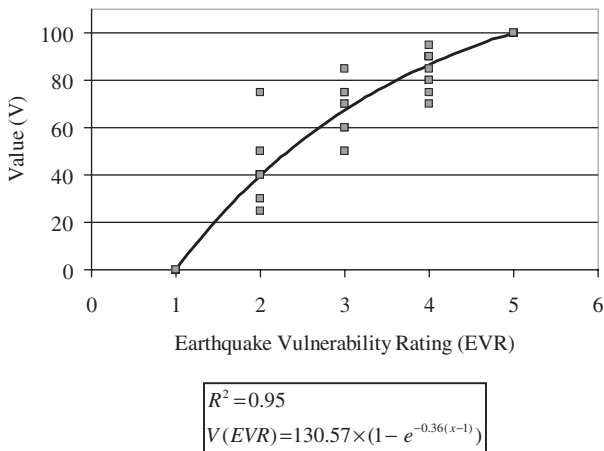
Figure 17. Value functions: Earthquake vulnerability rating and other disaster vulnerability rating.

Table 9. Gamble method results.

Set of Performance Measures	Avg. Sum	Std. Sum	t-stat	Threshold t-value (t_{95})	Conclusion
NBI Ratings	1.205	0.231	2.80	2.26	Reject H_0
Traffic Safety	0.885	0.153	-2.38	2.26	Reject H_0
Protection from Extreme Events	1.445	0.504	2.79	2.26	Reject H_0
Overall Goals	1.452	0.490	2.92	2.26	Reject H_0
Condition Ratings	1.285	0.770	1.17	2.26	Do not Reject H_0

the sense that no known deterministic polynomial algorithm exists. This means that the time requirement for the optimal solution grows exponentially with the size of the problem.

As an example of exponential growth, an optimization algorithm that takes 1 minute on a test dataset of 1,000 bridges might take roughly 3 days on an inventory of 12,000 bridges (the average of all states inventories) and roughly 45 days on an inventory of 50,000 bridges (the largest state inventory, Texas). In contrast, a polynomial algorithm's times for the same problem might be 1 minute for 1,000 bridges, 2.5 hours for 12,000 bridges, and 1.7 days for 50,000 bridges. For a suitably defined problem, solution times can be even faster than that of the polynomial. If the solution method could be reduced to simple sorting, it would have $n(\log n)$ execution time that would run in 1 minute for 1,000 bridges, 13 minutes for 12,000 bridges, and 31 minutes for 50,000 bridges. Obviously, this faster behavior is preferred because it allows the user of the system to quickly investigate the consequences of several policy alternatives.

There are two classes of methods that exist to solve the network optimization problem: exact methods (or algorithms) and heuristics. Exact methods are guaranteed to arrive at the optimal solution but are typically associated with lower computational speeds. In contrast, heuristic methods strive to achieve "good" approximate (i.e., near optimal) so-

Table 10. Scaling constants.

Performance Measure		Scaling Constant (k_i)
NBI Ratings	Deck condition rating	0.34
	Superstructure condition rating	0.45
	Substructure condition rating	0.42
Traffic Safety	Geometric rating	0.40
	Inventory rating	0.49
Protection	Scour vulnerability rating	0.38
	Fatigue vulnerability rating	0.42
	Earthquake vulnerability rating	0.37
	Other vulnerability rating	0.29
Overall	Preservation of bridge condition	0.55
	Traffic safety enhancement	0.25
	Protection from extreme events	0.24
	Agency cost minimization	0.27
	User cost minimization	0.14
Preservation	NBI condition ratings	0.42
	Health index	0.53
	Sufficiency rating	0.34

lutions rather quickly. Therefore, there is a trade-off between the accuracy and computational speed of the solution methods. The largest network we are dealing with has 50,000 bridges. Each bridge could have as many as five possible interventions, including the do-nothing option. This means that there are about a quarter million items, or 0-1 variables, for the knapsack problem. This is a huge integer programming problem.

The research team believes that heuristics are most appropriate for the network optimization problem because the exact methods would not have practical solution times on full-size bridge inventories. We have explored the best methods in the literature and tailored these methods to suit our specific problem. The research team focused on balancing the mathematical precision of these methods with practical issues such as computational speeds. The subsequent sections summarize the most promising methods we propose for the network optimization problem.

3.2.1.1 Using Domain Knowledge

The optimization problem can be made more computationally tractable by using domain knowledge, such as partitioning the network problem. Bridges could fall into one of the following three groups: (1) bridges that have no deficiencies and need no action, (2) bridges with severe deficiencies where only replacement or some other specific treatment is suitable, and (3) the in-between bridges. This would basically reduce the number of bridges with multiple choices, and the optimization methods will then be focused on the third group of bridges. The optimization process will also take advantage of screening methods to further reduce the problem size by identifying economically unattractive candidates.

3.2.1.2 Selection of an Appropriate Solution Method

The performance of a solution method can be evaluated on the basis of

- Accuracy of the solution method (i.e., how close is the solution to the true optimal solution?),
- Computational speed (i.e., how long does it take to solve?),

Table 11. Certainty equivalent results.

Performance Measure	Avg. of Values of Cert. Equiv.	Std. dev.	*t*-stat	*t*-95 (*df* = 10)	Conclusion	Functional Form
Deck Condition	69.31	7.75	8.26	2.23	Reject H_0	$u(x) = e^{\wedge}(c\ v(x))$
Superstructure Condition	66.62	8.28	6.65	2.23	Reject H_0	$u(x) = e^{\wedge}(c\ v(x))$
Substructure Condition	64.27	9.69	4.88	2.23	Reject H_0	$u(x) = e^{\wedge}(c\ v(x))$
Culvert Condition	64.16	8.59	5.47	2.23	Reject H_0	$u(x) = e^{\wedge}(c\ v(x))$
Health Index	38.85	20.55	-1.80	2.23	Do not reject H_0	$u(x) = v(x)$
Sufficiency Rating	50.60	15.11	0.13	2.23	Do not reject H_0	$u(x) = v(x)$
Geometric Rating	54.16	14.04	0.98	2.23	Do not reject H_0	$u(x) = v(x)$
Inventory Rating	54.48	16.02	0.93	2.23	Do not reject H_0	$u(x) = v(x)$
Operating Rating	52.03	16.48	0.41	2.23	Do not reject H_0	$u(x) = v(x)$
Scour Vulnerability	63.50	17.91	2.50	2.23	Reject H_0	$u(x) = e^{\wedge}(c\ v(x))$
Fatigue—Concrete	66.21	16.46	3.27	2.23	Reject H_0	$u(x) = e^{\wedge}(c\ v(x))$
Fatigue— Steel	60.61	16.24	2.17	2.23	Do not reject H_0	$u(x) = v(x)$
Earthquake Vulnerability	59.07	16.36	1.84	2.23	Do not reject H_0	$u(x) = v(x)$
Other Vulnerability	63.62	16.88	2.68	2.23	Reject H_0	$u(x) = e^{\wedge}(c\ v(x))$

- Robustness of the method (i.e., how sensitive is its performance to variation in inputs?), and
- Simplicity (i.e., will the prospective users understand it well enough to use it effectively?).

The accuracy depends on the type of solution method being implemented and the problem structure. The computational speed depends on a number of factors, including the type of method (i.e., theoretical computational complexity), coding language, coding efficiency, specific problem instances (i.e., realistic datasets), network parameters and computer configuration. We aim to balance the theoretical precision and an appropriate level of practicality while selecting the solution method. This is discussed in detail and quantified in the computational experiments in Section 3.2.5.

3.2.2 Incremental Utility-Cost (IUC) Ratio Heuristic

The general knapsack problem is a type of integer program that is difficult to solve mathematically. However, we can fruitfully address this problem by carefully reducing the complexity of what we try to accomplish to focus on what is important and practical to bridge engineers. A very common example

Table 12. Values of the constant *c*.

Performance Measure	Constant c
Deck Condition	0.017
Superstructure Condition	0.014
Substructure Condition	0.012
Culvert Condition	0.012
Scour Vulnerability	0.011
Fatigue—Concrete	0.014
Other Vulnerability	0.011

occurs when we consider a single objective and single constraint, such as a budget constraint, as essential and regard the rest as targets that may or may not be met. This reduction of complexity gives us the incremental benefit-cost (IBC) method, which is used in Pontis for its program simulation and is also widely used in pavement management systems. An important aspect of the IBC method is that it produces near-optimal—and not guaranteed-optimal—solutions. It is possible in principle to take an IBC solution, investigate variations thereof, and come up with a somewhat better solution. However, the IBC method guarantees that if the solution is not optimal, the maximum amount of sub-optimality (the additional total benefit that is possible but was not found) is limited to the benefit of the largest candidate selected. For a real-size problem, this is within the margin of uncertainty in the budget constraints and other inputs, so the small sub-optimality is considered acceptable as a practical matter.

The IUC heuristic is based on concepts similar to those of the IBC method. The benefit or reward of any project can be measured in terms of the total utility, which is a function of multiple performance measures. The agency costs and reward values can then be used to compute the ratios, IUC_M. The subscript M refers to "multiple" performance measures. The combination of the performance measures selected to compute the total utility would depend on the decision maker's concern.

The theoretical appeal for the IUC heuristic is that it is based on so-called *greedy* heuristics for the knapsack problem. The classical method to solve a linear relaxation of the knapsack problem starts with arranging items in decreasing order of reward-to-weight ratios and then scanning down the list to fill the knapsack.

The IUC heuristic takes slightly different forms for different formulations of the problem. The formulation of the knapsack
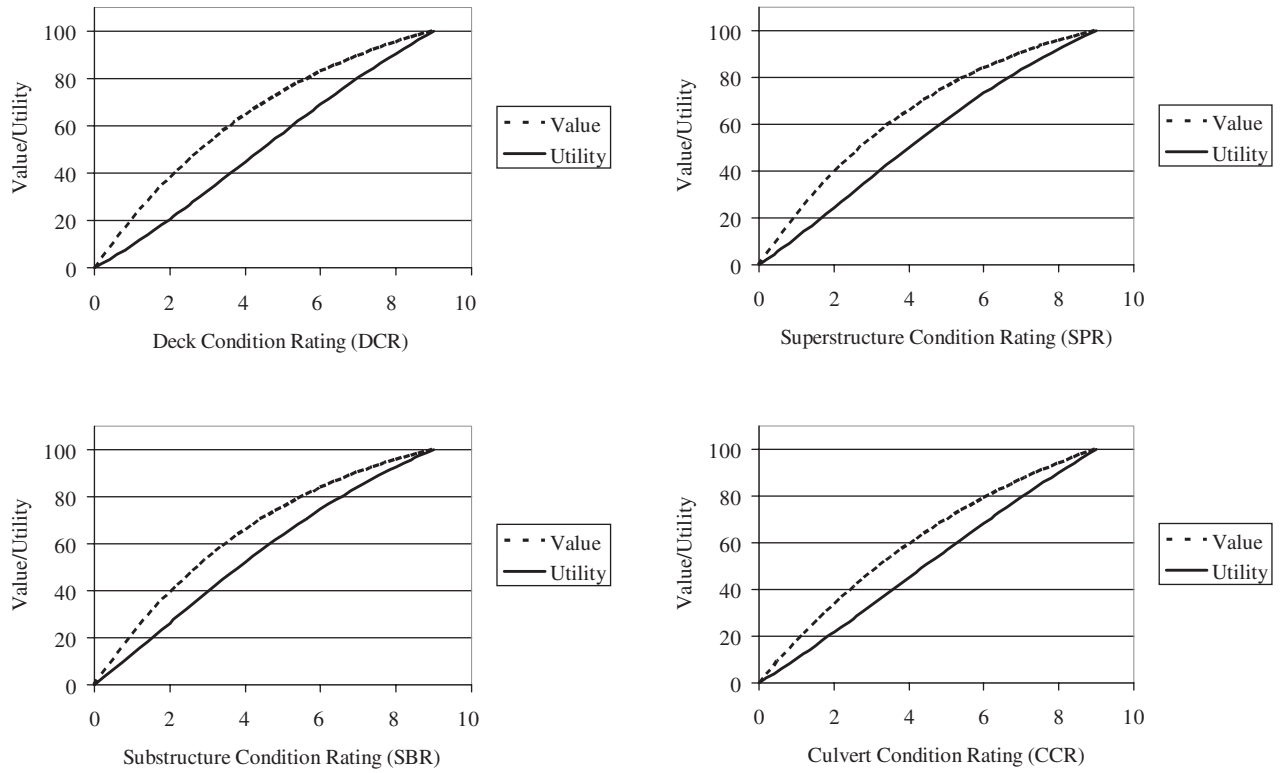
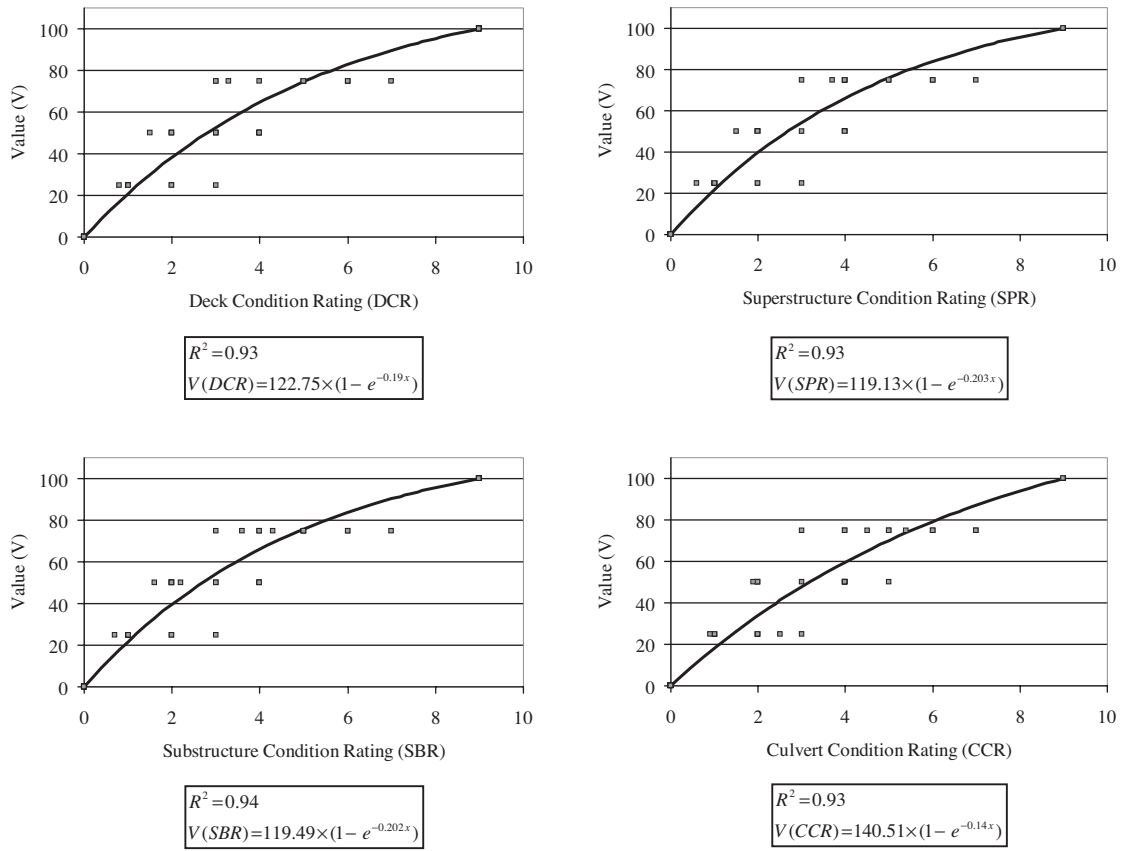**Figure 18. Utility and value functions, preliminary results: Deck condition rating, superstructure conditioning rating, substructure condition rating, and culvert condition rating.**

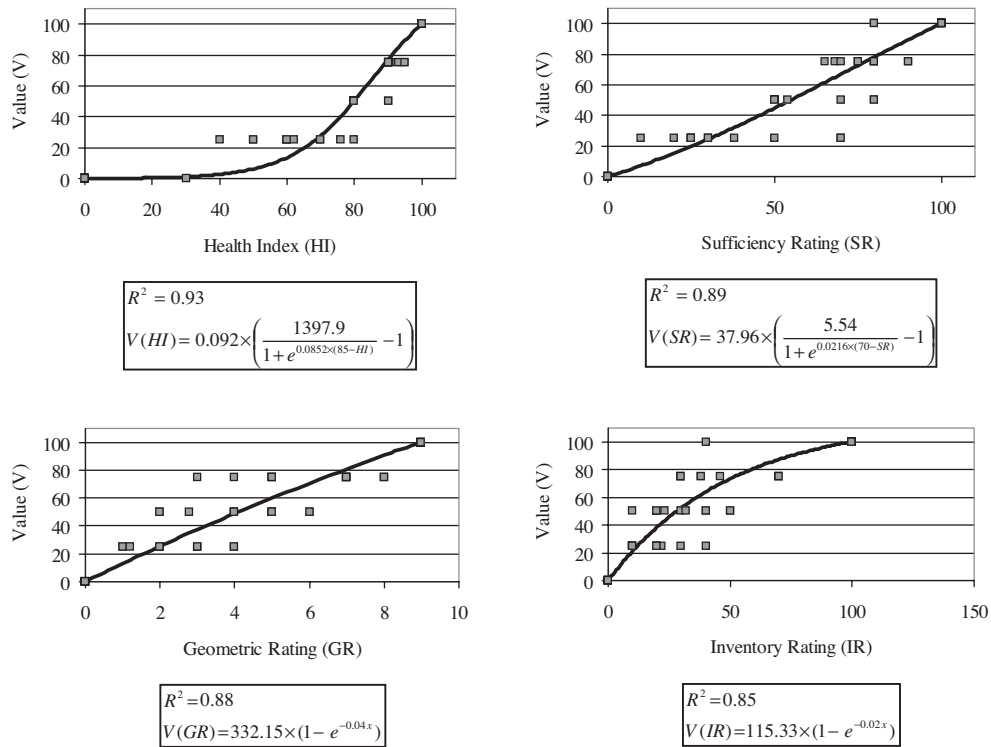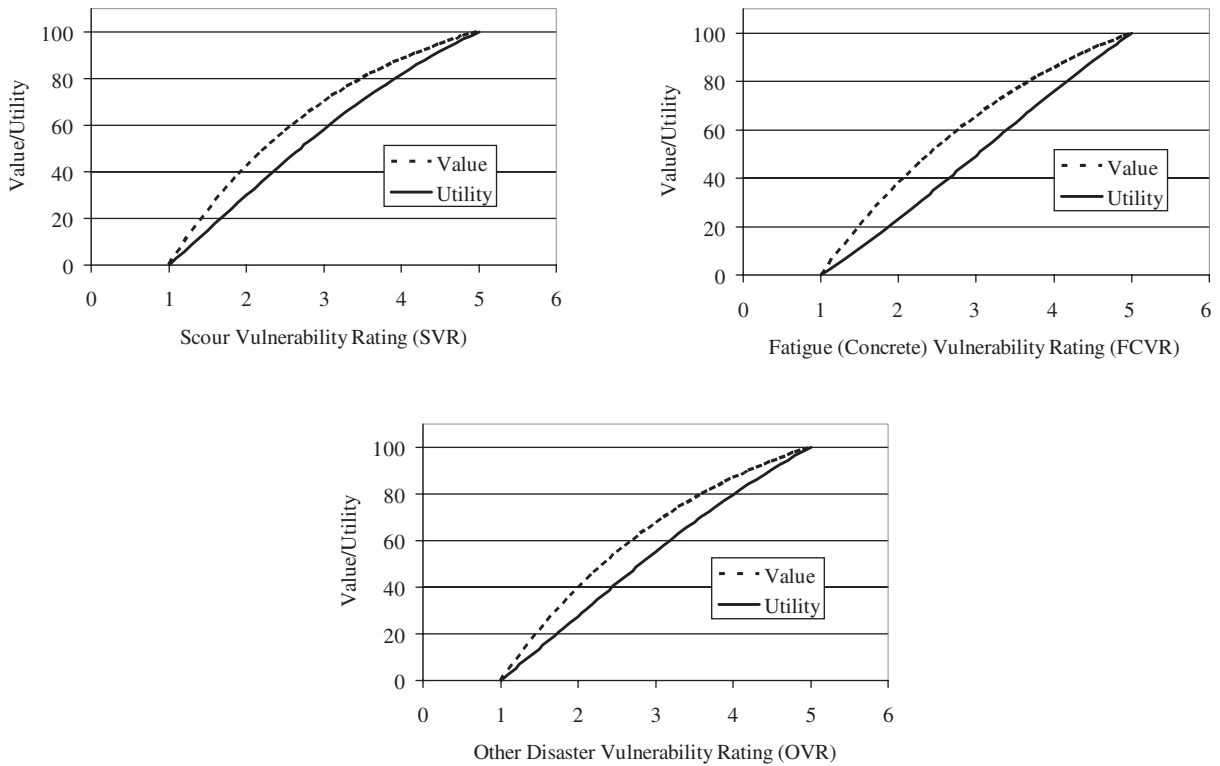**Figure 19. Utility and value functions, preliminary results: Scour vulnerability rating, fatigue (concrete) vulnerability rating, and other disaster vulnerability rating.**

Table 13. Utility functions: Final results.

Performance Measure	Utility Function
Deck Condition	Linear
Superstructure Condition	Linear
Substructure Condition	Linear
Culvert Condition	Linear
Health Index	Same as *value* function
Sufficiency Rating	Same as *value* function
Geometric Rating	Same as *value* function
Inventory Rating	Same as *value* function
Operating Rating	Same as *value* function
Scour Vulnerability	Linear
Fatigue (Concrete)	Linear
Fatigue (Steel)	Same as *value* function
Earthquake Vulnerability	Same as *value* function
Other Vulnerability	Linear

problem could contain single or multiple constraints depending on the decision maker's concern. The following subsections describe the heuristic for both single-constraint and multi-constraint scenarios. Illustrative examples are used to clarify and highlight its use in the bridge management decision-making context. The description of the heuristic starts with the economic concept of diminishing marginal returns—a concept that is fundamental to the heuristic.

3.2.2.1 Diminishing Marginal Returns

The law of diminishing marginal returns is a concept describing the economic relationships among alternative uses of the same investment capital. According to the law of diminishing marginal returns, each incremental investment produces a less-than-proportionate increase in benefits.

In our problem, each bridge has several alternative candidates with varying levels of investment and performance benefits. If funding is constrained, it is desirable to find the highest-benefit use for the money. If more funding becomes available, then additional investment can be made in the same bridges to increase the benefit. If the benefits of the various alternative candidates on a bridge are plotted against costs, the curve in Figure 20 is a typical result. When interpreting this example, "benefit" is defined as the savings in life-cycle cost of doing something, relative to the do-nothing candidate, or the increase in utility of doing something rather than doing nothing. If benefit is positive, this means that the discounted future cost savings exceeds the initial cost. Therefore, a positive benefit is desired.

If the scope of work on the bridge is upgraded from maintenance to repair, then the additional cost is $350,000 and the additional benefit is $300,000, which means that the marginal return, or IBC ratio, is 0.86. Similarly, if the scope of work is upgraded from rehabilitation to replacement, then the cost increases by $400,000 while the benefit increases by only $100,000, which means that the IBC ratio is 0.25. Under the law of diminishing returns, more expensive alternatives have progressively smaller IUC ratios. In other words, the first dollar gives the greatest benefit and the last dollar gives the smallest benefit. Therefore, in a program with a very large or unconstrained budget, the last alternative to be considered is that with the highest additional cost but the lowest additional benefit relative to the penultimate alternative. Generally, the last alternative will have the lowest IUC ratio.

To understand why this curve must always be concave downward, imagine a situation where cost of repair exceeds that of rehabilitation. If this were true, then rehabilitation would have higher benefits at lower cost, so it would always be a more economical choice.

Because of the competition that exists among a large number of candidate investments in any real bridge inventory, any
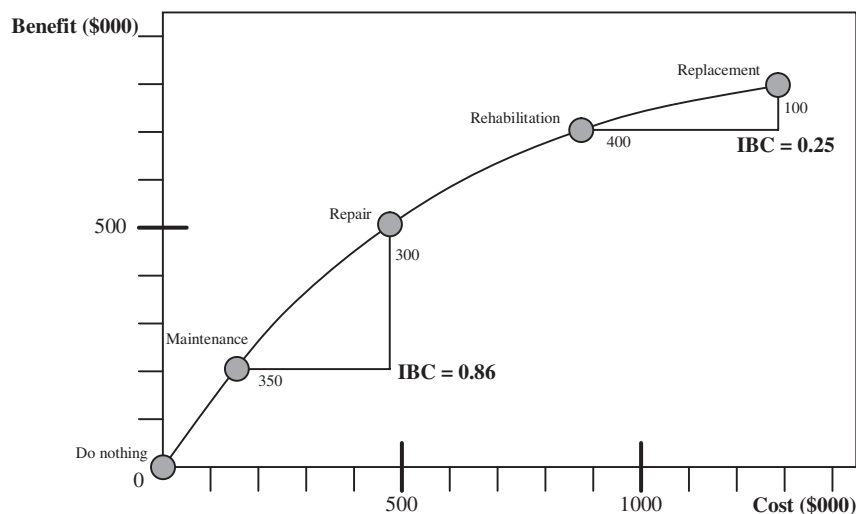
Figure 20. Diminishing marginal returns.

candidate that has benefits that are too low, or costs that are too high, to fit the diminishing marginal returns curve will be less attractive than other investments on the same bridge or other bridges. In other words, bridge maintenance projects behave like normal economic goods (rather than Giffen goods). Bridge maintenance models, as they have been developed in practice, with discounting, will practically always behave mathematically like normal goods.

3.2.2.2 Multi-Choice Knapsack Problem, Single Constraint

With only one constraint, on budget or performance (but not both), the heuristic maintains a list of investment candidates sorted by the ratio of change in utility, divided by change in cost (IUC_M). In this simple, well-known case where only the one performance measure involves life-cycle costs or benefits, this ratio is the IBC. On each bridge, a set of alternative candidates is defined, starting with do-nothing at zero cost and zero benefit, and ending with total replacement at maximum cost and maximum benefit. The rule of diminishing marginal returns is essential to the heuristic, so candidates failing to satisfy this rule are eliminated from consideration. The general steps of the heuristic are as follows:

1. Screen the candidates for diminishing marginal returns or diminishing IUC_M on each bridge.
2. Candidates of all bridges are combined, and the joined list is sorted by decreasing IUC_M.
3. Select the do-nothing candidate for each bridge.
4. Process the candidate list in IUC_M-sorted order. At each stage, the constraint is checked (budget or performance).
5. Replace each previously selected candidate with the next candidate on the same bridge, and then update the total cost and performance.
6. End the heuristic after scanning through the complete list or after the performance constraint is satisfied or the budget

constraint becomes too tight to allow the next candidate to be added.

The flowchart in Figure 21 shows the heuristic that uses the IBC ratio to solve a multi-choice knapsack problem with a budget constraint. It starts with a selection of do-nothing for each bridge, which has zero cost and zero benefit. Investments are added to the program in order of decreasing IBC ratio until the budget is exhausted.

Depending on the convention chosen, the stopping criterion may allow the final candidate to exceed the budget or may require that the candidate fit within the remaining budget. In the linear approximation of the knapsack problem, the final candidate is trimmed so that it fits the remaining budget exactly. The examples given here use the convention that all candidates must fit entirely within the budget in order to be selected. The remaining budget, if not used, will lead to the failure to realize the full benefit in the objective function. This is referred to as the *integrality gap* in the knapsack problems. An alternative way to reduce the integrality gap is to continue scanning down the list and try to fit smaller candidate projects.

For the case of a minimum performance constraint, an alternative way to use this heuristic would be to use a performance statistic instead of cost in the denominator of the IBC ratio. The heuristic, in this case, starts with the highest-performance investment on each bridge and moves downward through the sorted list until the constraint is violated.

The computationally intensive part of the IUC heuristic is a sorting algorithm that was selected for its computational efficiency in updating the candidate list. Often, it is convenient to maintain a data structure, such as a binary tree, that allows individual bridges to be modified without re-sorting the whole list. Many common operations, such as changing an individual candidate or moving the budget or performance constraint, can be performed without re-sorting the list, giving instantaneous performance for even very large problems.

Prepare candidate list
Prepare each bridge
Bridge-level analysis
Screen for DMR[1]

Assemble candidates
Sort by IUC[2] descending
Select DN[3] for every bridge

Select candidate with next highest IUC (upscope)

Update network cost and performance

Is budget gone?

no

yes

Done

[1]DMR   = Diminishing marginal returns
[2]IUC   = Incremental utility-cost ratio
[3]DN   = Do nothing
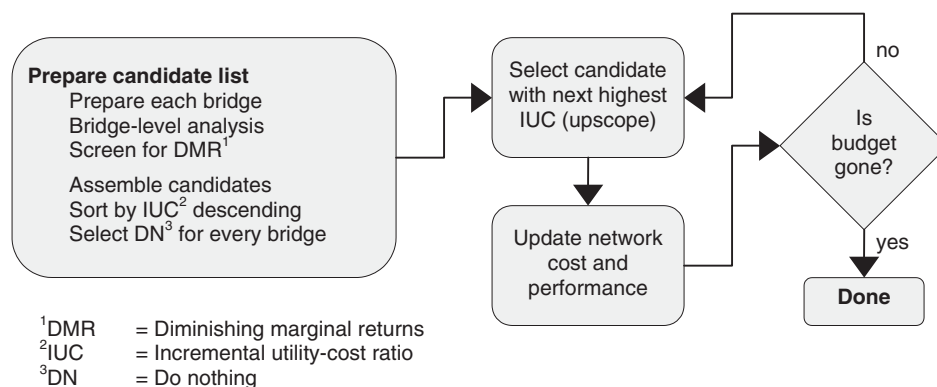
Figure 21. Flowchart of IUC heuristic for MCKP with a budget constraint.

An example of the MCKP with a budget constraint follows. Table 14 lists four bridges with 10 alternative candidates (Alt). Each bridge has a do-nothing alternative labeled "0," which has zero incremental cost and zero benefit (relative to itself) by definition. Life-cycle cost is calculated by the bridge-level analysis for each alternative. Benefit is the life-cycle cost of do-nothing minus the life-cycle cost of the alternative being considered. The IBC ratio is the change in benefit divided by the change in cost, relative to the next less expensive alternative on the same bridge. By definition, the do-nothing alternatives do not have an IBC ratio because there is no less expensive alternative. Note that this example is based on a single objective for illustration purposes, but this can be generalized using a utility function to incorporate multiple objectives.

The candidates in this example can be placed in priority order by sorting by IBC, as shown on the right side of Table 14. The right-most column is the cumulative cost of the four-bridge program as each increment of funding is added, if investments are selected in order of IBC ratio. Please note that cumulative values are not just the cumulative sum of the cost column. This is because when we determine the cumulative amount of money for the bridge network and select any candidate for a bridge on the list, we also need to de-select the previously selected candidate for that bridge. For example, if Alt #2 of Bridge #1 is added to the program (seventh row of the table), then the $700,000 cost of Alt #2 is added, but this replaces Alt #1, whose $200,000 cost is subtracted. This is a net increase in cost of $500,000, which increases the cumulative value from $600,000 to $1,100,000.

If no funding is available, the do-nothing candidate must be selected for all four bridges, so the total program cost is zero. If $1.7 million is available, there are adequate funds to perform Alt #1 on Bridges #2–3, and there is also enough to upscope Bridge #1 to Alt #2. If $2.6 million is available, then there are adequate funds to upscope Bridge #3 and to perform the work on Bridge #4.

Most agencies have uncertainty in both funding and project readiness, so it is common to overprogram by using a budget level larger than the amount actually anticipated. So when there is a residual, as in the case of a $1.7 million budget where $100,000 is left over, the extra amount is typically ignored. At any given budget level, total benefits are maximized by following this priority list, within a reasonable level of uncertainty.

A useful property of this heuristic is that it is not necessary to re-sort or make any other changes in the IBC-sorted candidate list in response to changes in the budget constraint. This facilitates the development of graphics to illustrate the relationships between performance and funding.

3.2.2.3 Multi-Choice Knapsack Problem, Multiple Constraints

As shown in Figure 22, the IUC heuristic can be generalized to incorporate multiple constraints if we recognize an order of priority among the constraints. The quantitative level of constraints is uncertain, and the achievability of a combined set of constraints is also uncertain. For example, if we have a total budget constraint of $1,000, but require that the 12,000-bridge inventory have a health index of 99, this is likely to be impossible mathematically. In the face of uncertainty, it is necessary to set priorities: certain constraints must be met absolutely, while others should be met only if possible or may be relaxed if necessary. This type of problem can be solved to yield a near-optimal result using a variation on the same procedure used for the single-constraint problem. The strategy is to divide

Table 14. Example MCKP with budget constraint.

Candidates grouped by bridge

Bridge	Alt	Cost	LCC	Benefit	IBC
1	0	0	2400	0	--
1	1	200	2000	400	2.00
1	2	700	1400	1000	1.20
2	0	0	3000	0	--
2	1	500	2550	450	0.90
3	0	0	2600	0	--
3	1	400	2000	600	1.50
3	2	600	1850	750	0.75
4	0	0	1900	0	--
4	1	800	1340	560	0.70

Sorted by IBC

Bridge	Alt	IBC	Cost	Cum
1	0	--	0	0
2	0	--	0	0
3	0	--	0	0
4	0	--	0	0
1	1	2.00	200	200
3	1	1.50	400	600
1	2	1.20	700	1100
2	1	0.90	500	1600
3	2	0.75	600	1800
4	1	0.70	800	2600

All economic quantities in $000s.
Alt = alternative.
LCC = life-cycle costs.
Cum = cumulative cost of the four-bridge program as each increment of funding is added, if investments are selected in order of IBC ratio.

¹DMR = Diminishing marginal returns
²IUC = Incremental utility-cost ratio
³DN = Do nothing
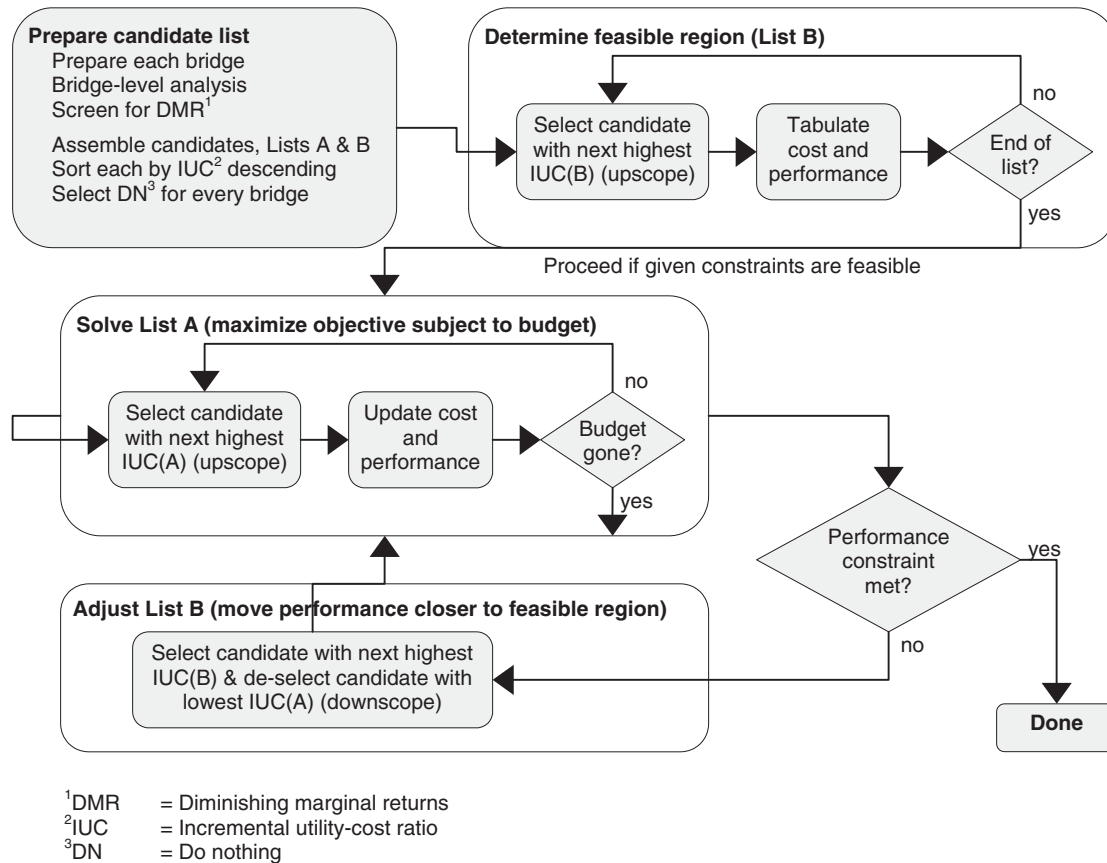
Figure 22. Flowchart of IUC heuristic for MCKP with multiple constraints.

it into two separate single-constraint knapsack problems that are solved in tandem:

- List A: simple budget-constrained problem

$$IUC(A) = \frac{\Delta \text{Utility}}{\Delta \text{Cost}} \tag{3-3}$$

- List B: feasible region problem

$$IUC(B) = \frac{\Delta \text{Performance}}{\Delta \text{Cost}} \tag{3-4}$$

List B describes the trade-off between the two competing constraints, giving the maximum performance level achievable at any given cost level. Therefore, it represents an ordinary MCKP that maximizes performance subject to a budget constraint and can be solved using the ordinary IUC algorithm described earlier. The numerator of $IUC(B)$ can be a utility function incorporating multiple performance measures, if desired. List A is the underlying optimization problem with only the budget constraint.

If there is no feasible solution for the combination of budget and performance constraints given, this is detected in List B before entering the main iteration of the procedure. Typically,

the first pass through the List A optimization will fail to satisfy the performance constraint, so List B is used to find the candidate most responsible for the violation of the constraint. Then List A is solved again. This repeats until both constraints are finally met. If the performance constraint is not binding, the optimal solution will be one of the candidate combinations found on List A but would not necessarily be visited by the IUC algorithm on List B.

An example of the MCKP with multiple constraints follows. Figure 23 presents the IUC heuristic for a small network. The network consists of four bridges, and each bridge candidate is evaluated based on life-cycle benefit (i.e., the costs of do-nothing minus that of do-something) and health index benefit (do-something minus do-nothing). Costs reported are in thousands of dollars. The problem is to determine optimal candidate selections to maximize the network life-cycle benefits subject to two constraints: budget available is 1,200,000 and health index benefit threshold is 60 units. The descriptions of the steps are as follows.

Step 0:
- Start with selecting do-nothing for each bridge.
- Select the first candidate in List A (Bridge 1, Candidate 1) and de-select do-nothing for Bridge 1.

List of alternative Candidates

Bridge	Alt	Cost	LCC	LCB	ILBC	HI	HIB	IHBC
1	0	0	2400	0	--	70	0	--
1	1	200	2000	400	2	85	15	0.075
1	2	700	1400	1000	1.2	90	20	0.010
2	0	0	3000	0	--	60	0	--
2	1	500	2550	450	0.9	90	30	0.060
3	0	0	2600	0	--	60	0	--
3	1	400	2000	600	1.5	80	20	0.050
3	2	600	1850	750	0.75	85	25	0.025
4	0	0	1900	0	--	90	0	--
4	1	800	1340	560	0.7	95	5	0.006

Alt = Candidate alternative (0 = do-nothing)
Cost = Agency cost of the Candidate
LCC = Life cycle cost
LCB = Life cycle benefit (do-nothing minus do-something LCC)
ILBC = Incremental benefit/cost using LCB
HI = Health index at end of horizon
HIB = Health index benefit (do-something minus do-nothing HI)
IHBC = Incremental benefit/cost using HI

Feasible Region

Bridge	Alt	Cost	HIB	IHBC
1	0	0	0	--
2	0	0	0	--
3	0	0	0	--
4	0	0	0	--
1	1	200	15	0.075
2	1	500	30	0.060
3	1	400	20	0.050
3	2	600	25	0.025
1	2	700	20	0.010
4	1	800	5	0.006

Cost= 1300
HIB= 70

Using List B, it is possible to determine the minimum budget constraint that is compatible with any given condition constraint, using the ordinary IBC algorithm. For example, if the condition constraint were 70, then 1300 is the smallest budget constraint that will give a feasible solution.

Maximize LCB subject to total Cost<=1200 and total HIB>=60

| Step | List A | | | | | List B | | | | | Network results and remarks |

Step 0

List A

Bridge	Alt	LCB	ILBC	Cost
1	0	0	--	0
2	0	0	--	0
3	0	0	--	0
4	0	0	--	0
1	1	400	2	200
3	1	600	1.5	400
> 1	2	1000	1.2	700
2	1	450	0.9	500
3	2	750	0.75	600
4	1	560	0.7	800

List B

Bridge	Alt	Cost	HIB	IHBC
1	0	0	0	--
2	0	0	0	--
3	0	0	0	--
4	0	0	0	--
1	1	200	15	0.075
2	1	500	30	0.060
3	1	400	20	0.050
3	2	600	25	0.025
> 1	2	700	20	0.010
4	1	800	5	0.006

Cost= 1100
HIB= 40
LCB= 1600

Begin with a do-nothing program, then use List A to upscope to the budget constraint. This is the same method as the ordinary one-constraint IBC method. Since the condition constraint is not met, we must continue to the next step.

Step 1a

List A

Bridge	Alt	LCB	ILBC	Cost
1	0	0	--	0
2	0	0	--	0
3	0	0	--	0
4	0	0	--	0
1	1	400	2	200
3	1	600	1.5	400
> 1	2	1000	1.2	700
2	1	450	0.9	500
3	2	750	0.75	600
4	1	560	0.7	800

List B

Bridge	Alt	Cost	HIB	IHBC
1	0	0	0	--
2	0	0	0	--
3	0	0	0	--
4	0	0	0	--
1	1	200	15	0.075
2	1	500	30	0.060
> 3	1	400	20	0.050
3	2	600	25	0.025
1	2	700	20	0.010
4	1	800	5	0.006

Cost= 600
HIB= 35
LCB= 1000

Downscope the lowest selected Candidate on List B. This lowers both the cost and condition, but increases the efficiency of condition improvement by eliminating the least efficient Candidate in the list.

Step 1b

List A

Bridge	Alt	LCB	ILBC	Cost
1	0	0	--	0
2	0	0	--	0
3	0	0	--	0
4	0	0	--	0
1	1	400	2	200
3	1	600	1.5	400
1	2	1000	1.2	700
> 2	1	450	0.9	500
3	2	750	0.75	600
4	1	560	0.7	800

List B

Bridge	Alt	Cost	HIB	IHBC
1	0	0	0	--
2	0	0	0	--
3	0	0	0	--
4	0	0	0	--
1	1	200	15	0.075
2	1	500	30	0.060
> 3	1	400	20	0.050
3	2	600	25	0.025
1	2	700	20	0.010
4	1	800	5	0.006

Cost= 1100
HIB= 65
LCB= 1450

In List A, upscope back up to the budget constraint to complete the step. Since the total HIB of 65 now exceeds the constraint of 60, this is a feasible near-optimal solution.

Step 2a

If a feasible solution had not been found in the previous step, the algorithm would continue through more steps. In each step, downscope the worst Candidate in List B, and then upscope Candidates in List A to return to the budget constraint.

Step 2b

...

Figure 23. Example IUC heuristic for a small network.

- Continue scanning List A and select Bridge 3, Candidate 1, and de-select do-nothing for Bridge 3.
- Select the next candidate in List A (Bridge 1, Candidate 2) and de-select Candidate 1 for Bridge 1.
- Stop because next candidate would violate the budget constraint (see shaded selections in Figure 23).
- Compute HIB = 40 (see List B, Column HIB). Since the health constraint is not met, go to the next step. Note that the total benefit for selected candidates is 1,600.

Step 1a:
- Select the first candidate in List B (Bridge 1, Candidate 1) and de-select Candidate 2 for Bridge 1. This lowers the cost and condition temporarily, but increases the efficiency of condition improvement.

Step 1b:
- Select the next candidate in List B (Bridge 2, Candidate 1) and de-select do-nothing for Bridge 2. The final selections are shown.
- Compute HIB = 65 and total benefit = 1,450. Stop, because the health constraint is now satisfied. Note that from Step 0 to Step 1b, the HIB is increased (from 40 to 65) at the cost of decreasing the total benefit (from 1,600 to 1,450). This reflects the trade-off between the total benefits and health index benefit.

3.2.3 Lagrangian Relaxation Heuristic

This section describes the Lagrangian relaxation heuristic and provides an example to illustrate the method for a small bridge network. The Lagrangian method is based on relaxing the constraints and incorporating them into the objective function. It is an iterative procedure that makes use of Lagrangian multipliers, as described in the next section. In effect, this method starts with the best (i.e., having the highest objective function value) but infeasible solution and iterates to find a feasible solution. It imposes, on any proposed solution, a penalty whose magnitude is proportional to the extent to which the constraint is violated.

3.2.3.1 Theoretical Foundation

Consider the following MCMDKP:

$$\max z = \sum_{k=1}^{n}\sum_{j\in L_k} r_{jk} x_{jk}$$

$$s.t. \quad \sum_{k=1}^{n}\sum_{j\in L_k} a_{ijk} x_{jk} \le b_i, \quad i = 1, 2, \ldots, m$$

$$\sum_{j\in L_k} x_{jk} = 1, \quad k = 1, 2, \ldots, n$$

$$x_{jk} \in \{0,1\} \quad k = 1, 2, \ldots, n, j \in L_k \quad (3\text{-}5)$$

where

r_{jk} = reward or value associated with candidate j for bridge k;

x_{jk} = 1 if candidate j is selected for bridge k, 0 otherwise;

i = subscript for size constraints (e.g., budget constraint and condition constraint);

a = nonnegative coefficients representing the size constraints (e.g., cost and health index); and

b = threshold values (e.g., budget and minimum health index).

Use of variable names a, b, and index i is needed to generalize the procedure for number of constraints m.

One candidate is to be selected for each bridge (i.e., group) to maximize the value obtained. The fundamental result that makes the Lagrangian multipliers applicable to discrete optimization problems is given by Everett's theorem (Everett 1963) as follows:

Let $\mu_1, \ldots, \mu_m$ be m nonnegative Lagrangian multipliers and $x_{jk}^* \in \{0,1\}$ be a solution of

$$\max\left\{\sum_{k=1}^{n}\sum_{j\in L_k} r_{jk} x_{jk} - \sum_{i=1}^{m}\mu_i \sum_{k=1}^{n}\sum_{j\in L_k} a_{ijk} x_{jk}\right\}$$

$$= \left\{\sum_{k=1}^{n}\sum_{j\in L_k}\left(r_{jk} - \sum_{i=1}^{m}\mu_i a_{ijk}\right) x_{jk}\right\} \quad (3\text{-}6)$$

Then, the binary variables x_{jk}^* are also a solution to:

$$\max z = \sum_{k=1}^{n}\sum_{j\in L_k} r_{jk} x_{jk}$$

$$s.t. \quad \sum_{k=1}^{n}\sum_{j\in L_k} a_{ijk} x_{jk} \le \sum_{k=1}^{n}\sum_{j\in L_k} a_{ijk} x_{jk}^* \quad (3\text{-}7)$$

If the Lagrangian multipliers are known, the solutions are:

$$x_{jk}^* = 1 \text{ if } r_{jk} - \sum_{i=1}^{m}\mu_i a_{ijk} > 0$$

$$= 0 \text{ otherwise} \quad (3\text{-}8)$$

The solution is feasible if the Lagrangian multipliers can be computed such that the following terms are nonnegative:

$$b_i - \sum_{k=1}^{n}\sum_{j\in L_k} a_{ijk} x_{jk}^*$$

The solution is optimal if the following condition holds:

$$\sum_{i=1}^{m}\mu\left(b_i - \sum_{k=1}^{n}\sum_{j\in L_k} a_{ijk} x_{jk}^*\right) = 0 \quad (3\text{-}9)$$

3.2.3.2 Procedure

The following heuristic from Moser et al. (1997) is based on the heuristic of Magazine and Oguz (1984), but is organized such that the group constraints are satisfied.

Input:

Element values	$r_{jk}, k = 1, \ldots, n; j \in L_k$
Element weights	$a_{ijk}, i = 1, \ldots, m; k = 1, \ldots, n, j \in L_k$
Knapsack sizes	$b_i, i = 1, \ldots, m$

Output:

Selected elements	$x_{j'k}, k = 1, \ldots, n; j \in L_k$

Step 0: Initialization and normalization
- Step 0.1: Reset Lagrangian multipliers.

$$\mu_i = 0, \quad i = 1, \ldots, m$$

- Step 0.2: Select most valuable elements. Find the index j' of the most valuable element in each group, $k = 1, \ldots, n$, and select this element.
- Step 0.3: Normalize weights.

$$a_{ijk} = \frac{a_{ijk}}{b_i} \quad i = 1, \ldots, m, \quad k = 1, \ldots, n, \quad j \in L_k \qquad (3\text{-}10)$$

- Step 0.4: Compute constraint violation.

$$y_i = \sum_{k=1}^{n} a_{ij'k} \quad i = 1, \ldots, m \qquad (3\text{-}11)$$

Step 1: Relax the constraint violation. Repeat until $y_i \leq 1$, $i = 1, \ldots, m$, or no more elements can be exchanged.
- Step 1.1: Determine the most violated constraint. Find the index i' of the largest $y_i > 1$, $i = 1, \ldots, m$.
- Step 1.2: Find element to exchange. Compute the increase δ_{jk} of the Lagrangian multiplier $\mu_{i'}$ for all nonselected elements in every group relative to the selected element of the group:

$$\delta_{jk} = \left(r_{j'k} - r_{jk} - \sum_{i=1}^{m} \mu_i \left(a_{ij'k} - a_{ijk} \right) \right) \Big/ \left(a_{i'j'k} - a_{i'jk} \right) \qquad (3\text{-}12)$$

Find the group k^0 and the index j^0 of the element to which the smallest δ_{jk} belongs.
- Step 1.3: Reevaluate multipliers and constraint violation.

$$\mu_{i'} = \mu_{i'} + \delta_{j^0 k^0} \qquad (3\text{-}13)$$

$$y_i = y_i - a_{ij'k^0} + a_{ij^0 k^0}, \quad i = 1, \ldots, m \qquad (3\text{-}14)$$

- Step 1.4: Exchange the selected element. Remove the selected element, index j', of group k'. Make the element with index j^0 the new selected element for group k^0.

Step 2: Improve solution. Repeat until no more elements can be exchanged.

- Step 2.1: Compute the knapsack value increases Δ_{jk} for all nonselected elements in every group relative to the value $r_{j'k}$ of the selected element:

$$\Delta_{jk} = r_{jk} - r_{j'k}, \text{ if } r_{j'k} - r_{jk} > 0 \text{ and } y_i - a_{ij'k} + a_{ijk} \leq 1, \ i = 1, \ldots, m$$

$$= 0 \text{ otherwise} \qquad (3\text{-}15)$$

- Step 2.2: Find the best exchangeable element. Find the group k^L and the index j^L for the largest knapsack value increase Δ_{jk}.
- Step 2.3: Exchange the selected element. Remove the selected element, index j', of group k^L. Make the element with index j^L the new selected element in group k^L.

Step 3: Compute the result.

$$y_i = \sum_{k=1}^{n} a_{ij'k} \quad i = 1, \ldots, m \qquad (3\text{-}16)$$

If $y_i \leq 1$, $i = 1, \ldots, m$, the problem is solvable and the results are elements with index $j'k$ for $k = 1, 2, \ldots, n$; otherwise, either the problem is not solvable or no solution can be found. The heuristic can be schematically represented, as shown in Figure 24.

3.2.3.3 Example

Consider a simple problem with three bridges and a budget constraint. The multiple objectives are to maximize life-cycle benefits and health index benefits. The list of alternative candidates is shown in Table 15. The do-nothing alternative is indicated as "0." Cost is in thousands of dollars, LCB is the life-cycle benefit, and HI is the health index at the end of program period if the candidate is implemented. Let the budget be $1,600,000. The first step is to compute the total utility values for each candidate. To do this, the single-attribute utility values for health index and life-cycle benefit must be computed individually. For simplicity, a linear utility for life-cycle benefit and an S-shaped utility function for health index are assumed. The functional form of the health index utility function used for this example is shown in Figure 25.

The functional form of the health index utility can be developed using the methods described in Chapter 2. It should be noted that the term *utility* has been deliberately used loosely in this discussion for simplicity. The use of the term *value function* is more appropriate here because utility functions are used in the presence of uncertainty in performance criteria, which is not considered in this example.

Next, the relative weights for this example are assumed to be as follows: $W_{LCB} = 0.7$, and $W_{HI} = 0.3$. These relative weights can be developed using the methods discussed in Chapter 2.

¹ Objective function
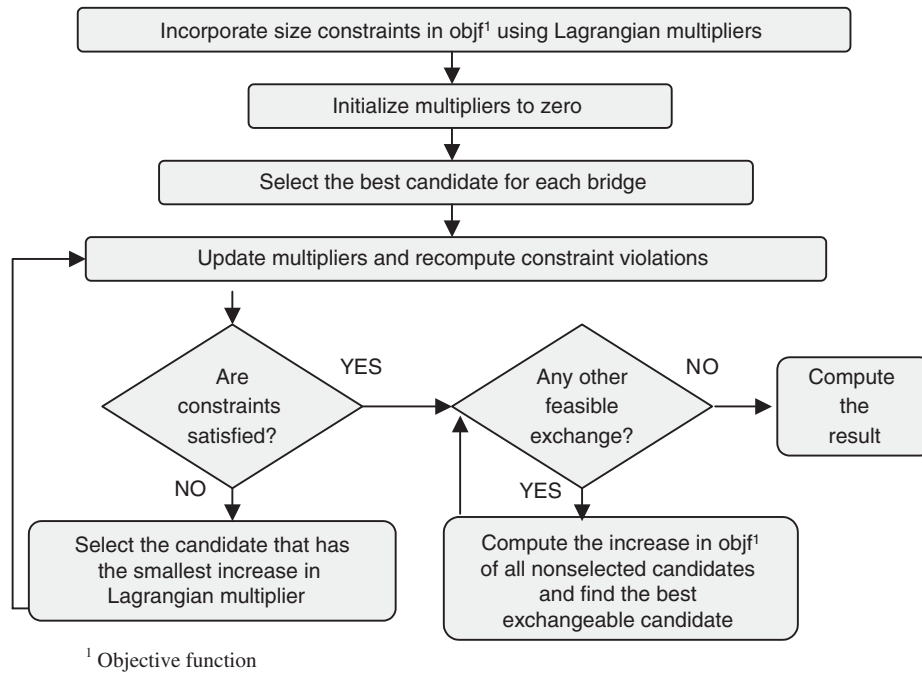
Figure 24. Schematic representation of the Lagrangian relaxation heuristic.

Then, the total utility (TU) and change in utility (Δ Utility) for each candidate are calculated:

$$TU = 0.7*\{U(\text{life-cycle benefit})\}$$
$$+ 0.3*\{U(\text{health index})\} \tag{3-17}$$

$$\Delta\text{ Utility} = TU(\text{candidate})$$
$$- TU(\text{do-nothing for that bridge}) \tag{3-18}$$

The results of these steps are summarized in Table 16.

Using the notations described earlier the Δ Utility is the "reward" r_{jk} of a candidate. Once these values are computed, the normalization step of the Lagrangian heuristic can be undertaken.

Step 1: Initialize and Normalize.   The Lagrangian multiplier is initiated to zero. In this example, only one Lagrangian

multiplier is needed because there is only one constraint (budget). The best candidate (i.e., the highest benefit or life-cycle benefit) is selected for each bridge. Costs are normalized by dividing cost values by the budget. The results of this step are shown in Table 17.

Step 2: Compute Violation and Exchange Selections. Using Equation 3-11, the constraint violation is computed as follows:

$$y = 1.531 (>1; \text{violated}) \tag{3-19}$$

The increase in the multiplier δ_{jk} is calculated for all nonselected candidates for each bridge relative to the selected candidate using Equation 3-12 given in the previous section. The candi-

Table 15. Example Lagrangian heuristic for a small network.

Bridge (k)	Alt (j)	Cost (a_{jk})	LCB_{jk}	HI_{jk}
1	0	0	0	58
1	1	450	250	64
1	2	1100	950	91
2	0	0	0	21
2	1	500	500	96
3	0	0	0	35
3	1	700	400	82
3	2	850	600	66

Alt = alternative.
LCB = life-cycle benefit.
HI = health index.

$$Utility = 102 \times (1 - e^{-0.0003 \times HI^2})$$
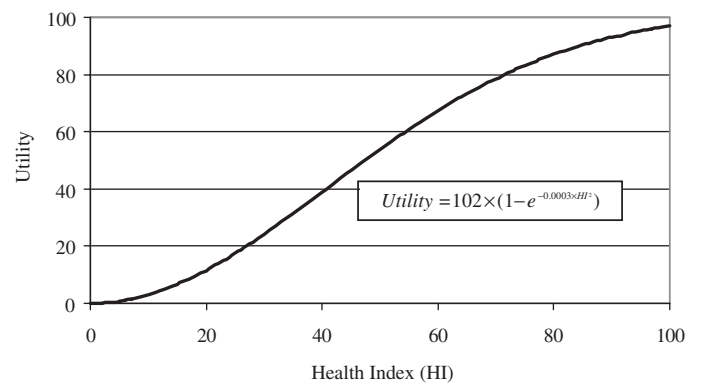
Figure 25. Utility function of health index for the small network example.

Table 16. Calculation of utilities for small network example.

Bridge (k)	Alt (j)	HI	U(HI)	LCB	U(LCB)	TU	ΔUtility (r_{jk})
1	0	58	67	0	0	20	0
1	1	64	75	250	25	40	20
1	2	91	95	950	95	95	75
2	0	21	13	0	0	4	0
2	1	96	97	500	50	64	60
3	0	35	33	0	0	10	0
3	1	82	90	400	40	55	45
3	2	66	77	600	60	65	55

Alt = alternative. HI = health index. U = utility. LCB = life-cycle benefit. TU = total utility.
Δ Utility = change in utility.

date with the smallest increase in multiplier is selected. The results of this step are shown in Table 18.

Step 3: Recompute the Multiplier and Violations. The constraint violation is recomputed as $y = 1$. Also, the multiplier is recomputed as $0 + 1035.3 = 1035.3$. The iterations stop when the constraint is satisfied and no more feasible exchanges are possible. The total utility benefit across the network is 135 (sum of utility benefits for selected candidates), which incorporates the life-cycle benefits and the health index benefits. The life-cycle benefit of the network is \$1,450,000, and the cost of selected candidates is \$1,600,000. The average health index of the network after implementation of the selected candidates is 74. The total health index benefit (sum of [health index of selected candidate − health index of do-nothing] for all bridges) of the network is 108.

3.2.4 Pivot and Complement Heuristic

"Pivot and complement" is a heuristic for finding approximate solutions to large arbitrary 0/1 programming problems (Balas and Martin 1980).

3.2.4.1 Theoretical Foundation

The theoretical foundation of the heuristic is based on the fact that the following 0/1 program (P),

$$\max \{ rx \,|\, Ax \le b, \, x_k = 0 \text{ or } 1, \, k = 1, 2, \dots, n \} \quad (3\text{-}20)$$

where A is $m \times n$, and b is $m \times 1$, is equivalent to the following linear problem (LP):

$$\max \{ rx \,|\, Ax + y = b, 0 \le x \le e, \, y \ge 0, \, y_i \text{basic},$$

$$\forall i = 1, 2, \dots, m \} \quad (3\text{-}21)$$

where $e = (1, \dots, 1)$ is a vector of size n. The notations have similar meaning as those described in the previous section. A is the matrix of coefficients a, and y is a slack variable to convert the constraints in standard form in the context of a simplex algorithm for linear programming.

3.2.4.2 Procedure

The steps of the heuristic are as follows:

1. Solve the linear programming relaxation to find an optimal basic solution in the sense of the simplex method for linear programs with upper-bounded variables. If the solution is integer, then stop.
2. Find a "good" feasible 0-1 solution by applying three types of methods:
   - Pivoting: This is aimed at putting into basic all the nonbasic slack variables y_i at a minimal cost in dual feasibility. Three different types of pivoting actions are used.
   - Complementing: Complementing a variable x_k means moving x_k from one of its bounds to the opposite bound. This is done to remove any occasional primal infeasibilities.

Table 17. Lagrangian heuristic example: Step 1 results.

Step	Bridge (k)	Alt (j)	Norm-Cost (na_{jk})
	1	0	0
1	1	1	0.281
	1	2	0.688
	2	0	0
	2	1	0.313
	3	0	0
	3	1	0.438
	3	2	0.531

Bridge (k)	Selected Alt
1	2
2	1
3	2

Table 18. Lagrangian heuristic example: Step 2 results.

Step	δ_{jk}	
	δ_{01}	109.1
2	δ_{11}	135.4
	δ_{02}	192.0
	δ_{03}	103.5
	δ_{13}	106.7

Bridge (k)	Selected Alt
1	2
2	1
3	0

- Rounding and truncating: At certain stages, to find a feasible 0-1 solution, rounding and truncating are used. Rounding a solution means rounding all fractional 0-1 variables to the nearest integer value. Truncating a solution means replacing all fractional values of 0-1 variables by 0.
3. Improve the current 0-1 solution by single, double, or triple complements (i.e., complement one, two, or three variables).

3.2.5 Computational Experiments

The heuristics to solve the network-level optimization problem were coded and implemented. This section describes these experiments, including the datasets, coding language, and computer configuration used, as well as the results obtained. The computational experiments are useful because they provide a way to test the heuristics on realistic-size bridge networks. The heuristics can be tested for the computational times taken to arrive at the results and also for the accuracy of the results (i.e., how close the results are to actual optimal solutions). On the basis of these results, an appropriate heuristic can be selected for implementation in the final software product. In order to test the heuristics for accuracy, we also need to know the true optimal solution that is obtained using an exact method. The state-of-the-art mathematical optimization technology ILOG CPLEX and Concert technology were used to arrive at the optimal solutions. This is explained in the subsequent sections.

3.2.5.1 Datasets

Sample datasets were prepared for bridge network sizes of 100; 1,000; 9,265; 12,000; and 50,000. The 12,000-bridge file represents an average database size for a state's inventory (including local bridges). The 50,000-bridge file represents the largest possible state inventory (that of Texas). The data files were generated using the Florida DOT Project Level Analysis Tool, which used Florida's actual inventory of 11,295 bridges and fully realistic deterioration, cost, and other models. The models are similar to those in Pontis 2.0 but include certain refinements that render them more realistic for bridge-level analysis. These are not exactly the same

as the bridge-level model in the present study, but they should produce results that are similar to that of the present study and provide a means to carry out realistic speed tests. To generate the smaller files, the inventory was scanned in order by bridge identifier, stopping when the target number of bridges was reached. For the larger cases, the list of bridges was cycled again, but the sample selection was carried out in a deliberate random fashion to ensure that there was significant statistical variation in the costs, benefits, and health indices of the selected bridges, and also to ensure that the values of cost, benefit, and health index do not vary together.

Three types of candidates were generated: Do-nothing, preservation, and replacement. Do-nothing, by definition, has zero cost and zero benefit in every case. Preservation is defined using a needs identification model similar to Pontis but refined by a minimum project size and a quantity prediction model developed for Florida for project-level analysis. Depending on the condition and deficiencies of a given bridge, its preservation candidate may include maintenance, repair, rehabilitation, and/or replacement of any or all elements. It may also include functional improvements. Replacement is immediate total replacement of the bridge. Benefit is computed on the basis of a life-cycle cost model that includes current and forecast future maintenance, repair, rehabilitation, improvement, and replacement costs, and user costs. The benefit of any candidate is computed by relating its life-cycle cost to that of the do-nothing candidate. So, the do-nothing candidate, by definition, has a zero incremental benefit relative to itself, while the benefits of preservation and replacement may be either positive or negative depending on whether discounted future life-cycle cost savings exceed initial costs. Benefits greater than zero are preferable. Also, each candidate has an associated health index value. This is the ending health index at the end of the year.

A screening process was then applied, which removed do-something candidates with nonpositive benefits and candidates inconsistent with diminishing marginal returns. So, preservation candidates were eliminated if replacement costs were the same or less. Replacement candidates were eliminated if the preservation candidate had equal or higher benefits. Also, any preservation candidates were eliminated if the replacement candidate had a higher IBC ratio. In a realistic bridge inventory, the screening tests can make the problem more tractable by reducing the number of bridges with multiple choices. It is important to take advantage of this in the optimization methodologies.

3.2.5.2 Problem Structure

The datasets can be used to test a simple but useful representative example of a multi-constraint model: Maximize the total

network benefit subject to two constraints: (1) the total cost may not exceed a given budget and (2) the average health index at the end of the year may not be below a given standard. Note that we are using life-cycle benefit rather than utility in the objective function, but this does not affect the speed of the optimization.

3.2.5.3 Computer Configuration

All computational experiments were implemented on computers with the following configuration for consistency in computational speeds: Intel Pentium 4 CPU 3.40 GHz, 1.00 GB of RAM. All the computational times reported in this chapter are based on the performance of computers that have this configuration.

3.2.5.4 Coding Language—Visual Basic for Applications (Heuristics)

Visual Basic for Applications (VBA) is a programming language that is built into many popular software packages, including Microsoft Excel. The heuristics are coded using VBA for consistency with the overall software product, which will also be VBA based.

3.2.5.5 ILOG CPLEX—Concert Technology (Exact Method)

ILOG CPLEX is state-of-the-art mathematical optimization technology that uses advanced algorithms and analytical techniques to solve optimization problems. ILOG Concert technology, released in October 2000, is a set of C++ and Java objects for representing large optimization problems. The Concert technology was used to write a code that represents the bridge network optimization problems. These problems were then solved using CPLEX. CPLEX provides high-performance optimizers for solving a variety of optimization problems, including mixed-integer programming problems. The CPLEX mixed-integer optimizer features branch and bound techniques, a variety of cutting plane strategies and node-selection strategies, and what is claimed to be the world's fastest implementations (ILOG CPLEX not dated). CPLEX provides true optimal solutions (ILOG CPLEX 2003).

3.2.5.6 Preliminary Computational Speed Test Results for Heuristics

The network optimization problem was solved using the IUC, Lagrangian, and pivot and complement heuristics to maximize total network benefits subject to a budget constraint and a health index constraint. The heuristics were implemented for different network sizes. The linear programming relaxation,

which is the first part of the pivot and complement heuristic, was solved to get the computational times. The total computational times for pivot and complement are the lower-bound estimates based on linear programming relaxation times and computational complexities.

Figure 26 shows the effect of bridge network size on the computational times for the heuristics. The computational times are shown in seconds. The IUC heuristic is much faster than the other two heuristics. The computational time for the Lagrangian heuristic grows with the network size much more rapidly than that of the IUC heuristic. This reflects the computational complexities of the two methods. The process time for the pivot and complement heuristic is very large and grows much more quickly with the network size compared with the computational time of the ICU heuristic. Based on these results, the IUC and Lagrangian heuristics were selected for a detailed computational analysis.

3.2.5.7 IUC and Lagrangian versus CPLEX Concert

In order to determine the performance of heuristics in terms of accuracy of solution, we need to know the optimal solution. The optimal solution was found using the CPLEX Concert technology. The optimization was implemented for different bridge network sizes. Table 19 summarizes the results of these computational experiments. Using data from Table 19, Figure 27 charts the computational times of the heuristics and the CPLEX Concert. The IUC heuristic is the fastest method computationally for network optimization.

Using the values in Table 19, we can compute the accuracy of the heuristics. The accuracy of a heuristic is measured based on how close the objective function value (i.e., total network benefits) is to the optimal objective function value. It is equal to 100 minus the percentage difference in the objective function value between the heuristic solution and optimal solution. Figure 28 shows the accuracy of heuristics across different sizes of the bridge network. In terms of accuracy, both the IUC
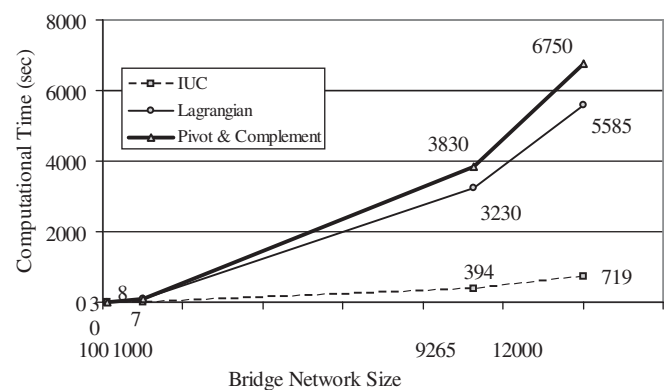
Figure 26. Preliminary computational speed test results.

Table 19. Network optimization results for CPLEX Concert, IUC, and Lagrangian heuristics.

Network Parameters	Network Size	100	1000	9265
	Budget ($)	**3,136,021**	**65,557,000**	**1,064,698,026**
	Health Index Threshold	**82.9**	**84.2**	**86.3**
CPLEX Concert (Exact)	Final Cost ($)	3,134,338	65,555,673	1,064,697,984
	Final Health Index	84.45	86.32	88.47
	Final Benefit ($)	2,473,164	352,279,924	3,910,275,499
	Computational Time (sec)	0	4	701
IUC Heuristic	Final Cost ($)	3,130,405	65,286,999	1,063,039,267
	Final Health Index	84.43	86.30	88.47
	Final Benefit ($)	2,472,730	352,109,043	3,909,466,703
	Computational Time (sec)	0	7	394
Lagrangian Heuristic	Final Cost ($)	3,130,405	65,556,919	1,064,697,380
	Final Health Index	84.43	86.37	88.48
	Final Benefit ($)	2,472,730	352,260,752	3,909,776,254
	Computational Time (sec)	3	82	3230

heuristic and the Lagrangian heuristic perform excellently. The Lagrangian heuristic has a slightly higher accuracy and much slower computational speed. The average accuracy for the IUC heuristic is 99.97% and for the Lagrangian heuristic is 99.99%. It becomes computationally hard to use the exact method for a 50,000-bridge network because the memory requirements grow exponentially. The Lagrangian heuristic takes a long time for the 50,000-bridge network. An estimate of computational time based on the mathematical complexity would be more than a day. An important observation is that the IUC heuristic takes 44% less time than the exact CPLEX Concert technology at the expense of a negligible loss of accuracy, for the 9,265-bridge case.

Figure 29 compares the IUC solution objective function value (i.e., total network benefits) with the optimal solution. This is the same as 100 minus the accuracy in Figure 28. A negative value indicates that the total network benefits for IUC solution is *below* the benefits of the optimal solution. These values should always be nonpositive because the IUC solution is
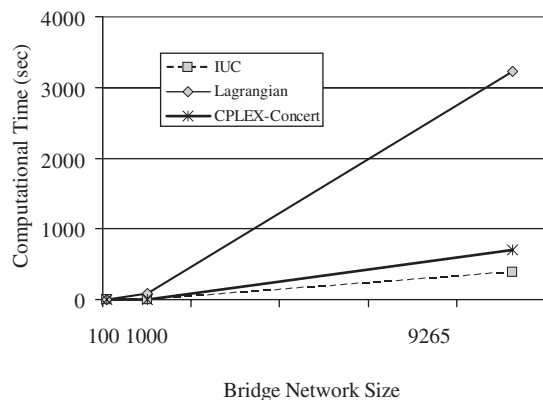
near optimal. Also plotted is the percentage difference in the average network health index between the IUC solution and the optimal solution. A negative value would indicate that the average network health index of the IUC solution is below the health index of the optimal solution. The average network health index for the IUC solution is very close to the health index for optimal solution (within 0.01%).

A similar plot for the Lagrangian heuristic in Figure 30 shows that the objective function values are very close to those of the optimal solution and that the average network health index is within 0.07% of the network health index of the optimal solution.

3.2.5.8 Evaluating the Robustness of Heuristics

The heuristics can be tested for robustness by carrying out a sensitivity analysis. We conducted experiments to analyze the effect of changing the threshold parameters like budget and health index on the performance of the heuristics to test for robustness. Performance of heuristics is evaluated in terms of accuracy and computational time, as in the previous section. Three different scenarios were considered:

- Scenario 1: Budget 25%, health index 25%
- Scenario 2: Budget 35%, health index 25%
- Scenario 3: Budget 25%, health index 40%

The percentages refer to how far the threshold parameters are from the less binding end of the range. The theoretical range for budget can be from zero to the total cost of do-everything candidates for all bridges. Health index thresholds can range from the network health index resulting from do-nothing for all bridges to the network health index resulting from do-everything for all bridges. The higher side of budget and lower side of health index are less binding. Therefore, the first scenario

Figure 27. Computational time for heuristics and CPLEX Concert.
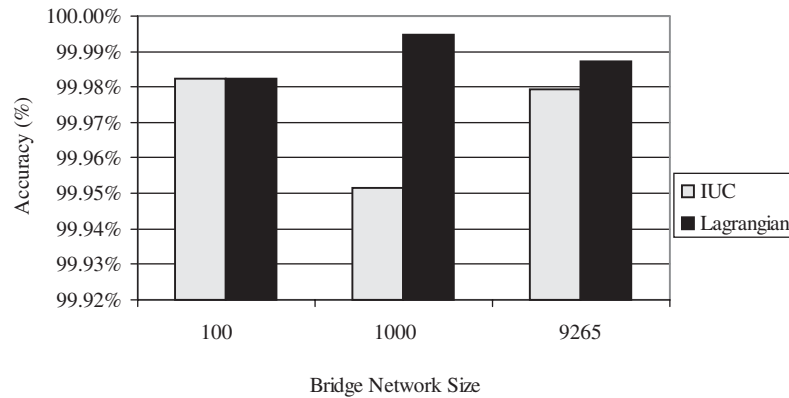
Figure 28. Accuracy of heuristics.

has a budget 25% less than the do-everything value and a health index threshold 25% more than the do-nothing value. Table 20 shows the network parameter values for these three scenarios.

Experiments were conducted on different sizes of bridge networks for each of the above scenarios. Figure 31 shows computational times of the IUC heuristic for different scenarios. The experiments were also conducted on a 50,000-bridge network. The average computational time across three scenarios was 3,606 seconds (not shown in the figure due to disproportionate scaling). The computational times are about the same under different scenarios.

To compute the accuracy, the same network optimization problems were also solved using CPLEX Concert for optimal solutions. Figure 32 shows the accuracy for the IUC solutions under the three scenarios. The accuracy varies from 96.82% to 100%. The IUC accuracy for Scenario 2 for the 9,265-bridge network is 100%, which means that it obtains the optimal solution. The average network health index was within 1.41% of the optimal solution for all scenarios.

Similar plots for the Lagrangian heuristic are shown in Figures 33 and 34. As shown in these figures, the Lagrangian

heuristic has similar computational times under different scenarios but is more sensitive to accuracy, especially for small networks. The accuracy varies from 84.94% to 99.99%.

The average performance of the IUC and Lagrangian heuristics across all experiments except for the 50,000-bridge network is presented in Figures 35 and 36. (The Lagrangian heuristic was not tested on the 50,000 network.) The average computational time for the 50,000-bridge network was 3,606 seconds for the IUC heuristic. The average accuracy for the IUC heuristic is 99.62% and for the Lagrangian heuristic is 97.88%.

3.2.5.9 Example to Illustrate Differences in the Bridge Candidates Selected by IUC and CPLEX

A 100-bridge network is taken to illustrate the solutions obtained from IUC and CPLEX. In this network, there are 46 bridges that have only one candidate—do-nothing. The ending health indices for these bridges are shown in Table 21. The
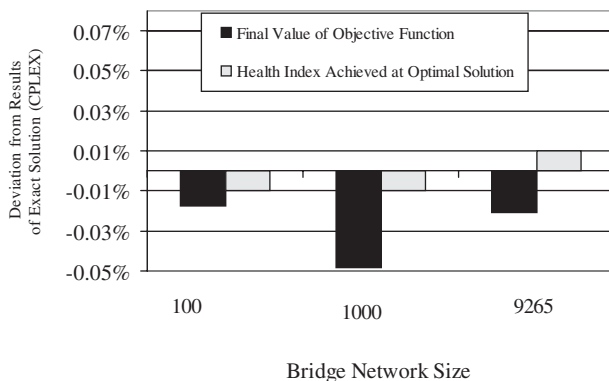
Figure 29. Deviation of IUC solutions (value of objective function and achieved health index) from optimal solutions.
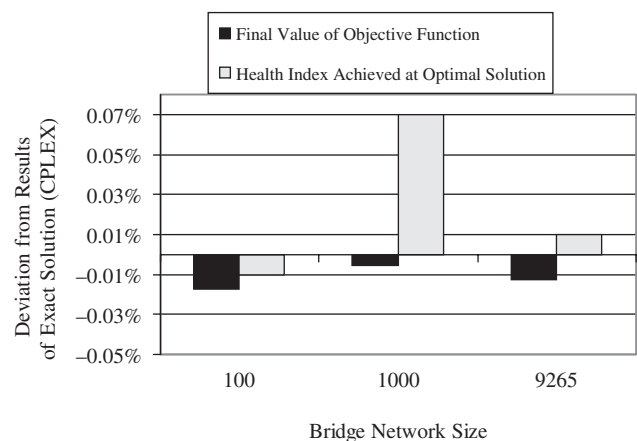
Figure 30. Deviation of Lagrangian solutions (value of objective function and achieved health index) from optimal solutions.

Table 20. Scenarios for sensitivity analysis.

Scenario Number	Network Parameters	Network Size				
		100	1000	9,265	12,000	50,000
Scenario 1	Budget ($)	3,136,021	65,557,000	1,064,698,026	1,305,334,742	5,613,742,982
	Health Index Threshold	82.91	84.18	86.31	85.12	82.54
Scenario 2	Budget ($)	2,717,884	56,816,066	922,738,289	1,131,290,109	4,865,243,918
	Health Index Threshold	82.91	84.18	86.31	85.12	82.54
Scenario 3	Budget ($)	3,136,021	65,557,000	1,064,698,026	1,305,334,742	5,613,742,982
	Health Index Threshold	83.56	84.98	87.01	85.81	83.17

other 54 bridges have two candidates each. Table 22 shows the costs, benefits, and post-improvement health indices of these other 54 bridges in the network. Highlighted in Table 22 are the solutions (i.e., selected candidates) of each method.

The IUC heuristic orders candidates in decreasing order of IUC values and therefore selects the candidate for Bridge 69, which has a high IUC. CPLEX, in contrast, selects the candidate for Bridge 74, which has a smaller IUC than Bridge 69 but has a higher benefit and higher cost. Also, the *integrality gap* (or the leftover funds, defined as the difference in the total budgeted amount and the total expenditure for the selected candidates) is reduced to a greater degree by the CPLEX than by the IUC heuristic, as shown in Figure 37.

The total benefits of the IUC solution are 0.02% lower than that of CPLEX, and the average health index of the network of the IUC solution is 0.01% lower than that of CPLEX.

3.2.5.10 Evaluating IUC Heuristic for Concavity Assumption

IUC heuristic implicitly assumes that the benefit-cost curve for a given bridge is concave. This assumption and its effect on the solution was an important point in the research that aroused considerable discussion. Some concerns that were raised include the possible adverse impact of the IUC screening

process on the quality of the solution and overall network benefits. In other words, is it possible to arrive at a better solution by including those candidates that do not conform to the concavity assumption? Could some jurisdiction have a higher percentage of such bridges? Do the screened-out candidates have any special characteristics? This section discusses the presence and effect of any candidates that do not conform to the concave-shaped benefit-cost curve in order to address the questions identified above.

The screening process for the IUC heuristic is based on diminishing marginal returns. As discussed previously, the law of diminishing marginal returns is a concept describing the economic relationships among alternative uses of the same investment capital. In economic theory, this law states that as the amount of any one input is increased, holding all other inputs constant, the amount that the output increases for each additional unit of the expanding input will generally decrease. In this context, if the costs and benefits of alternative investments for a bridge are plotted on a graph, a typical curve is concave, where each incremental investment produces a less-than-proportionate increase in benefits.

An investigation of the datasets containing the alternative bridge investment candidates was carried out to reveal the quantity of such candidates. Table 23 shows the numbers of candidates that do not satisfy the concavity assumption. As shown in the table, only a small percentage of candidates—less
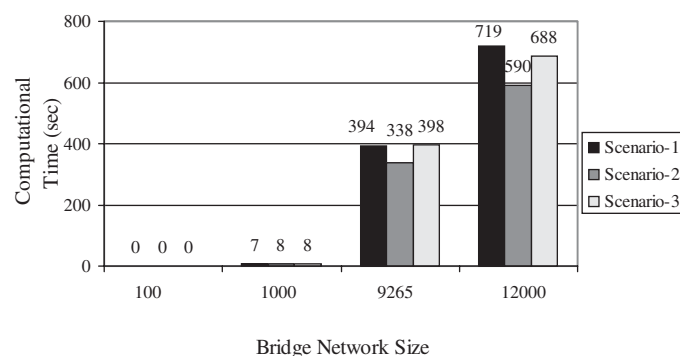
Figure 31. Computational time performance of the IUC heuristic for different constraint scenarios and network sizes.
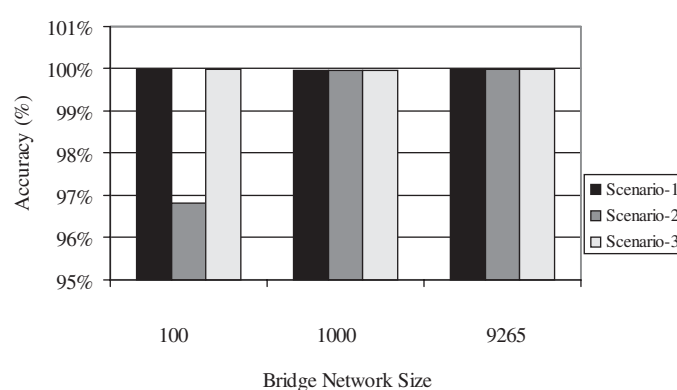
Figure 32. Accuracy performance of the IUC heuristic for different constraint scenarios and network sizes.
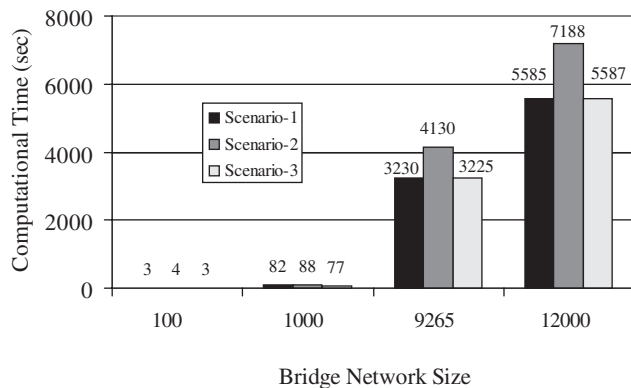
Figure 33. Computational time performance of the Lagrangian heuristic for different constraint scenarios and network sizes.

Figure 35. Average computational time performance of the IUC and Lagrangian heuristics.

than 4.8% of the inventory—are screened out. The next question addressed was whether this percentage of nonconforming candidates can be considered negligible. A detailed investigation was therefore carried out to quantify and test the effect of these candidates on the quality of the solution. Specifically, a series of computational experiments tested the performance of the IUC heuristic in terms of accuracy (i.e., quality of the solution) when the concavity assumption does not hold.

First, the true optimal solutions were computed for the entire bridge inventory (i.e., including the nonconforming candidates). The accuracy of the IUC heuristic was then tested by comparing this true optimal solution for pre-screened datasets (i.e., without assumption of concavity of utility-cost curve) with IUC solutions. The true optimal solutions were found using CPLEX for three different network sizes. Experiments were conducted for three different scenarios by varying the stringency of the budget and condition constraints.

A check was carried out to ascertain whether the CPLEX solution contained any of the screened-out candidates. It was found that the CPLEX solution did not contain any of the screened-out candidates. This means that the screened-out candidates did not compete well against the "normal" candidates (i.e., those that did not violate the concavity assumption). Furthermore, the average accuracy was 99.6%, which indicates that the IUC solution was very close to the true optimal solution. Therefore, the IUC screening process is not detrimental to the solution.

3.2.6 Conclusions and Recommendations

The network-level model aims to optimally select candidate projects from a networkwide candidate list to yield maximum network benefits subject to multiple constraints. The network benefit is measured with multiple criteria, and the constraints can be budgetary limitations and/or performance constraints. The network-level optimization model can be used to study the impact of various funding

Figure 34. Accuracy performance of the Lagrangian heuristic for different constraint scenarios and network sizes.
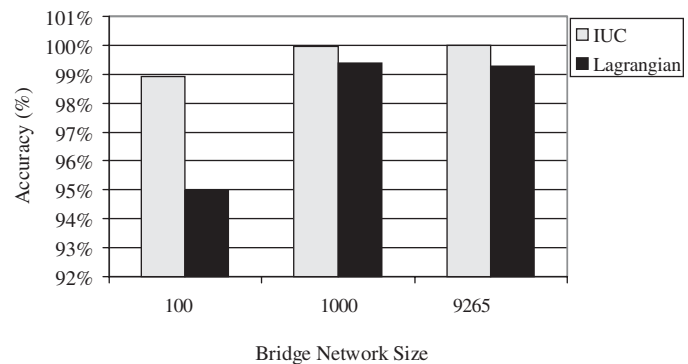
Figure 36. Average accuracy performance of IUC and Lagrangian heuristics.

Table 21. Ending health indices for bridges with only the do-nothing candidate.

Bridge ID	HI	Bridge ID	HI	Bridge ID	HI
9	64.5	34	86.2	59	95.7
12	72.4	35	89.9	60	90.3
15	76.7	39	89.9	61	76.9
16	81.2	40	87.3	62	63.4
17	81.8	43	92.3	70	83.1
20	84.2	44	78.8	73	82.1
21	84.3	45	90.4	82	78.9
22	87.7	46	91.1	83	81.7
23	87.6	49	90.8	86	85
25	88	50	85.1	87	87.1
26	88.7	51	91.6	88	76.3
29	73.9	52	91.7	91	83.5
30	92.3	53	78.8	92	86.8
31	92.4	57	87.8	94	85.1
32	70	58	86.6	99	80.7
33	70.9				
34	86.2				

HI = health index.

Table 22. Costs, benefits, and post-improvement health indices of the other 54 bridges in the network.

	Selected by IUC only
	Selected by CPLEX only
	Selected by both

Bridge ID	Cost of Candidate 2	Benefit of Candidate 2	HI - Do Nothing	HI - Candidate 2	IUC of Candidate 2
1	29065.2	15851.2	77.6	82	0.545
2	18287.3	10920.4	78.7	82.1	0.597
3	29791.3	1256.04	76	80.4	0.042
4	18284.6	4886.37	79.4	81.4	0.267
5	15457.1	21679.6	71.5	75	1.403
6	81043.2	60176.9	56.7	98.8	0.743
7	878170	604136	45.4	98.8	0.688
8	23435.6	20047.4	76.9	79.6	0.855
10	11011.1	14127.9	87.2	89.9	1.283
11	7980.39	4242.63	93.1	95.2	0.532
13	49002.6	10833	79.5	82.9	0.221
14	56276.8	78837.4	85.3	90.7	1.401
18	9916.87	33977.6	87.4	91.4	3.426
19	14502.2	10059.9	92.6	95.7	0.694
24	14898.1	219.17	87.8	90.3	0.015
27	59717.5	26271	91.1	94.9	0.440
28	30513.7	29379.8	91.4	96.6	0.963
36	751680	540718	83.1	98.8	0.719
37	751680	491025	87.3	98.8	0.653
38	40630.8	37755.1	92.5	97	0.929
41	5206.68	7383.46	86.6	87.9	1.418
42	4681.37	4277.72	87.1	88	0.914
47	13247.5	1426.5	84.7	87.5	0.108
48	37393.2	14624.9	83.9	86.8	0.391
54	54905.1	20704.7	74.1	77.6	0.377
55	22188.9	2310.16	82.9	86.1	0.104
56	25886.7	4198.69	81	84.3	0.162
63	64512	3222.53	63.2	98.6	0.050
64	94176	20397.8	81.2	98.8	0.217
65	109987	178676	67.8	99	1.625
67	15635.9	39503.8	85.3	91.3	2.526
68	32478.5	28637	85.6	90.4	0.882
69	8963.61	6138.93	75.8	78.1	0.685
71	35151.8	31631.7	82.9	88.6	0.900
72	18091.4	6774.86	75	79.8	0.374
74	12896.7	6574.02	73.7	77.2	0.510
75	27377.2	4374.48	77.3	81.6	0.160
76	30473	1059.6	74.2	77.2	0.035
77	30304.4	18010.5	72.3	76.8	0.594
78	28421.3	9390.91	83.9	88	0.330
79	50808.7	6322.93	82.6	87.7	0.124
80	46460.8	8371.22	83.4	88.7	0.180
81	17998.6	7090.39	84.9	88.1	0.394
84	12924.8	7217.52	72.4	76.4	0.558
85	17950.1	14343.6	83.8	88	0.799
89	29887.5	3907.28	75.4	82.3	0.131
90	28790.3	5438	86.7	90.4	0.189
93	12246	4847.19	85.7	87.9	0.396
95	33856.7	9178.39	85.8	89.8	0.271
96	33610.3	11785.5	86	90.2	0.351
97	42988.2	5598.21	86.2	90.4	0.130
98	41036.3	3745.94	81.3	85.5	0.091
100	136598	82028.1	69.3	98.8	0.601
66	37918.2	102040	67.6	77.7	2.691

HI = health index.

levels on network performance. It can also be used to estimate funding needed to achieve user-specified condition targets and acceptable risk levels. The problem was formulated as a multi-choice multi-dimensional knapsack problem (MCMDKP).

A strategy was developed to select an appropriate heuristic for implementation in the final software product. This strategy was based on evaluating the heuristics in terms of computational speed, accuracy, robustness, and simplicity. Based on this strategy, a number of computational experiments that were progressively designed on the basis of the preliminary results were conducted. In order to test the accuracy of the heuristics, it was necessary to obtain the optimal solution. This was done using an exact method that was implemented using CPLEX Concert technology. The heuristics were coded in Excel using Visual Basic for Applications (VBA).

The datasets used for the experiments were generated using Florida DOT's Project Level Analysis Tool, which uses Florida DOT's actual inventory and fully realistic deterioration, cost, and other models. Network sizes of 100; 1,000; 9,265; 12,000; and 50,000 were constructed. Three types of candidates were generated: do-nothing, preservation, and replacement. The benefit of any candidate was computed by relating its life-cycle cost to that of the do-nothing candidate. These datasets provided a simple but useful representative example of an MCMDKP. The network optimization then aimed to maximize total network benefits subject to a budget constraint and a health index performance constraint. Screening strategies and domain knowledge were advantageously used to improve computational tractability of the optimization problem.

The preliminary results analyzed the effect of bridge network size on computational speeds of the heuristics. The implemen-

Total $ Spent

Figure 37. Total cost of candidates selected by IUC and exact (CPLEX) algorithms.

tations on different network sizes showed that the computational times for the Lagrangian heuristic grow with the network size much more rapidly than for the IUC heuristic. These results were consistent with the theoretical computational complexities of the heuristics. The IUC heuristic was found to be the fastest. More detailed experiments were designed to test the performance of the IUC and Lagrangian heuristics under varying input conditions.

The accuracy of the heuristics was tested for networks of 100; 1,000; and 9,265 bridges. Accuracy was defined as how close the objective function value (i.e., total network benefits) was to the true optimal solution. The average accuracies were found to be 99.97% and 99.99% for the IUC and Lagrangian heuristics, respectively, both considered excellent. The average network health index was within 0.01% and 0.07% of the optimal solution for the IUC and Lagrangian heuristics, respectively.

The next step was to test the heuristics for robustness—that is, the sensitivity of performance of the heuristic to changing network parameters, such as budget and health index threshold value. Sensitivity analyses were carried out to test the heuristics under three different scenarios by changing the network parameter values. The computational times for both the IUC and Lagrangian heuristics were found to be similar for different

scenarios. However, the Lagrangian heuristic was found to be more sensitive to parameters than the IUC heuristic was. This was somewhat counterintuitive at the beginning. However, a close examination of the underlying heuristic structures and the results of the computational experiments indicated that although the structure of the Lagrangian heuristic is well suited for multi-constraint problems, it does not capture the multi-choice aspect of the problem very well. Also, for the specific problem of bridge network optimization that we have and for the realistic problem instances, the IUC heuristic is more accurate than the Lagrangian heuristic. The average accuracy of the IUC heuristic was 99.62%, and that of the Lagrangian heuristic was 97.88%.

The computational experiments provided useful insights into the optimization heuristics, reflecting appropriateness and applicability to the bridge network optimization problem. The IUC heuristic performed excellently in terms of both computational speed and accuracy. The average computational time for a 12,000-bridge network (which is the average database size for a state's inventory, including local bridges) was found to be 666 seconds, or about 11 minutes. The average accuracy was found to be 99.62%. This is very close to the true optimal solution. The IUC heuristic was also found to be very robust in terms of its performance under scenarios with different network parameters. The IUC heuristic is also relatively easy to understand, and its underlying concept (the IBC heuristic) is well known in the bridge community. CPLEX Concert, the exact algorithm, also performs excellently because it is very robust and provides true optimal solutions. However, the computational times are much higher for CPLEX Concert than for the IUC heuristic (for example, in the case of the 9,265-bridge inventory, it is

Table 23. Numbers of candidates that do not satisfy the concavity assumption.

Bridge Network Size	Total # of Candidates	# of Candidates screened out due to concavity assumption
100	160	5
1000	1605	76
9265	13,896	655

78% higher). Furthermore, CPLEX Concert is a commercial technology that would require purchase of a software license. The expenditure of agency funds for the CPLEX Concert license fee may not be justified given that the extra gain in average accuracy is less than 0.5% over the IUC heuristic.

Another nonquantitative consideration comes into play as we consider how the optimization algorithm will fit into the software deliverable. The computation times described here are all for a total re-computation of the optimal solution. However, in the proposed software, most of the common usage scenarios do not require total re-computation if an incremental computation is possible. For example, if we want to plot utility versus cost, we need only the change in utility that would result from small changes in budget; we do not need to redo the entire optimization for each budget level. The IUC method provides a simple and practically instantaneous way of computing such information by virtue of the underlying data structure of its sorting algorithm. Incremental adjustments are also possible with the Lagrangian method and the CPLEX Concert software, but these are more complex to develop and not as fast.

On the basis of these findings, the research team concluded that the IUC heuristic is most suited for the network optimization problem in bridge management. The research team therefore proceeded to implement this heuristic in the software product.

3.3 Bridge-Level Optimization

3.3.1 Preservation Models

The preservation model framework consists of a collection of submodels that work together to serve the life-cycle cost framework. Some of the models are used in more than one stage of life-cycle costing. Certain models are adapted directly from Pontis and are intended to give the same results as Pontis, while others are developed specifically for project-level analysis. The main models are as follows:

- **Deterioration:** Predicts future conditions of elements on a bridge on the basis of the most recent inspection and a possible candidate implemented during the program horizon.
- **Action effectiveness:** Predicts the outcome of an intervention on each element or on the bridge as a whole. This is somewhat more generalized than the approach in Pontis because actions can be associated with any performance measure and not just condition. However, the preservation discussion in this section focuses just on condition.
- **Cost estimation:** Computes the direct cost of preservation work on the bridge by aggregating over all elements using their predicted quantities and unit costs. Indirect costs are calculated separately.

- **Economic failure:** Describes the limiting behavior of an element if deteriorated conditions are not corrected, in the form of unplanned actions (e.g., emergency repairs).
- **Long-term cost:** Estimates life-cycle costs beyond the end of the program horizon based on conditions predicted at the end of the horizon.

All of the models herein described are intended to work in concert with the AASHTO CoRe Element Guide and agency customizations such as non-CoRe elements and sub-elements. A condition state inspection that separates severity from extent of deterioration is essential for some of these models.

3.3.1.1 Deterioration Model

As the primary cause of preservation needs in a bridge inventory, deterioration is the main driver of the life-cycle cost analysis in the bridge-level model. The system uses the same Markovian model as Pontis.

A Markovian model assumes that the probability of making a transition from one condition state to another depends only on the initial state and not on past conditions or any other information about the element. Thus, the model is expressed as a simple matrix of probabilities, which can be manipulated by matrix multiplication.

Inspections and predicted conditions are all assumed to occur at the beginning of the year before any action is taken. The model predicts conditions described as "base year" using the integer number of years from the inspection date to January 1 of the first program year. The agency sets the base year as a configuration option and may opt to use the condition from the most recent inspection as the base year condition and deactivate the intervening deterioration model.

For example, if the base year is 2007, and the latest element inspection on the subject bridge is August 1, 2004, then the deterioration model will simulate 2 years of deterioration to estimate conditions as of January 1, 2007. This discipline of timing conventions is established to make sure deterioration is applied consistently regardless of inspection intervals and schedules.

The left side of Table 24 shows a typical transition probability matrix. The rows are condition states at the beginning of the year, and the columns are condition states at the end of the year. Therefore, the table indicates, for example, that for all units of the element starting the year in state 1, 96.93% will remain in state 1 after one year, and 3.07% will go to state 2.

The rows of a transition probability matrix must always sum to 100%. (For this purpose, the final row includes the failure probability.) Since the matrix describes the change in condition when no rehabilitation action is taken, the probabilities from any condition state to superior states should always be

Table 24. Example transition probability matrix.

TRANSITION PROBABILITIES						PREDICTED CONDITIONS				
	ToState						Year	State1	State2	State3	State4
FromState	1	2	3	4		2001	100.00	0.00	0.00	0.00
1	96.93	3.07	0.00	0.00		2002	96.93	3.07	0.00	0.00
2	0.00	96.37	3.63	0.00		2003	93.95	5.93	0.11	0.00
3	0.00	0.00	92.38	7.62		2004	91.07	8.60	0.32	0.01
4	0.00	0.00	0.00	87.06		2005	88.27	11.09	0.61	0.03
						2006	85.56	13.39	0.96	0.08
Failure probability	12.94%					2007	82.94	15.53	1.38	0.15
All amounts in percent						2008	80.39	17.52	1.83	0.26
						2009	77.92	19.35	2.33	0.40
						2010	75.53	21.04	2.86	0.57
						2011	73.21	22.59	3.40	0.79

zero. After an element reaches the worst state, it is assumed to stay there for most purposes. However, there is a concept of element failure, described in a later section, that is used in the life-cycle cost analysis as a penalty for allowing an element to stay in the worst state.

Conditions at any future period can be predicted with a Markovian model by simple matrix multiplication. The right side of Table 24 shows an example of how a starting position changes over 10 years. The condition in 2005, for example, was calculated from 2004 as follows.

State 1:

$$88.27\% = 91.07\% \times 96.93\%$$ (3-22)

State 2:

$$11.09\% = 91.07\% \times 3.07\% + 8.60\% \times 96.37\%$$ (3-23)

State 3:

$$0.61\% = 8.60\% \times 3.63\% + 0.32\% \times 92.38\%$$ (3-24)

State 4:

$$0.03\% = 0.32\% \times 7.62\% + 0.01\% \times 100\% \text{ (stays in state 4)}$$ (3-25)

Note that the sum for each year must be 100%. It is possible to derive transition probabilities if the median number of years between transitions is known. Often, this is an easy way to develop a deterioration model from expert judgment. If it takes T years for 50% of a population of elements to transition from one state to the next, then the probability, P, in a 1-year period of staying in the starting condition state can be calculated as follows:

$$P = 0.5^{(1/T)}$$ (3-26)

So, if it takes a median of 6 years to transition from state 1 to state 2, then the transition probability of staying in state 1 is 89%. If we assume that all the rest of the element deteriorates to state 2, then the transition probability from state 1 to state 2 is $(1 - P) = 11\%$.

3.3.1.2 Action Effectiveness Model

The action effectiveness model is similar to the deterioration model and similar to Pontis. Markovian transition probabilities are used to express the condition immediately following the action, assumed to be at the beginning of the implementation year. These probabilities are then multiplied by the do-nothing transition probability matrix to forecast farther into the future. Because scope items are expressed in terms of action types applied to multiple elements, action outcomes are expressed in the same way when performance measures are calculated. For example, each scope item can have a predicted health index and life-cycle cost calculated by aggregating over preservation actions applied to affected elements and condition states.

Table 25 shows an example where action type "MRR Concrete Elements" was applied to a reinforced concrete girder. Three of the condition states have MRR actions belonging to this action type. In state 4, there are two actions of this type, but replacement has the lower long-term cost as calculated in Pontis and is therefore selected for this example.

3.3.1.3 Cost Estimation Model

Each candidate has an initial cost that is assumed to occur at the beginning of the implementation year. For the do-nothing candidate, the initial cost is always zero. For auto MRR&I and custom candidates, initial cost is made up of two components:

- Direct costs, which are always the sum of the direct costs of the scope items in the project.

Table 25. Example of the action effectiveness model.

ELEMENT 110 - R/Conc Open Girder (Environment 4)
FailProb: 12.94%

INPUTS BY CONDITION STATE AND ACTION

FromState	Action	Action Type	Prob1	Prob2	Prob3	Prob4	LTCost
1	>> 0 DN	Do nothing	96.93	3.07	0.00	0.00	119
2	0 DN	Do nothing	0.00	96.37	3.63	0.00	322
	>> 1 Seal&Patch	MRR - Concrete Elemen	47.25	50.89	1.86	0.00	370
3	0 DN	Do nothing	0.00	0.00	92.38	7.62	788
	>> 1 Cln&Patch	MRR - Concrete Elemen	61.79	34.48	3.53	0.19	860
4	0 DN	Do nothing	0.00	0.00	0.00	87.06	1925
	1 Rehab	MRR - Concrete Elemen	64.22	28.54	6.77	0.48	1857
	>> 2 Replace	MRR - Concrete Elemen	96.93	3.07	0.00	0.00	1332

Do-something probabilities in this table include the next year's deterioration, as in Pontis.
LTCost = Long-term cost calculated in Pontis

Apply action type <MRR Concrete elements> to this element
RESULTS BY YEAR

Year	State1	State2	State3	State4
Base 2005	0.00	21.51	58.56	19.94
2006	0.00	20.73	54.88	24.40
2007	0.00	19.97	51.45	28.58
2008	0.00	19.25	48.25	32.50
2009	0.00	18.55	45.27	36.18
2010	71.80	26.16	1.94	0.09
2011	69.60	27.42	2.74	0.23
2012	67.46	28.56	3.53	0.44
2013	65.39	29.59	4.30	0.71
2014	63.39	30.53	5.05	1.04

Conditions are shown at the end of the indicated year.

- Indirect costs, consisting of maintenance of traffic (MOT), mobilization, and engineering (including design, construction engineering, and inspection).

Typically, cost components that vary directly with quantity of work go into the direct unit cost, and remaining cost components are considered indirect.

Pontis provides the ability to give each MR&R action separate unit costs for direct and indirect costs. This has rarely been used so far, because the distinction does not affect network-level analysis. For bridge-level analysis, however, it is highly desirable to begin to take advantage of this feature. The present study does not incorporate separate direct and indirect cost factors. However, individual agencies could generate such factors using their maintenance management system data.

Table 26 presents a set of guidelines used by Florida DOT for bridge management work (Sobanjo and Thompson 2004). The software provides an option to decompose the Pontis MRR unit costs into their components using these factors. Then at the bridge level, an indirect cost model is based on the types of work performed, number of lanes, and other relevant factors.

Individual bridges may have considerations that cause their direct costs to be higher or lower than average. Therefore, each scope item in a custom candidate offers the ability to override

the default cost. Indirect costs may also be entered manually to override the cost value provided in the model.

3.3.1.4 Economic Failure Model

The concept of "failure" is a distinctive and fundamental part of the Pontis analytical framework. As such, the present study strove to appropriately incorporate this concept into the bridge-level model. In the Pontis network optimization, the role of the failure concept is to help develop policies that

Table 26. Florida DOT guidelines for indirect costs.

Indirect cost by work type (% of direct cost)		
Type or element group	MOT	Mobilization
Painting	20	20
Railing	25	10
Joints	25	10
Deck	25	10
Superstructure	20	10
Bearing	15	10
Substructure	10	15
Culvert	20	10

MOT = maintenance of traffic.

generally do not permit failure. At the bridge level, however, factors such as constrained funding, the typical rule of 10 years of deliberate inaction following a project, and the normal uncertainty in deterioration prediction present the possibility that failure could occur in isolated cases. So, the model needs a valid way of defining what realistically we mean by "failure" and quantifying the effect of failure on life-cycle costs, particularly the effect on road users and the cost of emergency repairs or replacement.

The Pontis long-term cost equation is not in itself sufficient to create a bounded model because the cost-minimizing solution to the equation is to choose the do-nothing action, whose cost is zero. If this policy were to be followed, bridges would not merely gather in the worst-condition state, but would proceed to an even worse state, denoted as the failure state. The failure state is defined as an intolerable condition, even worse than any that would normally be observed in an inspection, where the element no longer satisfies its intended purpose. If this happens, a life-cycle cost penalty is incurred to reflect the user disruption and the cost of mandatory repairs. Because of the failure cost element in the Pontis network optimization model, the economic failure model has three requirements:

- It must prevent the optimization from recommending a do-nothing action in the worst defined condition state, so failure cannot occur.
- It must reflect the relative importance of each element to the continuing functionality of the bridge, or the relative level of damage that would be caused if the element were to fail.

- It must reflect the impact of element failure on the road users.

A research task on failure costs (Thompson 2003) was completed by Florida DOT to calculate the minimum failure cost (assumed to be an agency cost) required to satisfy the first requirement above, and the maximum agency and user failure costs to satisfy the second and third requirements. The maximum cost was estimated from "failure scenarios," descriptions of the economic impact of an element failure if it were to occur. Table 27 shows an example of failure scenarios for a few selected elements.

For the current study, it is assumed that each agency will develop failure unit costs in a manner appropriate to the way it uses its bridge management system. Agencies that do not use Pontis or prefer not to incorporate failure risk into their life-cycle cost analysis have the option of defining a mandatory action for the failed state, which would have the same effect as incorporating the failure cost. Alternatively, they can use the minimum tolerable conditions feature of the optimization model (described earlier).

In the bridge-level life-cycle cost analysis, failure is a part of the probabilistic outcome prediction for any candidate. The model assumes that failure, though improbable under normal conditions, can happen on specific bridges. It is not necessary to predict the specific bridge elements that would fail; it is only necessary to recognize the possibility of failure and include the expected value of the life-cycle cost impact in the analysis for every bridge. If a program is well funded and conditions remain relatively good, the failure cost should be small and insignificant. However, if a program is poorly funded, if dete-

Table 27. Examples of element economic failure scenarios.

Element key	Short name	Number of bridges	Total quantity in inventory (metric)	Average bridge deck area (sq.m)	Average ADT on bridge	Average trucks ADTT on bridge	Average detour distance (km)	Average detour time (hrs)	Element repl cost per unit (metric)	Failure Scenario	Duration of impact (days)	Risk of impact (pct)	Agency cost of failure	User cost of failure
298	Pile Jacket Bare	648	16130	2881	11130	1053	18.0	0.684	500.00	Piles go to severe environment	30	100	500	0
299	Pile Jacket/Cath Pro	64	1866	5335	15667	1753	16.1	0.232	1000.00	Piles go to severe environment	30	100	1024	0
300	Strip Seal Exp Joint	455	22619	3343	16856	1366	6.7	0.101	400.28	All elements go to severe environment	30	100	498	3694
301	Pourable Joint Seal	5556	350819	1286	15431	1629	13.2	0.276	242.79	All elements go to severe environment	30	100	320	8411
302	Compressn Joint Seal	2567	211146	2607	24204	2709	7.7	0.132	498.71	All elements go to severe environment	30	100	558	5501
303	Assembly Joint/Seal	312	12316	8533	25046	2554	7.3	0.110	1378.02	All elements go to severe environment	30	100	1501	9803
304	Open Expansion Joint	400	38947	5374	14170	1315	14.1	0.238	997.42	All elements go to severe environment	30	100	1047	4098
310	Elastomeric Bearing	4523	350227	2632	22905	2481	8.2	0.147	330.00	Replace entire element; all trucks detoured	1	100	330	196
311	Moveable Bearing	1240	53438	3401	21344	2362	11.2	0.186	300.00	Replace entire element; all trucks detoured	1	100	300	330
312	Enclosed Bearing	9	52	5383	25558	3225	2.6	0.063	3700.00	Replace entire element; all trucks detoured	1	100	3700	1327
313	Fixed Bearing	1102	38722	2680	21575	2407	11.7	0.191	1340.00	Replace entire element; all trucks detoured	1	100	1340	563
314	Pot Bearing	136	2935	7523	20312	1715	6.0	0.079	1800.00	Replace entire element; all trucks detoured	1	100	1800	293
315	Disk Bearing	1	0	1853	4260	213	1.0	0.014	300.00	Replace entire element; all trucks detoured	1	100	300	0
320	P/S Conc Appr Slab	5	6	1373	9109	505	3.4	0.055	13000.00	Replace entire element; all trucks detoured	7	100	13000	6940
321	R/Conc Approach Slab	7545	15380	1817	17341	1884	10.6	0.215	13000.00	Replace entire element; all trucks detoured	7	100	13000	55238
330	Metal Rail Uncoated	222	25968	1567	9993	683	13.5	0.283	423.25	Increase in accident severity user cost	30	100	423	54
331	Conc Bridge Railing	6740	2E+06	2177	18268	1976	11.3	0.231	1640.50	Increase in accident severity user cost	30	100	1641	43
332	Timb Bridge Railing	203	8510	149	864	49	33.3	0.897	109.91	Increase in accident severity user cost	30	100	110	13
333	Other Bridge Railing	2621	351205	1478	15754	1352	12.0	0.225	136.86	Increase in accident severity user cost	30	100	137	74
334	Metal Rail Coated	363	74450	1075	7300	570	15.3	0.284	423.25	Increase in accident severity user cost	30	100	423	60

rioration proceeds more rapidly than normal, or if the bridge structure is particularly vulnerable to natural or human-made disasters, the failure becomes more likely, and the failure cost is high.

When failure occurs on an element with some nonzero probability, the model adds a failure risk penalty to the life-cycle cost. This penalty is calculated as follows:

$$\text{Failure risk cost} = \text{WorstProb} \times \text{FailProb} \times \text{FailCost} \\ \times \text{Quantity} \qquad (3\text{-}27)$$

where

WorstProb = probability of the worst defined condition state, calculated using the deterioration model;

FailProb = failure probability, a part of the deterioration model;

FailCost = unit failure cost, or the cost of a mandatory action in the failed state; and

Quantity = total element quantity, summed over all states in the most recent inspection.

After any action is taken, during the waiting and rest periods of the life-cycle cost analysis, failure risk cost occurs each year. This is based on the quantity predicted to be in the worst-condition state at the beginning of the preceding year. This allows one additional year to make the transition to the failed state. Each of these costs is then discounted to present value.

The do-nothing candidate has failure risk costs for every year of the program because work is assumed to occur only after the program horizon. Table 28 shows an example of the analysis for the do-nothing case. The failure risk cost in this table is the amount added to total risk each year—that is, the added contribution to life-cycle costs as long as conditions are not improved.

3.3.1.5 Long-Term Cost Model

In the bridge-level optimization framework, it is possible in theory to extend the program horizon out in time as far as desired. However, execution time for the algorithm grows exponentially with the length of the program horizon, and agencies seldom have any mandate to program bridge work more than 10–20 years into the future. After that, sources of uncertainty increase, thereby reducing the value of any analysis for periods beyond that horizon.

However, life-cycle costs remain significant well beyond the end of the normal program horizon. Most agencies use relatively low real discount rates of 5% or less. Therefore, it is important in a valid life-cycle cost model to have some reasonable indication of the magnitude of future costs, sensitive to conditions at the end of the program horizon. If conditions are left relatively high at the end of the program, then long-term costs should be low, and vice versa.

Pontis contains an ideal tool for estimating long-term costs, in the form of its optimization equation. In Pontis, long-term cost, L_{ia}, is an estimate of life-cycle cost over an infinite time horizon, calculated separately for each present condition state, assuming that a given policy is followed. It is calculated as follows:

$$L_{ia} = C_{ia} + \alpha \sum_j P_{aij} L_{j\bar{a}(j)} \qquad (3\text{-}28)$$

where

C_{ia} = unit cost of action a (fixed + variable) when the element is in state i;

α = discount rate for costs incurred 1 year in the future;

P_{aij} = transition probability of an element to be in state j in 1 year given state i and action a this year (Pontis deterioration model);

Table 28. Example failure risk analysis.

ELEMENT 110 - R/Conc Open Girder (Environment 4)

Quantity:	129 m
Failure cost:	697 $/m
Failure probability:	12.94 %

Do-nothing deterioration results by year

Year	State 1	State 2	State 3	State 4	Failure risk	Failure cost
2007	0.0	24.0	71.3	4.7	0.0	0
2008	0.0	22.1	66.1	11.8	0.6	541
2009	0.0	20.3	61.3	18.4	1.5	1371
2010	0.0	18.7	56.8	24.5	2.4	2140
2011	0.0	17.2	52.6	30.2	3.2	2853
2012	0.0	15.8	48.7	35.5	3.9	3513
2013	0.0	14.6	45.1	40.3	4.6	4125
2014	0.0	13.4	41.8	44.8	5.2	4692
2015	0.0	12.3	38.7	49.0	5.8	5217
2016	0.0	11.3	35.8	52.9	6.3	5702
2017	0.0	10.4	33.1	56.5	6.8	6152

68

$\bar{a}(j)$ = optimal action for state j (the action giving the lowest long-term cost); and

$L_{j\bar{a}(j)}$ = long-term cost that would be calculated next year if state j occurs and if the optimal action for that state is selected (calculated recursively by the same equation).

The long-term cost equation is recursive because it depends on a term that itself is calculated according to the same equation. It is not circular, however, because the long-term cost term on the right side is for 1 year later than the left side. When fully expanded, the equation is potentially an infinite series, because the time horizon of the analysis is not strictly limited. However, because of discounting, the contribution of each subsequent term is less than the previous one and ultimately approaches zero.

Pontis simplifies the problem by assuming that in the long term, the equation reaches a steady state in which the conditions and actions remain in the same proportions from 1 year to the next. The probability of any given state in year t is equal to the probability of the same state in year $t+1$. In other words, for each meter of girder moving out of a particular condition state, another meter moves in to replace it.

The bridge-level model developed in the present study assumes that the Pontis optimal policy is followed and that, therefore, the long-term cost as calculated above is incurred, starting at the end of the final rest period for a candidate. Long-term costs increase with condition state. A full example of Pontis optimization inputs and outputs is shown in Table 29. If the intervention and its subsequent rest period (typically 10 years) of deterioration leave a large fraction of an element in the worst state, then it is likely that further major work will be needed in the near future, and long-term costs will therefore be high.

In effect, the model relies on Pontis to perform the life-cycle cost analysis for all subsequent work. In Pontis, this cost is typically updated whenever the deterioration and cost models are modified, so the Excel program merely reads the result from actmodls.ltcost in the Pontis database. If an agency does not have Pontis, it can obtain long-term unit costs either from an in-house study of another agency with similar conditions or from simulation for a long period (say 100 years) of deterioration and consistent state-responsive actions.

3.3.2 Functionality Models

Each bridge is examined for deficiencies that could affect the level of service provided to road users. When such deficiencies are found, the economic consequences, in terms of user costs, are estimated and added to the life-cycle cost of the structure. Functional improvements may be undertaken to eliminate these user costs. Three types of functional needs are modeled:

- **Deficient roadway width,** which works together with deficient approach alignment to create excess accident risk relative to a bridge constructed according to design standards. A width deficiency is recognized if the bridge roadway width is less than the required width. Required and design widths can be calculated as follows:

Required width $= (2 \times \text{desired shoulder width})$

$$+ (\text{number of lanes} \times \text{desired lane width}) \quad (3\text{-}29)$$

Design width $= (2 \times \text{design shoulder width})$

$$+ (\text{number of lanes} \times \text{design lane width}) \quad (3\text{-}30)$$

Table 29. Example of Pontis optimization inputs and outputs.

ELEMENT 234 - R/Conc Cap (Environment 3)
FailCost $11846.00/m.; FailProb: 9.24%

FromState	Action		Sub-category	ToProb1	ToProb2	ToProb3	ToProb4	VarCost	FixCost	LTCost
1	>> 0	DN	000-Do nothing	38.00	2.00	0.00	0.00	0.00	0.00	30.09
	1	Misc	403-Patch minor spalls	38.00	2.00	0.00	0.00	85.07	29.77	204.33
2	>> 0	DN	000-Do nothing	0.00	36.46	3.54	0.00	0.00	0.00	327.18
	1	Seal&Patch	403-Patch minor spalls	46.55	51.59	1.86	0.00	109.37	38.28	362.22
3	>> 0	DN	000-Do nothing	0.00	0.00	94.55	5.45	0.00	0.00	813.62
	1	Cln&Patch	303-Clean rebar and patch	39.20	51.44	8.95	0.41	486.07	170.13	925.05
4	0	DN	000-Do nothing	0.00	0.00	0.00	90.76	0.00	0.00	2,418.84
	1	Rehab	203-Rehab concrete	61.25	37.42	1.33	0.00	1,215.19	425.31	1,819.51
	>> 2	Replace	141-Replace beam	98.00	2.00	0.00	0.00	1,117.97	331.29	1,599.35

Actions denoted by >> are optimal (lowest long-term cost) for the condition state.
The 'ProbN' columns are transition probabilities after the indicated action is taken, to each possible resulting state.
VarCost is the variable (direct) unit cost of the action.
FixCost is the fixed (indirect) unit cost of the action.
LTCost is the long-term cost from the Pontis network optimization.

Bridges under a given length threshold have required width based on the approach-road width. The desired widths are established by the agency as level-of-service standards.

- **Impaired vertical clearance,** forcing certain trucks to find an alternate and presumably longer route. A deficiency is identified by comparing roadway vertical clearance with level-of-service standards.
- **Inadequate load capacity,** forcing certain trucks to find an alternate and presumably longer route. A deficiency is identified by comparing bridge operating rating with level-of-service standards.

A bridge may have any or all of these types of needs. Whenever a functional need is established, a deficiency flag is created. A scope item may then be created to respond to the deficiency.

Level-of-service standards may vary by traffic volume class. As traffic grows on a roadway, the roadway traffic volume may cross from one volume class to another from 1 year to the next. This may change its level-of-service standard for any or all deficiencies. A roadway that is not deficient in a low-volume class may be considered deficient when it moves into a high-volume class.

A functional need is relieved by performing a functional improvement or replacement. The possible actions are as follows:

- **Widening:** Relieves the width deficiency on the roadway on the bridge.
- **Raising:** Relieves the vertical clearance deficiency on all roadways under the bridge.
- **Strengthening:** Relieves the operating rating deficiency on the roadway on the bridge.
- **Replacement:** Relieves width deficiencies on and under the bridge, approach alignment deficiency, vertical clearance deficiency on and under the bridge, and operating rating deficiency.

The initial direct agency costs of these actions are calculated from a unit cost per square meter of deck. Functional improvements reduce or eliminate excess user costs. The excess amount of user costs is determined by comparing the existing bridge with a replacement bridge constructed according to the design standards. Functional improvements may not address all the functional needs. Therefore, it is possible for a bridge to continue to have functional deficiencies and user costs even after functional improvements have been performed.

The do-nothing candidate allows no actions, including functional improvements, to occur during the program horizon. Its life-cycle cost model assumes that either the excess user costs continue at a constant level forever after the end of the program or the structure is replaced just after the end of the

program, whichever gives a lower life-cycle cost. This is done to provide a consistent basis for comparing candidates.

3.3.2.1 User Costs and Traffic Growth

When a functional deficiency is found to exist on a bridge, the effect on road users is represented as a user cost. This user cost is calculated for each year of the deficiency, discounted to present value, and added to life-cycle cost. Functional improvements may eliminate certain user costs, so that they do not occur in the year of the action or any following years.

User costs are proportional to traffic volume, so they change each year because of traffic growth. In most cases, the model interpolates the traffic volume for any given year based on a constant growth rate between the most recent average daily traffic (ADT) and the future ADT provided in the roadway table of the bridge management system. The complete formula for forecasting average daily traffic, V_{ry}, is as follows:

$$V_{ry} = V_{r0} \times \left(\frac{V_{rn}}{V_{r0}} \right)^{\left(\frac{Y - Y_{r0}}{Y_{rn} - Y_{r0}} \right)}$$

(3-31)

where

V_{r0} = most recent actual traffic volume estimate (NBI item 29, "adttotal" in the roadway table),

Y_{r0} = year of most recent traffic volume estimate (NBI item 30, "adtyear" in the roadway table),

V_{rn} = forecast future traffic volume (NBI item 114, "adtfuture" in the roadway table),

Y_{rn} = year of forecast traffic volume (NBI item 115, "adtfutyear" in the roadway table), and

Y = current year.

If the most recent ADT is missing or zero, the effect is to turn off the entire user cost model. If any other variables needed for the traffic growth calculation are missing, the model uses the most recent ADT directly.

To provide a uniform basis for comparing candidates, the model adheres to the following conventions:

- User costs are discounted to present value as of the beginning of the base year, under the assumption that they occur at the beginning of the year with which they are associated.
- No user costs are recognized prior to the first year of the program (i.e., the base year).
- In the remaining years prior to the implementation year of an intervention, user costs are calculated based on existing functional deficiencies in the inventory.
- User cost savings resulting from a candidate begin in the same year as the intervention containing the functional improvement.

- Starting with the intervention year, up to the last year of the program, user costs are based on any uncorrected functional deficiencies. Certain deficiencies (e.g., roadway width under a bridge) can be corrected only by replacement. Also, custom candidates can exclude needed improvements.
- After the end of the program horizon, the model assumes that either the remaining excess user costs continue forever at a constant level (without traffic growth) or the structure is immediately replaced, whichever gives a lower life-cycle cost.

The final point here is important because user costs normally increase over time because of traffic growth. This raises a complication in the bridge-level analytical process. If the traffic growth rate exceeds the real interest rate, then user costs can grow unbounded until the bridge is improved or replaced. For life-cycle activity profiles that do not include replacement within the planning horizon, it is necessary to find a way to provide bounds to the growing user cost so that valid comparisons can be made using net present value analysis.

In the optimization approach, the decision to not relieve a deficiency is interpreted as a decision to delay the functional improvement until the next decision point. In other words, we continue to delay the improvement until finally it competes favorably with other programmatic needs such that it is able to be funded. If the improvement cannot be funded within the program horizon, it is modeled as occurring in the year after the end of the program horizon.

The entire computation of functional needs and user costs occurs in the functional improvement worksheet of the bridge-level software. It is performed separately by roadway on and under the bridge. All of the calculations are Excel worksheet formulas, so in advanced mode it is possible to follow the calculations and even to change them.

3.3.2.2 Accident Risk

Accident risk costs occur if the bridge roadway width is deficient according to the level-of-service standards. The model used for this cost was developed in Florida (Thompson, Najafi, Soares, and Choung 1999) using a statewide bridge database and a statewide crash database. Accident costs are calculated as follows:

$$\text{Weight\%} \times 365 \times \text{AADT} \times \text{AccCost}$$
$$\times (\text{CurrRisk} - \text{ImprRisk}) \quad (3\text{-}32)$$

where

Weight% = user cost weight given as an agency option,

AADT = annual average daily traffic for the year analyzed,

AccCost = unit cost per accident,

CurrRisk = current accident risk as described below, and

ImprRisk = improved accident risk as described below.

The accident unit cost was derived from the results of a literature review conducted for Florida DOT in 1998. It is typical in public policy analysis for regulatory and investment purposes to use the "willingness-to-pay" approach, which includes the tangible costs of an accident (such as medical care, property damage, insurance and legal expenses, employer costs, lost productivity, and travel delay) plus the intangible costs (such as pain and suffering, loss of enjoyment of life, inconvenience, and the premium associated with risk aversion). This methodology is well established in the safety literature.

Current and improved accident risks were calculated in the Florida study from a statistical regression model based on bridge characteristics. Accident risk is calculated as follows:

$$\text{Risk} = (\text{Coef}1 + \text{Coef}2 \times \text{Lanes} \times \text{Length} + \text{Coef}3$$
$$\times \text{Narrowness} \times \text{AADT}) \div 1000 \div \text{AADT} \quad (3\text{-}33)$$

where

Coef1 = 886 for urban arterials and −377 for all other roads,

Coef2 = 0.7323,

Coef3 = coefficient determined from Table 30,

Lanes = number of lanes on the roadway (NBI item 28, "roadway.lanes"),

Length = length of the bridge (in meters, NBI item 49, "bridge.length"),

Narrowness = lanes ÷ traveled way width (in meters, NBI item 51, "roadway.roadwidth"), and

AADT = annual average daily traffic for the year of analysis.

In this case, deck condition is NBI item 58, "bridge.dkrating." Approach alignment is NBI item 72, "inspevnt.appralign." Functional class (NBI item 26, "roadway.funcclass") is used in the determination of Coef1: values 14 and 16 are urban arterials.

This model is more recent than the one used in Pontis and gives a far more realistic response to changes in the inputs.

Table 30. Coefficient for deck condition and approach alignment, Coef3.

Deck condition	Good approach alignment (>6)	Bad approach alignment (<=6)
Good deck condition (>6)	0.3904	0.5031
Bad deck condition (<=6)	0.4531	0.7899
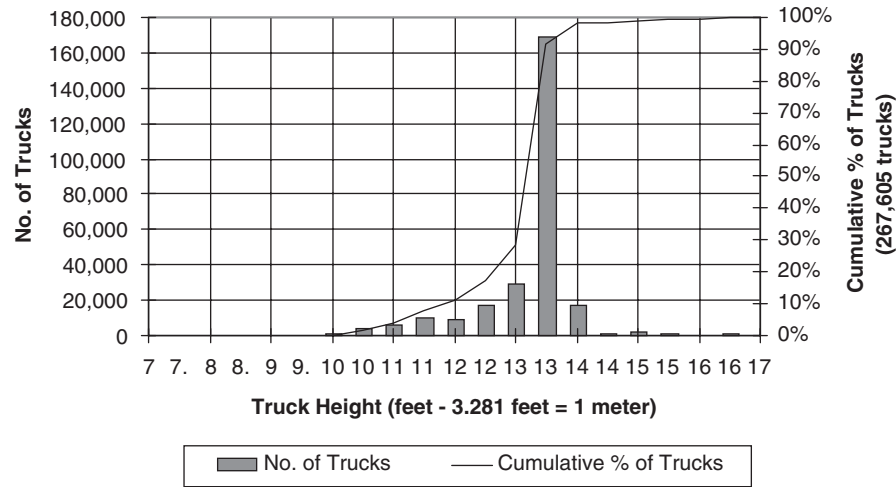
Figure 38. Example truck height histogram for Interstates.

3.3.2.3 Vertical Clearance

Truck detour costs occur if the roadway vertical clearance (NBI item 10, "roadway.vclrinv") is deficient according to the level-of-service standards. Trucks that are too high to pass under the bridge are forced to detour, presumably on a longer route. The user cost of this is calculated as follows:

$$\text{User Cost} = \text{Weight\%} \times 365 \times \text{AADT} \times \text{DetCost} \times \text{Truck\%}$$
$$\times (\text{CurrDet\%} - \text{ImprDet\%}) \qquad (3\text{-}34)$$

where

$\text{Weight\%}$ = user cost weight given as an agency option,
AADT = annual average daily traffic for the year analyzed,
DetCost = detour cost per truck (described below),
$\text{Truck\%}$ = fraction of trucks in the AADT (NBI item 109),
$\text{CurrDet\%}$ = percentage of trucks detoured by the current bridge, and
ImprRisk = percentage of trucks detoured by the improved bridge.

To determine the percentage of trucks detoured by any given vertical clearance restriction, a truck height histogram was developed in research conducted in Florida (Sobanjo et al. 2004). Truck height data were compiled from measurements taken by a "light curtain" vehicle profiler and a laser range finder at carefully selected locations around the state of Florida. Histograms and reverse cumulative frequency charts were drawn to represent the data collected. The reverse cumulative frequency chart gives the proportion (i.e., probability) of vehicles greater than a specified bridge height and therefore would need to detour.

To incorporate the truck height distribution into the user cost model, piecewise curvilinear functions were fitted to the

reverse cumulative frequency chart by regression. Sample results are shown in Figure 38 for Interstate roadways. Tables 31 and 32 show the equations derived for each segment of the piecewise functions. This is the only known truck height model developed specifically for bridge management system use.

3.3.2.4 Load Capacity

Truck detour costs occur if the bridge operating rating (NBI item 64, "bridge.orload") is deficient according to the level-of-service standards. Trucks too heavy to pass over the bridge are forced to detour, presumably on a longer route. The user cost of this is calculated as follows:

$$\text{User Cost} = \text{Weight\%} \times 365 \times \text{AADT} \times \text{DetCost} \times \text{Truck\%}$$
$$\times (\text{CurrDet\%} - \text{ImprDet\%}) \qquad (3\text{-}35)$$

Table 31. Truck height piecewise curves for Interstate roadways.

Height Range (Feet)	Percentage Detoured
< 9.65	100.00
9.65–13.00	$855.91 - 223.430x + 22.199x^2 - 0.74236x^3$
13.00–14.00	$(1.0956\text{E}+56)x^{-48.683}$
14.00–16.10	$14.567 - 0.9046x$
> 16.10	0.00

Table 32. Truck height piecewise curves for non-Interstate roadways.

Height Range (Feet)	Percentage Detoured
< 7.30	100.00
7.30–13.50	$-26.275 + 34.692x - 2.3894x^2$
13.50–14.00	$138.860 - 9.886x$
> 14.00	0.000

where

Weight% = user cost weight given as an agency option,
AADT = annual average daily traffic for the year analyzed,
DetCost = detour cost per truck (described below),
Truck% = fraction of trucks in the AADT (NBI item 109),
CurrDet% = percentage of trucks detoured by the current bridge, and
ImprRisk = percentage of trucks detoured by the improved bridge.

To determine the percentage of trucks detoured by any given operating rating restriction, a truck weight histogram was developed by research in Florida (Sobanjo et al. 2004). Histograms and reverse cumulative frequency charts of truck weights were generated using data from weigh-in-motion equipment located around the state of Florida. These histograms and charts were used as estimates of the fraction of trucks detoured by load restrictions.

To incorporate the truck weight distribution into the user cost model, piecewise curvilinear functions were fitted to the reverse cumulative frequency chart by regression. Sample results are shown in Figure 39 for Interstate roadways. Tables 33 and 34 show the equations derived for each segment of the piecewise functions.

3.3.2.5 Truck Detour Cost

Each time a truck is detoured, it experiences vehicle operating costs associated with the added detour distance and travel time costs associated with the added detour time. These costs are incurred per truck for vertical clearance and load capacity deficiencies as follows:

Table 33. Truck weight piecewise curves for Interstate roadways.

Weight Range (Pounds)	Percentage Detoured
< 10,000	100.00
10,000–80,000	$102.24 - (8.982\text{E-}05)x - (1.4336\text{E-}08)x^2$
80,000–91,000	$18.976 - (2.083\text{E-}04)x$
> 91,000	0.00

$$DetCost = VOC \times BypLen + TT \times BypLen/BypSpd \qquad (3\text{-}36)$$

where

VOC = unit vehicle operating cost per kilometer of detour,
BypLen = detour distance (in kilometers, NBI item 19, "roadway.bypasslen"),
TT = unit travel time cost per hour of detour, and
BypSpd = speed on the detour route (in kilometers per hour, not in the NBI, "roadway.det_speed").

If speed is missing in the bridge management system database, default speeds by functional class are specified on the configuration worksheet. The economic parameters VOC and TT were developed in a literature review in Florida (Thompson, Najafi, Soares, and Choung 1999).

3.3.3 Candidate Definition

The bridge-level model has four types of candidates, each with its own set of conventions for scoping and life-cycle activity profiles, as illustrated in Figure 40. Do-nothing has just one candidate, describing the case where no work is done in any year of the program horizon. Auto MRR&I, custom, and auto replace each have life-cycle activity profiles, costs, and
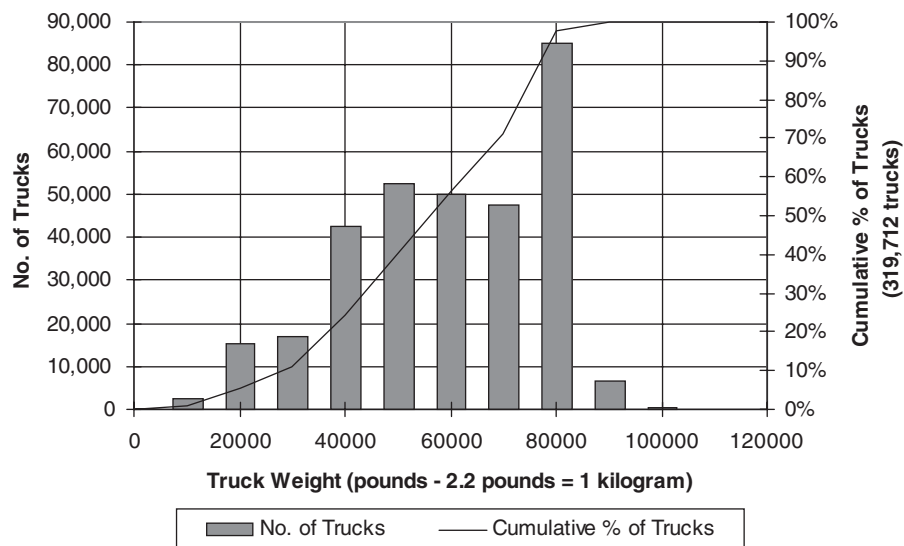
Figure 39. Example truck weight histogram for Interstates.

Table 34. Truck weight piecewise curves for non-Interstate roadways.

Weight Range (Pounds)	Percentage Detoured
< 3,700	100.00
3,700–85,000	$107.26 - (1.9743\text{E-}03)x + (6.5265\text{E-}09)x^2 + (2.2256\text{E-}14)x^3$
> 85,000	0.00

performance measures for each program year. Up to three custom candidates can be defined by the software user.

Each candidate, except do-nothing, may have scope items. Replacement candidates have only one scope item, bridge replacement. Custom candidates have the same set of scope items for implementation in each program year (but different costs and performance), while auto MRR&I candidates may have different scope items each year. Each scope item corresponds to one action type applied to the bridge. It is possible for auto MRR&I or custom candidates to have more than one scope item affecting any given element of the bridge, and each scope item may affect more than one element.

3.3.3.1 Do-Nothing

Do-nothing represents the "base case" of the bridge-level model, the scenario against which all other candidates are

compared. The economic benefits of any candidate are computed by subtracting its life-cycle cost from that of do-nothing. Benefits in terms of other performance measures are also computed relative to the do-nothing candidate. As the life-cycle activity profile diagram in Figure 41 shows, the do-nothing candidate has three cost components:

- **Failure risk,** representing the possibility of economic failure during the program period. It grows over time because of the continual nature of bridge deterioration.
- **User cost,** recognized if there are any functional deficiencies on the bridge. It grows during the program horizon because of traffic growth. User cost is assumed to continue without growth following the end of the program horizon, unless replacement gives a lower life-cycle cost.
- **Long-term cost,** representing future costs after the program horizon, as a function of ending conditions.
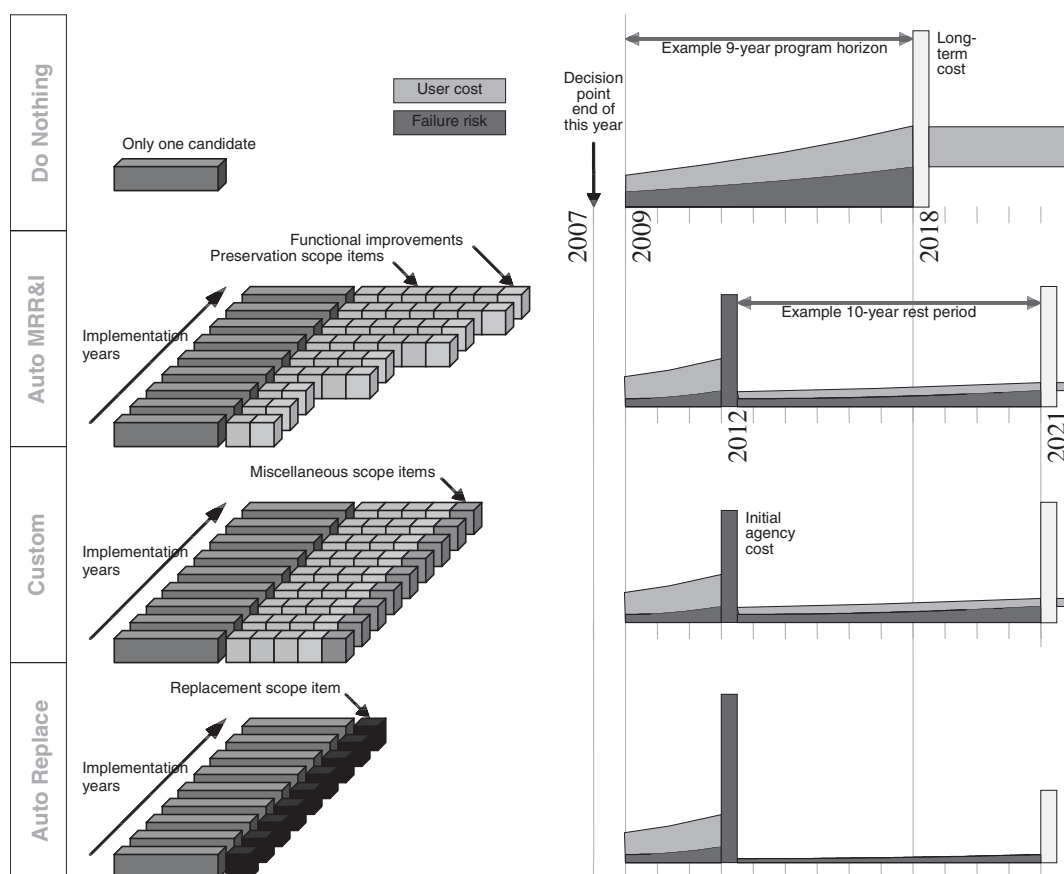
Figure 40. Candidate types and typical life-cycle activity profiles.
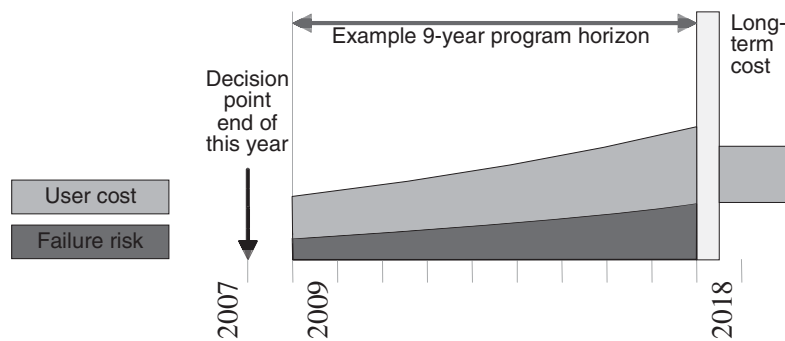
Figure 41. Life-cycle activity profile for do nothing.

The decision-support tool developed in the present study always generates a do-nothing candidate automatically and presents it to the maintenance planner for comparison with other candidates.

3.3.3.2 Auto MRR&I and Custom Candidates

Most of the analytical capability of the system is concerned with the auto MRR&I (maintenance, repair, rehabilitation, and improvement) candidate and custom candidates created from it, shown schematically in Figure 42. The bridge-level model has automated procedures to create a reasonable first-cut candidate scope based on preservation and functional needs forecast for each year of the program.

The engineer is encouraged to create custom candidates by making a copy of an auto MRR&I candidate in a particular year and then modifying it by adding, deleting, or editing scope items. Custom candidates can have any type of scope item other than bridge replacement, including a miscellaneous scope item with a user-defined cost and benefit. Preservation scope items can have a user-specified cost from which the model

calculates the benefit. These candidates contain all of the major cost components:

- **Failure risk,** representing the possibility of economic failure during the program period. This occurs in the program years before implementation and during the 10-year rest period following implementation.
- **User cost,** recognized if there are any functional deficiencies on the bridge. If the candidate contains any functional improvement scope items, the user cost is reduced or eliminated. (Only replacement is guaranteed to eliminate all excess user costs.) If user cost is not completely eliminated, it is assumed to continue without growth following the end of the program horizon. This long-term user cost is capped at the discounted bridge replacement cost.
- **Initial agency cost,** the actual cost of the work to be done during the first and subsequent interventions. This is assumed to occur at the start of each intervention's program year. Custom candidates can have user-specified cost components.
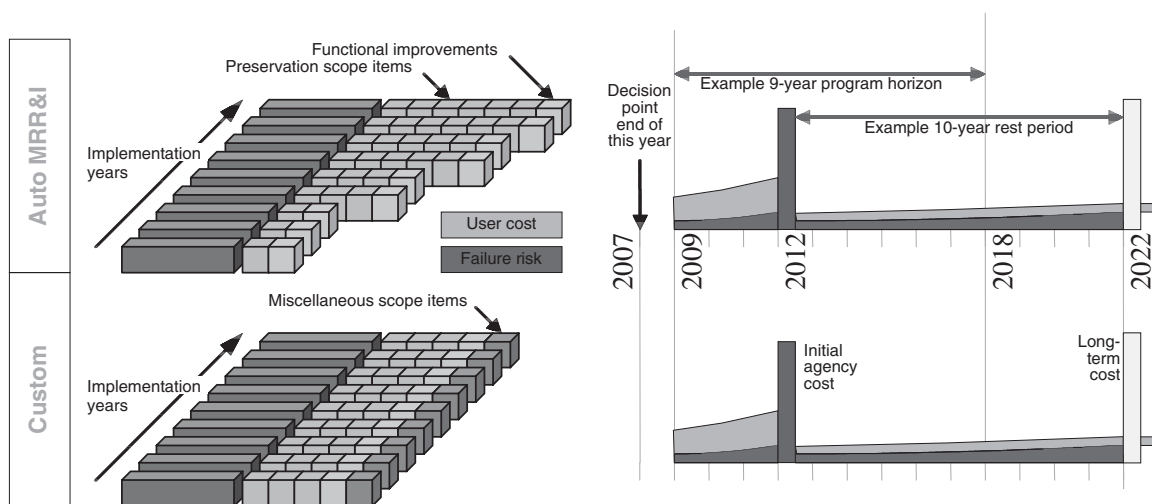
Figure 42. Auto MRR&I and custom life-cycle activity profiles.

- **Long-term cost,** representing future preservation costs after the final rest period, as a function of ending conditions.

The decision-support tool always creates an auto MRR&I candidate ("AutoMRRI") for each bridge to give the software user a consistent picture of forecast needs. The software user is given the option to create custom candidates by modifying AutoMRRI or another custom candidate.

Preservation Actions. Each element and condition state in the AASHTO CoRe Element Guide has a set of feasible MRR actions defined for it. These same actions and their preservation models are used in the bridge-level model.

The CoRe elements have a large number of defined MRR actions. This is important for distinguishing the costs and effectiveness of actions. However, it can greatly complicate the use of a bridge-level decision-support tool by an engineer working at this level of detail where a single intervention may contain 20–30 actions selected from 50–80 possibilities. To make the model more user friendly, similar types of actions are grouped together over multiple condition states and elements without losing the mathematical rigor of the underlying models. Suppose, for example, that a bridge has painted steel girders, floor beams, stringers, and bearings, all at various levels of deterioration. With enough extent of total deterioration, we would want the models to generate a total paint system replacement candidate, which automatically assigns the appropriate MRR action (with appropriate costs and benefits) for each condition state of each steel element. At lower levels of deterioration, we would still want the engineer to have the option to specify paint system replacement for a bridge as a whole, or specify a lesser painting approach (e.g., zone painting) and have the system automatically choose the correct MRR action and quantity for each state of each element.

As another example, suppose a bridge has reinforced concrete girders, abutments, caps, and columns, with varying degrees of spalling and rebar exposure. We would want the engineer to be able to specify concrete MRR for the bridge as a whole and have the model generate the appropriate MRR actions and quantities for the individual condition states of each concrete element. Or the engineer might specify a particular quantity of concrete repair for the bridge and want the model to assign this work to the most appropriate elements and condition states for the purpose of estimating future life-cycle costs, recognizing that the crew in the field will not limit itself to deterioration noted in the inspection, but will assess the concrete and repair whatever deterioration it actually finds.

Each scope item is defined as the application of one preservation action type to all the elements to which it is applicable. For example, we might have several steel elements in various combinations of condition states, but we present only one scope item for MRR of steel coatings on the bridge. Behind the scenes, this scope item is generated by selecting appropriate MRR actions for each condition state of each element, applying the corresponding unit costs and benefits, then adding up the results in dollar terms.

Action Types. To accomplish this sort of functionality, we use a classification scheme that recognizes when actions on two or more different elements are logically the same to the engineer (i.e., same crew skills, materials, and equipment requirements, but possibly differing units, costs, and action effectiveness). We also use a set of conventions for designating how different types of actions are to be used—for example, to specify that a coating replacement action is meant to be applied to the entire element and is priced accordingly, taking economies of scale into account.

The analytical database has an action type "Table" to hold this classification scheme. For the CoRe elements and actions, we use the following preservation action types:

TSR—Total System Replacement of:

- Bridge,
- Superstructure,
- Deck structure,
- Wearing surface,
- Steel coating,
- Expansion joints,
- Railings, and
- Bearings.

MRR—Maintenance, Repair, Rehabilitation or Replacement (depending on condition state) of portions of:

- Deck elements,
- Steel elements,
- Steel coating,
- Concrete elements,
- Timber elements,
- Expansion joints,
- Bearings,
- Railings, and
- Other elements.

A clear distinction is made here between actions that apply to whole quantities of elements (i.e., TSR) and actions that apply only to deteriorated portions of elements (i.e., MRR). Each of these action types represents a distinct approach to the preservation of a set of related elements on a bridge.

Every MRR action in the model has one corresponding MRR action type, as is the case in Pontis. The classification

Table 35. Action type classification example for element 107.

State	Action	MRR Action Type	TSR Action Type
Element 107 - Painted Steel Girder			
1	0 - Do nothing	Do nothing	
1	1 - Surface clean	Routine maintenance	
2	0 - Do nothing	Do nothing	
2	1 - Surface clean	MRR steel coating	
2	2- Clean and paint	MRR steel coating	
3	0 - Do nothing	Do nothing	
3	1 - Spot blast, clean and paint	MRR steel coating	
4	0 - Do nothing	Do nothing	
4	1 - Spot blast, clean and paint	MRR steel coating	
4	2 - Replace paint system	MRR steel coating	Steel coating replacement
5	0 - Do nothing	Do nothing	
5	1 - Rehab unit	MRR steel elements	
5	2 - Replace unit	MRR steel elements	Bridge replacement Superstructure replacement

scheme serves to group similar actions for more convenient handling. Costs and benefits are computed by summing the affected elements and condition states.

TSR actions are all replacement activities whose cost does not vary with the existing condition of the element. Each element needs only one unit cost for each applicable action, and the action always restores the element to 100% perfect condition. The TSR actions we want to consider are already defined in the CoRe Element Guide, but are mixed in with the rest of the MRR actions. So, the change we are making here is to flag and organize the TSR actions for special handling appropriate for the bridge-level analysis.

Action Type Examples. Tables 35, 36, and 37 show examples of preservation action type classifications. Table 35, for painted steel girders, has 13 MRR actions that are combined into just four MRR action types: do-nothing, routine maintenance, MRR steel coating, and MRR steel elements. If

the engineer specifies MRR steel coating for an element of this type, the model will know to apply an appropriate action to each of condition states 2 to 4, based on the actual condition of the element. When there are two MRR actions for a given state (as is the case for states 2 and 4), the one with lowest life-cycle cost is used.

In the same example, the engineer might specify the TSR steel coating replacement instead. This happens to be defined in the CoRe Element Guide as condition state 4, action 2. This action is applied to the entire element, using the unit cost given for condition state 4, action 2. The long-term cost model (described earlier in this report) also gets the data it needs from this MRR action.

A few restrictions may be apparent from this example. If the engineer selects TSR steel coating replacement as a scope item in a given candidate, the user cannot also select MRR steel coating. Logically, these are two different approaches to the same set of coating needs, so double-counting of actions

Table 36. Action type classification example for element 14.

State	Action	MRR Action Type	TSR Action Type
Element 14 - Protected Concrete Deck with AC Overlay			
1	0 - Do nothing	Do nothing	
2	0 - Do nothing	Do nothing	
2	1 - Repair potholes and substrat	MRR deck elements	
3	0 - Do nothing	Do nothing	
3	1 - Repair potholes and substrat	MRR deck elements	
3	2 - Replace overlay	MRR deck elements	
4	0 - Do nothing	Do nothing	
4	1 - Repair potholes and substrat	MRR deck elements	
4	2 - Replace overlay	MRR deck elements	
5	0 - Do nothing	Do nothing	
5	1 - Replace overlay	MRR deck elements	Deck overlay replacement
5	2 - Replace deck	MRR deck elements	Bridge replacement Superstructure replacement Deck structure replacement

Table 37. Action type classification example for element 300.

Element 300 - Strip Seal Expansion Joint			
State	Action	MRR Action Type	TSR Action Type
1	0 - Do nothing	Do nothing	
2	0 - Do nothing	Do nothing	
2	1 - Patch, reset, and clean	MRR expansion joints	
3	0 - Do nothing	Do nothing	
3	1 - Replace gland and/or patch	MRR expansion joints	
3	2 - Replace joint	MRR expansion joints	Bridge replacement Superstructure replacement Deck structure replacement Expansion joint replacement

must be avoided. Also, each TSR action type can point to at most one MRR action type in each element.

Table 36, for protected concrete deck with asphaltic concrete (AC) overlay, shows three different TSR actions that all use state 5, action 2, for various types of structure replacement. Finally, Table 37, for strip seal expansion joint, shows that a TSR deck structure replacement also includes replacement of the expansion joints (as well as approach slabs and railings) on the bridge.

By taking the perspective of the bridge as a whole, we deal with several important issues: that certain actions implicitly include actions on other elements; that actions separately defined for different elements may actually be the same action to the bridge engineer and can be presented in a more compact, consolidated form; and that economies of scale may exist at the bridge level that are not evident at the element level.

Scoping Example. When the bridge-level model generates the AutoMRRI candidate, it follows a well-defined sequence of steps to list all possible scope items, reduce them to cost-effective components, and select an appropriate set of action types. All the intermediate results of this calculation are shown on the scoping worksheet. Consider a bridge with the list of elements as shown in Table 38.

The first step is to generate a list of all possible scope items. This is done by reviewing the list of feasible MRR actions for each possible condition state of each element and noting which

action types are represented. Then a list of MRR actions is compiled, sorted first by action type and then by element and condition state (omitting do-nothing and routine maintenance actions). The quantity in each condition state is computed. Table 39 shows the result.

Each group of lines with a common action type makes up a scope item. Each element and condition state appears in at most one MRR scope item, but may appear in multiple TSR scope items. Each TSR scope item lists only one condition state and action for each element, but shows the entire element quantity. This is because TSR actions, by definition, apply to the entire element regardless of condition.

We now perform a benefit-cost analysis of the actions and scope items. For MRR scope items, the procedure is very much like what Pontis does, considering each action separately and computing the benefit by subtracting the action's long-term cost from the do-nothing long-term cost. If the action has a positive benefit, it is considered to be "worth it" and will receive further consideration. Actions with nonpositive benefits are eliminated from their scope items.

One important difference between this method and Pontis is that we only consider variable costs in the calculation of benefit. Normally, the long-term cost of an action in Pontis includes the initial cost of the action, both fixed and variable components. However, at this stage of the analysis we assume that small changes in scope do not affect fixed costs, but only affect variable costs. Another way to look at it is to take the

Table 38. Element list for scoping example.

Element/Environment	Quantity	Units	Condition at start of intervention year				
			1	2	3	4	5
39/4 - Unp Conc Slab/AC Ovl	190.00	sq.m	0.00	23.71	30.78	21.76	23.75
204/4 - P/S Conc Column	5.00	ea	0.00	46.95	26.10	26.95	0.00
215/4 - R/Conc Abutment	29.00	m	83.02	14.12	2.47	0.39	0.00
234/4 - R/Conc Cap	14.00	m	0.00	73.64	21.34	5.02	0.00
301/4 - Pourable Joint Seal	41.00	m	17.12	27.71	55.17	0.00	0.00
321/4 - R/Conc Approach Slab	2.00	ea	74.00	22.11	3.48	0.41	0.00
331/4 - Conc Bridge Railing	28.00	m	76.87	20.07	2.74	0.32	0.00
396/4 - Other Abut Slope Pro	89.00	sq.m	58.80	33.04	6.46	1.70	0.00

Table 39. Example scope items and their action components.

Action Type	Element/Environment	State/Action	Quantity	Units
TSR Deck structure	39/4 - Unp Conc Slab/AC Ovl	5/2 - Replace	190.00	sq.m
TSR Deck structure	321/4 - R/Conc Approach Slab	4/1 - Replace	2.00	ea
TSR Deck structure	301/4 - Pourable Joint Seal	3/1 - Replace	41.00	m
TSR Deck structure	331/4 - Conc Bridge Railing	4/2 - Replace	28.00	m
TSR Wearing surface	39/4 - Unp Conc Slab/AC Ovl	5/1 - Overlay	190.00	sq.m
TSR Wearing surface	321/4 - R/Conc Approach Slab	3/1 - Overlay	2.00	ea
TSR Joints	301/4 - Pourable Joint Seal	3/1 - Replace	41.00	m
TSR Railings	331/4 - Conc Bridge Railing	4/2 - Replace	28.00	m
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	2/1 - Repair	45.04	sq.m
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	3/1 - Repair	58.48	sq.m
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	4/1 - Repair	41.34	sq.m
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	5/2 - Replace	45.13	sq.m
MRR Other - 321/4	321/4 - R/Conc Approach Slab	2/1 - Seal cracks	0.44	ea
MRR Other - 321/4	321/4 - R/Conc Approach Slab	3/1 - Overlay	0.07	ea
MRR Other - 321/4	321/4 - R/Conc Approach Slab	4/1 - Replace	0.01	ea
MRR Joints	301/4 - Pourable Joint Seal	2/1 - Cln & Reseal	11.36	m
MRR Joints	301/4 - Pourable Joint Seal	3/1 - Replace	22.62	m
MRR Railings	331/4 - Conc Bridge Railing	2/1 - Seal & Patch	5.62	m
MRR Railings	331/4 - Conc Bridge Railing	3/1 - Cln & Patch	0.77	m
MRR Railings	331/4 - Conc Bridge Railing	4/2 - Replace	0.09	m
MRR Concrete	204/4 - P/S Conc Column	2/1 - Seal & Patch	2.35	ea
MRR Concrete	204/4 - P/S Conc Column	3/1 - Cln & Patch	1.30	ea
MRR Concrete	204/4 - P/S Conc Column	4/1 - Rehab	1.35	ea
MRR Concrete	215/4 - R/Conc Abutment	2/1 - Seal & Patch	4.09	m
MRR Concrete	215/4 - R/Conc Abutment	3/1 - Cln & Patch	0.72	m
MRR Concrete	215/4 - R/Conc Abutment	4/1 - Rehab	0.11	m
MRR Concrete	234/4 - R/Conc Cap	2/1 - Seal & Patch	10.31	m
MRR Concrete	234/4 - R/Conc Cap	3/1 - Cln & Patch	2.99	m
MRR Concrete	234/4 - R/Conc Cap	4/2 - Replace	0.70	m
MRR Other - 396/4	396/4 - Other Abut Slope Pro	2/1 - Rehab & Prot	29.40	sq.m
MRR Other - 396/4	396/4 - Other Abut Slope Pro	3/1 - Rehab	5.75	sq.m
MRR Other - 396/4	396/4 - Other Abut Slope Pro	4/1 - Rehab	1.51	sq.m

perspective of the field crew. If they set up on the worksite and find additional needs not noted in the inspection, they will find it much more cost-effective to spend extra time to take care of those needs right away rather than planning to revisit the site at a later time. That is because the fixed costs of mobilization and traffic control are already spent at that point. This is one of the important reasons why work quantities estimated by Pontis tend to be smaller than the quantities actually performed in the field.

TSR scope items are handled in a somewhat different way, reflecting the fact that they are accepted or rejected for the whole element and not subdivided by condition state. The action's long-term cost is computed for the entire quantity of the element, still based on variable costs. For the benefit computation, the base case is the average long-term cost based on existing conditions. This is computed for each condition state of each element, using that condition state's do-nothing long-term cost. Table 40 shows the set of computations for this example.

Table 40. Base case computation for TSR scope items.

Average do-nothing long-term cost (DNLTC)										
	Condition at start of intervention year					Do-nothing full long term unit cost (DNLTC)				Average
Element	1	2	3	4	5	1	2	3	4	5	DNLTC
39/4 - Unp Conc Slab/AC Ovl	0.00	23.71	30.78	21.76	23.75	144.31	228.51	309.88	396.77	743.22	412.43
204/4 - P/S Conc Column	0.00	46.95	26.10	26.95	0.00	322.57	1052.57	3715.60	10701.82	0.00	4348.24
215/4 - R/Conc Abutment	83.02	14.12	2.47	0.39	0.00	129.21	454.12	1028.18	5354.44	0.00	217.66
234/4 - R/Conc Cap	0.00	73.64	21.34	5.02	0.00	107.21	364.85	859.76	2396.64	0.00	572.39
301/4 - Pourable Joint Seal	17.12	27.71	55.17	0.00	0.00	353.62	526.07	2028.56	0.00	0.00	1325.47
321/4 - R/Conc Approach Slab	74.00	22.11	3.48	0.41	0.00	8.05	20.92	603.27	15128.21	0.00	93.00
331/4 - Conc Bridge Railing	76.87	20.07	2.74	0.32	0.00	11.26	31.85	81.00	220.42	0.00	17.96
396/4 - Other Abut Slope Pro	58.80	33.04	6.46	1.70	0.00	6.91	13.26	26.95	140.56	0.00	12.58

Elements in worse condition have higher do-nothing long-term costs, which increase the benefit of the scope item. After computing the benefit for each element within a TSR scope item, the elements are summed to arrive at a total benefit. If this benefit is not positive, the scope item receives no further consideration. Table 41 is the full set of computations for the example bridge.

Scale Feasibility. After the above steps of elimination, a list of scope items is gathered and tested against a set of scale feasibility thresholds. Functional improvement scope items are also included at this point. Scale feasibility converts a probabilistic view of quantity and cost into a deterministic view. For example, costing of deck replacement is carried out on the basis of the unit cost of replacing the entire deck, but in the deterioration forecast there is a probability (which is less than 1) that the deck will be in a state where replacement is needed. The combination of the TSR deck replacement action and the scale feasibility rule bring the cost estimate back up to full cost and ensure that the action is programmed only if it is very likely to be needed.

To calculate the scale of a scope item, we compare the cost of MRR actions against the maximum possible cost of the same scope item if the elements it affects are fully deteriorated. For MRR scope items, we consider only the condition states having positive benefits, but for TSR scope items as well as for MRR deck, we consider all condition states because the action is applied to the entirety of the element.

The maximum point on the cost scale for each element is computed simply from the quantity of element and the variable unit cost of the action corresponding to TSR bridge replacement. The scale of a scope item, then, is computed by summing the MRR actions over all elements affected by the scope item, then dividing that sum by the corresponding sum of maximum costs. For each action type, a scale feasibility threshold is established by agency policy. The suggested values are given in Table 42. These thresholds are applied to each scope item, as shown in Table 43. Several scope items in the example failed to reach the minimum size.

Dominance. A final consideration is what happens when a TSR scope item has positive benefits and satisfies the scale

Table 41. Full set of example scoping computations for a bridge.

MRR Actions implied by scope items			B/C Computation						
Action Type	Element/Environment	State/Action	VarUnitCo	FixUnitCo	LTCost	VarLTC	DNLTC	NetBen	WorthIt
TSR Deck structure	39/4 - Unp Conc Slab/AC Ovl	5/2 - Replace	230.15	103.57	478.03	374.46	412.43	7214	yes
TSR Deck structure	321/4 - R/Conc Approach Slab	4/1 - Replace	8965.52	4034.48	13008.05	8973.57	93.00	-17761	yes
TSR Deck structure	301/4 - Pourable Joint Seal	3/1 - Replace	167.44	75.35	796.46	721.11	1325.47	24779	yes
TSR Deck structure	331/4 - Conc Bridge Railing	4/2 - Replace	113.14	50.91	175.31	124.40	17.96	-2980	yes
TSR Wearing surface	39/4 - Unp Conc Slab/AC Ovl	5/1 - Overlay	280.63	126.29	757.78	631.49	412.43	-41621	no
TSR Wearing surface	321/4 - R/Conc Approach Slab	3/1 - Overlay	26.07	11.73	49.29	37.56	93.00	111	no
TSR Joints	301/4 - Pourable Joint Seal	3/1 - Replace	167.44	75.35	796.46	721.11	1325.47	24779	yes
TSR Railings	331/4 - Conc Bridge Railing	4/2 - Replace	113.14	50.91	175.31	124.40	17.96	-2980	no
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	2/1 - Repair	37.12	16.71	282.34	265.63	228.51	-1672	no
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	3/1 - Repair	74.24	33.41	417.53	384.12	309.88	-4342	no
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	4/1 - Repair	148.48	66.82	612.07	545.25	396.77	-6139	no
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	5/2 - Replace	230.15	103.57	741.53	637.96	743.22	4751	yes
MRR Other - 321/4	321/4 - R/Conc Approach Slab	2/1 - Seal cracks	4137.93	1862.07	6008.90	4146.83	20.92	-1825	no
MRR Other - 321/4	321/4 - R/Conc Approach Slab	3/1 - Overlay	26.07	11.73	49.29	37.56	603.27	39	yes
MRR Other - 321/4	321/4 - R/Conc Approach Slab	4/1 - Replace	8965.52	4034.48	13008.05	8973.57	15128.21	50	yes
MRR Joints	301/4 - Pourable Joint Seal	2/1 - Cln & Reseal	58.83	26.48	490.68	464.20	526.07	703	yes
MRR Joints	301/4 - Pourable Joint Seal	3/1 - Replace	167.44	75.35	796.46	721.11	2028.61	29574	yes
MRR Railings	331/4 - Conc Bridge Railing	2/1 - Seal & Patch	101.83	45.82	159.27	113.45	31.85	-459	no
MRR Railings	331/4 - Conc Bridge Railing	3/1 - Cln & Patch	452.55	203.65	674.77	471.12	81.00	-299	no
MRR Railings	331/4 - Conc Bridge Railing	4/2 - Replace	113.14	50.91	175.31	124.40	220.42	9	yes
MRR Concrete	204/4 - P/S Conc Column	2/1 - Seal & Patch	185.19	64.81	882.87	818.06	1052.57	551	yes
MRR Concrete	204/4 - P/S Conc Column	3/1 - Cln & Patch	370.37	129.63	3594.89	3465.26	3715.60	327	yes
MRR Concrete	204/4 - P/S Conc Column	4/1 - Rehab	3703.70	1296.30	5468.55	4172.25	10701.82	8800	yes
MRR Concrete	215/4 - R/Conc Abutment	2/1 - Seal & Patch	109.37	38.28	451.51	413.23	454.12	167	yes
MRR Concrete	215/4 - R/Conc Abutment	3/1 - Cln & Patch	364.56	127.59	1019.16	891.57	1028.18	98	yes
MRR Concrete	215/4 - R/Conc Abutment	4/1 - Rehab	1215.19	425.31	2031.05	1605.74	5354.44	423	yes
MRR Concrete	234/4 - R/Conc Cap	2/1 - Seal & Patch	109.37	38.28	390.13	351.85	364.85	134	yes
MRR Concrete	234/4 - R/Conc Cap	3/1 - Cln & Patch	486.07	170.13	955.15	785.02	859.76	223	yes
MRR Concrete	234/4 - R/Conc Cap	4/2 - Replace	1117.97	391.29	1616.47	1225.18	2396.64	823	yes
MRR Other - 396/4	396/4 - Other Abut Slope Pro	2/1 - Rehab & Prot	25.67	8.99	42.16	33.17	13.26	-585	no
MRR Other - 396/4	396/4 - Other Abut Slope Pro	3/1 - Rehab	25.67	8.99	44.12	35.13	26.95	-47	no
MRR Other - 396/4	396/4 - Other Abut Slope Pro	4/1 - Rehab	25.67	8.99	42.00	33.01	140.56	163	yes

VarUnitCo = Variable unit cost
FixUnitCo = Fixed unit cost

LTCost = Pontis long-term cost
VarLTC = Variable long-term cost
DNLTC = Do-nothing long-term cost

NetBen = Net life cycle benefit of including the action
WorthIt = Indicates positive benefits

Table 42. Suggested scale feasibility thresholds.

Name of Action Type	Scale Feasibility Threshold
Do nothing	0
TSR Replace Structure	50
IMP Widen	15
IMP Raise	15
IMP Strengthen	15
IMP Scour Mitigation	15
IMP Seismic Retrofit	15
IMP Fatigue Mitigation	15
TSR Superstructure	30
TSR Deck Structure	30
TSR Wearing Surface	15
TSR Steel Coating	15
TSR Expansion Joints	30
TSR Railings	30
TSR Bearings	30
MRR Deck Elements	5
MRR Steel Elements	5
MRR Steel Coating	5
MRR Concrete Element	5
MRR Timber Elements	5
MRR Expansion Joints	5
MRR Bearings	5
MRR Railings	5
MRR Other Elements	5
Routine Maintenance	0
Temporary Cribbing	0
Remove Structure	0
Custom Action	0

thresholds, but affects elements that are also affected by other TSR or MRR scope items. In the current example, TSR deck structure and TSR joints both meet all the criteria, but only one of these can be implemented because replacing the deck also includes replacing the joints.

The solution is to select the highest-type feasible TSR scope item whenever this situation occurs. In other words, we do not select a TSR joint replacement scope item if TSR deck structure is feasible.

Indirect Cost Model. Indirect costs, especially mobilization and maintenance of traffic, play a significant role in bridge-level bridge management, representing often half or more of the cost of a project. In network-level bridge management, it is possible to ignore fixed costs or make them variable as in Pontis. This, however, is not valid at the bridge level.

Fixed costs are known to be complicated to estimate and in practice require inputs that aren't found in a bridge management system. In the past, this has been a barrier to even considering them in bridge management system analysis. Without considering fixed costs, however, a bridge management system will generate frequent unrealistically small interventions, underestimate costs, and overestimate economic benefits of small activities. To alleviate this problem, bridge management systems sometimes introduce project cost thresholds, or "clumping rules," that have even less economic validity than a simple indirect cost model would have.

The bridge-level model of the Multi-Objective Optimization System (MOOS) software estimates indirect cost using a simple model, taking advantage of the separation of variable and fixed unit costs that is already allowed, but seldom used, in Pontis. We use the variable unit cost for most of the analysis at the scope item level and add the fixed costs back at the candidate

Table 43. Application of scale feasibility thresholds to the example bridge.

Scope Item scale feasibility and performance						
Action Type	VarCost	MRRI	MaxCost	Scale	MinScale	ScaleFeas
IMP Widening	156160	156160	531944	29.4	15	**yes**
IMP Strengthening	86864	86864	531944	16.3	15	**yes**
TSR Deck structure	71693	29829	71693	41.6	30	**yes**
TSR Wearing surface	53372	24444	61660	39.6	15	**yes**
TSR Joints	6865	4456	6865	64.9	30	**yes**
TSR Railings	3168	929	3168	29.3	30	**no**
MRR Deck	22539	22539	43729	51.5	5	**yes**
MRR Other - 321/4	75	75	17931	0.4	5	**no**
MRR Joints	4456	4456	6865	64.9	5	**yes**
MRR Railings	10	10	3168	0.3	5	**no**
MRR Concrete	10121	10121	147520	6.9	5	**yes**
MRR Other - 396/4	39	39	4968	0.8	5	**no**

VarCost = Variable cost
MRRI = MRR and improvement needs
MaxCost = Maximum deteriorated cost

Scale = Relative extent of needs
MinScale = Minimum needs threshold
Scale Feas = Indicator of whether the need is large enough

level after scope items have been selected. The simple indirect cost model recognizes just the most obvious observations:

- Mobilization and maintenance of traffic are the largest fixed cost items, and each has its own distinct behavior.
- Mobilization and maintenance of traffic both depend, in some nonlinear way, on the size of the project. We drastically simplify this relationship to give each one just a fixed portion and a linear portion that depends on direct costs.
- The fixed portion of maintenance of traffic depends on the number of lanes of roadway affected by the work. We determine which lanes are affected in a simple way by classifying a candidate as above the bridge, below the bridge, or both.
- Work on decks, joints, and barriers produce higher fixed costs than other types of work.

We use these observations of indirect costs to give us some rudimentary but necessary behavior that has more validity than simply setting a floor on total costs. But more importantly, this model represents a placeholder for improved indirect cost models that can be developed in future research.

To fully specify the basic indirect cost model, we establish a few definitions. First, we define the cost basis used for computing the relationship between direct and indirect costs, by classifying the individual scope item components according to whether they affect traffic on top of the bridge, underneath the bridge, or both. Table 44 gives the general pattern.

For example, a functional improvement scope item affects traffic in the lanes both above and below the bridge, so it has two separate traffic control installations and counts double toward indirect costs. We distinguish superstructure, substructure, and deck elements by the value of "elemdefs.etypkey" in Pontis, as follows:

Substructure: Etypkey = 4, 13, 21, 22, 23, 24, 25
Superstructure: Etypkey = 10, 12, 14, 15, 16
Deck: Etypkey = All others, including joints, barriers, and approach slabs

For superstructure elements, we have to decide whether they affect the bottom lanes of a bridge (e.g., girders) or both top

and bottom (e.g., thru-trusses). We do this with NBI item 43b (design type, main span) as follows:

Bottom type: NBI item 43b = 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 18, 19, 22
Top+Bottom type: NBI item 43b = All others

Having classified the direct costs of every component of every scope item in this way, we sum up the total cost basis for the top and bottom of the bridge.

The fixed portion of maintenance of traffic is computed as a cost per affected lane, times the number of affected lanes. If the top cost basis is non-zero, then all the lanes on the structure are included. If the bottom cost basis is non-zero, then all the lanes under the structure are included.

The variable portion of maintenance of traffic is computed as a multiplier of total cost basis, computed as follows: top cost basis + bottom cost basis + deck cost basis. We count the deck twice here because traffic control costs are higher for work on the deck.

Mobilization is computed as the sum of the top and bottom cost bases, times a multiplier, plus a fixed cost. This is a very simple model that agencies can enhance using their own estimation procedures and experience. While there is currently no solid research basis for this or any other indirect cost model, this at least will serve as a first step in the direction of better methods.

Relationship to Pontis Simulation Rules.   A reader familiar with Pontis may notice that the action classification scheme and scale feasibility thresholds proposed here address many of the same issues that motivate the Pontis simulation rules. Pontis added a set of features in release 4.0 to try to generate a more realistic set of projects in its program simulation than was possible in earlier versions. These rules are designed to be used in a "black box" mode where the engineer cannot intervene in the bridge-level decision process because a great many bridges are processed all at once.

This rule-based handling of candidate definition is not sufficient for a bridge-level model, in which the engineer interacts with candidates individually. Aside from the ability to generate reasonable candidates, the bridge-level tool needs to offer the ability for the engineer to manipulate and change the results of candidate scoping decisions without losing the

Table 44. Classification of scope item components as on or under the bridge.

	Functional Improvements	Substructure Elements	Superstructure Elements		Deck Elements
			Bottom Type	Top+Bottom Type	
Top	X			X	X
Bottom	X	X	X	X	

underlying conceptual relationship among related elements and condition states.

For example, the bridge painting rules in Pontis establish the concept of a zone painting action without explicitly representing such an action in the database. In a bridge-level tool, however, it is necessary to be able to define and store the zone painting action as a distinct object that the engineer can view and manipulate (e.g., to change the quantity of painting). This is more consistent with the way the engineer and the paint crew may think about the action ("paint whatever needs to be painted on the bridge").

Another difference in perspective comes from the desire, in the Pontis program simulation, to make each project as realistic as possible without user interaction. Realism is important, but at the bridge level, it is provided by allowing the engineer to review each candidate individually, giving more realism than any set of rules could provide. Therefore, the requirement for a relatively elaborate set of decision rules in a program simulation does not exist in a bridge-level tool, where a simpler set of database relationships is sufficient. The more rigorous use of the action type classification scheme, and the addition of TSR action types, gives the maintenance planner the expressive power and flexibility that he or she needs, while at the same time simplifying the presentation and greatly reducing the amount of user input required in order to define a candidate.

Unlike the Pontis simulation rules, the action type classification system does not force actions into or out of the program. It merely consolidates them for the purpose of presentation and evaluation. Candidate selection is based on considerations of performance measures, economic concerns, and the judgment of the maintenance planner. So the classification scheme does not act as a set of constraints on the bridge-level optimization. Instead, the set of minimum tolerable conditions, discussed earlier, plays this role.

Custom Candidates. Custom candidates allow the engineer to specify any quantity of any action, which therefore does not have to conform to the exact quantities implied by action classification and scale feasibility thresholds.

If the maintenance planner adds scope items or changes their quantities, the bridge-level model will estimate the quantities of each MRR action for each condition state by applying actions to states where they are feasible in a way that minimizes life-cycle cost (i.e., starting with the highest life-cycle benefit-cost ratio). The result gives an estimate of the best possible life-cycle cost outcome that can be achieved from the given quantity of work.

Continuing the example presented earlier, the scoping process generates a table where scope items are divided into MRR action components for each affected element and condition state. After calculating costs and benefits for each component as in the earlier example, we calculate a benefit-cost ratio, as shown in Table 45. The total variable cost of MRR

concrete work is $10,121, with a net benefit of $11,545 and a benefit-cost ratio of 1.14. If the maintenance planner decides to spend only $5,000, the models calculate the reduced benefits of this scope item by removing the components that contribute least to total benefits relative to their cost—in other words, the components with lowest benefit-cost ratios.

In this example, $5,000 is enough to perform the work on element 215, state 4, plus almost all the work on element 204, state 4. Benefits for the marginal component (element 204, state 4) are computed by prorating. So we have $423 in benefits for element 215 and $8,574 in benefits for element 204, for a total benefit of $8,997 and a benefit-cost ratio for the whole scope item of 1.80.

If the engineer or models raise the quantity of an action beyond the quantity of deteriorated condition states present for which the action type is feasible, the extra quantity still incurs a cost but not a benefit. This is especially important with the TSR actions, which by definition replace the entirety of elements even if portions are still in condition state 1.

Pontis-Generated Candidates. The bridge-level software tool offers the ability for a maintenance planner to select from Pontis one or more work candidates or work items on a bridge (inspector-generated or Pontis-generated) and import them to the bridge-level model as custom candidates. From there, the maintenance planner can investigate variations on scope or timing and use them in the network-level model.

The translation from Pontis work candidates to the new model's scope items is fairly straightforward as long as both systems are using the same action type classification scheme and of course the same element, state, and action definitions. Consolidation would occur when Pontis has multiple work candidates with the same action type (on different elements, for example).

Since the evaluation capabilities of the bridge-level model are not exactly the same as Pontis 4.0, the results can be expected to differ, especially with regard to the new performance measures calculated in the new system and the new classification of MRR actions. Compared with the Pontis Bridge Analysis Screen, the new project-level dashboard offers a greater degree of flexibility and a more user-friendly presentation of results and trade-offs.

3.3.3.3 Replacement

The replacement candidate, shown in Figure 43, normally provides an upper bound on the cost and effectiveness of work that can be done on a bridge. The bridge-level model does not have features to analyze traffic requirements and their impact on the design of a replacement bridge (e.g., adding lanes). Therefore, the model is fairly simple. It computes initial costs from a swell factor (i.e., a multiplier reflecting the fact that replacement bridges are usually longer and wider than what

Table 45. Example data used in computing custom candidates.

MRR Actions implied by scope items					
Action Type	Element/Environment	State/Action	VarCost	NetBen	B/C
TSR Deck structure	39/4 - Unp Conc Slab/AC Ovl	5/2 - Replace	43729	7214	0.1650
TSR Deck structure	321/4 - R/Conc Approach Slab	4/1 - Replace	17931	-17761	-0.9905
TSR Deck structure	301/4 - Pourable Joint Seal	3/1 - Replace	6865	24779	3.6094
TSR Deck structure	331/4 - Conc Bridge Railing	4/2 - Replace	3168	-2980	-0.9408
TSR Wearing surface	39/4 - Unp Conc Slab/AC Ovl	5/1 - Overlay	53320	-41621	-0.7806
TSR Wearing surface	321/4 - R/Conc Approach Slab	3/1 - Overlay	52	111	2.1267
TSR Joints	301/4 - Pourable Joint Seal	3/1 - Replace	6865	24779	3.6094
TSR Railings	331/4 - Conc Bridge Railing	4/2 - Replace	3168	-2980	-0.9408
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	2/1 - Repair	1672	-1672	-1.0000
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	3/1 - Repair	4342	-4342	-1.0000
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	4/1 - Repair	6139	-6139	-1.0000
MRR Deck	39/4 - Unp Conc Slab/AC Ovl	5/2 - Replace	10387	4751	0.4574
MRR Other - 321/4	321/4 - R/Conc Approach Slab	2/1 - Seal cracks	1830	-1825	-0.9971
MRR Other - 321/4	321/4 - R/Conc Approach Slab	3/1 - Overlay	2	39	21.6997
MRR Other - 321/4	321/4 - R/Conc Approach Slab	4/1 - Replace	73	50	0.6865
MRR Joints	301/4 - Pourable Joint Seal	2/1 - Cln & Reseal	668	703	1.0517
MRR Joints	301/4 - Pourable Joint Seal	3/1 - Replace	3787	29574	7.8085
MRR Railings	331/4 - Conc Bridge Railing	2/1 - Seal & Patch	572	-459	-0.8013
MRR Railings	331/4 - Conc Bridge Railing	3/1 - Cln & Patch	347	-299	-0.8620
MRR Railings	331/4 - Conc Bridge Railing	4/2 - Replace	10	9	0.8487
MRR Concrete	204/4 - P/S Conc Column	2/1 - Seal & Patch	435	551	1.2663
MRR Concrete	204/4 - P/S Conc Column	3/1 - Cln & Patch	483	327	0.6759
MRR Concrete	204/4 - P/S Conc Column	4/1 - Rehab	4991	8800	1.7630
MRR Concrete	215/4 - R/Conc Abutment	2/1 - Seal & Patch	448	167	0.3739
MRR Concrete	215/4 - R/Conc Abutment	3/1 - Cln & Patch	262	98	0.3747
MRR Concrete	215/4 - R/Conc Abutment	4/1 - Rehab	137	423	3.0849
MRR Concrete	234/4 - R/Conc Cap	2/1 - Seal & Patch	1128	134	0.1189
MRR Concrete	234/4 - R/Conc Cap	3/1 - Cln & Patch	1452	223	0.1538
MRR Concrete	234/4 - R/Conc Cap	4/2 - Replace	785	823	1.0478
MRR Other - 396/4	396/4 - Other Abut Slope Pro	2/1 - Rehab & Prot	755	-585	-0.7756
MRR Other - 396/4	396/4 - Other Abut Slope Pro	3/1 - Rehab	148	-47	-0.3187
MRR Other - 396/4	396/4 - Other Abut Slope Pro	4/1 - Rehab	39	163	4.1897

VarCost = Total variable cost
NetBen = Net life cycle benefit of including the action
B/C = Benefit/cost ratio

they replace) and a unit cost per deck area. For life-cycle cost computations following replacement, it is assumed that the replacement bridge has the same elements in the same quantities as the old bridge, starting in new condition. Other than the swell factor and user cost model, no costs or benefits are recognized because of a larger or better-constructed bridge.

Replacement contains all of the major cost components:

- **Failure risk,** representing the possibility of economic failure during the program period. This occurs in the program years before the first intervention and during the periods between interventions.
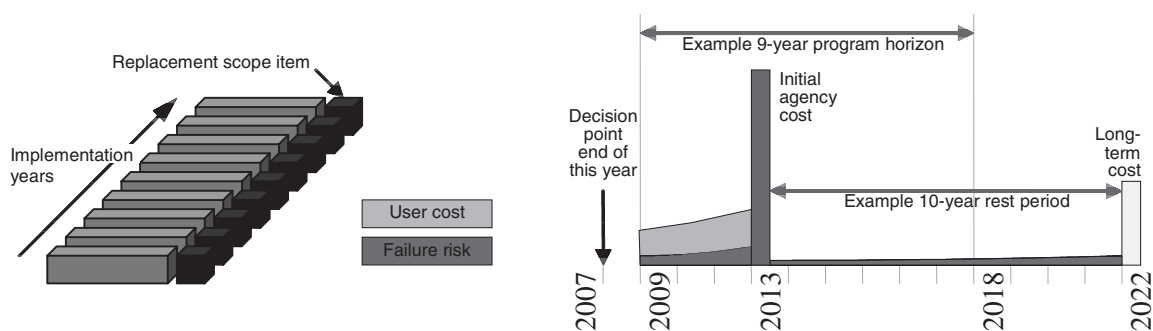

Figure 43. Replacement life-cycle activity profile.

- **User cost,** recognized if there are any functional deficiencies on the bridge. Replacement is assumed always to remedy all functional needs, so there are no excess user costs following replacement.
- **Initial agency cost,** the actual cost of the work to be done during each intervention. This is assumed to occur at the start of the program year assigned to each intervention.
- **Long-term cost,** representing future preservation costs after the final rest period, as a function of ending conditions.

The deterioration model and traffic growth model are the reasons why costs of the first two components increase over time.

3.3.4 Evaluation and Optimization

After the outputs and outcomes of a candidate are fully analyzed, the remaining step is to establish a basis for comparing the candidate with other candidates. We use performance measures and a utility function to do this. Table 46 lists all of the performance measures used in the system and organizes the data needed for evaluation and optimization. This table indicates examples of the settings used for each performance measure to govern how they participate in the bridge-level optimization. As described earlier, the optimization can operate in a mode where timing of interventions is governed by worst tolerable performance thresholds, or it can select the optimal timing based on a utility function (or a combination of both). Optimal scoping is always based on the utility function. If the worst tolerable performance feature is activated and if any performance measure is below its worst tolerable performance threshold, then the optimal timing is considered to be immediate.

Each performance measure may participate in a utility function for selecting optimal candidates. Utility functions are always oriented so that higher values are considered "good" and lower values "bad." If a performance measure is given a nonzero weight, then it is considered in the bridge-level

Table 46. Performance measure table.

Field	High Level	Low Level	Sense	Worst Tolerable	Remedies	Utility Weight	Base Case	Outcome	Benefit	Scaled Benefit	Utility
NBI Serviceability											
Deck geometry (68)	3	0	1		21		7	3	2	0.222	0.00
Inventory rating (66)	41	0	1		23		32.6	99.0	66.4	1.620	0.00
Operating rating (64)	41	0	1		23		53.5	99.0	45.5	1.110	0.00
Condition and sufficiency											
Deck condition	9	0	1				5	9	4	0.444	0.00
Superstructure condition	9	0	1				6	9	3	0.333	0.00
Substructure condition	9	0	1				7	9	2	0.222	0.00
Culvert condition	9	0	1				N	9	0	0.000	0.00
Sufficiency rating	100	0	1			20	83.7	100.0	16.3	0.163	3.26
Health index	100	0	1	80		30	41.9	100.0	58.1	0.581	17.44
Vulnerability assessment											
Scour	6	0	1	2	24		N	6	0	0.000	0.00
Fatigue	6	0	1	2	26		N	6	0	0.000	0.00
Seismic	6	0	1	2	25		N	6	0	0.000	0.00
Other	6	0	1	2	27		N	6	0	0.000	0.00
Smart flags											
Steel fatigue (356) [Fatig]	3	0	-1	2	26		0	0	0	0.000	0.00
Pack rust (357) [PkRust]	4	0	-1	3	31		0	0	0	0.000	0.00
Deck cracking (358) [DkCrak]	4	0	-1	3	31,32		0	0	0	0.000	0.00
Soffit (359) [Soffit]	5	0	-1	4	31,32		0	0	0	0.000	0.00
Settlement (360) [Setlmt]	3	0	-1				0	0	0	0.000	0.00
Scour (361) [Scour]	3	0	-1		24		0	0	0	0.000	0.00
Traffic impact (362) [Traflmp]	3	0	-1		31		0	0	0	0.000	0.00
Section loss (363) [SecLoss]	4	0	-1	3	31		0	0	0	0.000	0.00
Life cycle cost ($000)											
Initial cost	1716423	0	-1				0	1716423	-1716423	-1.000	0.00
Agency life cycle cost	1716423	0	-1				7352448	2626020	4726428	2.754	0.00
User life cycle cost	1716423	0	-1				0	0	0	0.000	0.00
Total life cycle cost	1716423	0	-1			50	7352448	2626020	4726428	2.754	137.68
Custom performance measures											
Custom1									0	0.000	0.00
Custom2									0	0.000	0.00
Custom3									0	0.000	0.00
Utility function											158.33

optimization. It is permissible for different groups of bridges to use different sets of weights.

One feature of note in the performance measure table is that smart flags can participate in the bridge-level optimization through the worst tolerable performance function, the utility function, or both. Another notable feature is that an agency can define its performance measures based on any available data about the bridge.

In each pass through the bridge-level optimization loop, a worst tolerable performance calculation and a utility function calculation occur using the performance measure table. The outcome prediction, assuming that the subject intervention takes place, occurs in various ways appropriate to each performance measure:

- NBI serviceability ratings and vulnerability assessments are taken directly from the bridge inventory, subject to any changes made by earlier interventions (if in a recursive call to the optimization algorithm).
- Condition and sufficiency assessments are forecast to the end of the intervention year using the element-level Markovian deterioration model, then converted to NBI indicators using the FHWA NBI translator program.
- Smart flags are taken directly from the most recent element inspection, subject to any changes made by earlier interventions.

- Life-cycle costs are calculated, starting at the beginning of the intervention year and discounted to that point, using the procedures described in Chapter 2.
- Custom performance measures are envisioned to be defined as Excel worksheet formulas with access to any available bridge-related data, so they are calculated by updating the formulas.

A base case prediction is also made for each performance measure, assuming that no action is taken that year or in any future year until the end of the program horizon. Interventions taken in any earlier, higher-level recursions are assumed to be the same for both the intervention outcome and the base case. Benefits are calculated by subtracting the base case result from the intervention outcome for each performance measure.

The calculation of utility function is shown in the final column of the projected performance table. This will incorporate all the functionality of scaling, amalgamation, and application of the weights given for each performance measure.

After calculation of the final utility function value, the candidate is compared first with other candidates (having different scoping approaches) for the same year. If it turns out to be the highest-utility candidate for the year, it is then compared with the best candidates for other years to determine the optimal candidate.

CHAPTER 4

Conclusions

In striving to provide bridge managers with a useful decision-support mechanism, the present study developed a bridge management system methodology and an accompanying application tool that can enable bridge engineers to define overall optimality of any investment decision in terms of a desired combination of selected performance measures and to make investment choices on the basis of optimal forecast performance. This was done through the conduct of several tasks as summarized in this chapter.

4.1 Performance Measures

This study, on the basis of past research, established a set of performance measures that includes the vulnerability of bridge structures to natural and human-made hazards. These measures can be added to Pontis by agencies that already have vulnerability data or have the intention of collecting such data. Vulnerability of structures is an important dimension of the bridge management problem that has not been addressed comprehensively at the bridge inventory level thus far. In general, the list of performance measures provided in the software module is much more comprehensive than any agency would use in practice. This study provides some guidance on how a bridge engineer can ascertain the appropriateness of a given set of multiple performance measures established for decision-making purposes and how the bridge engineer can choose any specific individual performance measure for inclusion in the set of performance measures. Many agencies may find it sufficient to use life-cycle cost, one condition measure, and one vulnerability measure for most purposes. Adding a condition measure such as health index to the objective function of the optimization has the effect of giving condition extra weight to reflect community preferences for the perception of safety and the general state of repair of highly visible public infrastructure. Even this one change in typical bridge management practice would provide very significant benefits and would greatly increase the usefulness of bridge management systems.

4.2 Development of the Multi-Criteria Utility Function

This study showed that it is feasible for bridge managers, using the performance measures that they have established, to adopt multi-criteria decision-making methodologies. The study determined that a convenient and easily implemented methodology involves combining all performance criteria into a single criterion (also referred to as a "utility") through the processes of weighting, scaling, and amalgamation. With regard to weighting, the study presented a variety of ways through which bridge managers can establish a preferential order for the performance measures of interest. Three recommended weighting methods were demonstrated using data generated from a survey of bridge experts (the NCHRP 12-67 project panel). At a future time, these weighting methods can be replicated by an agency to update the existing default weights so that they are more reflective of the agency's internal practices. The results of the study also showed that there are a number of feasible methods by which the bridge engineer can reduce all the different performance measures (and their different dimensions) into commensurate units so that their respective levels (consequences of a bridge action in terms of the performance measure) can easily be compared or added to each other. For such scaling of the performance measures, two different scenarios were considered: where the consequences of bridge actions are known with certainty and with uncertainty. For each scenario, at least one scaling process was discussed in this study. The functional form for the multi-criteria utility function was ascertained using the gamble method of scaling. For the benefit of engineers who may wish to replicate the scaling process for a different set of performance measures, the scaling methods were comprehensively described and illustrated using data generated from a survey of bridge experts (the NCHRP 12-67 project panel).

The process of amalgamation, or combining the weighted and scaled values of the bridge performance measures into a

single metric that can be interpreted for purposes of evaluation, is a key aspect of the study. The study investigated a number of alternative methods for amalgamation and ultimately selected one that is simple and easily understandable, yet robust enough to duly account for, in an impartial and balanced manner, the preferences of decision makers (with regard to the performance measures) as well as the consequences of the alternative bridge actions in terms of the performance measures. In the selected method, the bridge engineer determines the overall desirability or utility of a bridge action simply as a mathematical "summation" (i.e., additive or the multiplicative) of the weighted and scaled performance levels accrued from that action. The study discussed and demonstrated, using real data, the necessary tests that can be carried out by the bridge engineer to ascertain the appropriateness of the additive or the multiplicative form for a given bridge evaluation problem.

Interaction with, and direct questionnaire surveys from, bridge experts (the project panel) yielded valuable data that helped develop a set of performance measures, default weights, and value and utility functions that were incorporated as default parameters into MOOS, the final software product. However, the software allows users the flexibility to input updated lists of performance measures, relative weights, and utility and value functions to reflect new situations or agency practice or merely to investigate the sensitivity of the resulting solution (i.e., recommended investment decision) to changes such input parameters.

4.3 Selection of Bridge Actions Through Optimization

After presenting the methods by which a bridge engineer can measure the consequence of alternative bridge actions in an impartial, rational, and commensurate manner that considers multiple performance measures, the study demonstrated that if each bridge action (including the do-nothing candidate) is associated with a certain value of such a multi-criteria consequence, it is possible to carry out a multi-criteria–based (and therefore, optimal, or cost-effective) selection of actions for a specific bridge over its lifespan or for a network of bridges over a specified programming period. The developed bridge-level model helps the bridge engineer to schedule specific bridge interventions over the remaining life of a bridge, whereas the network-level model selects candidate projects from a networkwide candidate list to yield maximum network benefits subject to multiple constraints. The network benefit is measured with multiple criteria, and the constraints could be budgetary limitations and/or performance constraints. The impact of various funding levels on network performance can be studied using the network-level optimization model. The network-level model can also be used to estimate funding needed to achieve user-specified condition targets and accept-

able risk levels. The network-level optimization was formulated as a multi-choice multi-dimensional knapsack problem (MCMDKP).

4.4 Identification and Evaluation of Network-Level Solution Approaches

After a careful investigation of alternative approaches for solving the MCMDKP, the researchers determined that three alternative solution approaches were satisfactory: the IUC, Lagrangian, and pivot and complement approaches. The researchers further investigated the suitability of these approaches for agencywide bridge management. Such further investigation included a series of computational experiments in which heuristics were implemented using real field data. The heuristics were evaluated on the basis of four criteria: computational speed, accuracy, simplicity, and robustness. Accuracy was based on comparing heuristic solutions with true optimal solutions derived using CPLEX Concert technology, a state-of-the-art commercial optimization software package. The datasets used for the experiments were from the inventory, deterioration, and cost data files of Florida DOT's Project Level Analysis Tool. Tests were carried out on bridge networks sizes of 100; 1,000; 9,265; 12,000; and 50,000. At least three types of bridge actions (or "candidates") were considered: do-nothing, preservation, and replacement. The utility of any candidate action was determined in relation to the utility of the do-nothing candidate.

By implementing the solution algorithms using bridge networks of varying sizes, the study established that the computational times for the Lagrangian heuristic grows with the network size in a more rapid manner than the IUC heuristic, a finding that seemed consistent, in theory, with the computational complexities of those heuristics. Of all three heuristics, the IUC was found to be the fastest.

The accuracy of the heuristics was tested for network sizes of 100; 1,000; and 9,265 bridges. Accuracy is defined as how close the objective function value (i.e., total network benefits) is to the true optimal solution (determined using CPLEX). It was determined that the average accuracies were 99.97% and 99.99% for IUC and Lagrangian heuristics, respectively, and the corresponding average network condition (in terms of their health index) was within 0.01% and 0.07%, respectively, of the optimal solution.

The study evaluated the stability of each heuristic's performance by changing performance parameters such as agency cost constraint (i.e., budgetary limit) and threshold value for the average facility condition (i.e., health index). It was determined that the performance of the IUC and Lagrangian heuristics (in terms of computational times) was quite stable across the different scenarios. Compared to the IUC heuristic, the

Lagrangian heuristic was found to be more sensitive to parameters in terms of accuracy. Also, the IUC heuristic was found to be more accurate than the Lagrangian heuristic (99.62% versus 97.88%).

Overall, the computational experiments provided valuable information regarding the appropriateness of the heuristics for the optimization solution approaches with respect to bridge management. The IUC heuristic was found to perform excellently. Its average computational time for a 12,000-bridge network, which is roughly the average network size, was 666 seconds (11 minutes), and its average accuracy was 99.62%. The IUC heuristic is also sufficiently robust and relatively easy to comprehend, and it has a close conceptual interpretation to that of classic IBC (and thus would be more familiar to the bridge management community). Although CPLEX Concert yields a true optimal solution and is very robust, it has higher computational times than the best heuristic (IUC). Given that the extra gain in average accuracy is very small (less than 0.5%), the benefit of additional computational time yielded by the CPLEX Concert may not be justified.

Overall, the results of the computational experiments were unequivocal: Of the three approaches, the IUC heuristic was consistently found to be the best approach in terms of computational speed, accuracy, robustness, and simplicity. Furthermore, the IUC method provides the simplest and quickest way to compute the optimal solution in the case of small changes in the input parameters without the need to redo the entire optimization for each budget level. With the conclusion that the IUC heuristic is most suited for the bridge optimization and decision-making problem under consideration, this heuristic was selected for coding in the application tool (i.e., software product) that was developed as part of this research.

4.5 Application Tool (Software Package)

The application tool that evolved from this project is a software module, MOOS, that can be used by bridge agencies to quickly and interactively identify and implement balanced bridge investment decisions at the network and bridge levels. In effect, the research product provides valuable management decision support for the business processes of goal setting, budgeting, and programming through efficient and interactive features and user-friendly graphical interfaces. The tool can help shed more light on the inherent trade-offs between any pair of performance measures of interest, such as facility vulnerability and agency cost, and will therefore help bridge engineers, analysts, managers, and elected officials understand what levels of each performance criterion can be "bought" for a given funding level, and also the cost to achieve any given level of a given performance criterion.

4.6 Bridge Deterioration

Testing of the software tool provided an opportunity to directly test and evaluate from a new perspective several analytical tools that have, for some time, been in use for other purposes. For example, the software uses FHWA's NBI translator program to convert forecasts of element condition states produced by the deterioration model into predicted NBI condition measures. This conversion process does not work well. The NBI translator was developed and validated primarily to help agencies avoid duplicate collection of condition data while having a neutral impact on federal funding eligibility. Many states consider the translator to work well for this purpose. However, the translator has several shortcomings that, in the researchers' opinion, make it unsuitable for a predictive planning application:

- It is not able to make effective distinctions in the highest (6 to 9) and lowest (0 to 3) NBI condition ranges. While this might not seriously jeopardize federal funding eligibility of deteriorated bridges, it seriously impacts the accuracy of condition prediction for bridges in very good condition, which most state bridge inventories have.
- It is overly sensitive to small fractions of elements in poor condition states. A Markovian deterioration model always projects at least a small amount of deterioration from each condition state to each worse state each year. For an element with five condition states, it takes only 4 years before a tiny but nonzero fraction is forecast to reach the worst-condition state. The translator tends to give too much weight to this occurrence, resulting in deterioration rates that are too rapid when expressed as NBI condition ratings.

In view of these problems, we found that the NBI condition ratings, and the sufficiency ratings derived thereof, were not sufficiently accurate in the software module. Florida DOT encountered the same issues when using a performance measure based on NBI ratings in its programming and budgeting decision-support tool and is consequently developing a new approach to NBI translation that mitigates some of these problems. A national effort to develop a translator that is more suitable for long-range planning applications would significantly enhance the utility of the product.

Another issue that was uncovered in the testing of the software module is the difficulty of developing network-level solutions that sufficiently improve conditions when expressed as a health index. The researchers investigated a wide range of utility weights, budget levels, rest periods, and other variables to fully understand the behavior of the model and the reasons for the "resistance" to improvement at high condition levels.

The researchers' experience with the software module in the present study, as well as previous experience with Pontis models, suggests that the primary cause of this problem is the rapid deterioration rate between condition states 1 and 2 for many of the bridge elements. While only the California deterioration model was used in the software trials for the present study, the same issue has been observed in models used by other states. It is expected that FHWA's Long-Term Bridge Performance Project will shed further light on this issue and will ultimately lead to the development of improved deterioration models. In the meantime, some of the states may have acquired adequate element-level data to enable the development of more accurate deterioration models for at least the phase between condition states 1 and 2 (the most common in most states' inventories). An effort that combines inspections from several states could also help model the change in deterioration rates as a function of bridge age among other variables.

In addressing this issue, it is possible with the analytical process and software module developed in the present study to be flexible with the assumptions underlying the Markovian decision process in Pontis. The Pontis network optimization requires that transition probabilities remain constant for bridges of any age because that constancy is inherent in the Markovian decision process. However, the new deterioration models are not as restrictive and can use any functional form. For example, a model that delays the onset of Markovian deterioration or that postpones the recognition of such deterioration until condition state probabilities reach a threshold level is likely to improve the models.

4.7 Cost Models

A significant contribution of the current research is a bridge-level model framework that recognizes the difference between fixed and variable costs. For a long time, Pontis has not used such information in its cost models. The framework developed in the present study duly places greater emphasis on the bridge-level perspective, where the behavior of fixed costs is critical.

The estimation of fixed costs is significantly influenced by the effective representation of scale economies and the relationships among treatments applied at the same time to the same bridge. The new action classification system used in the present study, as well as the new approach to project evaluation described on the scoping worksheet in the software module, provides the basic information required for fixed cost computations.

An excellent topic for future research is a rational model for estimating fixed costs, especially for mobilization and traffic control, using commonly available input data such as traffic and geometrics. While the precision of such a model may be limited by data availability, the bridge-level model would benefit greatly from any substantial level of precision that can be obtained in such a cost model.

4.8 Deployment and Implementation

While a frank assessment is given here of the limitations of the approach and the future research needed to improve it, the researchers believe that the analytical framework and software tool are sufficiently ready that they can be implemented right away. The implementation will substantially improve current bridge management practice, particularly when used in conjunction with Pontis or another suitable bridge management system.

With regard to the software tool, the best step to take at this time for guiding further development is pilot testing the tool in actual decision-making contexts in a group of transportation agencies. This pilot testing will uncover potential improvements and unmet needs, a necessary step toward widespread acceptance of any new model framework.

The tool, in its present form, can be deployed with Pontis by AASHTO, or its software implementation can be developed further. Through implementation experience of end users, a need may be realized for additional new software features, or even the removal of some existing features, to give the optimal balance of realism and simplicity for a successful widespread adoption of the system.

References

American Association of State Highway and Transportation Officials (AASHTO), *Guidelines for Bridge Management Systems.* 1992.

Balas, E., and Martin, C. H., "Pivot and Complement—A Heuristic for 0-1 Programming." *Management Science,* Vol. 26, No. 1, 1980.

Booch, G., Rumbaugh, J., and Jacobson, I., *Unified Modeling Language.* Addison Wesley, Reading, Massachusetts, 1998.

Cambridge Systematics, Inc., Parsons Brinckerhoff Quade & Douglas, Inc., Ray Jorgenson Associates, Inc., and Thompson, P. D., *Transportation Asset Management Guide.* American Association of State Highway and Transportation Officials, 2002.

Dalkey, N. C., and Helmer, O., "An Experimental Application of the Delphi Method to the Use of Experts." *Management Science,* Vol. 9, No. 3, INFORMS, 1963.

Everett III, H., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources." *Operations Research,* Vol. 11, No. 3, 1963.

Fowler, M., *UML Distilled.* Addison Wesley, Reading, Massachusetts, 2000.

Freville, A., "The Multidimensional 0-1 Knapsack Problem: An Overview." *European Journal of Operational Research,* Vol. 155, No. 1, 2004.

Goicoechea, A., Hansen, D. R., and Duckstein, L., *Multiobjective Decision Analysis with Engineering and Business Applications.* John Wiley & Sons, New York, New York, 1982.

Hobbs, B., and Meier, P., *Energy Decision and the Environment: A Guide to the User of Multicriteria Methods.* Kluwer Academic Publishers, Boston, Massachusetts, 2000.

Hyman, W., *NCHRP Report 511: Guide for Customer-Driven Benchmarking of Maintenance Activities.* Transportation Research Board, 2004.

ILOG CPLEX, "CPLEX Mixed Integer Optimizer." http://www.ilog.com/products/cplex/product/mip.cfm, 2005.

ILOG CPLEX, *ILOG CPLEX 9.0 User's Manual,* December 2003.

Jacobson, I., Ericsson, M., and Jacobson, A., *The Object Advantage: Business Process Reengineering with Object Technology.* Addison-Wesley, Reading, Massachusetts, 1995.

Keeney, R. L., and Gregory, R. S., "Selecting Attributes to Measure the Achievement of Objectives." *Operations Research,* Vol. 53, No. 1, 2005.

Keeney, R. L., and Raiffa, H., *Decision with Multiple Objectives—Preferences and Value Tradeoffs.* Wiley, 1976.

Li, Z., and Sinha, K. C., *Multicriteria Highway Programming Incorporating Risk and Uncertainty: A Methodology for Highway Asset Management System.* Technical Report FHWA/IN/JTRP-2003/21, West Lafayette, Indiana, 2004.

Lorie, J. H., and Savage, L. J., "Three Problems in Rationing Capital." *Journal of Business,* Vol. XXVIII, No. 4, University of Chicago Press, 1955.

Magazine, M. J., and Oguz, O., "A Heuristic Algorithm for the Multidimensional Zero-One Knapsack Problem." *European Journal of Operational Research,* Vol. 16, 1984.

Moser, M., Jokanovic, D. P., and Shiratori, N., "Algorithm for the Multidimensional Multiple-Choice Knapsack Problem." *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* Vol. E80-A, No. 3, 1997.

Organization for Economic Cooperation and Development, *Performance Indicators for the Road Sector.* Paris, France, 2001.

Poister, T. H., *NCHRP Synthesis of Highway Practice 238: Performance Measurement in State Departments of Transportation.* Transportation Research Board, 1997.

Saaty, T. L., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.* McGraw Hill, 1980.

Schneider, G., and Winters, J. P., *Applying Use Cases: A Practical Guide* (2nd Edition). Addison-Wesley, 2001.

Sobanjo, J. O., and Thompson, P. D., *Project Planning Models for Florida's Bridge Management System: Final Report.* Florida Department of Transportation, www.pdth.com/images/projplan.pdf, 2004.

Thompson, P. D., *Failure Cost Analysis for Pontis, Technical Memorandum and Dataset.* Florida Department of Transportation, www.pdth.com/images/failcost.zip, 2003.

Thompson, P. D., Najafi, F. T., Soares, R., and Choung, H. J., *Florida DOT Pontis User Cost Study: Final Report.* Florida Department of Transportation, www.pdth.com/images/fdotuser.pdf, 1999.

APPENDIX A

Review of Related Research Publications

A.1 Summary of the Information Search

To fully exploit the availability of research findings on various areas of the research, a thorough search of available information was conducted. This involved the collection and collation of relevant domestic and international research. The information search covered the following areas:

- Description of Current BMS Practice at State Highway Agencies
- Optimization Methodologies Used for BMS Decision Making
- Bridge Performance Measures
- Probabilistic Cost and Performance Models for Bridge Management
- Risk and Uncertainty Issues Facing Bridge Managers
- Methodologies for Decision Making when Faced with Multiple Objectives
- Knapsack Problems and their Algorithms and Heuristics

The dimensions of the problem statement for this research project, which guided the information search, are as follows:

- It has both bridge-level and network-level perspectives.
- It utilizes bridge inventory data and condition deterioration models.
- For both the bridge-level and the network-level perspectives, it is possible to address multiple objectives and multiple constraints. Objectives and constraints can be economic and non-economic. A constraint can be addressed within an objective function, and a criterion in an objective function can be treated as a constraint.
- The bridge-level perspective is capable of identifying the best action to take in a particular year.
- The network-level perspective can address all the bridges on a highway network or subset of the network for a series of successive periods (Gurenich and Vlahos 1999). Single- or multi-objective, multi-period optimization, without any constraints (i.e., with unconstrained needs) or with one or more constraints is the generalization of the network-level perspective.

The information search was carried out on the basis of applicability, conclusiveness of findings, and usefulness for the development of the multi-objective bridge management system (BMS) optimization methodology. The information includes areas of probabilistic modeling of bridge costs, service life performance, and risk from extreme events. Sources that were searched for published material on the subject areas include TRB, ASCE, the Australian Road Research Board, the Transportation Research Information Services (TRIS), the National Technical Information Service (NTIS), the Organization for Economic Cooperation and Development (OECD), the Transportation Association of Canada (TAC), the Transport Research Laboratory (TRRL), and the World Bank. In addition, textbooks and journals in the area of optimization and multi-criteria decision making were searched.

The information search reviewed work done by Gruver et al. (1976), who developed methods for maximizing user benefits and reducing accidents in order to identify the best improvements, including bridge work, at multiple sites. Marginal analysis was used for optimization, and the model could handle multiple constraints. Since then, the incorporation of life-cycle costs, often including user costs, has become widespread in BMSs and other highway needs models. Hyman and Hughes (1983) demonstrated that it was possible to simulate bridge needs and condition deterioration over multiple budget periods based on a combination of two objectives: minimizing life-cycle costs and minimum tolerable conditions that reflect bridge safety. Also, they developed a probabilistic, element-level approach to defining life-cycle cost profiles.

The information search also covered multi-period optimization modeling such as that by Farid et al. (1988), who helped

develop North Carolina DOT's BMS. The researchers incorporated levels of service for clear deck width, under clearance, load capacity, and condition while simultaneously using incremental benefit-cost analysis to select the best actions in each budget period. Similarly, Chen and Johnston (1987) helped North Carolina DOT adopt the Algorithm for Selection of Optimal Policy developed by Kulkarni et al. (1980) to determine optimal levels of service for competing objectives (preservation of investment, safety, aesthetics, etc.) subject to a budget constraint. Al-Subhi et al. (1990) further refined North Carolina's BMS by using 0-1 integer programming with multiple constraints in a simulation framework to assess bridge needs over time. The objective function consisted of maximizing the reduction in equivalent uniform annual costs.

Jiang and Sinha (1989) showed that multi-stage (i.e., multi-period) problems could be solved by a combination of two different types of mathematical programming methods: (1) integer programming to select the best actions subject to a budget constraint for a particular period and (2) dynamic programming to identify the best path to reach the next period. In a subsequent study for Indiana DOT, Jiang and Sinha proposed an approach that combined ranking and optimization to select projects. The ranking procedure consisted of the analytic hierarchy process, which addressed multiple criteria consisting of effectiveness of investment, remaining service life, structural condition rating, and bridge safety and community impact. The optimization method was similar to the previous study and focused on effectiveness measured as a reduction in disutility before and after performing an activity.

For purposes of deterioration modeling and programming, Golabi et al. (1982) utilized Markovian approaches for pavement management at Arizona DOT. Subsequently, the FHWA funded the development of the Pontis BMS, which was implemented under the guidance of a technical advisory committee with members from several states. In the technical report, Optima et al. (1993) explained that Pontis uses element-level Markovian deterioration models. Condition states and alternative actions were later adopted by AASHTO as Commonly Recognized (CoRe) Elements. Pontis uses Markovian Dynamic Programming—a Markovian model married to a linear program or an equivalent optimization procedure—to determine optimal maintenance policies for each bridge element assuming that bridges remain in service indefinitely. The other part of Pontis includes an improvement optimization procedure that uses incremental benefit-cost analysis within a multi-period simulation procedure. Benefits are based on avoidable user costs—accidents, travel time, and vehicle operating costs. More recently, Thompson et al. (1999a) developed an object-oriented method for the Ontario BMS. This system simultaneously keeps track of a bridge's status in both a project-level model and a network-level optimization.

Several different formulations of multi-objective optimization and near optimization were found to exist in the literature. Techniques include variants of the optimal control problem, such as dynamic programming and the calculus of variations. Other methods involve integer programming (e.g., different formulations of the knapsack problem), goal programming, neural networks, and evolutionary algorithms such as genetic algorithms. Some of these methods have exact solutions and can be solved in polynomial (finite) time. The time required to solve some other formulations, especially certain 0-1 integer programming problems, increases exponentially with the problem size. Very large problems involving tens of thousands of bridges and numerous alternatives to be considered over a long planning horizon are not tractable. Many multi-objective optimization procedures require heuristics to obtain a near optimal solution or to satisfice. Some solution methods involve meta-heuristics, which consists of applying one optimization technique to establish a parameter or solution space that, in turn, is addressed by another optimization technique.

The information search included a review of the different ways by which vulnerability, within the context of risk and uncertainty, can be incorporated as a performance measure for decision making. Shirolé and Loftus (1992) identified the most significant failure modes in a study of highway bridges in New York. These failure modes were hydraulic, overload, steel structural details, collision, concrete structural details, and earthquake. Kuprenas et al. (1998) developed a seismic retrofit program based primarily on seismic risk. Frangopol et al. (2000) examined the optimization of network-level bridge maintenance planning with the goal of ensuring an adequate level of safety at the lowest possible life-cycle cost. At the network level, the approach minimizes the expected maintenance cost of a bridge stock while maintaining the lifetime reliability of each bridge above an acceptable target level.

The information search also covered different ways of combining multiple objectives. These methods included weighting, scaling, and amalgamation. Weighting involves applying weights to different commensurable criteria so that the criteria can be added together. Scaling involves applying a scale to objectives that are not compatible. The creation of a utility function is a well-known scaling method for multi-attribute decision making, as described in Keeney and Raiffa (1993). Amalgamation involves combining different objectives that do not have a linear relationship. The Caltrans Bridge Health Index can be considered an example of amalgamation of individual element conditions into an overall bridge condition rating.

The information search points to the following methods that are applicable to various aspects of the multi-objective optimization procedure for bridge management:

- Markovian element-level condition deterioration models, especially for the AASHTO CoRe elements.

- Economic and non-economic criteria in the objective functions or constraints. Economic criteria address life-cycle and user costs.
- A mathematical programming procedure capable of addressing multiple objectives and multiple constraints in order to address the network-level problem. Constraints may consist of minimum tolerable conditions or thresholds.
- Exact or heuristic methods, as appropriate, to obtain an optimal, near optimal, or satisfactory solution.
- Incorporating the optimization procedure into some multi-staged method for addressing multiple time periods such as simulation or dynamic programming.
- Effective consideration of multiple criteria using weighting, scaling, and amalgamation to formulate an objective function consisting of such criteria.

Details of the information search are provided in the next section.

A.2 Details of the Information Search

This section provides further details to the material presented in the above summary and documents the entire information search efforts carried out as part of the research study. The information search included the current state of BMS practice at state highway agencies with emphasis on specific packages prioritization criteria in use, vulnerability assessment procedures, and criteria for bridge program optimization. The information search also includes optimization methodologies used in BMS and other facility management systems, mathematical programming models, and meta-heuristic approaches for BMS decision making. Past research on probabilistic BMS performance and cost models were reviewed, and risk and uncertainty issues in bridge decision making were identified. Past work on various aspects of bridge decision making involving multiple objectives—such as weighting methods for multiple objectives, scaling of multiple criteria, and amalgamation methods for multiple criteria—were also reviewed. Existing literature on knapsack problems and their exact algorithms and heuristics were also identified and studied for relevance to the BMS optimization problem. Finally, performance measures for bridge decision making were identified from past research and reports.

A.2.1 State of BMS Practice at State Highway Agencies

As part of the information search for the present study, an informal survey of a small number of state DOTs selected to give a broad cross section of the industry was conducted to obtain a general sense of the current state of practice in the use of BMSs to prioritize bridge needs and optimize state bridge programs. Of the ten state DOTs contacted, nine responded

to five questions. A summarized discussion of the responses is as follows:

A.2.1.1 BMSs in Use

Many states license Pontis, but not all of them currently use it for their bridge management processes. In an informal and random survey of ten state DOTs, it was found that four currently use Pontis for managing their bridge network. One state has a system based on Pontis under development and uses inspection summary reports to prioritize its projects. Another state is in the process of reviewing its current BMS and is likely to replace it with a modified version of Pontis. One state uses Bridgit, while two others use systems that they had developed in-house. Six of the states have been using a BMS for over 7 years (range of 7 to 14 years). Specifically, the BMS survey of the sample of states showed the following:

- Pontis participating states (i.e., states that participated in the development of Pontis): Arizona, California, Georgia, Kansas, Minnesota, New York, Ohio, Pennsylvania, Virginia, and Washington.
- Pontis user states: California, Georgia, Kansas, and Minnesota.
- Pontis under development: Virginia (currently uses summary reports).
- Just beginning Pontis implementation: Arizona.
- Currently considering Pontis: Pennsylvania.

A survey covering all 50 states was conducted in spring 2006 by the FHWA Office of Policy.

A.2.1.2 Prioritization Criteria Used

The survey results suggest that all states use structural condition and geometric/functional deficiency in prioritizing their bridge project needs. The next most often used categories of prioritization criteria are (a) vulnerability to floods and scour and (b) corridor improvement. Also used by some states as criteria are vulnerabilities to fracture, earthquakes, collision, or fire. Except for structural condition and geometric/functional deficiency criteria, which appear to be based on the BMS, states are generally using in-house programs and priority ranking systems for selecting bridge projects. One obvious conclusion that can be drawn is that states do not use their BMSs for selecting or prioritizing projects for state "improvement" programs in addressing these other criteria. Only one state indicated security against terrorist attacks as a highest-priority criterion to be included in its BMS.

A.2.1.3 Bridge Vulnerability Assessment Procedures

Most states indicated that they have routine vulnerability assessment procedures for floods and scour. Three of the nine

states indicated that they have a vulnerability assessment procedure for earthquakes, two states indicated that they have one for vulnerability to fracture, and two states indicated that they have one for vulnerability to terrorist attack.

A.2.1.4 Criteria for Bridge Program Optimization

The survey clearly indicated that the states favor minimizing life-cycle costs as the most desired objective for their bridge programs. Maximizing structural condition and capacity, as well as minimizing vulnerability to natural events, tied for the next most favored objective. Other favored objectives, in order of preference by states, are minimizing geometric deficiency, improving level of service, and minimizing vulnerability to terrorist attacks. One state emphasized bridge system preservation as an important criterion.

A.2.1.5 Bridge Program Performance Measures

The survey indicated that a sufficiency rating (federal or other), structural condition indicator, and health index were the most favored bridge program performance measures. Some cost-related performance measures mentioned were initial and life-cycle costs, economic benefit, or benefit-cost ratio. Suggested safety-related performance measures were adequate geometrics and risk of bridge failure. Three states indicated level of service as a performance measure.

Of the performance measures not currently handled by their BMSs, cost-related performance measures appeared to be of highest priority for future inclusion in the states' BMSs. These included life-cycle cost and incremental benefit-cost analysis of alternative projects, as well as programs. Risk and total needs reduction and multiple asset optimization over time were some other desired performance measures for future inclusion in the states' BMSs.

A.2.2 Optimization Methodologies Used in Facility Management Systems

This section discusses various optimization methodologies that have been used in bridge management by various highway agencies and researchers in various mathematical programming models and meta-heuristic approaches.

A.2.2.1 Use of Optimization in BMS and Other Management Systems

Over the past few decades, there has been considerable research done in bridge management optimization in the United States and abroad. This has been done largely in response to research requests by agencies seeking to maximize returns for bridge investments. The present section discusses details of past studies, including performance measures that were being optimized at the time of study, whether the study involved network- or bridge-level optimization, the number and nature of constraints, the use of economic analysis and life-cycle costing in the optimization procedure, whether user costs were considered, and how. For each study in this section, the discussion also includes use of the incremental benefit-cost algorithm in mathematical programming techniques, the nature of network optimization (whether it was sought to maximize some continuous variable such as percentage of bridges in good condition, or whether it was sought to determine what actions to perform and at which bridge within a given timeframe). For each past study that utilized or described BMS optimization techniques at both the network level and the bridge level, the information search went further to identify whether the two levels were considered independently of each other, whether the network level fed its outputs to the project level, or vice versa.

The Highway Investment Analysis Package. In an early study that addressed optimization problems for entire networks on an aggregated basis, Gruver et al. (1976) designed a computer model based on microeconomic theory to analyze highway investments, including bridges on road sections and limited highway networks. The accompanying software package, called Highway Investment Analysis Package, addresses either one of two objective functions: (1) maximize user benefits (consisting of vehicle operating costs, travel times, and accidents) or (2) reduce accidents (measured in one of several ways). The computer model uses marginal analysis to select among alternatives and staged improvements at each analysis site. The selection process can handle a broad set of constraints.

Life-Cycle Cost Analysis Model for Bridge Replacement. Considerable work has been done in the optimization of decision making, specifically for BMSs. Hyman and Hughes (1983) developed a computer model for life-cycle cost analysis of statewide bridge replacement needs for Wisconsin DOT's state highway system plan. This was one of the earliest efforts to address the network-level, intertemporal optimization problem regarding the repair versus replacement of bridges under conditions of uncertainty. The computer model selected the least-cost option—repair or replace—for each structure on Wisconsin's highway network in each year over a user-defined planning horizon. Expert elicitation was used to define alternative life-cycle activity profiles involving different bridge components (decks, bearings, abutments, etc.) for various combinations of bridge configurations and materials. Analysts had to establish probabilities for alternative life-cycle activity profiles for different groupings of span configurations and materials. These life-cycle activity profiles were randomly assigned to different types of structures on Wisconsin's high-

way network. The computer model calculated the discounted present value of life-cycle costs of the repair and replacement options and selected the option with the minimum cost, although analysts could set a minimum tolerable threshold to reflect an unsafe condition of the bridge. Thus, the model addressed two objectives: minimizing life-cycle costs of the actions selected and avoiding deterioration that was unsafe. Forecasting models predicted the condition of each bridge, and each year the simulation model would determine the cost-minimizing set of actions for all bridges on the network. The model was useful because, like the present study, it sought to make available a means of investigating the effect of different bridge repair and replacement policies on some performance measure over the planning horizon. Using the model, the number of bridges requiring deck or bearing replacement in each year could be determined. From the bridge-level output, it was also possible to plot the optimum path for bridge condition assuming the least-cost action was always taken. If a few bridges reached an unsafe condition triggering replacement, then the path for bridge condition was near optimal with respect to two objectives: minimum life-cycle cost and ensuring bridge safety. The Wisconsin study spawned further research at other states, as discussed in subsequent sections of this appendix.

Optimal Resource Allocation for State Highway Bridge Maintenance, Rehabilitation, and Replacement Using Incremental Benefit-Cost Analysis. In a series of improvements to the Wisconsin model, researchers at North Carolina State University used the concept of incremental benefit-cost analysis for optimal allocation of state highway bridge maintenance, rehabilitation, and replacement funds in North Carolina (Farid et al. 1988). The reasons for the extension were threefold: to assess the feasibility of incorporating user costs into the objective function (user costs are defined as avoidable travel time costs, accident costs, and vehicle operating costs); to calculate avoidable user costs based on level of service for the approach roadway alignment, clear deck width, vertical clearance, and load capacity; and to use incremental benefit-cost analysis or another procedure to determine the near optimal or optimal bridge actions subject to a budget constraint for each year into the future. First, an effort was undertaken to assess the feasibility of using incremental benefit-cost analysis to optimally allocate a limited budget to maintenance, rehabilitation, and replacement actions for bridges. The project involved a thorough investigation of the issues involved in applying INCBEN, an algorithm for performing incremental benefit-cost analysis, to bridges. Procedures were developed for defining future maintenance, rehabilitation, and replacement costs, and then INCBEN was applied to a sample of bridges subject to a budget constraint to determine the optimal—or, more strictly speaking, near-optimal—set of actions.

In an earlier study at North Carolina State University, researchers used level of service as a measure of bridge performance to formulate objectives and subsequently investigated optimal improvement actions and times and budget prediction (Chen and Johnston 1987). After establishing the feasibility of assessing avoidable user costs and selecting bridge actions based on benefit-cost analysis, the researchers developed a model that selected the least-cost action—maintenance, rehabilitation, or replacement—for each bridge on the basis of agency (i.e., maintenance) costs and user costs. The overall model included a means of incrementally "aging" each bridge over time; predicting the condition of the deck, superstructure, and substructure; and determining maintenance, rehabilitation, and replacement needs for each year over the planning horizon.

The incremental benefit-cost algorithm that was used in the 1988 North Carolina study was developed at the Texas Transportation Institute (TTI) for purposes of evaluating safety countermeasures (McFarland et al. 1983a) but was subsequently found useful in applications for other functional areas of highway asset management, such as bridges and pavements. That research showed that multiple objectives weighted by dollars could be entered into the expression for incremental benefits. TTI also developed dynamic and integer programs that produced nearly identical optimum solutions (Subramanian and Schafer 1983; McFarland et al. 1983b). The researchers stated that of the three methods, only the incremental benefit-cost algorithm was capable producing a priority ranking subject to a budget constraint. It has come to be recognized that when the objectives are expressed quantitatively and assigned suitable weights, the incremental benefit-cost algorithm can also address problems involving multiple objectives or criteria even if they are not expressed in dollars. Like the TTI study, the present study also considers the incremental benefit-cost ratio concept. Also, the present study duly recognizes the different dimensions of various performance measures and consequently adopts the concept of utility by which all measures may be converted into non-dimensionless units.

The incremental benefit-cost concept developed by TTI was later espoused by Hudson et al. (1987) in *NCHRP Report 300: Bridge Management Systems.* In the report, the researchers did not specifically address the concept of multi-objective optimization, but implicitly addressed multi-criteria decision making in the form of a decision tree. This report was originally intended to provide a blueprint for the development of a BMS that uses deterministic deterioration models and ultimately evolved into the Bridgit Bridge Management System, which used probabilistic deterioration models.

Network-Level Bridge Budget Forecasting and Resource Allocation. In yet another North Carolina DOT study, Al-Subhi et al. (1990) developed a network-level bridge budget forecasting and allocation module. The inputs to the model

included yearly budget constraints, performance objectives, and policies. The user also specified immediate needs for deficient bridges, level-of-service goals (acceptable or desirable), minimum allowable condition ratings for bridge elements, the highest condition rating at which a bridge element should be rehabilitated, and unit costs. The model used 0-1 integer linear programming with multi-choice constraints to optimize decisions each year in the analysis horizon by maximizing the reductions in equivalent uniform annual costs. The model was used to predict the condition of a network of bridges over time given different yearly budget levels.

Network-Level Deterioration Monitoring Using Markovian Models. In the early 1980s, efforts in the area of pavement management yielded advances such as the development of Arizona's pavement management system (PMS) (Golabi et al. 1982) that had a large influence on BMS methodologies. The most distinguishing feature of the Golabi et al. work was the introduction of Markovian deterioration models in a Markovian dynamic programming framework that allowed the optimization of infrastructure facility actions on the basis of Markovian dynamic programming. In the Arizona PMS, the network optimization system consists of two interrelated models: a short- and long-term model. In the short-term model, the network performance is expressed in terms of proportions. The objective of the network optimization system is to identify the least-cost actions that would maintain a preestablished proportion of road sections in condition states desired by policy makers and not a greater proportion in undesirable states. A linear program is used to find the solution to the short-term model, which is a steady-state optimal policy. Such goals reflected the desire of the Arizona DOT decision makers to be able to influence the time taken for the network to reach the optimal steady state and to be able to impose different short- and long-term performance standards. The long-term model finds a policy that minimizes the long-term expected costs subject to a variety of constraints, including performance regarding acceptable and unacceptable states.

The applicability of the Arizona PMS Markovian dynamic programming to bridge management—indeed, bridge components—was aptly recognized in the FHWA Bridge Management Systems Demonstration Program report (O'Connor and Hyman 1989), which stated that the Arizona PMS approach could be modified and applied to determine future bridge deck needs. The O'Connor and Hyman report provided a comprehensive review of approaches to BMSs that had evolved by the late 1980s.

In furtherance to its Bridge Management System Demonstration Program, FHWA funded the development of network optimization for bridge improvements, which is known as Pontis (Optima et al. 1993). In their work, Optima et al. developed two models. The first model employed Markovian dynamic programming at the element-type level to determine long-term, least-cost, steady-state maintenance policies for each significant component of each type of bridge on the highway network. A linear program was used to determine the least-cost, steady-state condition of each type of bridge element. (Note: it was later replaced with a recursive procedure that achieved the same optimization.) A unique feature of Pontis is the use of expert opinion in deriving the initial transition probabilities for condition states and subsequent use of biennial bridge inspection data with Bayesian updating techniques to revise the transition probabilities. The second optimization procedure within Pontis concerns the optimal choice of improvements at any given time—whether to strengthen, widen, or replace. That part of Pontis determines whether there are avoidable user costs resulting from the existing level of service represented by clear deck width, vertical clearance, and load capacity, much the same as the North Carolina Model. The model compares the discounted present value of the avoidable user costs with the discounted present value of the maintenance costs of keeping the bridge in service in perpetuity consistent with the optimal maintenance policy. In a manner similar to the Wisconsin and North Carolina BMSs, Pontis utilizes a simulation procedure. Each bridge is "aged" a year at a time, and future conditions and needs are projected. It is possible to perform a variety of sensitivity analyses by varying the budget constraints in each year. Finally, Pontis addresses a variety of issues important to decision makers but outside of the optimization frameworks. The user can set flags to identify serious conditions such as scour and risks associated with fracture critical features.

Guidelines for Establishing Bridge Management Systems. By the time the initial version of Pontis had been developed, enough experience with BMSs had accumulated that the states commissioned the development of a set of guidelines on BMSs (Hyman and Thompson 1992). The guidelines discuss the minimum requirements for a BMS. These requirements fall into two fundamental classes, one addressing organizational issues and the other the analytical needs of a BMS. The analytical needs include maintenance cost minimization and multi-period optimization. The guidelines state that the optimization procedure needs to account for all the bridges on the network, or any subset, and address structure deterioration, traffic growth, agency and user costs, and the efficiency and effectiveness of agency expenditures in meeting agency objectives.

Incorporation of Optimization Techniques in Bridge Management Decision Making. The information search on BMS optimization practice also showed that while some studies sought to optimize the percentage of bridges in a certain desirable condition, others sought to determine the best set

of actions to carry out, at which bridge and in which year. It was also seen that the consideration of economic analysis and life-cycle costing was prevalent in most BMS optimization procedures, and those conceived after 1990 mostly tended to incorporate user costs into the overall picture. The use of the incremental benefit-cost procedure (initially developed for safety management applications) has been embraced by most BMS optimization methodologies, but a few researchers have explored the use of mathematical programming techniques with some success. The subsequent section provides a more detailed discussion of past work on mathematical programming not necessarily within a BMS context.

At an international level, there have been significant efforts at incorporating optimization into bridge management practice. The National Bridge Investment Analysis System (NBIAS) and PlanOpt decision-support systems were developed for FHWA and the Swedish National Road Administration, respectively. Both systems feature interactive "what-if" analysis based on optimization and simulation of the process of budget allocation for the national stock of bridges over a long-term period. Also, both systems simulate the influence of policy factors on the network performance over a substantial time horizon for multiple measures of effectiveness. In Finland, the BMS utilizes both network- and bridge-level models (Söderqvist and Veijola 1999). A ranking system is used for the selection of bridge projects based on the individual repair index (a function of bridge structural condition, damage class, and repair urgency). The project-level model uses the recommendations and goals from the network-level model to decide on the repair measures in individual repair projects. Thompson et al. (1999a) presented an object-oriented method for the Ontario BMS. This system makes use of an object-oriented software architecture to simultaneously keep track of a bridge's status in both a project-level model and a network-level model. Whenever new information is introduced at either the project level (e.g., a new inspection) or the network level (e.g., a change in the budget constraint), both models are updated accordingly. At its core, the network model is a single-criterion incremental benefit-cost algorithm, but is architected to respond to changes in budget constraints without repeating the entire analysis. This makes it very efficient in analyzing the funding versus performance trade-off.

It is clear from the above discussion that for purposes of the present study, valuable lessons can be learned from past research and reports that have highlighted key aspects of bridge management optimization practice in the United States and abroad. It was seen that performance measures that were sought to be optimized largely consisted of some form of bridge condition—expressed as a structural index, sufficiency rating, or some other index to indicate the overall or elemental health of the bridge—and constraints were typically single (e.g., an annual budgetary constraint). The information search revealed that most studies had either bridge-level or network-level methodologies. For the relatively few that had both methodologies, it was typically sought to obtain results at the bridge level first before proceeding to the network level, with the exception of Pontis policy optimization and the BMS of Finland.

A.2.2.2 Mathematical Programming Models

In past research, several mathematical programming methods have been proposed to allocate resources to achieve a certain objective in the best possible manner. These methods have been as diverse as the intentions of the researchers who used them and have included integer programming, dynamic programming, linear programming, goal programming, and so forth.

The classical optimal control problem, a fundamental aspect of microeconomic theory, inspired the approach to the Wisconsin BMS optimization model and has been discussed extensively by Intriligator (1971) in his book *Mathematical Optimization and Economic Theory*. The control problem consists of allocating scarce resources among competing ends over an interval of time and choosing optimal time paths (i.e., trajectories) for certain control variables from a given class of time paths, called the "control set." The objective function, which is to be maximized, is a mapping of control trajectories to points on the real line. Some formulations involve multiple criteria in the objective function—for example, the selection of the optimal set of co-state variables representing how each variable of interest changes over time.

There are three traditional approaches to the control problem: the calculus of variations, the maximum principle, and dynamic programming. All three approaches are interrelated. Indeed, dynamic programming, which can handle discrete decisions, is a general case of the calculus of variations and implies the maximum principle's conditions. The equations of both the calculus of variations and the maximum principle are continuous or piecewise continuous functions. Also, the maximum principle and dynamic programming pertain to the same type of control problem. Dynamic programming is derived from Bellman's equations and the principal of optimality—an optimal policy has the property that whatever the initial state and decision are, the remaining decisions must also be an optimal policy with respect to the state resulting from the first decision.

The simulation approach used in the Wisconsin model was considered more tractable for solving an intertemporal optimization problem that addressed life-cycle and safety concerns involving a collection of more than 10,000 bridges. In contrast, various approaches to the optimal control problem either forsake discrete solutions appropriate to individual bridges for continuous solutions (the calculus of variations and the maximum principle) or run into the curse of dimensionality

for large-scale problems involving tens of thousands of bridges over periods of 10 to 50 years (dynamic programming and the maximum principle). However, as discussed below, dynamic programming has proved valuable in solving multi-stage optimization problems when it is used to focus on the intertemporal aspect of the bridge management problem. Over the past three decades, a large literature on optimal control applied to multiple objectives has evolved.

Sinha et al. (1981) developed and applied goal programming techniques to achieve optimum allocation of federal and state funds for highway system improvement and maintenance. Like the present study, their work involved a multi-objective framework (four system objectives were used), and six alternative activities were considered for each facility. While goal programming is a useful technique for multi-criteria optimization, it involves the use of the percentage of facilities in certain condition states rather than discrete decision variables (as envisaged in the present study).

In a study that implicitly utilized the concepts advocated in the Intriligator text, Jiang and Sinha (1989) developed an optimization model for the Indiana BMS based on dynamic programming and integer linear programming. The model selected projects while maximizing the effectiveness or benefit to the system subject to the constraints of available budget over a given program period. Markovian chain transition probabilities of bridge conditions were used to predict or update bridge conditions at each stage of the dynamic programming. The use of dynamic programming, in combination with integer linear programming and the Markovian chain, facilitates efficient management of the BMS that comprises several hundreds of bridges. It was mentioned that application of dynamic programming ensured that the results were optimal not only for a program period but also for the subperiods. Dynamic programming was described as a way of looking at a problem that may contain a large number of interrelated decision variables so that the problem is regarded as though it consisted of a sequence of problems, each requiring the determination of only one (or a few) variables. Dynamic programming reaches global optima rather than local optima. But a major limitation is that if there are too many variables, then there are computational problems relating to the storage of information as well as the time it takes to perform the computation.

The dynamic programming procedure divided the state and federal budgets of each year into several possible spending portions, and then integer linear programming selected projects by maximizing yearly system effectiveness subject to different budget spending. The effectiveness was measured in terms of the coefficient of safety condition, the coefficient of community impact in terms of detour length and bridge average daily traffic (ADT), and the change in the area under performance curves achieved by the activity. In terms of dynamic programming, each year of the program period was consid-

ered a stage. The federal and state budgets were taken as state variables. Each activity of a bridge is a 0-1 decision variable of the dynamic programming as well as the integer linear programming. The effectiveness of the entire system was used as the return of the dynamic system. The dynamic programming problem for the Indiana BMS optimization was solved by developing recurrence equations, and the branch-and-bound method was used to solve the integer linear programming problem.

In a follow-up study for the Indiana BMS, Jiang and Sinha (1990) proposed an approach to combine the ranking and optimization techniques to select projects. The ranking model was developed using the analytic hierarchy process (AHP), which helps the decision maker to set the relative order of importance of different projects, a useful tool when subjective judgments are involved. Utility curves were developed for various evaluation criteria, which in turn were used to compute the expected utility of each alternative project. A four-strata hierarchy (including an overall goal of the ranking method, objectives that bridge managers would like to achieve, evaluation criteria with utility curves, and individual bridges) was developed, and the criteria weights were obtained by applying the AHP. The multiple criteria used were the following: effectiveness of investment, remaining service life, structural condition rating, bridge safety, and community impact. A similar optimization model was used as described earlier with a modification in the way effectiveness was measured. The effectiveness was replaced by the change or reduction in disutility of a bridge before and after performing the activity. In a subsequent study, it was shown that the Indiana BMS can be used to conduct trade-off analyses by varying the model parameters (such as funding levels) to analyze the effect of various spending policies on bridge condition and other performance measures (Vitale et al. 1996).

Harper et al. (1990) utilized linear programming techniques to optimize decisions for bridge management. Their model also took due cognizance of the fact that optimization parameters are not deterministic but vary considerably and typically follow a certain probabilistic distribution. The module consisted of three network optimization solution models: long-term (steady-state) goal-setting model, the multi-year (short-term) planning model, and the financial model. These were based on a Markovian decision model using linear programming techniques. Bridges were stratified according to bridge type, climate, and functional class, and a separate linear program was solved for each stratum. The models work on major bridge segments, deck, superstructure, and substructure. The long-term model first established the steady-state performance goals that provide targets for the multi-year and financial models. The steady-state model takes inputs as desirable and undesirable condition states, proportions in these states, maximum and minimum allowable proportions, and Markovian

transition probabilities. The model optimized for proportion of segments in a given condition state receiving a given action and the average cost for each segment. The multi-year model determined the optimal maintenance policy for each year in the planning horizon. The financial model imposed a network-wide budget constraint across all strata.

In a multi-criteria optimization model based on goal programming, Ravirala et al. (1996) analyzed multiple goal-oriented scenarios for a bridge capital improvement program. Various goals, objectives, and constraints were identified and formulated in a goal program. The model determined an optimal multi-year program that minimized the weighted sum of treatment costs and deviations from goals. The goals defined in the study included an annual budget goal for each geographical class of bridges and an annual average system condition goal for each bridge component for each geographical class. The model was solved as a linear program and was applied to bridges in New York State.

Guignier and Madanat (1999) presented a Markovian decision model for joint optimization of maintenance and improvement activities to aid budget allocation. The infinite horizon model was used to study steady-state policies while relaxing the assumption of age homogeneous condition state transition probabilities. The model also allowed for carry over of annual budget, which could be spent more efficiently in later years. Facility-specific representation was used in the model because the improvements were selected for individual facilities, whereas maintenance could be optimized at the network level. The computational complexity issues were discussed, and the joint optimization was thought of as a significantly larger problem due to a higher number of decision variables and constraints. However, the authors asserted that it was not a critical issue since it was a planning problem, which would be solved once every year.

At least one study (Li and Sinha 2004) utilized the Lagrangian relaxation technique for optimization of improvement actions in a multi-functional context (bridges, pavements, congestion, and safety). The authors developed a methodology for optimal project selection for overall asset management. A set of goals and a set of performance criteria under each goal were identified, and multi-attribute utility functions were developed in order to do the trade-off analysis. This was done for two alternative scenarios of risk and certainty. The authors formulated the optimization problem as a multi-choice, multi-dimensional knapsack problem.

A.2.2.3 Meta-Heuristic Approaches for BMS Decision Making

The 1990s were characterized by an upsurge in the investigation and use of nontraditional techniques to arrive at optimal control of resources for highway facility management. These techniques have included neural network, fuzzy logic, and genetic algorithms.

Mohamed et al. (1995) used artificial neural networks (ANNs) to optimize available resources to generate the group of bridge improvements that minimize the loss of network benefits. The bridge problem was perceived to have two dimensions: the time dimension (which seems to be consistent with the bridge-level module in the present study) and the network dimension (which seems to be consistent with the network-level module in the present study). A dynamic programming model was used to handle the time dimension, and a two-layer ANN was developed for the network dimension. Each neuron receives many inputs, which are converted to a single output by using activation and output functions. The number of neurons in the second layer of the ANN was equal to the total number of bridges times the number of activities for each bridge. The network is supplied with the loss and initial cost associated with each alternative of each bridge and the available budget. Once the network reached a steady state, the output of neuron (0 or 1) indicated which activities were to be carried out. The authors argued that the ANN was advantageous because a high speed of operation could be achieved by parallel implementation in hardware or software.

Fairly recent work in pavement management that may be transferable to bridge management indicates that genetic algorithms can be applied to neural networks in order to optimize the process of setting values for parameters regarding the neural network solution, which is usually left to the judgment of the network developer. In a study that analyzed an evolutionary neural network model for the selection of pavement strategy, Taha and Hanna (1995) showed that such judgmental approach can lead to slow convergence and/or poor performance regarding unseen instances. Their research applied genetic algorithms as a search technique to design the best neural network model to develop an optimum maintenance strategy for flexible pavements. They described both an evolutionary learning system using gradient descent learning and a genetic algorithm to determine the network connection weights. The input vector consists of factors that affect the selection of a flexible pavement maintenance strategy. The output vector consists of different pavement maintenance strategies available. The authors used Brainmaker Professional, a commercially available software package, to develop the neural network model. One hundred unseen cases were used to test and validate the model, and only six cases were misclassified. The average error rate was 0.024. While that research involved a single objective rather than multiple objectives, it had multiple criteria as inputs. Furthermore, the research indicates that genetic algorithms and neural networks can be combined to handle multi-objective optimization problems.

Genetic algorithms were also used by Pilson et al. (1999) to solve the multi-objective optimization of pavement scheduling

problems at both the network and bridge level. The authors contended that pavement management is ideally suited for directed random search heuristics, such as genetic algorithms. They first explored the use of a genetic algorithm to address the project-level pavement management problem and discussed cases for single- and multi-objective optimization. Then the authors showed how to solve the general network problem using project-efficient surfaces. The authors contend that using efficient surfaces to break down the network problem into project subproblems holds a great deal of promise for overcoming some of the existing problems in pavement management. A similar approach may be useful for bridge management.

In a research that addressed optimization of resources for pavement management, Fwa et al. (2002) used a genetic algorithm procedure to solve multi-objective network-level pavement maintenance resource allocation problems. This method is in distinct contrast to single-objective optimization methods that are typical of most research in pavement resource optimization. Two genetic algorithm methods for finding an optimal solution were applied. The work explored finding a Pareto optimal solution set and a rank-based fitness evaluation. The authors concluded that the robust search characteristics and multi-solution handling capability of genetic algorithms were well suited to optimization analysis. A numerical example of pavement maintenance optimization involving two and three objectives were explored, compared, and evaluated.

Chan et al. (2003) applied a genetic algorithm optimization technique to simultaneously satisfy the objectives of headquarters and districts or regional offices with respect to pavement condition, a methodology that has potential application in bridge management. Their approach sought to explicitly recognize that different regions have different characteristics and priorities and that applying a common formula as the basis for resource allocation may not achieve the best results for the district networks and the agency network as a whole. The virtues of their multi-objective optimization procedure were illustrated using an example involving a headquarters and three regions. A two-staged genetic algorithm was used to optimize the allocation of funds to the three regions for pavement maintenance. The results were compared with traditional allocation systems and shown to yield better overall performance measures at a network level.

The concept of genetic algorithms was also utilized by Hegazy et al. (2004) in their study that carried out optimization of repair actions over facility life-cycle using genetic algorithms. Like the Pilson et al. (1999) study discussed above, the Hegazy et al. (2004) study utilized a comprehensive framework for a bridge deck management system that aims to integrate project-level and network-level decisions into one model so that the costs are optimized at both levels. While such an attempt is indeed laudable, it may be problematic to implement such methodologies for a potentially large bridge network with multiple repair alternatives and a fairly large number of years. The constraints considered were yearly budget limits, minimum allowable condition state for individual bridges, and the network. The solution representation in the form of chromosomes was achieved by constructing a string of (N by T) elements, where N is the number of bridges and T is the planning horizon. Each element had a value of 0 to 3 corresponding to repair options. An initial population of chromosomes (i.e., feasible solutions) was generated, and crossover and mutation (GA operators) were applied to improve the solutions. The methodology was applied to a small network where the algorithm reached near-optimal solution.

Fuzzy logic techniques have also seen implicit applications in bridge inspection and, consequently, decision making for the best intervention (Tee et al. 1988).

The information search showed that the use of nontraditional techniques, such as neural networks, fuzzy logic, and genetic algorithms, for bridge management optimization shows considerable promise, given its apparent success with bridge deck management and pavement management. This offers encouragement to look beyond traditional methods to solve particularly hard problems such as the one in the present study.

A.2.3 Probabilistic Cost and Performance Models for Bridge Management

This section presents the information gleaned through a comprehensive search of publications and other documentation related to the development of probabilistic cost and performance models for bridge repair or rehabilitation. These publications include work already done by state departments of transportation (DOTs), academic institutions, private organizations, FHWA, the American Association of State Highway and Transportation Officials (AASHTO), and the National Cooperative Highway Research Program (NCHRP). The present study does not include development of cost models. However, it is useful to review past modeling efforts, particularly cost models and probabilistic performance models, to gain further insights into the bridge management state of practice.

A.2.3.1 Cost Models

Sobanjo and Thompson (2001) provided useful information on cost of bridge projects in Florida, while Adams and Barut (1998) presented data on bridge maintenance, rehabilitation, and repair (MR&R) cost for the Wisconsin DOT. In a similar study, Chengalur-Smith et al. (1997) developed agency cost models for bridge MR&R activities. Also, Abed-Al-Rahim and Johnston (1991) developed methodologies and procedures for estimating the unit costs of bridge replacements in North Carolina, while Gannon et al. (1995) provided cost models for concrete bridge protection and rehabilitation with national

data. Ballou et al. (1997) also provided information on bridge rehabilitation costs. These studies focused on analysis of network-level cost data. Studies that focused on developing cost models for specific treatments include those of Tam and Stiemer (1996), who developed a bridge corrosion cost model for coating maintenance, and Wipf et al. (1987), who concentrated on cost analysis for activities that strengthen existing bridges.

Thompson and Markow (1996) reported that in Pontis, the cost information is based on the average unit cost of activities classified by element, condition, state, and type of work. They further reported that at most DOTs, the management of historical contract replacement cost data has been automated, but only a few states have automated bridge maintenance cost data. Nearly all states with automated bridge-level data collection have computerized contract and maintenance cost estimation. Thompson and Markow further found that most states have the capabilities to estimate the cost components of projects, but few have procedures to track and update cost factors that might be used for network-level and project-level cost estimating. Also, it was established that in the development of cost factors, most states use subjective agency experience rather than the results of historical data analysis. The researchers stated that at that time, only California, Maryland, Minnesota, Mississippi, Nebraska, New York, Texas, Washington, and Virginia systematically tracked the costs developed in their design processes and periodically updated their bridge repair cost models. In general, the most difficult unit costs to develop across the surveyed agencies were those of maintenance, repair, and rehabilitation.

In a more recent NCHRP report, a methodology for bridge life-cycle cost analysis (BLCCA) was described (Hawk 2003). The methodology introduced vulnerability and the uncertainty cost analysis to yield a more realistic estimate of life-cycle cost. The BLCCA package is designed for application to individual bridges. The project-level focus of BLCCA distinguishes it from other currently available methodologies such as Pontis or Bridgit. In BLCCA, default values are provided for cost parameters, and it is stated that the user will benefit from the development and the use or parameters specific to the structure and environment in question. Such parameters include bridge repair costs.

Many state agencies have reliable data on bridge replacement cost, but only a few have reliable data on bridge maintenance costs (Thompson and Markow 1996). Even though some states collect data on preservation activity cost, the data collected is often state specific or incompatible with the needs of nation-wide studies. Therefore, many studies have called for development of bridge preservation activity cost databases for the benefit of future research on bridge costing. For instance, a University of Wisconsin study (Adams and Barut 1998) affirms that many historical records of bridge activities are general-

izations or aggregations of specific preservation actions used by BMSs such as Pontis. Therefore, the costs of specific bridge preservation activities are not easily determined from historical records. A similar situation was encountered in the development of cost estimates for Florida (Sobanjo and Thompson 2001), where a major problem was the inability to match the historical cost data record for rehabilitation actions on a bridge to the deteriorated state of the bridge when the action was performed.

Jiang et al. (2000) analyzed optimal life-cycle costing with partial observability. They argued that determination of structure costs in an "optimal" manner requires taking into account lifetime management policies concerning inspection and maintenance costs in addition to construction cost. Recent literature on lifetime costing reveals that Markovian decision process models are being used to incorporate management policies into structural reliability analysis. The authors presented an optimization model that uses a partially observable Markovian decision process.

A.2.3.2 Performance Models

Past literature on probabilistic performance modeling has included Markovian process models, Bayesian decision models, and survivor curves. Markovian theory assumes that change in condition from one state to another depends only on its current state. Markovian process models are frequently used as a means of incorporating uncertainty into condition prediction. Bayesian theory allows for combining both subjective and objective data to develop predictive models using regression analysis (Butt 1991). Survivor curves represent the percentage of highways that remain in service as a function of time (McNeil et al. 1992).

Markovian Process Models. Markovian process models are developed from estimates of probability that a given condition state will either stay in the same state or move to another state. The probability of each of these events is estimated based on historical field data or the experience of agency personnel. For instance, Washington DOT started to use Markovian transition probabilities of pavement condition states in the early 1970s; Indiana DOT used the Markovian chain for bridge performance prediction and bridge management in the 1980s (Jiang et al. 1988); Arizona DOT used the Markovian process for pavement performance prediction in the 1980s and improved the transition probability matrices by introducing the concept of pavement probabilistic behavior curves (Wang et al. 1994); and Ohio DOT developed Markovian deterioration models using Monte Carlo simulation for pavement performance analysis (Tack and Chou 2001). Pavement or bridge conditions can be predicted at any point in the future as long as the initial condition state and transition matrix are known.

Using the probability transition matrices, an agency can also develop pavement performance models by calculating plotted points based on matrix multiplication. The Markovian process assumes time homogeneity of the transition probabilities, which may not be realistic for pavement or bridge performance. One remedying measure to this limitation is to incorporate the use of zones within which the transition process is stationary.

Bayesian Regression Analysis. In Bayesian regression analysis, both subjective and objective data are used to develop prediction models. An example of this approach was provided in a research project in the state of Washington (Kay et al. 1993). In this project, by using both the subjective opinions of experienced personnel and objective data obtained from mechanistic models, new models were developed to relate pavement fatigue life as a function of asphalt consistency, asphalt content, asphalt concrete proportion, and base course density. Using Bayesian regression analysis, the model parameters were found to be random variables with associated probability distributions.

Bulusu (1996) examined Bayesian approaches and econometric methods for modeling bridge performance. The Bayesian approach was considered to combine expert opinion and observed data. Previous models were developed using observed data, and the condition states of bridge elements were predicted using the transition probabilities estimated from a regression-based approach. The resulting transition probabilities were modified because of various changes made by bridge management experts at INDOT. The developed binary probit models considered the discreteness of the condition states and related the deterioration level with explanatory variables, such as region, cumulative ADT, and structure type. Transition probabilities were updated for two successive inspection periods. Binary probit models that were developed as a function of various explanatory variables estimated the probability of a bridge element deteriorating to the next condition state. The models were used in forecasting bridge condition. The information search also showed some use of Markovian techniques in infrastructure performance modeling (Optima et al. 1993; Jiang et al. 1988; Guigner and Madanat 1999). Morcous et al. (2002) investigated the use of case-based reasoning in bridge performance modeling.

A.2.4 Risk and Uncertainty Issues in Bridge Decision Making

The concept of risk and uncertainty comes from the inability to know what the future will bring in response to a given action today. The input variables could also inherit a certain extent of uncertainty. The probability distribution of the outcomes is known under risk and unknown under uncertainty. Risk can be subjective or objective. Subjective risk is based on personal perception that may be related to consequences of failure as well as the ability or inability to control the situation. Objective risk is based on theory, experiment, or observation.

A.2.4.1 Risk Assessment and Management

Essential elements of risk assessment and risk management (Ezell et al. 2000) can be summarized in the following questions.

For risk assessment:

- What can go wrong?
- What is the likelihood that it will go wrong?
- What are the consequences?

For risk management:

- What can be done?
- What are the associated trade-offs in terms of cost, risks, and benefits?
- What are the impacts of current management decisions on the future?

Limited resources need to be allocated to maximize utility, safety, and condition of bridges. During the service of a bridge, various risk factors might influence the following:

- The performance,
- Maintenance and repair (M&R) and operating cost,
- Longevity and fitness of the facility, and
- Cost-effective management of bridges.

Therefore, it is important to incorporate the risk factors that affect the necessity and timing of various activities to mitigate these risks.

Risk Factors. The risk factors could be extreme events such as environmental disasters (e.g., earthquakes) or human-made hazards (e.g., terrorist attacks). There could be accidental risks due to collision or overloading. There could also be risk of failure due to everyday deterioration or the agency's failure to maintain the facility. It is important to incorporate these risk factors into the life-cycle cost or objective function for cost-effective management. The following failure modes were identified as most significant in terms of potential damage they can cause to highway bridges in New York State (Shirolé and Loftus 1992):

- Hydraulic,
- Overload,

- Steel structural details,
- Collision,
- Concrete structural details, and
- Earthquake.

These failure modes were identified as a part of the Bridge Safety Assurance (BSA) program that provides a systematic method to reduce vulnerability of the state's bridges to all potentially significant modes of failure. Screening, classification, and rating schemes were presented to provide a measure of bridges' vulnerability to failure.

Hastak and Baim (2001) identified risk factors that influence the cost-effective management, operation, and maintenance of infrastructure as well as how and when in the project life cycle the risk factors impact associated facility costs. Risk factor was defined as a factor that has the potential to adversely influence the life-cycle cost. Based on the failure modes, broad risk factors were identified that link the various physical symptoms together. Some of the common risk factors identified were management focus, forecast and calculation, design concept and details, deterioration and natural hazards, communication, material selection, quality control, construction, maintenance practices, and training of inspection personnel.

Engineering risks and uncertainty are related to the interaction of environmental factors and bridge characteristics causing partial or complete loss of functionality, collateral damage, or both. Hawk (2003) identified some of the following common factors:

- A condition-related reduction in load capacity, life, or both;
- Seismic vulnerability;
- Bridge scour;
- Overloads; and
- Collisions.

Chang and Shinozuka (1996) presented a framework to combine the discounted cost for seismic retrofit and damage/repair cost from seismic events for more realistic life-cycle cost estimation for bridges. Life-cycle costs consisted of four components:

$$C = C_1 + C_2 + C_3 + C_4 \qquad (A\text{-}1)$$

where

C = Total life-cycle costs,
C_1 = planned costs and owner's costs,
C_2 = user costs associated with C_1,
C_3 = unplanned costs and owner's costs (including expected value of repair costs resulting from earthquake damage), and
C_4 = user costs associated with C_3.

The framework incorporates the benefit-cost trade-off associated with seismic retrofit. The costs of seismic upgrading are reflected in C_1 and C_2, and the benefits are in terms of reduced unplanned costs C_3 and C_4. C_3 was calculated as follows:

$$C_3 = \sum_{t=1}^{T} \sum_{k=1}^{K} G(x, d_k, t) [r_k C_1(x)] z(t) \qquad (A\text{-}2)$$

where

G = probabilistic condition or performance index,
x = vector of design parameters,
d_k = damage state,
r_k = percentage of replacement (i.e., initial) cost, and
$z(t)$ = discount factor.

The factor G was based on natural deterioration rate, condition-improving effects of maintenance, and seismic retrofit activity and hazard rate (i.e., seismic ground intensity). This index deterioration was modeled as a Markovian chain in which the transition probabilities relate to anticipated performance of the structure in seismic events.

In a study that developed a seismic retrofit program for Los Angeles bridges, Kuprenas et al. (1998) used site evaluation and seismic analysis to select bridges to be included in the program. The following groups of bridges were eliminated:

- Bridges with reinforced concrete box structures that were restrained by the surrounding soil. These bridges were not considered vulnerable to major damage or collapse during earthquakes.
- Bridges supported by reinforced concrete pier walls and abutments. These bridges were considered structurally sound.

The bridges selected for the program were prioritized based on a seismic risk value scoring equation:

$$R_S = 0.5(F_C) + 0.2(F_O) + 0.15(F_T) + 0.15(F_A) \qquad (A\text{-}3)$$

where

R_S = Seismic risk (the higher the value, the higher the priority of the project),
F_C = replacement cost,
F_O = overall rating,
F_T = equivalent traffic, and
F_A = year built.

The authors also discussed several design problems and deficiencies found in these bridges. Stein et al. (1999) presented a method for assessing the risk associated with scour threat to bridge foundations. The authors mentioned that foundation information is important in evaluating scour vulnerability,

which was not available. The risk of scour failure for 1 year was calculated as follows:

$$\text{Risk} = K * P[\text{Rebuild Cost}] + [\text{Running Cost}]$$
$$+ [\text{Time Cost}] \tag{A-4}$$

where

K = risk adjustment factor based on foundation type and span and

P = probability of failure for 1 year.

The method was based on the data contained in the National Bridge Inventory. The authors pointed out that this method is most useful as a relative measure for prioritization and not as an absolute measure.

In analyzing strategies for integrating seismic risk into BMSs, Small (1999) presented two approaches. The first approach was based on prioritization procedures developed by FHWA and state DOTs and employed a value-mapping approach to convert priority indices to economic measures. Since all the information was not available in the inventory and inspection databases, additional information requirements were documented. The measure-value approach can be expressed in a generalized manner, which is shown for assessment of costs as follows:

$$C = I\gamma\phi \tag{A-5}$$

where

I = priority index,

γ = function to map the index to a normalized scale, and

ϕ = value function.

The costs for seismic retrofitting were determined using a categorical normalization process as follows:

$$\text{Category} = (V)(\gamma) \tag{A-6}$$

$$\phi = f(\text{Category}) = \begin{array}{ll} 0 - negligible & (\text{where } V = 0) \\ 1/3 - (UC) - low & (\text{where } 0 < V \leq 3) \\ 2/3 - (UC) - moderate & (\text{where } 3 < V \leq 7) \\ UC - high & (\text{where } V > 7) \end{array}$$
$$\tag{A-7}$$

where

V = vulnerability (maximum of the superstructure and substructure vulnerability estimates from Oregon DOT) and

UC = seismic vulnerability retrofit unit cost.

This approach resulted in wide variances; therefore, an alternative risk-based procedure was presented in the form of a fragility curve. A fragility curve plots the probability that a damage state will exceed a minimum value, against the input ground motion. This approach requires that the fragility relationships could be defined and that the cost of retrofitting bridge classes could be estimated. The benefit-cost ratio could be computed where benefit was defined as the reduction in risk given by the following equation:

$$\Delta\text{Risk} = C_i \left[p_a(d_i|a_g) - p_b(d_i|a_g) \right] \tag{A-8}$$

where

C_i = Consequences of damage i,

p_a, p_b = probability of damage state i for configurations A (preretrofit) and B (postretrofit),

d_i = damage state i, and

a_g = spectral acceleration.

Given the detail of retrofit considerations, a component-level generation of fragility curves was considered more appropriate.

A probability-based method for evaluating the safety level of bridges was presented to help select bridges in need of strengthening against earthquakes (Monti and Nistico 2002). The method was based on probabilistic description of the bridge exceeding predefined performance levels (light damage, heavy damage, and near collapse) through the use of a damage function. A damage function represents the actual damage level for various values of peak ground acceleration (PGA). Damage level is considered to follow a probability distribution for each level of PGA. *Fragility curves* were determined by computing the probability of exceeding each performance level as a function of PGA. These fragility curves were then compared with a predefined target fragility curve to assess the safety level of bridges.

In a study that investigated the costs of vulnerability, Hawk (2003) suggested a stochastic approach to bridge life-cycle cost analysis to account for vulnerability costs. Vulnerability cost, VC, for hazard H was represented as expected value of annual extraordinary costs anticipated under a particular bridge management strategy, including both agency and user costs. It was computed as follows:

$\text{VC}(H) = E(c|H)$, the expected cost given that the hazard H has some impact

$$= \sum_i [c(h_i) \times p(h_i)] \tag{A-9}$$

where

$c(h_i)$ = estimated cost associated with a hazard event h_i of intensity i,

$p(h_i)$ = probability of hazard event h_i occurring in any single analysis period, typically expressed on an annual basis

i = the set of estimated intensities $\{i\}$ for hazard H.

Salem et al. (2003) carried out risk-based life-cycle costing and presented a new approach for estimating life-cycle costs and evaluating infrastructure rehabilitation and construction alternatives, taking into consideration the uncertainty involved in determining the service life. The approach was based on probability theory and simulation application. The uncertainty was introduced through the parameters of probability distributions fitted to pavement time-to-failure data. These parameters were input to the model using random sampling of variables. Monte Carlo simulation was then used to generate the output probability distributions of the associated life-cycle costs of different alternatives. These probability distributions can provide valuable information to decision makers regarding the probability of executing an alternative at or below a certain life-cycle cost.

In a study that addressed reliability-based structural design with a Markovian decision process (MDP) (Tao et al. 1995), two objectives were identified: minimizing maintenance costs and maintaining acceptable structural reliability. The authors sought to develop a synthesis of an optimal structural design and a maintenance/management policy over the design lifetime and accomplished their goal by integrating a MDP model and structural reliability theory. An MDP generates a long-term maintenance policy based on minimum expected lifetime cost with respect to the initial design. According to the authors, incorporation of a reliability model with MDP affords decision makers an opportunity to make future maintenance policies that result in the minimum discounted expected future cost of a bridge while maintaining acceptable reliability.

Frangopol et al. (2000) investigated the optimization of network-level bridge maintenance planning on the basis of minimum expected cost. Bridge management at the network level is concerned with ensuring an adequate level of safety at the lowest possible life-cycle cost. However, the combining of life-cycle cost analysis and bridge reliability analysis was limited to individual bridges. The researchers offered a framework for optimal network-level bridge maintenance planning that minimizes the expected maintenance cost of a bridge stock and maintains the lifetime reliability of each bridge above an acceptable target level. The framework supports the optimal allocation of resources to manage a stock of gradually deteriorating bridges. The framework uses methods that balance life-cycle maintenance cost and lifetime reliability. The approach is illustrated for a set of realistic bridges. Ages of the bridges vary and have time-dependent reliabilities. Simultaneously accounting for life-cycle costs and reliability over the lifetime of bridges has important practical implications in the development of the optimal network-level management strategy.

In a study that analyzed optimal life-cycle costing with partial observability, Jiang et al. (2000) determined that costing structures in an optimal manner requires taking into account lifetime management policies concerning inspection and maintenance costs in addition to construction cost. Recent literature on lifetime costing reveals that MDP models are being used to incorporate management policies into structural reliability analysis. The authors presented an optimization model that uses a partially observable MDP. The model reflects the uncertainty inherent in different inspection techniques. Environmental degradation from fatigue and corrosion influence the costs and uncertainties of these inspection procedures. The modeling procedure implies a management policy regarding the frequency and type of inspection and extent of repair. The authors illustrated their methodology using a steel girder highway bridge.

Zimmermann (1987) surveyed different approaches to using fuzzy sets in decision making, including multi-criteria decision making and capital budgeting under a form of uncertainty. The general and most simple version of the problem is to simultaneously satisfy both the objective function and constraints, where both are membership functions expressed as fuzzy sets. In the case of multi-attribute decision making, the author discusses how to express "fuzzy utilities" under uncertainty. Other topics addressed in the book include individual decision making in fuzzy environments, multi-person decision making in fuzzy environments, fuzzy mathematical programming (including fuzzy multi-staged programming), multi-criteria decision making in ill-structured situations, and decision support systems (including an interactive decision support system for fuzzy and semi-fuzzy multi-objective problems).

Cheng and Ko (2003) developed the Object-Oriented Evolutionary Fuzzy Neural Inference System to solve manifold, complex, and uncertain construction management problems. In particular, they combined fuzzy logic, genetic algorithms, and neural networks to simultaneously search for the fittest membership functions having the minimum fuzzy neural network structure and optimal parameters. They conducted a series of simulations to demonstrate potential application of their procedure and concluded that this system could solve a wide variety of construction management problems.

Crum and Derkinderen (1981) edited a publication that addresses the problem of capital budgeting under conditions of uncertainty. Articles are grouped into three sections: government intervention in the investment process, investment issues in complex environments, and capital allocation modeling. Articles in the last two sections are most relevant to multi-objective optimization for bridge management. These articles include capital rationing methods, multi-criteria approaches to decision making, and interactive multi-goal programming as an aid for capital budgeting and financial planning with multiple goals. Article topics that shed light on dealing with uncertainty or nonquantitative data include impact of stochastic project lives on the capital budgeting decisions and capital budgeting under qualitative data information.

A.2.5 Bridge Decision Making Involving Multiple Objectives

The past few decades have seen an increase in the amount of literature on multi-objective (or multi-criteria) decision making and how to combine different performance measures or objectives. The multi-objective methods can be differentiated based on the decision maker's preferences. If the decision maker does not know the preferences and just knows that "more is better," then vector optimization is used to determine *pareto optimal* solutions. But if the decision maker is aware of the preferences for objectives and if the existence of a value/utility function is assumed, then multiple objectives can be combined into a single objective. Because there are so many different methods, we have found it necessary to make some fine distinctions that differentiate the types of methods and provide a mathematical organizing framework. The main categories are weighting (a measure of the relative importance of different objectives), scaling (a means of finding a common system of measurement to use in combining objectives that are normally measured using incompatible units), and amalgamation (a means of combining measurements that might not reflect a linear best-to-worst scale).

A.2.5.1 Weighting Methods for Multiple Objectives

The relative weights among bridge performance objectives play a very influential role on the selection of treatments over the life cycle of a given bridge or in the selection of candidate bridges for a given network. Therefore, it is important to pay close attention to the investigation and choice of the most appropriate weighting schemes for the multi-objective optimization. A selection of viable weighting methods will be made available in the software so that the decision maker may choose his or her preferred method and see how the various methods differ. Among the possibilities are the following:

- **Equal weighting** (i.e., assigning the same weights to each objective) is simple and straightforward, as well as easy to implement, but does not capture the preference among different attributes.
- **Observer-derived weighting** (Hobbs and Meier 2000) estimates the relative weights of multiple goals by analyzing unaided subjective evaluations of alternatives using regression analysis. For each alternative, the decision maker is asked to assign scores of benefits under individual goals and a total score on a scale of 0 to 100. A functional relationship is then established using the total score as a response variable and the scores assigned under individual goals as explanatory variables through regression analysis. The calibrated coefficients of the model thus become the relative weights of the multiple goals. Psychologists and pollsters have shown preference for the observer-derived weighting method because it yields the weights that best predict unaided opinions.
- **Direct weighting** (Dodgson et al. 2001) asks the decision maker to specify numerical values directly for individual goals between 1 and 10 on an interval scale.
- **The AHP** allows considering objective and subjective factors in assigning weights to multiple goals (Saaty 1977). This method is based on three principles: decomposition, comparative judgments, and synthesis of priorities. The relative weights of individual decision makers that reflect their importance are first established, and then relative weights of individual decision makers for the multiple goals are assessed. The local priorities of the goals with respect to each decision maker are finally synthesized to arrive at global priorities of the goals. One criticism of this technique is the rank reversal of goals when an extra goal is introduced.
- **The gamble method** chooses a weight for one goal at a time by asking the decision maker to compare a "sure thing" and a "gamble." The first step is to determine which goal is most important to move from its worst to best possible level. Then, consider two situations: First, the most important goal is set at its best level, and other goals are at their least desirable levels. Second, the chance of all goals at their most desirable levels is set to p, and the chance of $(1 - p)$ is set for all goals at their worst values. If the two situations are equally desirable, the weight for the most important goal will be precisely p. The same approach is repeated to derive the weights for remaining goals with decreasing relative importance. The hypothetical probabilities for all goals in their best or worst cases are likely to vary for different assessors. This method has clear applicability to vulnerability and risk.

A.2.5.2 Scaling Multiple Criteria

Multi-objective optimization for bridge management will involve multiple noncommensurable goals that have different units; therefore, such optimization requires the decision maker to scale attributes. Value scaling can be viewed as a value function that translates a social, economic, or environmental attribute into an indicator of worth or desirability. A value function usually describes a decision maker's preferences regarding different levels of an attribute under certainty. The most preferred outcome is assigned a value of 1, and the worst a value of 0. A utility function—a more specific type of the value function—reflects both the decision maker's innate value of different levels of the attributes and the decision maker's attitudes toward risk, including risk prone, risk neutral, and risk averse.

A utility function for an established bridge performance criterion or attribute can be applied in three steps: create a single-attribute utility function for an attribute, characterize the probability distribution of the attribute for each alternative, and calculate the expected utility of the attribute for each alter-

native. The alternative with the highest expected utility value is most preferred by the decision maker.

The utility function for an attribute can be developed by following five steps: preparing for assessment, identifying relevant qualitative characteristics, specifying quantitative restrictions, choosing a utility function, and checking for consistency (Keeney and Raiffa 1993). At an early stage of the assessment, it is necessary for the bridge decision maker to determine whether the utility function is monotonic and whether the utility function is risk prone, risk neutral, or risk averse. After identifying the relative shape of the utility function, quantitative utility values corresponding to some attribute values, conventionally on a five-point scale, may need to be assessed. We propose to do this by first choosing attribute values at the five successive levels and then finding the corresponding utility values. The bridge attribute utility functions may also be calibrated by fixing utility values at 0, 0.25, 0.50, 0.75, and 1.00, then determining the attribute values associated with such levels. Before finishing the whole evaluation process, some simple consistency checks need to be conducted. For instance, the decision maker's certainty equivalent can be assessed using three consecutive points in a five-point assessment procedure.

Another approach to scaling that uses willingness to pay instead of calculating a utility function was the Algorithm for Selection of Optimal Policy (ASOP) (Kulkarni et al. 1980). It was originally designed to determine optimal levels of service for different types of attributes related to highway maintenance. However, under FHWA Demonstration Project 71, an investigation was performed to determine its applicability to bridges (Nash and Johnston 1985). Ultimately, North Carolina DOT adopted this procedure as a part of its BMS.

A.2.5.3 Amalgamation Methods for Multiple Criteria

Amalgamation is a process applied to yield a single-value index for an alternative bridge management action that involves multiple goals, thereby allowing several such alternatives to be ranked. This process has critical applications in multi-objective optimization for bridge management. Amalgamation methods can be categorized as no-preference, prior, posterior, interactive, and evolutionary methods.

For the present study, classical methods with no preference that were considered include the weighted sum method, the ϵ-constraint method, and the weighted Tchebycheff method. Methods can be either continuous or discrete with prior articulation of preferences. Methods mainly include goal programming, multi-attribute utility function, surrogate worth trade-off, and outranking. Posterior articulation of preferences can be done by data envelopment analysis. Interactive methods, which are used for situations where minimal a priori knowledge is available, are often characterized by progressive articulation of preferences. Popular interactive methods include the step method and compromise programming. Evolutionary algorithms mimic natural evolutionary patterns for ultimate optimization. These algorithms are well suited for situations characterized by nonlinearities and complex interactions among problem variables. The most commonly used evolutionary algorithms are genetic algorithms. The following paragraphs briefly discuss each of these methods.

The weighted sum method is the most widely used procedure that scales multiple objectives into a single objective by multiplying a weight with each objective. Setting relative weights for individual objectives is a central issue in applying the method and depends largely on the magnitude of each objective function. It is desirable to normalize them so that each has more or less the same scale of magnitude.

The ϵ-constraint method (Goicoechea et al. 1982) allows the user to arbitrarily select an objective function for maximization while specifying bounds on the remaining objectives, thereby alleviating the difficulties faced by the weighted sum method in solving problems having nonconvex solution space. Motivation for specifying bounds on the objective functions is often provided by the formulation requirements of the problem. Because the solution technique is used to solve for one objective function at a time, this method leads to an intermediate nondominated solution, and the global nondominated solution can only be obtained mathematically under some specific conditions.

Instead of using a simple weighted sum of the multiple objectives, **the weighted Tchebycheff method** (Steuer 1989) uses distance metrics for the amalgamation process, thereby providing theoretical grounds for goal programming. **Goal programming** is an approach to solve multi-criteria optimization problems when the relationship between multiple conflicting goals and decision variables can be expressed mathematically. This method requires the decision maker to provide relative weights and target levels of the conflicting goals. Alternatives are then ranked according to the weighted deviation from the goal: the smaller the deviation, the more preferred the alternative. The idea is to choose an alternative closest to the goals by minimizing a distance measure.

Another approach to amalgamation is **compromise programming,** a variation of **goal programming** (Zeleny 1973), which identifies solutions closest to the ideal solution as determined by some measure of distance. The solutions identified are called compromise solutions and constitute the compromise set. If the compromise set is small enough to allow the decision maker to choose a satisfactory solution, then the process is terminated. Otherwise, the ideal solution is redefined and the whole process is repeated.

Multi-attribute utility functions were also investigated for this study. Such functions capture a decision maker's preferences regarding levels of attributes and the attitude toward risk

for several attributes simultaneously (Keeney and Raiffa 1993). This is done by weighting and synthesizing single-attribute utility functions into a multi-attribute utility function in either additive or multiplicative form. The expected values of the multi-attribute utility function are then used to rank the alternatives, and the alternative with maximum expected utility value is then picked. Two assumptions are made for the multi-attribute utility functions: utility independence and preference independence. Utility independence means that each attribute's utility function does not depend on the levels of other attributes. Preference independence holds that the trade-off that one is willing to make between two attributes does not depend on the levels of other attributes.

Outranking methods, a class of multi-criteria decision making techniques that provide an ordinal ranking (sometimes only a partial ordering) of the alternatives, are exemplified by the Elimination and Choice Translating Algorithm (ELECTRE) method (Benayoun et al. 1966; Roy and Bertier 1971). ELECTRE establishes a set of outranking relationships among alternatives. One alternative is found to outrank another only if (i) the sum of normalized weights where the first alternative is better (i.e., the concordance index) exceeds a predetermined threshold value and (ii) the number of attributes where the second alternative is better by an amount greater than a tolerable threshold value (i.e., the discordance index) is zero. An extension of the ELECTRE method by incorporating uncertainty was discussed by Mahmassani (1981).

The step method (Benayoun and Tergny 1969) is the first interactive method introduced to solve for linear and nonlinear problems. The method assumes that the best compromising solution has the minimum combined deviation from the ideal point and that the decision maker has a pessimistic view of the worst component of all individual deviations from the ideal point. The technique essentially consists of two steps: (i) a nondominated solution in the minimax sense to the ideal point for each objective function is sought, and a pay-off table is constructed to obtain the ideal criterion vector, and then (ii) the decision maker compares the solution vector with the ideal vector of a pay-off table by modifying the constraint set and the relative weights of objective functions. The process terminates when the decision maker is satisfied with the current solution.

A.2.5.4 Some Studies in General Multi-Objective Optimization

Cohon (2003) provides a fairly comprehensive treatment of multi-objective programming and planning. The general multi-objective programming problem consists of a set of separate objective functions subject to a set of constraints. The objective functions are not added, multiplied, or combined in any way. In contrast to a single objective optimization problem, there is no single unique solution when there are multiple objectives. If a solution is optimal for one objective, it will generally be suboptimal for the remaining objectives. Besides providing a general formulation for the multi-objective optimization problem, this book discusses different multi-objective programming methods. Weighting the objectives to obtain a noninferior solution is the oldest solution technique. The constraint method involves optimizing one objective while constraining all of the others to specific values. The noninferior set estimation method finds extreme points and evaluates the properties of the line segments between them. Simplex methods for multi-objective optimization have been developed that do not convert the problem to one with a single objective. There is discussion on how to maximize a multi-attribute utility function subject to a set of constraints. Multi-attribute utility functions capture trade-offs in terms of the marginal rate of substitution, whereas multi-objective optimization reveals how much of one objective is given up in order to obtain a gain in another objective. Goal programming is also addressed, but the author contends it is not multi-objective optimization as defined above. Instead, goal programming uses a minimum distance criterion for what is best.

A.2.6 Knapsack Problems: Algorithms and Heuristics

The knapsack problem is a famous integer programming problem, and the large domain of its applications has greatly contributed to its fame. The knapsack problem is *NP-hard*. The multi-dimensional 0-1 knapsack problem (MDKP) is a special case of general 0-1 linear programs. Historically, one of the first examples was exhibited by Lorie and Savage (1955) as a capital budgeting problem. The multi-choice multi-dimensional knapsack problem (MCMDKP) can be stated as follows:

$$\max z = \sum_{k=1}^{n} \sum_{j \in L_k} r_{jk} x_{jk}$$

$$s.t. \quad \sum_{k=1}^{n} \sum_{j \in L_k} a_{ijk} x_{jk} \le b_i, \quad i = 1, \ldots, m$$

$$\sum_{j \in L_k} x_{jk} = 1, \quad k = 1, 2, \ldots, n$$

$$x_{jk} \in \{0, 1\} \quad k = 1, 2, \ldots, n \quad j \in L_k \qquad (A\text{-}10)$$

where

  n = number of classes,
  L_k = set of items for class k, and
  m = number of knapsack constraints (i.e., size constraints) with capacities b_i.

Each item $j \in L_k$ is associated with r_{jk} units of profit and a_{ijk} units of weight. The goal is to choose one item from each class such that the profit is maximized without exceeding the capacities of the knapsack. The second set of constraints is referred

to as the *choice constraints* or *group constraints.* If the number of size constraints is one and there is only one item in each class, then the problem reduces to a simple 0-1 knapsack problem.

Another variation of the knapsack problem is the multi-choice knapsack problem (MCKP). An MCMDKP is reduced to an MCKP when there is only one size constraint. Sinha and Zoltners (1979) presented the problem as a capital budgeting problem where investment opportunities fall into disjointed subsets. Exactly one project is to be selected from each subset of projects to maximize the profit gained. Since then, knapsack problems have been modeled in wide applications, including cutting stock problems, investment policy for the tourism sector, allocation of databases and processors in a distributed data processing, delivery of groceries in multi-compartment vehicles, multi-commodity network optimization, and daily management of a remote sensing satellite (Freville 2004).

Algorithms for these problems can be classified into two groups: exact algorithms and heuristics. The exact algorithms solve the problem to optimality. Heuristics strive to achieve near-optimal solutions quickly.

A.2.6.1 Exact Algorithms for Solving Knapsack Problems

The exact algorithms in the literature are based on a variety of solution methods, including dynamic programming, branch-and-bound approach, special enumeration techniques, and reduction schemes. Marsten and Morin (1977, 1978) combined dynamic programming and branch-and-bound approaches for solving the MDKP. Morin and Marsten (1976) demonstrated the use of branch-and-bound methods to reduce computational requirements in discrete dynamic programs. Relaxations and fathoming criteria were used for identification and elimination of irrelevant states, whose corresponding subpolicies could not lead to optimal policies, during the dynamic programming computation.

Morin and Esogbue (1974) presented a method to reduce dimensionality in finite dynamic programs. With the use of some mathematical properties of the functional equation of the dynamic programming, the M-dimensional state space was reduced to a one-dimensional search over an imbedded state space. Based on this concept, an algorithm was developed for nonlinear knapsack problems that recursively generates the complete family of undominated feasible solutions (Morin and Marsten 1976). Nauss (1978) presented two branch-and-bound algorithms for the MCKP based on two different relaxations for the bounding mechanism. The first algorithm was based on relaxing the generalized upper-bound constraints (multi-choice constraints) and placing them in the objective function, which reduces the problem to a 0-1 knapsack problem. The second algorithm used the linear relaxation of the

MCKP as the primary relaxation. The second algorithm was shown to perform better on a set of test problems. Sinha and Zoltners (1979) presented a branch-and-bound algorithm for the MCKP problem that featured quick solution of linear programming relaxation and its efficient, subsequent reoptimization as a result of branching. This algorithm performed well on the basis of a large set of test problems. Shih (1979) designed a linear programming–based, branch-and-bound method for MDKPs. The estimation of an upper bound and the branching rule at any node are based on the information provided by the solutions of the linear programming relaxations associated with each of the m single-constraint knapsack problems.

Balas and Zemel (1980) presented an algorithm for large 0-1 knapsack problems. It is based on three ideas. The first is to focus on what is called as the core problem—that is, the subproblem, a knapsack problem equivalent, defined on a particular subset of variables. The size of this core is usually a small fraction of the full problem size and does not seem to increase with the latter. While to precisely identify the core would require solving the knapsack problem, a satisfactory approximation can be found by solving the linear relaxation of the knapsack problem (LKP). The second idea is a binary search–type method for solving the LKP without sorting the variables. The computational complexity of this procedure is $O(n)$. This procedure also yields a convenient approximation to the core problem. The third idea is to use a simple heuristic that finds a 0-1 solution with a probability that increases exponentially with the size of the problem. If such a solution is found, the probability that it is optimal increases with n.

Solving the MCKP requires solving a subproblem that is a linear relaxation of the integer MCKP (Nauss 1978; Sinha and Zoltners 1979). Various $O(n)$ algorithms were developed for solving the linear MCKP by Zemel (1980, 1984) and Dyer (1984). Zemel (1987) also developed a linear time-randomizing algorithm for searching ranked functions that was generalized for the parametric knapsack problem. Gavish and Pirkul (1985) discussed various relaxations of the MDKP and proved theoretical relations between these relaxations. The Lagrangian, surrogate, and composite relaxations are used to reduce the MDKP to a single-constraint knapsack problem. Algorithm for computing surrogate multipliers, rules for reducing problem size, and an efficient branch-and-bound procedure were developed. Dyer et al. (1995) presented a hybrid dynamic programming and branch-and-bound algorithm for the MCKP. Lagrangian duality is used to compute tight bounds on every active node in the search tree. A reduction procedure is also used to reduce the problem size for the enumeration phase.

Martello and Toth (1997) proposed upper bounds for hard 0-1 knapsack problems by adding valid inequalities on the cardinality of an optimal solution and then relaxing it in a Lagrangian fashion. A branch-and-bound algorithm was

developed that incorporated a polynomial time-iterative technique to determine the optimal Lagrangian multipliers for the linear relaxation of the problem. This approach was combined with the concepts of surrogate relaxation and core problem to develop an efficient algorithm for a 0-1 knapsack problem (Martello et al. 1999). The core was enumerated through dynamic programming. Martello et al. (2000) provide an overview of techniques for solving hard knapsack problems, with special emphasis on the addition of cardinality constraints and dynamic programming.

The minimal algorithm (Pisinger 1995) is based on a partitioning algorithm and a dynamic programming algorithm to solve an MCKP. An $O(n)$ partitioning algorithm is used to derive the optimal solution to the linear relaxation of the problem. Then it is incorporated into the dynamic programming algorithm such that a minimal number of classes are enumerated, sorted, and reduced. An exact algorithm for large MDKP problems was presented by Pisinger (1999) based on the branch-and-bound method with a surrogate relaxation and dynamic programming algorithm for subset problems. Some difficult issues relating to the core problems were discussed by Pisinger (1999). A branch-and-bound algorithm was presented (Kozanidis and Melachrinoudis 2004) for a 0-1 mixed integer knapsack problem with linear multi-choice constraints. The algorithm solves at each node of the tree a linear relaxation using an adaptation of an algorithm for the linear MCKP.

A.2.6.2 Heuristics for Solving Knapsack Problems

The early approaches were based on "greedy" algorithms, which are fast and generally simple to implement. They basically employed the use of ratios of the profits to the weight coefficients to solve the single-constraint knapsack problem. Senju and Toyoda (1968) developed a *dual* heuristic for the MDKP. The heuristic started with the all-ones solution and set the variables to 0 one at a time according to increasing ratios until feasibility requirements were satisfied. Approximate solutions were reported for two large test problems. However, the usefulness of the heuristic was limited because the optimal solutions to these problems were unknown and were thought to be only achievable in an astronomical amount of time. Marsten and Morin (1977) found the optimal solutions and showed that the heuristic was very effective. Toyoda (1975) developed a *primal* method that started from the origin and set variables to 1 according to decreasing ratios until no more variables could be added without violating the constraints.

Dual multipliers have been employed to design more competitive greedy algorithms. Magazine and Oguz (1984) combined Senju and Toyoda's dual algorithm with a Lagrangian relaxation approach. The heuristic also provided upper bounds with approximate solutions at no additional cost. This heuristic was improved by Volgenant and Zoon (1990) by calculating the Lagrangian multipliers simultaneously and sharpening the upper bounds. The work by Magazine and Oguz was extended by Moser et al. (1997) by generalizing the heuristic for an MCMDKP. Pirkul (1987) developed a greedy heuristic based on surrogate duality to solve the MDKP. This is based on a descent procedure to determine the surrogate constraints. A linear relaxation of the surrogate problem is considered to help computational efficiency. Lee and Guignard (1988) developed a heuristic for MDKP that is controlled by three parameters that affect the trade-off between solution quality and computation time. It first uses a modified Toyoda method, then reduces the problem size by fixing a certain set of variables using the linear programming relaxation. Finally, it improves the solution by complementing certain sets of variables.

Pivot and complement is a heuristic for finding approximate solutions to large arbitrary 0-1 programming problems (Balas and Martin 1980). The heuristic has been reported to perform remarkably well, from the viewpoint of both the computational effort required and the quality of solutions obtained. The computational effort is bounded by a polynomial in the number of constraints and variables. The procedure starts with solving the linear relaxation and then performs a sequence of pivots aimed at putting all slacks into the basis at a minimal cost. Finally, the procedure improves the solution by using local search techniques and by complementing a certain set of variables.

Gens and Levner (1998) proposed a fast approximation heuristic for the MCKP that is guaranteed to be ⅘-bounded. The approach is based on a binary search. It was mentioned that the method is complementary and can be used to improve other fully polynomial approximation algorithms for knapsack problems. Zhang and Ong (2004) proposed a linear programming–based heuristic for solving bi-objective 0-1 knapsack problems. The method generated good approximations to the nondominated set efficiently.

Like other combinatorial optimization problems, knapsack problems have been investigated using metaheuristics. Chu and Beasley (1998) presented a heuristic based on genetic algorithms for the MDKP. The genetic algorithm is restricted to search only the feasible region of the solution space by using a heuristic operator to convert an infeasible solution to a feasible one. This operator is based on a greedy-like heuristic that uses the profit-to-weight ratios. To convert the MDKP to a single-constraint knapsack problem, the surrogate relaxation of the problem is considered. The surrogate duality approach of Pirkul (1987) is then used to determine the surrogate multipliers by solving the linear relaxation of the original MDKP. The heuristic was shown to provide good solutions with a modest computational effort.

Hanafi and Freville (1998) developed a heuristic based on a tabu search for the MDKP. Strategic oscillation and surrogate constraint information is used to balance the intensification

and diversification strategies. Vasquez and Vimont (2005) used a geometric constraint and cutting planes combined with a tabu search method to solve the MDKP.

A.2.7 Performance Measures for Bridge Decision Making

Performance is defined as the execution of required function. Performance indicators are quantitative or qualitative measures that directly or indirectly reflect the degree to which results meet expectations or goals (Poister 1997). Externally, the need for meaningful performance indicators in government has been underscored by resolutions taken by professional organizations, such as the Governmental Accounting Standards Board (GASB) (1999), the National Academy of Public Administration (NAPA) (1991), and the American Society for Public Administration (ASPA) (1992). The U.S. Congress also passed two pieces of legislation, Public Law 101-576 and Public Law 103-62, to build performance measurement into federal management processes. Internally, strategic management or total quality management processes within a governmental agency are impossible without the development and use of performance indicators to track progress in achieving strategic goals or to evaluate the success of continuous process improvement activities.

In the context of bridge management decision making, the establishment of performance measures is critical because it is sought to optimize the level of bridge interventions that would yield optimal values of such performance measures. Performance measures are needed in both bridge-level and network-level optimization. Some desirable properties for

performance measures of each objective are the following (Keeney and Raiffa 1993):

- **Completeness:** A set of performance measures is complete if it is adequate in indicating the degree to which the overall set of goals is met.
- **Operational:** Since the idea of decision analysis is to help the decision maker choose the best course of action, the performance measures must be useful and meaningful to understand the implications of the alternatives and to make the problem more tractable.
- **Non-redundancy:** Performance measures should be defined to avoid double-counting of consequences.
- **Minimal:** It is desirable to keep the set as small as possible to reduce dimensionality.

A.2.7.1 General Goals of Any Management System for Highway Assets

Goals are related to highway system performance in that they reflect different perceptions of what the highway system should achieve. Understanding different goals is critical to identifying different types of highway performance indicators that need to be included in the management process. Table A.1 summarizes an example set of goals and objectives identified by Cambridge Systematics (2000) and found to provide a solid and broad basis for a highway asset management process.

A.2.7.2 Performance Indicators Under System Goals

The purpose of establishing performance indicators is to enable transportation agencies to assess the degree to which

Table A.1. Example goals and objectives by category (from Cambridge Systematics 2000).

Goal Category	Goal	Objective
System Preservation	Preserve highway infrastructure cost-effectively to protect the public investment	Improve construction techniques and materials to minimize construction delays and improve service lives of highway assets
Operational Efficiency	Develop strategies that improve the transfer of people and goods by reducing delays and minimizing discomforts	Use economies of scale by providing for joint use of intermodal facilities
Accessibility	Ensure reasonable accessibility for all residences	Maintain access for population to reach specified services
Mobility	Ensure basic mobility for all residences by providing safe, efficient, and economical access employment, educational opportunities, and essential services	Make public transportation travel time competitive with automobiles
Economic Development	Address anticipated demand from increases in trade	Improve access to passenger and freight facilities to serve trade
Quality of Life	Ensure that highway investments are cost-effective, protect the environment, and promote energy efficiency	Provide opportunity for safe, enjoyable, and low-environmental-impact recreation
Safety	Maintain high standards of safety in the transportation system	Reduce motor vehicle–related fatalities, injuries, and property damages
Resource and Environment	Develop projects that are environmentally acceptable	Improve air quality through transportation measures

the selected investment program has been successful in terms of improved system benefits. Setting clear performance indicators and using the results of this evaluation to inform future investment choices and management decisions are essential to ensure that an agency's investment is producing intended outcomes. Table A.2 summarizes highway performance indicators currently used by state transportation agencies (Poister 1997).

It is seen that state transportation agencies tend to maintain various performance indicators for several goals, including system preservation, agency cost, operational efficiency, mobility, safety, and environment. In the current study, we will examine the details of performance indicators identified, refine the content, and consider data collection efforts needed to establish a final set of performance indicators under the study framework.

Table A.2. Some highway performance indicators used by state transportation agencies (from Poister 1997).

Category		Performance Indicator	State
System Preservation	Pavement	Percentage of highway miles built to target design	OR
		Average roughness or overall pavement index value for state highway, by functional class	CT, FL, IN, MN, NC, NY, PA, RI, VA
		Percentage of highways rated good to excellent	IN, MN, NY
		Percentage of roads with a score of 80 or higher on overall highway maintenance rating scale	FL, IN, MN, OR
		Percentage of total lane-miles rated fair or better	OR
		Miles of highway that need to be reconstructed	MN, NY, WA
	Bridge	Percentage of highway bridges rated good or better	IN
		Percentage of highway mainline bridges rated poor	IN, WA
		Number of bridges that need to be rebuilt	FL, IN, WA
Operational Efficiency	Construction, maintenance, and operation	Cost per lane-mile of highway constructed	AL, GA, FL
		Cost per unit of highway maintenance work completed; labor cost per unit completed	AZ, NC, MN, PA, WA
	Cost-effectiveness	Cost per percentage point increase in lane-miles rated fair or better on pavement condition	CA, VA
		Cost per accident avoided by safety projects	CA, VA
Accessibility	Automobile/ roadway	Percentage of population residing within 10 minutes or 5 miles of state-aided public roads	MN, OR
	Roadway	Percentage of bridges with weight restrictions	AZ
		Miles of bicycle-compatible highway rated as good or fair	IN
Mobility	Travel speed	Average speed versus peak-hour speed	MN
	Delay, congestion	Hours of delay	MN, NY
		Percentage of limited-access highways in urban areas not heavily congested during peak hours	IN, OR, NY
	Amount of travel	Vehicle-miles of travel (VMT) on state highways	PA
		Percentage of VMT on roads with high v/c ratios	AZ, NJ, PA
		Percentage of passenger-miles of travel (PMT) in private vehicles and public transit buses on roads with high v/c ratios	NJ
Economic Development	Support of economy by transportation	Percentage of wholesale and retail sales occurring in significant economic centers served by unrestricted market artery routes	MN
Quality of Life	Accessibility, mobility related	Percentage of motorists indicating they are satisfied with travel times for work and other trips	IN, MN, PA
Safety	Number of vehicle collisions	Vehicular accidents per million VMT	CA, IN, KS
		Fatalities or injury per 100 VMT	CA, IN, KS, OR
		Accidents involving injuries per 1,000 residents	KS
		Accidents involving pedestrians or bicyclists	IN
		Number of pedestrians killed on state highways	IN
	Roadway condition related	Percentage of change in miles in high-accident locations	PA
		Percentage of accident reduction due to highway construction or reconstruction projects	CA, OR, VA
		Reduction in highway accidents by safety improvement projects	IL
		Number of railroad crossing accidents	CA, IN
		Percentage of motorists satisfied with snow and ice removal, or roadside appearance	MN
	Construction related	Number of accidents in highway workzones	IN, NC
Resource and Environment	Fuel usage	Highway VMT per gallon of fuel	IN

A.2.7.3 Performance Measures—A Review of Past Practice

From the literature search carried out as part of the present study, it was seen that a wide variety of performance measures have been used in past bridge management practice and research. This included early legislative requirements that led to the development of performance measures, measures arising from the national studies funded by AASHTO, measures based on bridge condition, and measures based on user perceptions.

One of the earliest studies that highlighted the several performance measures of interest to the infrastructure manager was that of Juster and Pecknold (1976), who developed a multi-period investment programming procedure and software. They integrated a variety of performance measures for the Massachusetts Department of Public Works using a methodology that addressed regional equity, public acceptance, uncertainty, and legislative and funding constraints. The tool was designed for use in an iterative planning process appropriate to community decision makers, regional planning authorities, and state agencies. The tool uses substantial data inputs, illuminating the trade-offs between and among programs. These trade-offs are the basis for discussion, compromise, and final decision making.

A legislative impetus that engendered the need for developing infrastructure performance measures was the Government Performance and Results Act of 1993 (U.S. Public Law 103-62), which required each federal government agency to develop interrelated strategic and performance plans and to submit an annual performance report to the U.S. Congress. It was stipulated that the strategic plan should have at least a 5-year time horizon, be updated at least every 3 years, and contain the following: a mission statement, general goals and objectives (including those which are outcome related), a description of how the goals are to be achieved, and a list of factors outside the agency's influence that might affect the ability to achieve the strategic goals. The performance plan should establish annual performance goals; express objective, quantifiable, and measurable goals; establish performance indicators to be used in measuring or assessing outputs, service levels, and outcomes of each program activity; and provide a basis for comparing actual program results with the performance goals. It was further stipulated that each agency should submit an annual report to Congress regarding progress toward accomplishing the goals.

An FHWA report titled *Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges* (FHWA 1995) describes several performance indicators routinely used in bridge management, including condition ratings of bridge components (deck, superstructure, substructure, waterway, and culverts), a rating of deck geometry, and a sufficiency rating. FHWA maintains procedures and software for computing the component condition ratings from CoRe element inspection data (commonly known as the NBI Translator) and for combining several measures into the standard sufficiency rating.

The Governmental Accounting Standards Board (GASB 1999), which ushered in a new era of operational accountability, implicitly provided a set of performance measures required of public infrastructure. GASB promulgated a set of rules for agencies that manage public infrastructure, and those rules were intended to reveal whether the infrastructure was being properly maintained. The rules require the organizations to use one of two types of performance measures to account for the extent to which public infrastructure has decreased in value. The first measure is net asset value—the economic value of the infrastructure after taking into account depreciation. The agency must report both depreciation and net asset value each year. The second set of measures is based on condition of assets as determined by an approved asset management system, such as a bridge or pavement management system. The government must report the condition levels at which it intends to maintain the assets and provide a comparison of the annual dollar amount estimated to be required to maintain the assets at that condition level with actual expenditures for the last 5 years.

In followup reports, GASB presented an overview of a major effort of GASB to promote the use of performance measures for all types of state and local government activity. The report includes a framework for performance measurement that distinguishes between outcomes, outputs, inputs, and factors outside the control of government organizations. The overview summarized the results of preliminary efforts to establish performance measures that would be useful for different government functions, including road maintenance.

While GASB dwelt on performance measures of a general nature, efforts have been made elsewhere to identify such measures more specifically. In their 2000 AASHTO study that identified "commonly recognized" bridge elements, Thompson and Shepard (2000) described issues regarding the development and application of element-level bridge condition measures that have become widely adopted and are the basis for federally approved bridge inspection procedures. These commonly recognized elements, abbreviated as CoRe elements, generally (but not always) take on condition ratings of 1 to 5, have corresponding descriptions of the condition states for each bridge element, and identify alternative actions for addressing each condition state. The general pattern of the rating scale is 1 = protected; 2 = exposed; 3 = attacked; 4 = damaged; 5 = failed. Bridge inspectors allocate the total quantity of each element among the condition states. Thus, this condition rating system is capable of identifying the severity and extent of deterioration or damage of each bridge element. The CoRe elements were originally developed for use in BMSs such as Pontis and Bridgit, but one of the important applications of these

condition measures is the collection and analysis of performance data. It is easy, for example, to make queries against the bridge inventory and condition database and determine what fraction of bridge decks, steel girders, bridge railings, and so on across the network are in different condition states and how their condition has changed over time. This information is valuable for maintenance planning and programming. This article also describes the use of a bridge health index (which assigns an overall score of the health of a bridge on a 0–100 scale for each element using the CoRe element condition assessments) as a performance measure in the state of California.

Sivakumar et al. (2003) proposed bridge performance measures based on detailed element-level condition data for prioritizing bridges. Four individual and two composite indexes were proposed based on a safety perspective, measuring bridge attributes that have a direct or impeding effect on safety. The individual indexes were based on condition, live load, geometric features, and special events.

For purposes of maintenance quality assurance, Smith et al. (1997) carried out a research for NCHRP that presents a framework for maintenance performance measurement applicable to all types of inventory on a highway network, including bridges. It has been adopted by many states and provides valuable feedback to managers responsible for maintaining different aspects of the highway inventory in different parts of the state. It is also useful to chief executive officers who seek a report card regarding the condition of highway assets. The maintenance quality assurance process is based on taking a scientific random sample of highway sections, applying a set of measurement protocols for each type of asset to determine each asset's condition, and normalizing the score on a scale of 0 to 100. Assets might include pavements, shoulders, guardrail, fences, and bridges. In the past each state has developed its own measurement procedures for assessing the condition of assets found along the road. Consequently, there is no uniformity or basis for making comparisons from state to state, although certainly states do make comparisons among districts or other organizational units within their own jurisdiction. Once measurements have been taken for all roadway sections in the sample, it is possible to determine the percentage of sections that have scores that exceed or are below a certain number. In practice, the number selected is usually 80, which represents good condition.

In a study for Florida DOT, Thompson et al. (1999b) examined user costs as a performance measure. They used hypothesis testing and regression analysis techniques to quantify relationships between various bridge characteristics and the frequency and severity of crashes. Significant relationships for prediction of accident frequency were found and developed into a prediction model. Earlier research results by the Federal Highway Administration and the Florida Trucking Association were used for economic unit costs for travel time, vehicle operating cost, and accident cost. Compared with the models included in Pontis, the new models gave substantially smaller user costs that nonetheless were large enough to continue to justify most functional improvements. The new models also behaved much more intuitively in sensitivity analyses of the explanatory variables.

Hawk (2003) prepared a definitive guide (*NCHRP Report 483*) to the state of the practice in life-cycle cost analysis as applied to bridges. It categorizes the types of agency and user costs typically included in bridge life-cycle cost analysis, describes data sources, and reviews the various bridge-related applications. It also includes detailed computational procedures and software, with an emphasis on dealing with uncertainty in the input parameters. A common application of bridge life-cycle cost analysis is to generate an economic performance measure that combines many of the important quantitative decision factors in asset management. Standard methodologies can combine costs incurred at different points in time, can represent the outcomes of events with different probabilities of occurrence, and can incorporate the economic aspects of projects affecting the public. For evaluation of projects and programs, it is common for life-cycle costs to be defined in terms of avoided costs, the future costs that are predicted to occur if a project is not implemented and that can be avoided if the project occurs. Most BMSs contain rigorous procedures for computing avoided costs.

In a similar study that addressed the issue of identifying suitable performance measures for the purposes of benchmarking maintenance activities, Hyman (2004) presented and discussed the essential performance measures for customer-driven benchmarking for maintenance activities. The report identified four classes of performance measures necessary for benchmarking: outcomes, outputs, inputs, and uncontrollable factors. According to the author, the outcomes are customer oriented and should consist of both the results of customer surveys and technical measures of outcomes important to the customer (e.g., whether a bridge is posted for load or not). Inputs consist of labor, equipment, and materials. Outputs are measures of production or accomplishment, such as acres mowed per day. Uncontrollable factors are factors beyond the influence of the agency, such as weather, terrain, population density, and traffic. The approach to benchmarking taken in this report is essentially economic and focuses on what outcomes are achieved given the resources expended while controlling for weather and sometimes the level of output. This report includes a discussion of data envelopment analysis for purposes of identifying best performers when benchmarking. In this context, data envelopment analysis is a nonparametric optimization procedure for identifying an efficient frontier based on outcomes and input resources of each organizational unit being compared. The efficient frontier can be established using linear programming. Usually there are more than one and sometimes a large num-

ber of organizational units found on the frontier when bench-marking with data envelopment analysis. Uncontrollable factors and outputs can also be addressed within data envelopment analysis, principally by transformation of the outcome and input variables, but also through separate analyses.

The information search on performance measure for bridge management optimization showed that while legislative requirements encouraged the search for suitable performance measures (and indeed provided some measures of a general nature), studies have been carried out to identify a broad spectrum of performance measures that could be used in bridge management optimization. Such measures cover bridge condition and health, risk and vulnerability, and impacts on the user and the community.

A.2.7.4 Some Performance Measures for BMSs

To identify a performance measure for a bridge, it is useful to think in terms of the questions "what attributes of the bridge could be enhanced by undertaking some activity?" and "by doing nothing for a bridge, what attributes could suffer?" The same questions can also be asked for an entire network of bridges. The various bridge management performance measures identified and/or discussed in various published literature can be categorized as follows:

- Bridge condition;
- Remaining service life;
- Economic returns;
- Reliability;
- Risk of damage or failure;
- Geometric or functional adequacy;
- Highway Bridge Rehabilitation and Replacement (HBRR) program eligibility;
- Life-cycle cost;
- Community impact;
- User cost of safety, time, and vehicle operation; and
- Traffic capacity.

Bridge Condition.   The importance of bridge condition as a measure of performance has been implicitly or explicitly emphasized in the work done by Jiang et al. (1988), Ghosn and Moses (1991), Hearn et al. (1995), and Thompson et al. (1999a). The information search on the state of practice showed that a sufficiency rating (federal or other), a structural condition indicator, or a health index rating were the most favored bridge program performance measures. Bridge condition can be expressed in several ways—condition rating (such that of the National Bridge Inventory [NBI]), sufficiency rating, or bridge health index.

The bridge health index, a normalized weighted average of element conditions (Shepard and Johnson 2001; Scherschligt

and Kulkarni 2003), is an overall measure of the health of a bridge. It employs a 0–100 ranking for each element, and the ranking is derived from the CoRe element condition ratings. Weights are assigned to each element based on economic importance, which might be related to long-term costs, or element failure consequences. For example, a bridge railing would get a lower weight than girders that support a bridge. The bridge health index is a normalized expression of the sum over all elements of the current element value divided by the sum over all elements of the total element value. In California, for example, the state highway agency examines the percentage of bridges that have different health index scores, identifying what percentage of the bridges in each district have health index scores below 80 for purposes of performance assessment and resource allocation.

Remaining Service Life.   The remaining service life of a bridge is the time taken (typically in years) for the overall condition of the bridge to reach some terminal value at which a major improvement such as rehabilitation or reconstruction would be necessary. Various elements of a bridge could have their own remaining service lives. The consideration of an element's remaining service life as a performance measure arises from the fact that the remaining service life of a bridge may be conceptually increased when some improvement is done on the bridge.

Remaining service life may be considered as a performance measure for an individual bridge or for a network of bridges (average remaining service life).

Life-Cycle Cost.   Life-cycle costing, a technique founded on the principles of economic analysis, helps in the evaluation of overall long-term economic efficiency between competing alternative investment options (AASHTO 1986). The Federal Highway Administration (FHWA) has always encouraged the use of life-cycle cost analysis in all major investment decisions where such analyses are likely to increase the efficiency and effectiveness of investment decisions. Also, congressional interest in life-cycle cost analysis is manifested in the requirement that the secretary of transportation develop recommended life-cycle cost analysis procedures for National Highway System projects (Walls and Smith 1998).

Previous studies conducted by Indiana and elsewhere strongly suggest that more effective long-term investment decisions could be made at lower cost if life-cycle cost analysis were adopted properly. The information search showed that most bridge agencies place high premium on cost-related performance measures such as life-cycle costs.

Economic Returns.   Economic returns may be described as the "bang" earned for each "buck" of bridge investment. In highway infrastructure management, measures of economic returns are typically expressed in terms of a benefit-cost ratio,

net present value, or equivalent uniform annual return. The information search showed that agencies place a high emphasis on cost-related performance measures such as initial costs, life-cycle costs, or incremental benefit-cost ratios, even though very few agencies currently give such performance measures full consideration in their decision-making processes.

Risk and Vulnerability to Catastrophic Failures. The legally mandated biennial and interim inspections in the United States have been generally successful in identifying safety-related bridge component deficiencies. As a result, catastrophic bridge failures due to deteriorated conditions have been relatively uncommon. Bridges typically have a 50-year or longer design life, during which the design parameters and material characteristics that were state of the art when bridges were designed tend to become inadequate for significantly changed serviceability and performance needs. Therefore, bridges can become vulnerable to failure modes not sufficiently accounted for in their original design, such as hydraulic or seismic failure modes.

In the early 1990s, the New York State Department of Transportation (NYSDOT) conducted a national survey of the catastrophic bridge failures since 1950. This survey identified the following six most common modes of failure: hydraulic (scour/ice/debris), overload (design/posted), steel structural details, collision (vehicle/collision), concrete structural details, and earthquake (Shirolé and Holt 1991). Since September 11, 2001, security against terrorist attacks has been added to the list of potential vulnerabilities.

Factors contributing to each of these different failure modes are generally so unique and diverse that no meaningful relationships may exist between them. Therefore, bridge ranking values based on factors related to one failure mode cannot be directly compared with those related to another failure mode. All failure modes have inherently associated with them a certain degree of risk based on frequency of occurrence and consequence of failure-causing events. Also associated with the degree of risk is a consequent requirement for priority of prudent action needed to preclude the possibility of failure. Therefore,

as is recognized by the NYSDOT, it is possible to develop a system to rate bridges across different failure modes based on the type and urgency of needed action (Shirolé and Holt 1991) and (Shirolé and Malik 1995). An agency can use this type of approach to translate individual failure mode ranking into a common rating scale for the purposes of prioritizing needed actions across all significant modes of failure. The programmatic implications of developing this risk-based prioritization are in enabling the agency to make informed policy decisions in selecting appropriate performance measures to eliminate or mitigate vulnerabilities to catastrophic bridge failures.

The questionnaire made it clear that performance measures such as risk reduction are currently of importance to bridge agencies and need to be given due consideration in BMS optimization.

User Cost. User cost is an important performance measure because it somewhat addresses the effects of any bridge improvement (or lack thereof) on the facility users. User costs include crash costs (due, for instance, to a functionally inadequate bridge) and travel time and vehicle operating cost due to work zone detours (Son and Sinha 1997). The questionnaire carried out in the present study showed that some agencies place great importance on safety-related performance measures, such as adequate geometrics.

Other Performance Measures. Ghosn and Moses (1991) addressed the concept of a reliability index as a safety criterion for bridge management. Other measures include geometric or functional adequacy, HBRR program eligibility, community impact (which can be represented by detour length), and traffic capacity (expressed as traffic volume capacity, load inventory, or operating rating). States ultimately seek to make decisions not based on the impacts on a single facility type, but in a holistic manner that incorporates all highway facility types and functional areas. This is a basic feature of integrated highway asset management that was espoused earlier by Sinha and Fwa (1989) in the 1980s and is increasingly receiving attention at national and state levels.

Appendix A References

Abed-Al-Rahim, I. J., and Johnston, D. W., "Bridge Replacement Cost Analysis Procedures." *Transportation Research Record 1490: Management and Maintenance of Bridge Structures,* Transportation Research Board, 1991.

Adams, T. M., and Barut, J. F., "Establishing MR&R Costs for a Network-Level Bridge Management System." *Transportation Research Record 1642: Maintenance and Management of Bridge Structures,* Transportation Research Board, 1998.

Al-Subhi, K. M., Johnston, D. W., and Farid, F., "OPBRIDGE: An Integrated Bridge Budget Forecasting and Allocation Module at the State Level." *Transportation Research Record 1268: Highway Maintenance Operations and Research 1990,* Transportation Research Board, 1990.

American Association of State Highway and Transportation Officials (AASHTO), *Guide for Design of Pavement Structures,* Part 1, Chapter 3: Life Cycle Cost Analysis for Pavements. 1986.

American Society for Public Administration (ASPA), *Resolution Encouraging the Use of Performance Measurement and Reporting by Government Organizations.* 1992.

Balas, E., and Martin, C. H., "Pivot and Complement—A Heuristic for 0-1 Programming." *Management Science,* Vol. 26, No. 1, 1980.

Balas, E., and Zemel, E., "An Algorithm for Large Zero-One Knapsack Problems." *Operations Research,* Vol. 28, No. 5, 1980.

Ballou, D. P., Chengalur-Smith, I. N., and Pazer, H. L., "Modeling the Costs of Bridge Rehabilitation," *Transportation Research, Part A: Policy and Practice,* Vol. 31, No. 4, 1997.

Benayoun, R., Roy, B., and Sussman, B., "ELECTRE: Une Methode Pour Guider le Choix en Presence de Points de Vue Multiples." *SEMA* (Metra International), Note de Travail No. 49, Direction Scientifique, Paris, France, 1966.

Benayoun, R., and Tergny, J., "Critere Multiples en Programmation Mathematique: Une Solution Dans le Cas Lineaire." *RFAIRO,* No. 3, Paris, France, 1969.

Bulusu, S. V., *Improved Methodologies to Predict Bridge Deterioration.* M.S. Thesis, Purdue University, West Lafayette, Indiana, 1996.

Butt, A. A., *Application of the Markov Process to Pavement Management Systems at Network Level.* Ph.D. Dissertation, University of Illinois at Urbana Champaign, Illinois, 1991.

Cambridge Systematics, Inc., *NCHRP Report 446: A Guidebook for Performance-Based Transportation Planning.* Transportation Research Board, 2000.

Chan, W. T., Fwa T., and Tan, J., "Optimal Fund-Allocation for Multi-district Highway Agencies," *ASCE Journal of Infrastructure Systems,* Vol. 9, No. 4, American Society of Civil Engineers, 2003.

Chang, S. E., and Shinozuka, M., "Life-Cycle Cost Analysis with Natural Hazard Risk." *ASCE Journal of Infrastructure Systems,* Vol. 2, No. 3, American Society of Civil Engineers, 1996.

Chen, C., and Johnston, D. W., *Bridge Management Under a Level-of-Service Concept Providing Optimum Improvement Action, Time and Budget Prediction.* Center for Transportation Engineering Studies, Department of Civil Engineering, North Carolina State University at Raleigh, 1987.

Cheng, M.-Y., and Ko, C.-H., "Object-Oriented Evolutionary Fuzzy Neural Inference System for Construction Management." *Journal of Construction Engineering and Management,* Vol. 129, No. 4, 2003.

Chengalur-Smith, I. N., Ballou, D. P., and Pazer, H. L., "Modeling the Costs of Bridge Rehabilitation." *Transportation Research Part A,* Vol. 31, No. 4, 1997.

Chu, P. C., and Beasley, J. E., "A Genetic Algorithm for the Multi-dimensional Knapsack Problem." *Journal of Heuristics,* Vol. 4, No. 1, 1998.

Cohon, J. L., *Multiobjective Programming and Planning.* Dover Publications, Mineola, New York, 2003.

Crum, R. L., and Derkinderen, F. G. J. (Eds.), *Capital Budgeting under Conditions of Uncertainty.* Martinus Nijhoff Publishing, 1981.

Dodgson, J., Spackman, M., Pearman, A., and Phillips, L., *Multicriteria Analysis: A Manual.* Department for Environment, Food and Rural Affairs, London, United Kingdom, 2001.

Dyer, M. E., "An O(n) Algorithm for the Multiple-Choice Knapsack Linear Program." *Mathematical Programming,* Vol. 29, 1984.

Dyer, M. E., Riha, W. O., and Walker, J., "A Hybrid Dynamic Programming/Branch-and-Bound Algorithm for the Multiple-Choice Knapsack Problem." *Journal of Computational and Applied Mathematics,* Vol. 58, No. 1, 1995.

Ezell, B. C., Farr, J. V., and Wiese, I., "Infrastructure Risk Analysis Model." *ASCE Journal of Infrastructure Systems,* Vol. 6, No. 3, American Society of Civil Engineers, 2000.

Farid, F., Johnston, D. W., Chen, C., Laverde, M., and Rihani, B., *Feasibility of Incremental Benefit-Cost Analysis for Optimal Allocation of Limited Budgets to Maintenance, Rehabilitation, and Replacement of Bridges.* FHWA Report DP-71-02, Federal Highway Administration, 1988.

Federal Highway Administration, *Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges.* Report FHWA-PD-96-001, http://www.fhwa.dot.gov/BRIDGE/mtguide.pdf, 1995.

Frangopol, D. M., Gharaibeh, E. S., Kong, J. S., and Miyake, M., "Optimal Network-Level Bridge Maintenance Planning Based on Minimum

Expected Cost." *Transportation Research Record 1696: Fifth International Bridge Engineering Conference, April 3–5, 2000, Tampa, Florida, Volume 2,* Transportation Research Board, 2000.

Freville, A., "The Multidimensional 0-1 Knapsack Problem: An Overview." *European Journal of Operational Research,* Vol. 155, No. 1, 2004.

Fwa, T., Chan, W., and Hoque, K., "Multiobjective Optimization for Pavement Maintenance Programming." *ASCE Journal of Transportation Engineering,* Vol. 26, No.5, American Society of Civil Engineers, 2002.

Gannon, E. J., Weyers, R. E., and Cady, P. D., "Cost Relationships for Concrete Bridge Protection, Repair, and Rehabilitation." *Transportation Research Record 1490: Management and Maintenance of Bridge Structures,* Transportation Research Board, 1995.

Gavish, B., and Pirkul, H. "Efficient Algorithms for Solving Multiconstraint Zero-One Knapsack Problems to Optimality." *Mathematical Programming,* Vol. 31, No. 1, 1985.

Gens, G., and Levner, E., "An Approximate Binary Search Algorithm for the Multiple-Choice Knapsack Problem." *Information Processing Letters,* Vol. 67, No. 5, 1998.

Ghosn, M., and Moses, F., "Reliability and Load Modeling for Bridge Management." *Transportation Research Record 1290: Third Bridge Engineering Conference, March 10–13, 1991, Denver, Colorado, Volume 2,* Transportation Research Board, 1991.

Goicoechea, A., Hansen, D. R., and Duckstein, L., *Multiobjective Decision Analysis with Engineering and Business Applications.* John Wiley & Sons, New York, New York, 1982.

Golabi, K., Kulkarni, R., and Way, G., "A Statewide Pavement Management System." *Interfaces 12.6,* 1982.

Governmental Accounting Standards Board, *GASB Statement No. 34: Basic Financial Statements—and Management's Discussion and Analysis—for State and Local Governments.* Norwalk, Connecticut, 1999.

Gruver, J. E., Patron, F. P., Batchelder, J. H., and Juster, R. D., "Highway Investment Analysis Package." *Transportation Research Record 599,* Transportation Research Board, 1976.

Guignier, F., and Madanat, S., "Optimization of Infrastructure Systems Maintenance and Improvement Policies." *ASCE Journal of Infrastructure Systems,* Vol. 5, No. 4, American Society of Civil Engineers, 1999.

Gurenich, D., and Vlahos, N. J., "Network-Level Bridge Management Systems for National Road Administrations." Paper presented at 8th International Bridge Management Conference, 1999.

Hanafi, S., and Freville, A., "An Efficient Tabu Search Approach for the 0-1 Multidimensional Knapsack Problem." *European Journal of Operational Research,* Vol. 106, No. 2, 1998.

Harper, W. V., Lam, J., Al-Salloum, A., Al-Sayyari, S., Al-Theneyan, S., Ilves, G., and Majidzadeh, K., "Stochastic Optimization Subsystem of a Network-Level Bridge Management System." *Transportation Research Record 1268: Highway Maintenance Operations and Research 1990,* Transportation Research Board, 1990.

Hastak, M., and Baim, E. J., "Risk Factors Affecting Management and Maintenance Cost of Urban Infrastructure." *ASCE Journal of Infrastructure Systems,* Vol. 7, No. 2, American Society of Civil Engineers, 2001.

Hawk, H., *NCHRP Report 483: Bridge Life-Cycle Cost Analysis.* Transportation Research Board, 2003.

Hearn, G., Frangopol, D. M., and Chakravorty, M., "Calibration and Application of Deterioration Models for Highway Bridges." Paper presented at the 4th International Bridge Engineering Conference, San Francisco, California, 1995.

Hegazy, T., Elbeltagi, E., and El-Behairy, H., "Bridge Deck Management System with Integrated Life Cycle Cost Optimization." *Transportation Research Record 1866: Maintenance and Management of Pavement and Structures,* Transportation Research Board, 2004.

Hobbs, B., and Meier, P., *Energy Decision and the Environment: A Guide to the User of Multicriteria Methods.* Kluwer Academic Publishers, Boston, Massachusetts, 2000.

Hudson, S. W., Carmichael III, R. F., Moser, L. O., Hudson, W. R., and Wilkes, W. J., *NCHRP Report 300: Bridge Management Systems,* Transportation Research Board, 1987.

Hyman, W., *NCHRP Report 511: Guide for Customer-Driven Benchmarking of Maintenance Activities,* Transportation Research Board, http://www.trb.org/news/blurb_detail.asp?id=3256, 2004.

Hyman, W. A., and Hughes, D. J., "A Computer Model for Lifecycle Cost Analysis of Statewide Bridge Repair and Replacement Needs." *Transportation Research Record 899: Bridge Inspection and Rehabilitation,* Transportation Research Board, 1983.

Hyman, W. A., and Thompson, P. D., *American Association of State Highway and Transportation Officials (AASHTO) Guidelines for Bridge Management Systems,* Final Report for NCHRP Project 20-7, Task 46, Transportation Research Board, 1992.

Intriligator, M. D., *Mathematical Optimization and Economic Theory.* Prentice Hall, Englewood Cliffs, New Jersey, 1971.

Jiang, M., Corotis, R., and Ellis, J., "Optimal Life-cycle Costing with Partial Observability." *ASCE Journal of Infrastructure Systems,* Vol. 6, No. 2, American Society of Civil Engineers, 2000.

Jiang, Y., Saito, M., and Sinha, K. C., "Bridge Performance Prediction Model Using the Markov Chain." *Transportation Research Record 1180: Bridge Design and Testing.* Transportation Research Board, 1988.

Jiang, Y., and Sinha, K. C. "Approach to Combine Ranking and Optimization Techniques in Highway Project Selection." *Transportation Research Record 1262: Planning, Management and Economic Analysis 1990,* Transportation Research Board, 1990.

Jiang, Y., and Sinha, K. C., "Dynamic Optimization Model for Bridge Management Systems." *Transportation Research Record 1211: Concrete Bridge Design and Maintenance: Steel Corrosion in Concrete,* Transportation Research Board, 1989.

Juster, R. D., and Pecknold, W. M., "Improving the Process of Programming Transportation Investments." *Transportation Research Record 599,* Transportation Research Board, 1976.

Kay, R. K., Mahoney, J. P., and Jackson, N. C., *The WSDOT Pavement Management System—A 1993 Update.* Research Report WA-RD 274.1. Washington State Department of Transportation, 1993.

Keeney, R. L., and Raiffa, H., *Decisions with Multiple Objectives—Preferences and Value Tradeoffs* (reprint). Cambridge University Press, New York, New York, 1993.

Kozanidis, G., and Melachrinoudis, E., "A Branch & Bound Algorithm for the 0-1 Mixed Integer Knapsack Problem with Linear Multiple Choice Constraints." *Computers & Operations Research,* Vol. 31, No. 5, 2004.

Kulkarni, R., Finn, F., Golabi, K., Johnson, R., and Alveti, E., *NCHRP Report 223: Maintenance Levels-of-Service Guidelines.* Transportation Research Board, 1980.

Kuprenas, J. A., Madjidi, F., Vidaurrazaga, A., and Lim, C. L., "Seismic Retrofit Program for Los Angeles Bridges." *ASCE Journal of Infrastructure Systems,* Vol. 4, No. 4, American Society of Civil Engineers, 1998.

Lee, J. S., and Guignard, M., "An Approximate Algorithm for Multidimensional Zero-One Knapsack Problems—A Parametric Approach." *Management Science,* Vol. 34, No. 3, 1988.

Li, Z., and Sinha, K. C., *Multicriteria Highway Programming Incorporating Risk and Uncertainty: A Methodology for Highway Asset Management*

System. Technical Report FHWA/IN/JTRP-2003/21, West Lafayette, Indiana, 2004.

Lorie, J. H., and Savage, L. J., "Three Problems in Rationing Capital." *Journal of Business,* Vol. XXVIII, No. 4, University of Chicago Press, 1955.

Magazine, M. J., and Oguz, O., "A Heuristic Algorithm for the Multi-dimensional Zero-One Knapsack Problem." *European Journal of Operational Research,* Vol. 16, 1984.

Mahmassani, H. S., *Methodological Aspects of a Decision Aid for Transportation Choices under Uncertainty.* Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1981.

Marsten, R. E., and Morin, T. L., "A Hybrid Approach to Discrete Mathematical Programming." *Mathematical Programming,* Vol. 14, No. 1, 1978.

Marsten, R. E., and Morin, T. L., "Optimal Solutions Found for Senju and Toyoda's 0/1 Integer Programming Problems." *Management Science,* Vol. 23, No. 12, 1977.

Martello, S., Pisinger, D., and Toth, P., "Dynamic Programming and Strong Bounds for the 0-1 Knapsack Problem." *Management Science,* Vol. 45, No. 3, 1999.

Martello, S., Pisinger, D., and Toth, P., "New Trends in Exact Algorithms for the 0-1 Knapsack Problem." *European Journal of Operational Research,* Vol. 123, No. 2, 2000.

Martello, S., and Toth, P., "Upper Bounds and Algorithms for Hard 0-1 Knapsack Problems." *Operations Research,* Vol. 45, No. 5, 1997.

McFarland, W., Rollins J., and Dheri, R., *Documentation for Incremental Benefit-Cost Technique,* technical report prepared for the Federal Highway Administration by the Texas Transportation Institute, Texas A&M University System, TX, 1983a.

McFarland, W., Rollins, J., and Harris, F., *Documentation for Integer Programming Technique,* technical report prepared for the Federal Highway Administration by the Texas Transportation Institute, Texas A&M University System, College Station, TX, 1983b.

McNeil, S., Markow, M., Newmann, L., Ordway, J., and Uzarski, D., "Emerging Issues in Transportation Facility Management." *ASCE Journal of Transportation Engineering,* Vol. 118, No. 4, American Society of Civil Engineers, 1992.

Mohamed, H. A., Abd El Halim, A. O., and Razaqpur, A. G., "Use of Neural Networks in Bridge Management Systems." *Transportation Research Record 1490: Management and Maintenance of Bridge Structures,* Transportation Research Board, 1995.

Monti, G., and Nistico, N., "Simple Probability-Based Assessment of Bridges under Scenario Earthquakes." *Journal of Bridge Engineering,* Vol. 7, No. 2, American Society of Civil Engineers, 2002.

Morcous, G., Rivard, H., and Hanna, A. M., "Modeling Bridge Deterioration Using Case-Base Reasoning." *ASCE Journal of Infrastructure Systems,* Vol. 8, No. 3, American Society of Civil Engineers, 2002.

Morin, T. L., and Esogbue, A. M. O., "The Imbedded State Space Approach to Reducing Dimensionality in Dynamic Programs of Higher Dimensions." *Journal of Mathematical Analysis and Applications,* Vol. 48, 1974.

Morin, T. L., and Marsten, R. E., "An Algorithm for Nonlinear Knapsack Problems." *Management Science,* Vol. 22, No. 10, 1976.

Moser, M., Jokanovic, D. P., and Shiratori, N. "Algorithm for the Multidimensional Multiple-Choice Knapsack Problem." *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* Vol. E80-A, No. 3, 1997.

Nash, S. C., and Johnston, D. W., *Level of Service Analysis for Bridge Maintenance Activities in North Carolina.* Report FHWA/NC/88-003, Federal Highway Administration, 1985.

National Academy of Public Administration, *Performance Monitoring and Reporting by Public Organizations.* 1991.

Nauss, R. M., "The 0-1 Knapsack Problem with Multiple Choice Constraints." *European Journal of Operational Research,* Vol. 2, 1978.

O'Connor, D., and Hyman, W. A., *Bridge Management Systems.* Report FHWA DP-71-01R, Federal Highway Administration, Report for Demonstration Project 71, 1989.

Optima, Inc., Cambridge Systematics, and Federal Highway Administration Office of Technology Applications, *Pontis Version 2.0: A Network Optimization System for Bridge Improvements and Maintenance.* Report FHWA-SA-93-083, Federal Highway Administration, 1993.

Pilson, C., Hudson W. R., and Anderson, V., "Multiobjective Optimization in Pavement Management by Using Genetic Algorithms and Efficient Surfaces." *Transportation Research Record 1655: Pavement Management Systems and Other Pavement Design Issues,* Transportation Research Board, 1999.

Pirkul, H., "A Heuristic Solution Procedure for the Multiconstraint Zero-One Knapsack Problem." *Naval Research Logistics,* Vol. 34, 1987.

Pisinger, D., "A Minimal Algorithm for the Multiple-Choice Knapsack Problem." *European Journal of Operational Research,* Vol. 83, 1995.

Pisinger, D., "An Exact Algorithm for Large Multiple Knapsack Problems." *European Journal of Operational Research,* Vol. 114, 1999.

Poister, T. H., *NCHRP Synthesis of Highway Practice 238: Performance Measurement in State Departments of Transportation.* Transportation Research Board, 1997.

Ravirala, V., Grivas, D. A., Madan, A., and Schultz, B. C., "Multicriteria Optimization Method for Network-Level Bridge Management." *Transportation Research Record 1561: Maintenance and Management of Highway Structures,* Transportation Research Board, 1996.

Roy, B., and Bertier, B., "La Methode ELECTRE II: Une Methode de Classment en Presence de Criteres Multiples." *Note de Travail No. 142.* Direction Scientifique, Groupe Metra, France, 1971.

Saaty, T. L., "A Scaling Method for Priorities in Hierarchical Structures." *Journal of Mathematical Psychology,* Vol. 15, Academic Press, 1977.

Salem, O., AbouRizk, S., and Ariaratnam, S., "Risk-Based Life-Cycle Costing of Infrastructure Rehabilitation and Construction Alternatives." *ASCE Journal of Infrastructure Systems,* Vol. 9, No. 1, American Society of Civil Engineers, 2003.

Scherschligt, D., and Kulkarni, R. B., "Pontis-Based Health Indexes for Bridge Priority Evaluation." *Transportation Research E-Circular E-C049: 9th International Bridge Management Conference,* Transportation Research Board, http://www.trb.org/news/blurb_detail.asp?id=1885, 2003.

Senju, S., and Toyoda, Y., "An Approach to Linear Programming with 0-1 Variables." *Management Science,* Vol. 15, No. 4, 1968.

Shepard, R. W., and Johnson, M. B., "California Bridge Health Index: A Diagnostic Tool to Maximize Bridge Longevity, Investment," *TR News,* No. 215, July-August, Transportation Research Board, http://onlinepubs.trb.org/onlinepubs/trnews/trnews215full.pdf, 2001.

Shih, W., "A Branch and Bound Method for the Multiconstraint Zero-One Knapsack Problem." *The Journal of the Operational Research Society,* Vol. 30, No. 4, 1979.

Shirolé, A. M., and Malik, A. H., "Seismic Vulnerability Evaluation of New York State Bridges." *Restructuring: America and Beyond (Proceedings of Structures Congress XIII held in Boston, Massachusetts, April 2–5, 1995),* American Society of Civil Engineers, http://www.pubs.asce.org/WWWdisplay.cgi?9500991, 1995.

Shirolé, A. M., and Holt, R. C., "Planning for a Comprehensive Bridge Safety Assurance Program." *Transportation Research Record 1290: Third Bridge Engineering Conference, March 10–13, 1991, Denver, Colorado, Volume 1,* Transportation Research Board, 1991.

Shirolé, A. M., and Loftus, M. J., "Assessment of Bridge Vulnerability to Hydraulic Failures." *Transportation Research Record 1347: Maintenance of Highway Structures,* Transportation Research Board, 1992.

Sinha, K. C., and Fwa, T. F., "On the Concept of Total Highway Management." *Transportation Research Record 1229: Finance, Planning, and Administration,* Transportation Research Board, 1989.

Sinha, K. C., Muthusubramanyam, M., and Ravindran, A., "Optimization Approach for Allocation of Funds for Maintenance and Preservation of the Existing Highway System." *Transportation Research Record 826: Transportation Programming, User Benefit Analysis and Vehicle Licensing,* Transportation Research Board, 1981.

Sinha, P., and Zoltners, A. A., "The Multiple-Choice Knapsack Problem." *Operations Research,* Vol. 27, No. 3, 1979.

Sivakumar, B., Minervino, C. M., and Edberg, W., "New Bridge Performance Measures for Prioritizing Bridges." *Transportation Research E-Circular E-C049: 9th International Bridge Management Conference,* Transportation Research Board, http://www.trb.org/news/blurb_detail.asp?id=1885, 2003.

Small, E. P., "Examination of Alternative Strategies for Integration of Seismic Risk Considerations in Bridge Management Systems." Paper presented at 8th International Bridge Management Conference, 1999.

Smith, K. L., Stivers, M. L., Hoerner, T. E., Romine, A. R., *NCHRP Web Document 8: Highway Maintenance Quality Assurance,* Transportation Research Board, http://www.nap.edu/catalog/6346.html, 1997.

Sobanjo, J. O., and Thompson, P. D., *Development of Agency Maintenance, Repair, and Rehabilitation (MR&R) Cost Data for Florida's Bridge Management System: Final Report.* Florida Department of Transportation, www.pdth.com/images/fdotagcy.pdf, 2001.

Söderqvist, M., and Veijola, M., "Finnish Project Level Bridge Management System." Paper presented at the 8th International Bridge Management Conference, 1999.

Son, Y., and Sinha, K. C., "Methodology to Estimate User Costs in Indiana Bridge Management Systems." *Transportation Research Record 1597: Maintenance of Highway Pavements and Structures,* Transportation Research Board, 1997.

Stein, S. M., Young, G. K., Trent, R. E., and Pearson, D. R., "Prioritizing Scour Vulnerable Bridges Using Risk." *ASCE Journal of Infrastructure Systems,* Vol. 5, No. 3, American Society of Civil Engineers, 1999.

Steuer, R. E., *Multiple Criteria Optimization: Theory, Computation, and Application.* Robert E. Krieger Publishing Company, 1989.

Subramanian, B., Schafer, D. L., and Tyler, J. E., *Documentation for Dynamic Programming Technique.* Vol. 5 of a final report to Federal Highway Administration, 1983.

Tack, J. N., and Chou, Y. J., "Pavement Performance Analysis Applying Probabilistic Deterioration Methods." *Transportation Research Record 1769: Management, Monitoring, and Accelerated Testing,* Transportation Research Board, 2001.

Taha, M. A., and Hanna, A. S., "Evolutionary Neural Network Model for the Selection of Pavement Maintenance Strategy." *Transportation Research Record 1497: Artificial Intelligence and Geographical Information,* Transportation Research Board, 1995.

Tam, C. K., and Stiemer, S. F., "Development of Bridge Corrosion Cost Model for Coating Maintenance." *Journal of Performance of Constructed Facilities,* University of British Columbia, Vancouver, Canada, 1996.

Tao, Z., Corotis, R., and Ellis, J., "Reliability-Based Structural Design with Markov Decision Process." *ASCE Journal of Structural Engineering,* Vol. 121, No. 6, 1995.

Tee, A. B., Bowman, M. D., and Sinha, K. C., "A Fuzzy-Mathematical Approach for Bridge Condition Evaluation." *ASCE Journal of Civil Engineering Systems,* Vol. 5, No. 1, 1988.

Thompson, P. D., and Markow, M., *NCHRP Synthesis of Highway Practice 227: Collecting and Managing Cost Data for Bridge Managements Systems.* Transportation Research Board, http://www.trb.org/news/blurb_detail.asp?id=3330, 1996.

Thompson, P., Merlo, T., Kerr, B., Cheetham, A., and Ellis, R. "The New Ontario Bridge Management System." Paper presented at the 8th International Bridge Management Conference, 1999a.

Thompson, P. D., Najafi, F. T. Soares, R., and Choung, H. J., *Florida DOT Pontis User Cost Study: Final Report.* Florida Department of Transportation, www.pdth.com/images/fdotuser.pdf, 1999b.

Thompson, P. D., and Shepard, R., "American Association of State Highway and Transportation Officials (AASHTO) Commonly-Recognized Bridge Elements: Successful Applications and Lessons Learned." *Proceedings, National Workshop on Commonly Recognized Measures for Maintenance, San Antonio, TX,* 2000.

Toyoda, Y., "A Simplified Algorithm for Obtaining Approximate Solutions to Zero-One Programming Problems." *Management Science,* Vol. 21, No. 12, 1975.

Vasquez, M., and Vimont, Y., "Improved Results on the 0-1 Multidimensional Knapsack Problem." *European Journal of Operational Research,* Vol. 165, No. 1, 2005.

Vitale, J. D., Sinha, K. C., and Woods, R. E., "Analysis of Optimal Bridge Programming Policies." *Transportation Research Record 1561: Maintenance and Management of Highway Structures,* Transportation Research Board, 1996.

Volgenant, A., and Zoon, J. A., "An Improved Heuristic for Multidimensional 0-1 Knapsack Problems." *The Journal of the Operational Research Society,* Vol. 41, No. 10, 1990.

Walls, J., and Smith, M., *Life-Cycle Cost Analysis in Pavement Design—In Search of Better Investment Decisions.* Report FHWA-SA-98-179, Federal Highway Administration, http://isddc.dot.gov/OLPFiles/FHWA/013017.pdf, 1998.

Wang, K. C. P., Zaniewski, J., and Way, G., "Probabilistic Behavior of Pavements." *ASCE Journal of Transportation Engineering,* Vol. 120, No. 3, American Society of Civil Engineers, 1994.

Wipf, T. J., Erickson, D. L., and Klaiber, F. W., "Cost-Effectiveness Analysis for Strengthening Existing Bridges." *Transportation Research Record 1113: Bridge Maintenance, Corrosion, Joint Seals, and Polymer Mortar Materials,* Transportation Research Board, 1987.

Zeleny, M., "Compromise Programming." In Cochrane, J. L., and Zeleny, M. (editors), *Multicriteria Decision Making,* University of South Carolina Press, Columbia, South Carolina, 1973.

Zemel, E., "A Linear Time Randomizing Algorithm for Searching Ranked Functions." *Algorithmica,* Vol. 2, No. 1, 1987.

Zemel, E., "An O(n) Algorithm for the Linear Multiple Choice Knapsack Problem and Related Problems." *Information Processing Letters,* Vol. 18, No. 3, 1984.

Zemel, E., "The Linear Multiple Choice Knapsack Problem." *Operations Research,* Vol. 28, No. 6, 1980.

Zhang, C. W., and Ong, H. L., "Solving the Biobjective Zero-One Knapsack Problem by an Efficient LP-Based Heuristic." *European Journal of Operational Research,* Vol. 159, No. 3, 2004.

Zimmerman, H. I., *Fuzzy Sets, Decision Making and Expert Systems.* Kluwer Academic Publishers, Boston, 1987.

A P P E N D I X   B

Performance Measures Identified for This Study

B.1 General Goals

Although decision makers can define their own performance measures as long as they provide any pertinent data, Table B.1 summarizes the set of performance measures identified in this project for each goal. The following sections describe these performance measures in detail.

B.2 NBI Condition Ratings

Condition ratings are used to describe the existing, in-place bridge as compared with the as-built condition. The ratings are based on the evaluation of the materials and the physical condition of the deck, superstructure, and substructure. The condition evaluation of culverts is also included. These ratings vary from 0–9 under the current standard. The evaluation scale in the *Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges* (FHWA 1995) is as follows:

N *Not Applicable.*
9 *Excellent Condition.*
8 *Very Good Condition*—no problems noted.
7 *Good Condition*—some minor problems.
6 *Satisfactory Condition*—structural elements show some minor deterioration.
5 *Fair Condition*—all primary structural elements are sound but may have minor section loss, cracking, spalling, or scour.
4 *Poor Condition*—advanced section loss, deterioration, spalling, or scour.
3 *Serious Condition*—loss of section, deterioration, spalling, or scour have seriously affected primary structural components. Local failures are possible. Fatigue cracks in steel or shear cracks in concrete may be present.
2 *Critical Condition*—advanced deterioration of primary structural elements. Fatigue cracks in steel or shear cracks in concrete may be present, or scour may have removed substructure support. Unless the condition is closely monitored, it may be necessary to close the bridge until corrective action is taken.
1 *"Imminent" Failure Condition*—major deterioration or section loss present in critical structural components or obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic, but corrective action may put it back in light service.
0 *Failed Condition*—out of service and beyond corrective action.

The following condition ratings are based on the above evaluation scale:

Deck Condition Rating (NBI Item 58): This item describes the overall condition rating of the deck.
Superstructure Condition Rating (NBI Item 59): This item describes the physical condition of all structural members.
Substructure Condition Rating (NBI Item 60): This item describes the physical condition of piers, abutments, piles, fenders, footings, or other components.
Culvert Condition Rating (NBI Item 62): This item evaluates the alignment, settlement, joints, structural condition, scour, and other items associated with culverts. The rating code is intended to be an overall condition evaluation of the culvert.

B.3 Health Index

The bridge health index is a single numerical rating of 0 (worst possible condition) to 100 (best condition) that reflects element inspection data in relation to the asset value of a bridge or network of bridges. The formulas for computation of the bridge health index are as follows (Shepard and Johnson 2001):

$$HI = \left(\frac{\sum_e CEV}{\sum_e TEV} \right) \times 100\%$$

$$TEV = TEQ \times W_e$$

Table B.1. Performance measures.

GOAL	PERFORMANCE MEASURES
1. Preservation of Bridge Condition	(a) Condition Ratings (NBI 58-60, 62) (b) Health Index (c) Sufficiency Rating
2. Traffic Safety Enhancement	(a) Geometric Rating/Functional Obsolescence (b) Inventory Rating *or* Operating Rating
3. Protection from Extreme Events	(a) Scour Vulnerability Rating (b) Fatigue/Fracture Criticality Rating (c) Earthquake Vulnerability Rating (d) Other Disaster Vulnerability Rating (Collision, Overload, Human-Made)
4. Agency Cost Minimization	(a) Initial Cost (b) Life-Cycle Agency Cost
5. User Cost Minimization	(a) Life-Cycle User Cost

$$CEV = W_e \times \sum_i \left(QCS_i \times WF_i \right)$$

$$WF_i = 1 - \frac{i-1}{State\ Count - 1} \tag{B-1}$$

where

HI = health index,
CEV = current element value,
TEV = total element value,
TEQ = total element quantity,
QCS_i = quantity in condition state *i,*
WF_i = weighting factor for the condition state *i,* and
W_e = element weight.

The element weight W_e can be either the element's failure cost or the weight coefficient explicitly assigned to the element in Pontis multiplied by the cost of the most expensive action defined for that element (AASHTO 2001).

B.4 Sufficiency Rating

The sufficiency rating is a method of evaluating highway bridge data by calculating four separate factors to obtain a numeric value (i.e., percentage) that is indicative of bridge sufficiency to remain in service. One hundred percent represents an entirely sufficient bridge, and 0% represents an entirely insufficient or deficient bridge (FHWA 1995). The four factors are as follows:

1. Structural adequacy and safety (S_1): 55% max
   – Superstructure
   – Substructure
   – Culverts
   – Inventory rating

2. Serviceability and functional obsolescence (S_2): 30% max
   – Lanes on structure
   – Average daily traffic
   – Approach roadway width
   – Structure type
   – Bridge roadway width
   – Vertical clearance over deck
   – Deck condition
   – Structural evaluation
   – Deck geometry
   – Under-clearances
   – Waterway adequacy
   – Approach roadway alignment
   – Highway designation
3. Essentiality for public use (S_3): 15% max
   – Detour length
   – Average daily traffic
   – Highway designation
4. Special reductions (S_4): 13% max
   – Detour length
   – Traffic safety features
   – Structure type

Then sufficiency rating, *SR*, is calculated as follows:

$$SR = S_1 + S_2 + S_3 - S_4 \tag{B-2}$$

B.5 Geometric Rating

Geometric rating (NBI Item 68) is an overall rating for deck geometry based on two evaluations: (1) NBI Item 51, bridge roadway width, and (2) NBI Item 53, vertical over-clearance. Geometric rating varies from 0 to 9 (FHWA 1995).

B.6 Inventory Rating

Inventory rating (NBI Item 66) represents the design standard and is a load level that can safely use the existing structure for an indefinite period of time (FHWA 1995). The inventory rating is coded as a three-digit number to represent the total mass in metric tons of the entire vehicle measured to the nearest tenth of a metric ton.

B.7 Operating Rating

Operating rating (NBI Item 64) is a capacity rating representing the absolute maximum permissible load level to which the structure may be subjected for the vehicle type used in the rating (FHWA 1995). The operating rating is coded as a three-digit number to represent the total mass in metric tons of the entire vehicle measured to the nearest tenth of a metric ton.

B.8 Vulnerability Ratings—General Procedure

We propose a vulnerability rating procedure adopted from NYSDOT (1996a), as shown in Figure B.1. The vulnerability ratings are based on the likelihood and consequence of an event. The subjective measure for likelihood is based on a classifying process that is specific to the type of vulnerability considered. The consequence of failure is based on the type of failure the bridge is prone to and the exposure to the public that a failure would cause. The exposure parameter is a measure of the effect—in terms of traffic volume and functional classification—that a failure of a structure will have on the users of the bridge.

The procedure proposed here is a general procedure followed for various types of vulnerability assessments, such as scour, fatigue/fracture, earthquake, collision, and overload. The

Table B.2. Vulnerability scores.

Vulnerability Class	Likelihood Score
High	10
Medium	6
Low	2
Not Vulnerable	0

Failure Type	Failure Type Score
Catastrophic	5
Partial Collapse	3
Structural Damage	1

Traffic Volume	Traffic Volume Score
> 25,000 AADT	2
4,000–25,000 AADT	1
< 4,000 AADT	0

Functional Classification	Functional Classification Score
Interstate and Freeway	3
Arterial	2
Collector	1
Local Road & Below	0

AADT = annual average daily traffic.

procedure to assign a vulnerability class is specific to the type of vulnerability, which is discussed in subsequent sections. Table B.2 presents guidelines on assigning scores to arrive at a vulnerability score. For each vulnerability type, a vulnerability score is defined as follows:

$$\text{Vulnerability Rating Score} = \text{Likelihood Score} + \text{Consequence Score} \quad \text{(B-3)}$$

where

$$\text{Consequence Score} = \text{Failure Type Score} + \text{Exposure Score}$$

and

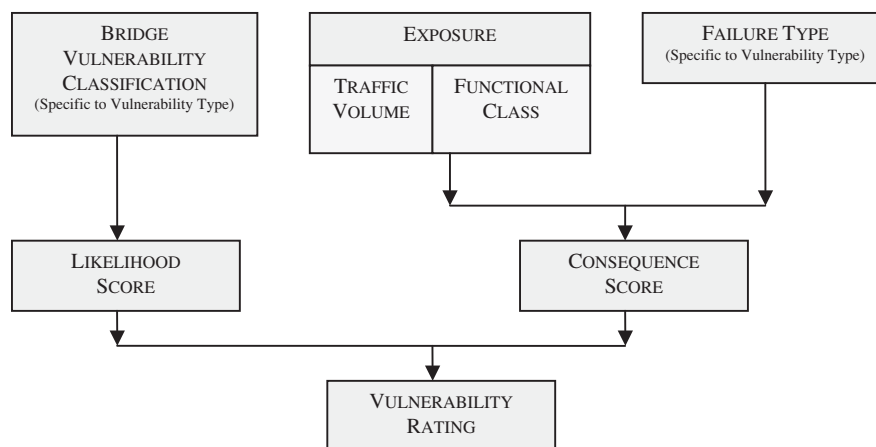$$\text{Exposure Score} = \text{Traffic Volume Score} + \text{Functional Classification Score}.$$

Figure B.1. Vulnerability rating procedure (from NYSDOT 1996a).

Table B.3. Vulnerability ratings.

Vulnerability Score	Vulnerability Rating
> 15	1 (most vulnerable)
13–16	2
9–14	3
< 15	4
< 9	5 (least vulnerable)

Vulnerability scores are then converted to vulnerability ratings on a scale of 1 to 5, as shown in Table B.3. Table B.4 presents the definitions of vulnerability ratings (O'Connor 2000).

Following are definitions of vulnerability failure types (NYSDOT 1996a):

- **Structural Damage:** The structure is vulnerable to localized failures that may be manifested in the form of excessive deformation or cracking in the primary superstructure or substructure members. Such defects are typically due to accumulated effects of temperature variations, low temperature, traffic load repetitions, and deicing chemicals. A failure of this type may be unnoticed by the traveling public but would require repair once it is discovered.
- **Partial Collapse:** The structure is vulnerable to major deformation or discontinuities of a span (which would result in loss of service to traffic on or under the bridge). This may be the result of tipping or tilting of the substructure causing deformations in the superstructure. A failure of this type may endanger the lives of those on or under the structure.
- **Catastrophic:** The structure is vulnerable to a sudden and complete collapse of a superstructure span or spans. This may be the result of a partial or total failure of either the superstructure or the substructure. A failure of this type would endanger the lives of those on or under the structure.

B.9 Description of Vulnerability Types

B.9.1 Scour Vulnerability Rating

The procedure to assess the scour vulnerability rating (Shirolé and Holt 1991; Shirolé and Loftus 1992) is divided into two sections: general hydraulic assessment and foundation assessment. In each section, specific parameters are examined and a value is assigned to describe the existing conditions. In the foundation assessment section, all of abutments and piers on a structure are evaluated, but the more critical of the two scores is used. The final score is then used to arrive at a high, medium, or low scour vulnerability rating. The schematic representation of the procedure is shown in Figure B.2.

B.9.2 Fatigue Vulnerability Rating—Concrete

The procedure to assess the concrete detail vulnerability rating consists of evaluating the superstructure and substructure and conducting a general assessment (NYSDOT 1997). The more critical of the superstructure and substructure scores is used to arrive at the vulnerability class. The schematic representation of the procedure is shown in Figure B.3.

B.9.3 Fatigue Vulnerability Rating—Steel

The procedure to assess the steel detail vulnerability rating consists of evaluating the superstructure and the substructure (NYSDOT 1999). The more critical of the two scores is then used to arrive at the vulnerability rating. The schematic representation of the procedure is shown in Figure B.4.

Table B.4. Definitions of vulnerability ratings.

Vulnerability Rating	Definition
1	Designates a vulnerability to failure resulting from loads or events that are likely to occur. Remedial work to reduce the vulnerability is an immediate priority.
2	Designates a vulnerability to failure resulting from loads or events that may occur. Remedial work to reduce the vulnerability is not an immediate priority but may be needed in the near future.
3	Designates a vulnerability to failure resulting from loads or events that are possible but not likely. This risk can be tolerated until a normal capital project can be implemented.
4	Designates a vulnerability to failure presenting minimal risk providing that anticipated conditions do not change. Unexpected failure can be avoided during the remaining service life of the bridge by performing normal scheduled inspections, with attention to factors influencing the vulnerability.
5	Designates a vulnerability to failure that is less than or equal to the vulnerability of a structure built to the current design standards. Likelihood of failure is remote.

GENERAL HYDRAULIC ASSESSMENT
- River slope/velocity
- Channel bottom
- Channel configuration
- Debris ice problem
- Near river confluence
- Effected by backwater
- Historic scour depth
- Historic maximum flood depth
- Adequate opening
- Overflow/relief available
- Simple spans

FOUNDATION ASSESSMENT

ABUTMENTS
- Scour countermeasures
- Abutment foundation
- Abutment location on river bend
- Angle of inclination
- Embankment encroachment

PIERS
- Scour countermeasures
- Pile foundation
- Skew angle
- Pier/pile bottom below streambed
- Pier width

SCOUR VULNERABILITY RATING

Figure B.2. Scour vulnerability rating.

SUPERSTRUCTURE ASSESSMENT
- Skew
- Curvature
- Redundancy
- Flagging history
- Deterioration of beams
- Reinforcement corrosion
- Concrete deterioration
- Unreinforced concrete members
- Deflection conditions
- Prestressed concrete members
- Posttensioned concrete members

SUBSTRUCTURE ASSESSMENT
- Deterioration of components
- Stress-related cracking at joints
- Stress-related condition of pier capbeams
- Cracking due to differential settlement
- Punching shear condition
- Hammerhead pier
- Solid pier
- Concrete rigid frame
- Pier bents with concrete piles
- Concrete pier caps

GENERAL ASSESSMENT
- Bridge location
- Year of construction
- Wearing surface
- Condition ratings
- Debris accumulation on concrete

FATIGUE VULNERABILITY RATING

Figure B.3. Fatigue vulnerability rating—concrete.

SUPERSTRUCTURE ASSESSMENT

PRIMARY MEMBER
- Redundancy
- Fatigue resistance
- ADTT and fatigue design
- Material toughness
- Distortion-induced cracking
- Truss

EXTERNAL CONNECTIONS
- Connection type:
  - Bearing
  - Framed
  - Suspended or hinged support

ACCUMULATED DAMAGE
- Ever flagged
- Reason flagged
- Painted steel
- Deterioration factors
- Condition rating factors

SUBSTRUCTURE ASSESSMENT

PRIMARY MEMBER
- Redundancy
- Fatigue resistance
- ADTT and fatigue design
- Material toughness
- Distortion-induced cracking
- Internal component connections

ACCUMULATED DAMAGE
- Ever flagged
- Reason flagged
- Painted steel
- Deterioration factors
- Condition rating factors

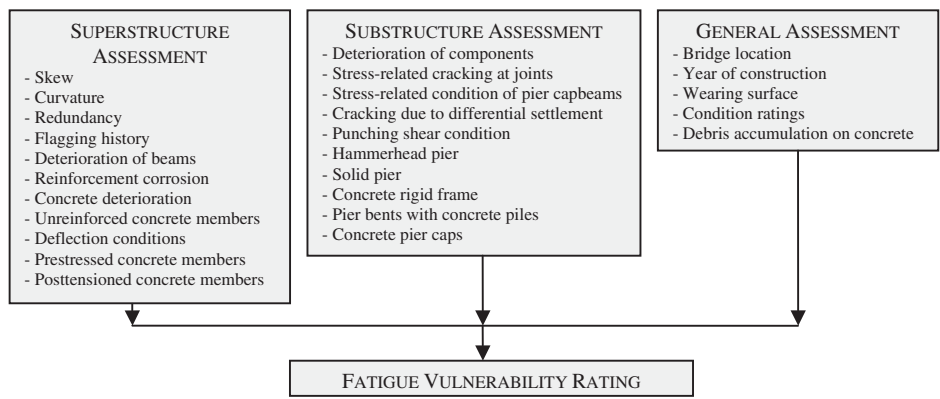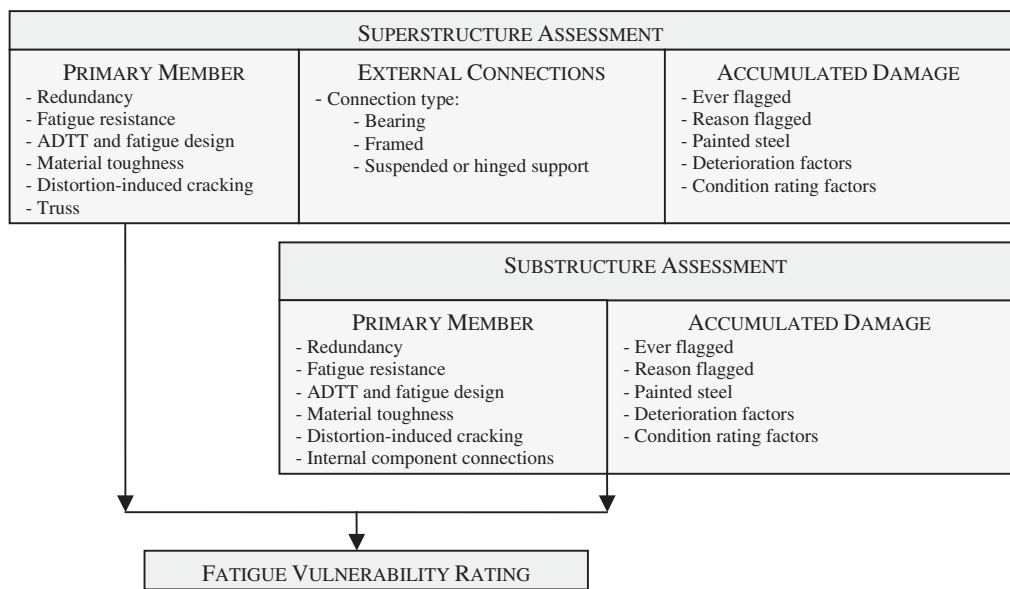### FATIGUE VULNERABILITY RATING

ADTT = average daily truck traffic.

Figure B.4. Fatigue vulnerability rating—steel.

B.9.4 Earthquake Vulnerability Rating

The earthquake vulnerability rating (NYSDOT 2002) is assessed based on the classification score, *CS:*

$$CS = V \times E \qquad (B\text{-}4)$$

where

V = a numerical measure of the structural vulnerability and
E = seismic hazard rating for the site.

The structural vulnerability measure is based on the following:

- Vulnerability score for connections, bearings, and seat widths
  - Bearing types
  - Support lengths
  - Support skew
- Pier vulnerability score
  - Pier design
  - Shear failure
  - Flexural failure
- Abutment vulnerability score
- Liquefaction vulnerability score

The seismic hazard rating is based on the design seismic acceleration coefficient and the soil profile type to allow for soil amplification effects.

The vulnerability assessment measures described in this section are adapted from the NYSDOT vulnerability manuals. These manuals are available at http://www.dot.state.ny.us/structures/manuals.html.

B.9.5 Other Vulnerability Rating

Other vulnerability rating, *OVR,* is calculated as the average of three ratings corrected to the nearest integer:

$$OVR = (COL + OVL + HM)/3 \qquad (B\text{-}5)$$

where

COL = collision vulnerability rating,
OVL = overload vulnerability rating, and
HM = human-made vulnerability rating.

B.9.5.1 Collision Vulnerability Rating

The collision vulnerability rating is based on the following (NYSDOT 1996b):

- Truck-on-bridge collision vulnerability,
- Superstructure vulnerability to truck-under-bridge collision,
- Pier vulnerability to truck-under-bridge collision,
- Superstructure vulnerability to water vessel collision,
- Pier vulnerability to water vessel collision,
- Superstructure vulnerability to train-under-bridge collision, and
- Pier vulnerability to train-under-bridge collision.

B.9.5.2 Overload Vulnerability Rating

The overload vulnerability rating is based on the following (NYSDOT 1996c):

- Load expectancy. This is to evaluate the likelihood that a load heavy enough to cause a failure will ever use the bridge.
- Structural capacity.
  - Resistance: This is to evaluate the capacity of a structure to resist applied loads. The operating rating parameter is modified to account for reserve capacity.
  - Condition: This is to include the effect of structural deterioration based on condition ratings.

B.9.5.3 Human-Made Vulnerability Rating

Human-made vulnerability rating is based on the criticality index of a bridge, which is assessed on the basis of the following (Rummel et al. 2002):

- Commerce.
  - Average and maximum daily truck traffic.
- Transportation needs.
  - Average and maximum daily traffic.
  - Maximum detour length and detour length for that bridge.
- Connectivity.
  - Average and maximum daily traffic on the intersecting route.
  - Interstate intersection.
- Navigational access.
  - Whether the bridge requires a Coast Guard permit.
- International access.
  - Whether the bridge borders on a neighboring country.
- Military movement.
  - Whether the bridge is located on the strategic highway network.
- Replacement/repair.
  - Type of superstructure.
  - Span length.

The criticality index is then converted to a vulnerability rating of 1, 2, 3, 4, or 5, where a higher rating represents a higher vulnerability.

Appendix B References

American Association of State Highway and Transportation Officials (AASHTO), *Pontis Technical Manual,* Release 4.0. 2001.

Federal Highway Administration, *Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges.* Report FHWA-PD-96-001, http://www.fhwa.dot.gov/BRIDGE/mtguide.pdf, 1995.

New York State Department of Transportation, *Hydraulic Vulnerability Manual.* Bridge Safety Assurance Program, New York State Department of Transportation, 1996a.

New York State Department of Transportation, *Collision Vulnerability Manual.* Bridge Safety Assurance Program, 1996b.

New York State Department of Transportation, *Overload Vulnerability Manual.* Bridge Safety Assurance Program, 1996c.

New York State Department of Transportation, *Concrete Details Vulnerability Manual.* Bridge Safety Assurance Program, 1997.

New York State Department of Transportation, *Steel Details Vulnerability Manual.* Bridge Safety Assurance Program, 1999.

New York State Department of Transportation, *Seismic Vulnerability Manual.* Bridge Safety Assurance Program, 2002.

O'Connor, J. S., "Bridge Safety Assurance Measures Taken in New York State." *Transportation Research Record 1696: Fifth International Bridge Engineering Conference, April 3–5, 2000, Tampa, Florida, Volume 1,* Transportation Research Board, 2000.

Rummel, T., Hyzak, M., and Ralls, M. L., "Transportation Security Activities in Texas." Paper presented at the 19th International Bridge Conference, 2002.

Shepard, R. W., and Johnson, M. B., "California Bridge Health Index: A Diagnostic Tool to Maximize Bridge Longevity, Investment," *TR News,* No. 215, July–August, Transportation Research Board, http://onlinepubs.trb.org/onlinepubs/trnews/trnews215full.pdf, 2001.

Shirolé, A. M., and Holt, R. C., "Planning for a Comprehensive Bridge Safety Assurance Program." *Transportation Research Record 1290: Third Bridge Engineering Conference, March 10–13, 1991, Denver, Colorado, Volume 1,* Transportation Research Board, 1991.

Shirolé, A. M., and Loftus, M. J., "Assessment of Bridge Vulnerability to Hydraulic Failures." *Transportation Research Record 1347: Maintenance of Highway Structures,* Transportation Research Board, 1992.

APPENDIX C

Sample Questionnaire

C.1 Direct Questioning Approach

This questionnaire solicits information on the relative weights across goals and across performance criteria within each goal using the direct questioning approach. Please assign a weight for each of the goals/criteria below such that they all add up to 100. A higher weight represents greater importance.

Set-1 (Preservation of Bridge Condition)

Performance Criterion	Weight
1. Condition Ratings	_____
2. Health Index	_____
3. Sufficiency Rating	_____

C.2 Analytic Hierarchy Process

This questionnaire solicits information on the relative weights across goals and across performance criteria within each goal using the analytic hierarchy process on the basis of pairwise comparisons. Here you are asked to compare two goals/criteria at a time in terms of their relative importance. Goal/criterion "importance" can generally be defined as follows: *Goal/Criterion X is more important than Goal/Criterion Y if the event (X is at its best value, Y is at its worst value) is preferred to the event (X is at its worst value, Y is at its best value).*

The ratios of importance you state should reflect your trade-off preferences between these goals/criteria. The scale in Table C.1 is suggested. Using this scale, please fill in the upper triangle of the matrix (Table C.2).

C.3 Gamble Method

This questionnaire solicits information on the relative weights across performance criteria using the gamble method. Here you are offered two situations:

- Receive a guaranteed reward or
- Play a gamble with a probability of gaining a high reward or nothing.

We then ask you whether you would like to receive your guaranteed reward *or* would you like to play the gamble.

Set 1 (Condition Ratings)

Step 1a. Consider the following two situations:

- Situation 1: The alternative yields the performance values with certainty (guaranteed).
- Situation 2: The alternative yields the performance values with given probabilities.

See Table C.3.

Are you satisfied with the guaranteed situation, or would you like to play the gamble? Please check one:

Guaranteed Situation _____
Play the Gamble _____

(Hint: We expect your answer to be guaranteed situation.)

Step 1b. So let us start with the scenario where you are satisfied with the guaranteed situation. Now we start increasing the probability p (i.e., the probability of achieving best levels in all performance criteria in Situation 2). As this probability increases, the gamble becomes more attractive.

How high would this probability p have to be in order for you to give up the guaranteed situation and be willing to play the gamble? In other words, at what probability p would you make the switch from the guaranteed situation to the gamble?

Step 2. Consider the two situations shown in Table C.4. How high would this probability p have to be in order for you to give

Table C.1. Analytic hierarchy process interpretations.

IF:	THEN X/Y ratio should be:
Goal X is extremely more important than Goal Y	9
Goal X is strongly more important than Goal Y	7
Goal X is moderately more important than Goal Y	5
Goal X is slightly more important than Goal Y	3
Goal X is equally important as Goal Y	1
Goal X is slightly less important than Goal Y	1/3
Goal X is moderately less important than Goal Y	1/5
Goal X is strongly less important than Goal Y	1/7
Goal X is extremely less important than Goal Y	1/9

Table C.2. Matrix 1 (bridge condition reservation).

		Y		
		Condition Ratings	Health Index	Sufficiency Rating
X	Condition Ratings	1		
	Health Index	--	1	
	Sufficiency Rating	--	--	1

up the guaranteed situation and be willing to play the gamble? In other words, at what probability p would you make the switch from guaranteed situation to the gamble?

Step 3. Consider the two situations shown in Table C.5. How high would this probability p have to be in order for you to give up the guaranteed situation and be willing to play the gamble? In other words, at what probability p would you make the switch from guaranteed situation to the gamble?

C.4 Direct Rating Method

This questionnaire solicits information on your preferences for a performance criterion to develop the single-criterion value function using the direct rating method. We need to develop value functions to establish a common scale for comparison because the performance criteria are measured in different units. This method is useful for developing value functions of the vulnerability ratings because of fewer possible levels for these ratings.

A value function relates the possible levels of the performance criterion X to values scaled from 0 to 100, where 100 corresponds to the most desired level. Please assign a value to each of the intermediate levels of the following performance criterion.

Table C.3. Step 1a.

Criterion	Situation 1 (Guaranteed)	Situation 2 (Gamble)	
		with probability 0.01% (p)	with probability 99.99% ($1 - p$)
Deck Condition Rating	Best level	Best level	Worst level
Superstructure Condition Rating	Worst level	Best level	Worst level
Substructure Condition Rating	Worst level	Best level	Worst level

Table C.4. Step 2.

Criterion	Situation 1 (Guaranteed)	Situation 2 (Gamble)	
		with probability 0.01% (p)	with probability 99.99% ($1 - p$)
Deck Condition Rating	Worst level	Best level	Worst level
Superstructure Condition Rating	Best level	Best level	Worst level
Substructure Condition Rating	Worst level	Best level	Worst level

Table C.5. Step 3.

Criterion	Situation 1 (Guaranteed)	Situation 2 (Gamble)	
		with probability 0.01% (p)	with probability 99.99% ($1 - p$)
Deck Condition Rating	Worst level	Best level	Worst level
Superstructure Condition Rating	Worst level	Best level	Worst level
Substructure Condition Rating	Best level	Best level	Worst level

Table C.6. Scour vulnerability ratings.

Scour Vulnerability Rating	Value
1	0
2	
3	
4	
5	100

For the scour vulnerability rating, *SVR*, let $V(SVR = 1) = 0$ and $V(SVR = 5) = 100$, as shown in Table C.6.

C.5 Midvalue Splitting Technique

This questionnaire solicits information on your preferences for a performance criterion using the midvalue splitting technique. Consider an example of a value function shown in Figure C.1. A value function relates the possible levels of the performance criterion X to values scaled from 0 to 100, where 100 corresponds to the most desired level. The process is as follows:

1. Start with setting the extreme points: $V(X_{best}) = 100$ and $V(X_{worst}) = 0$.

2. Find X_{50} such that you would be equally delighted with
   – An improvement of performance from worst level to X_{50} and
   – An improvement of performance from X_{50} to best level.
3. Find X_{25} such that you would be equally delighted with
   – An improvement of performance from worst level to X_{25} and
   – An improvement of performance from X_{25} to X_{50}.
4. Find X_{75} such that you would be equally delighted with
   – An improvement of performance from X_{50} to X_{75} and
   – An improvement of performance from X_{75} to best level.
5. Do a consistency check. Would you be equally delighted with
   – An improvement of performance from X_{25} to X_{50} and
   – An improvement of performance from X_{50} to X_{75}?
   If yes, the values are consistent, please continue. If no, go to Step 2.

Example for Deck Condition Rating

Using the midvalue splitting technique, please assess the value function for this performance criterion. Please mark three points on the graph (Figure C.2).
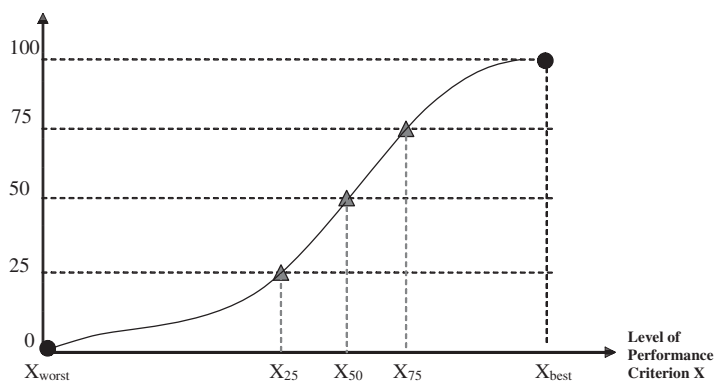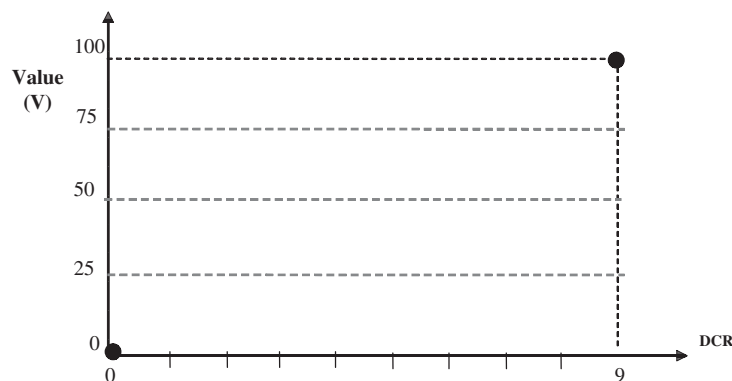

Figure C.1. Midvalue splitting technique.

Figure C.2. Graph for midvalue splitting technique (deck condition rating example).

Abbreviations and acronyms used without definitions in TRB publications:

AAAE	American Association of Airport Executives
AASHO	American Association of State Highway Officials
AASHTO	American Association of State Highway and Transportation Officials
ACI–NA	Airports Council International–North America
ACRP	Airport Cooperative Research Program
ADA	Americans with Disabilities Act
APTA	American Public Transportation Association
ASCE	American Society of Civil Engineers
ASME	American Society of Mechanical Engineers
ASTM	American Society for Testing and Materials
ATA	Air Transport Association
ATA	American Trucking Associations
CTAA	Community Transportation Association of America
CTBSSP	Commercial Truck and Bus Safety Synthesis Program
DHS	Department of Homeland Security
DOE	Department of Energy
EPA	Environmental Protection Agency
FAA	Federal Aviation Administration
FHWA	Federal Highway Administration
FMCSA	Federal Motor Carrier Safety Administration
FRA	Federal Railroad Administration
FTA	Federal Transit Administration
IEEE	Institute of Electrical and Electronics Engineers
ISTEA	Intermodal Surface Transportation Efficiency Act of 1991
ITE	Institute of Transportation Engineers
NASA	National Aeronautics and Space Administration
NASAO	National Association of State Aviation Officials
NCFRP	National Cooperative Freight Research Program
NCHRP	National Cooperative Highway Research Program
NHTSA	National Highway Traffic Safety Administration
NTSB	National Transportation Safety Board
SAE	Society of Automotive Engineers
SAFETEA-LU	Safe, Accountable, Flexible, Efficient Transportation Equity Act: A Legacy for Users (2005)
TCRP	Transit Cooperative Research Program
TEA-21	Transportation Equity Act for the 21st Century (1998)
TRB	Transportation Research Board
TSA	Transportation Security Administration
U.S.DOT	United States Department of Transportation