

Multi-Objective Optimization for Bridge Management Systems

Paul Thompson, Consultant (corresponding author)
258 Hardwick Ct, Castle Rock, CO 80108
303-681-2425 (phone); 303-265-9694 (fax); pdt@pdth.com

Kumares Sinha, Professor of Civil Engineering
Purdue University School of Civil Engineering
Civil Engineering Building, West Lafayette, IN 47907
765-494-2211 (phone); 765-496-1105 (fax); ksinha@purdue.edu

Vandana Patidar, Consultant
Jacobs Consultancy
555 Airport Boulevard, Suite 300, Burlingame, CA 94010
650-375-5354 (phone); 650-343-5220 (fax); vandana.patidar@jacobs-consultancy.com

Samuel Labi, Assistant Professor
Purdue University School of Civil Engineering
550 Stadium Mall Drive, West Lafayette IN, 47907
765-494-5926 (phone); 765-496-7996 (fax); labi@purdue.edu

Submitted on 31 March, 2008
Word count: 6,954

ABSTRACT

Optimizing investment funding levels and combinations of treatment types and timings, as an aid to management decision making, are vital functions of any bridge management system (BMS). Bridge managers and engineers are finding that their constituents require bridge projects and programs to perform not only as provided by least long-term cost solutions, but also to satisfy other objectives such as safety, minimum traffic flow disruption and risk. NCHRP Project 12-67, published as NCHRP Report 590, has developed a multiple objective optimization methodology and software to facilitate implementation of balanced decision-support practices at the network and bridge-levels.

Relative to existing bridge management system analytical frameworks, the new analysis includes several innovations: the use of utility theory and appropriate techniques of weighting, scaling, and amalgamation to combine multiple decision objectives; separation of bridge level, network-level, and program level components of the optimization so the engineer can work with each one separately yet maintain consistency in either a top-down or bottom-up management style; a framework for incorporating structure vulnerability and indirect costs in the analysis; the ability at the program level to apply both budget and performance constraints; and a digital dashboard presentation style to help engineers visualize the economic and performance tradeoffs.

The products of the study addressed many unmet needs in bridge management and opened the door to significant further research and development. Many aspects of the report are now being designed into AASHTO's Pontis 5.2 bridge management system.

OVERVIEW

Optimizing investment funding levels and combinations of treatment types and timings, as an aid to management decision making, are vital functions of any bridge management system (BMS). Bridge managers and engineers are finding that their constituents require bridge projects and programs to perform not only as provided by least long-term cost solutions, but also to satisfy other objectives such as safety, minimum traffic flow disruption and risk.

NCHRP Project 12-67, published as NCHRP Report 590, has developed a multiple objective optimization methodology and software to facilitate implementation of balanced decision-support practices at the network and bridge-levels.

The addition of multiple objective capability to a bridge management system, adds important realism and policy relevance. But it brings with it the potential for added complexity and data requirements, attributes that are not widely demanded in these systems. A challenge for the research team was to redefine the analytical framework of the system to break it into manageable pieces that better fit agency business processes and encourage a higher intensity of use of decision support features. Combined with a new generation of graphical user interface, the study products give an entirely new look and feel to the analysis, with the goal of complementing relevance with usability.

Inherent in multiple-criteria decision-making is the value tradeoff, i.e., the decision maker is faced with a problem of trading off the achievement of one criterion against another criterion due to conflicting objectives. In such a case, an important aspect of the decision making process is to be able to capture these value tradeoffs effectively. Multi-criteria methodologies were developed for the bridge decision making problem based on the concepts of value and utility theory. Mathematical strategies were advantageously used to handle multi-dimensionality of the problem in order to capture the decision makers' preference structures effectively in a practical manner.

The network level model aims to obtain the optimal selection of candidate projects from a network-wide candidate list to yield maximum network benefits subject to multiple constraints. The problem was formulated as a Multi-Choice Multi-Dimensional Knapsack problem. The most promising heuristics were explored and a strategy was developed that balanced theoretical precision with an appropriate level of practicality while selecting the solution heuristic for implementation in the final software product. This strategy was based on evaluating the heuristics in terms of optimality, computational speed, accuracy, robustness and simplicity. The computational experiments provided very useful insights into the optimization heuristics reflecting the appropriateness and applicability to the bridge program optimization problem. The Incremental Utility/Cost (IUC) heuristic performed very well both in terms of computational speed and optimality and was implemented in the software product.

The bridge level methodology includes a life cycle cost framework, preservation and functionality models, candidate definitions and their evaluation. In many ways this is similar to AASHTO's Pontis (2) approach, with similar models for network policy optimization, bridge element deterioration, and functional improvements. The methodology includes a bridge level optimization that maximizes utility by selecting from an array of scoping and timing alternatives. A recursive approach was selected that is consistent with input data available from existing bridge management systems, but that gives a deterministic result as is needed at the bridge level.

MULTIPLE OBJECTIVES

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 different 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

Selecting Performance Measures

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 (3):

- **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.

A large number of performance criteria can be derived from a bridge management database, including basic National Bridge Inventory items (4), Pontis analytical products (2), and new measures not currently accounted for in most bridge management systems, such as vulnerability to extreme events. The software product of the study provides a menu of 23 different performance measures, but many are completely or partially redundant with each other so the number of measures used in practice would be far less. The measures that have generated greatest interest among early users of the products are:

- Life cycle cost, as a single quantity or divided into agency and user costs;
- A condition measure, either health index (5) or some combination of National Bridge Inventory condition ratings; and
- A measure of vulnerability and/or risk.

It is the authors' conjecture based on their own experience, that a selection of just three measures in these categories would be considered adequate in a wide variety of applications.

Combining Performance Measures

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. When multiple criteria are considered together, it is often true that no dominant alternative will exist that is better than all other alternatives in terms of all of these criteria. For example, one cannot maximize service levels and at the same time minimize costs.

As a result, 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 tradeoffs 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:

- **Weighting** assigns relative weights to the multiple criteria, reflecting their relative importance.
- **Scaling** translates the decision makers' preferences for each performance criterion onto a 0-100 scale. This involves developing utility functions.

- Amalgamation combines the single criterion utility functions into one measure based on mathematical assumptions about the decision makers' preference structure. This involves deriving the functional forms of multi-criteria utility functions.

There are two general approaches for establishing utility function weights, distinguished by their context. One approach is to conduct a structured expert elicitation exercise that is intentionally removed from the context of projects and programs under consideration, to try to obtain more general insights into the relative importance of policy goals. In the research study several methods were evaluated and tested, some using the NCHRP Project Panel as the group of experts. The method preferred by the researchers and panel was a pairwise comparison method known as the Analytic Hierarchy Process (AHP, 6). This technique, which involves the construction of a hierarchy of "levels" and pair-wise comparisons at each level to estimate the relative weights, has been found to be capable of handling quantitative, qualitative, tangible and intangible criteria.

Another completely different approach is to experiment with the construction of real projects and programs in an optimization framework, adjusting the utility function weights iteratively to try to develop the most satisfactory program in terms of both project selection and forecast achievement of policy goals and objectives. This method was built into the study's software product. Both approaches are valid and can be used together: for example, the AHP can be used to generate an initial set of weights for the optimization model, weights that are later refined when evaluating actual programs of projects.

ANALYTICAL ARCHITECTURE

Much of the analytical structure of the study products was motivated by experience of the project panel and researchers in the use of AASHTO's Pontis bridge management system (2). Used by nearly all of the states, Pontis solidifies the *de facto* data requirements, concepts, and jargon of decision support for bridge management. Yet, it has been widely believed that the decision support features of Pontis are under-utilized, and that the perceived "black box" nature of the analysis might be a barrier to more widespread implementation.

Important progress on this issue was made by the Florida Department of Transportation when it developed a Project Level Analysis Tool (PLAT, 7) as an add-on to Pontis. The PLAT uses the Pontis database and predictive models, but presents the results from a bridge-level perspective as a set of scoping and timing alternatives for future work on the bridge. The probabilistic orientation of the network level was translated to a deterministic presentation of possible near-term futures for a bridge, using a graphical presentation to help engineers visualize current structural and economic health, and future possibilities.

Existence of the PLAT didn't compromise the network level perspective of bridge management at all. In fact, Florida later developed a companion tool for visualizing network level tradeoffs of performance vs. budget. But immediate interest in PLAT made it clear that an essential business process of bridge management was being missed by Pontis.

To better support project planning in a bridge management system, it would be necessary to decouple bridge level from program level analysis, and engineer the program level analysis so it can make full use of bridge level decisions. This architectural change would follow a pattern set by the Ontario Bridge Management System (8), where bridge level analytical products and decisions are stored in a database for later use by a program planning tool.

As a result, the analytical architecture of the NCHRP 12-67 study divides up the analysis as in Figure 1. The Analytical Database is a new feature separate from the bridge management system database, with the purpose of communicating bridge level results asynchronously to a separate network-level programming process.

BRIDGE-LEVEL ANALYSIS

At the bridge level, the analytical framework generates and evaluates a set of "candidates" which are alternative life cycle activity profiles each distinguished by a set of scope items and an implementation

year. The evaluation model estimates the initial cost and future costs and performance of each candidate. A utility function is then calculated from this information.

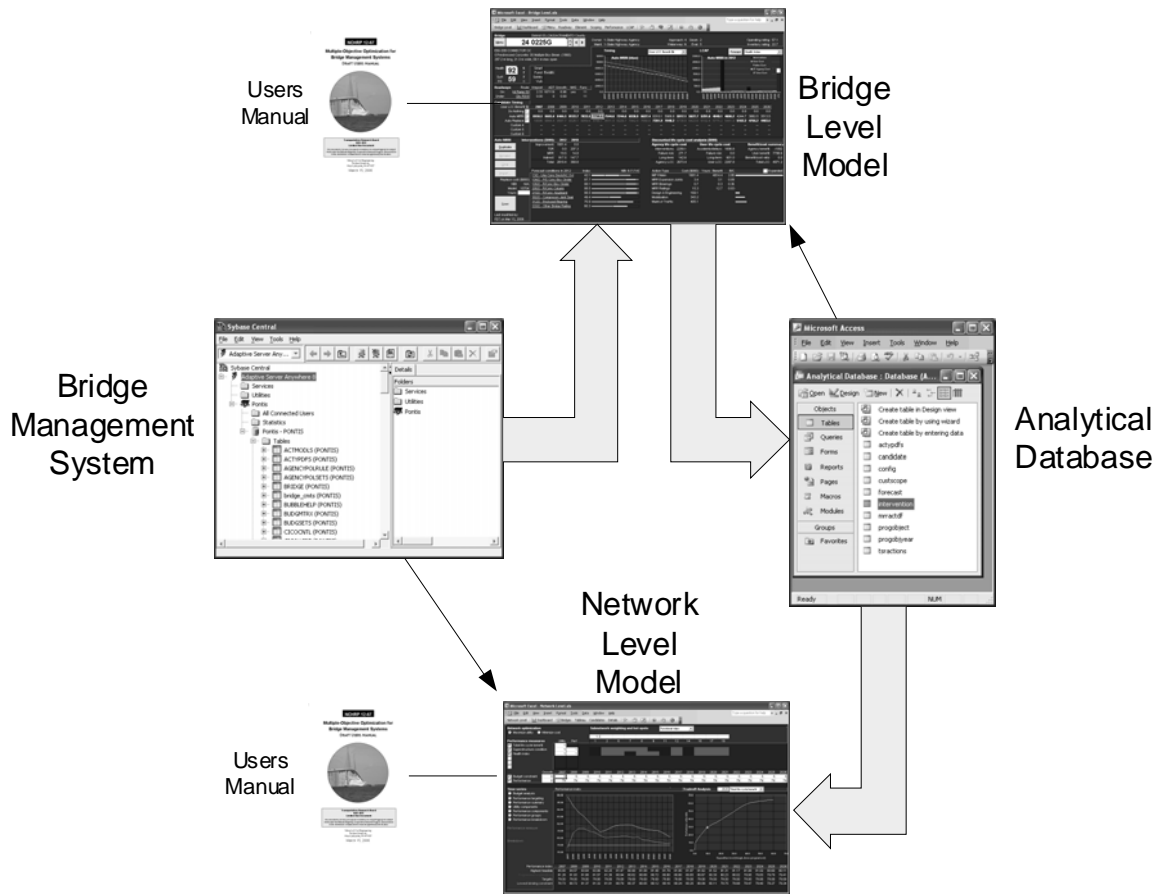


FIGURE 1 High-Level Architecture of the Report 590 Analysis Framework

Life Cycle Costs

Each life cycle activity profile is modeled as an infinite time series of annual cash flows representing various types of 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.

Figure 2 is a schematic depiction of a typical pattern of life cycle costs. The diagram shows an analysis conducted in 2007 with a view to developing a program to start in 2008. For the example candidate described in the diagram, the first intervention occurs in 2011, followed by a period of inaction until 2023. The significant time intervals and milestones are described in the following sections.

First waiting Period

From the start of 2008 to the start of 2011 is the initial waiting period, the time interval over which the bridge deteriorates gradually and its repair needs accumulate, prior to a first intervention. 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 occur.

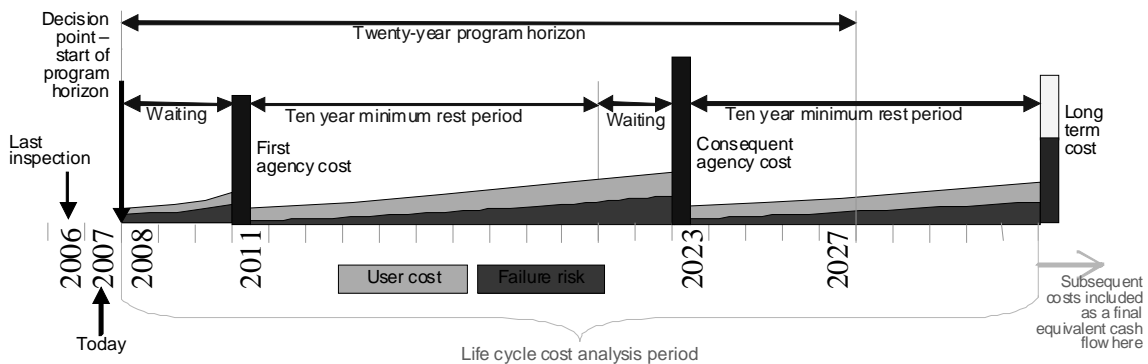


FIGURE 2 Example Life Cycle Activity Profile

First Intervention

At the start of 2011, the agency implements an intervention that may address some or all of the built up needs. With the intervention, the condition of elements is improved, and functional deficiencies and vulnerabilities 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. Also, any functional deficiencies that were not remedied by the first intervention would cause user costs to be incurred.

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, are 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 accordingly. No interventions may be programmed beyond the program horizon. As such, with the conventions adopted here, a waiting period may not extend beyond that point.

Long Term

The end of a life cycle activity profile, as modeled here, occurs at a point where the following year may either bring 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 a going concern in perpetuity.

Discounting

Since 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 (reduced in value) by an amount that depends on how far in the future it occurs. Naturally if a cost needs to be incurred, the analyst prefers to put it off as long as possible, because then it matters less to the analyst. The discount factor represents how much less it matters for each year that the cost can be delayed.

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 costs that occur far in the future.

Decision Variables

The most 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 whose duration 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. The following types of first interventions are analyzed:

- Do nothing – models the costs and performance of the structure if no programmed action is taken on the bridge any time during the analysis period.
- Auto MRR&I – an intervention that includes all cost-effective maintenance, repair, rehabilitation, and functional improvement actions, as indicated at the time of intervention.
- Auto Replace – an intervention that replaces the structure with a new, larger structure.
- Custom – up to three interventions whose scope is determined by the engineer.

To make the scope of work of each intervention easy to understand and analyze, the model framework defines a concept of scope items. Each scope item is a general class of work, such as painting or concrete repair, applied to all elements on the structure to which the work is applicable. For example, a scope item of “total paint system replacement” involves recoating all painted steel elements on the structure.

Optimization

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 then must decide what kind of intervention to undertake. Naturally, if the waiting period is extended, deterioration will continue and it is likely that the scope and cost of the first intervention will increase. Intervention scope and timing are therefore inter-related. If the program horizon is long enough, the decision becomes a multi-stage process. After the first intervention and subsequent rest period, the need may arise for another intervention. The scope and timing of the second intervention are conditional on the decisions made in the first intervention. So we end up with a tree structure of decisions, having potentially a very large number of branches.

The familiar tool of recursion provides a means to divide the problem into smaller parts that are easier to analyze and require less computation. The algorithm exploits the fact that each stage of the decision tree is similar to earlier stages. Setting aside rest periods where the only prescribed 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 scopes (types) of interventions to be considered: do-nothing, rehabilitation, or replacement (for the first intervention, custom candidates may also be considered). Do-nothing in this case means merely putting off consideration of any intervention to the next programming period (which is typically one year).
- The choice of intervention scope in every programming period 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 3. The optimization works in either or both of two modes:

- Optimizing, where all program years are investigated and the one with highest utility is chosen;
- or

- Worst tolerable performance (WTP), where an intervention is considered only in the first year that 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 fails, but may select a candidate in an earlier year if the latter has higher utility. The WTP mode requires an intervention to be taken even if no candidate has a positive benefit (that is, a utility greater than that of the do-nothing candidate).

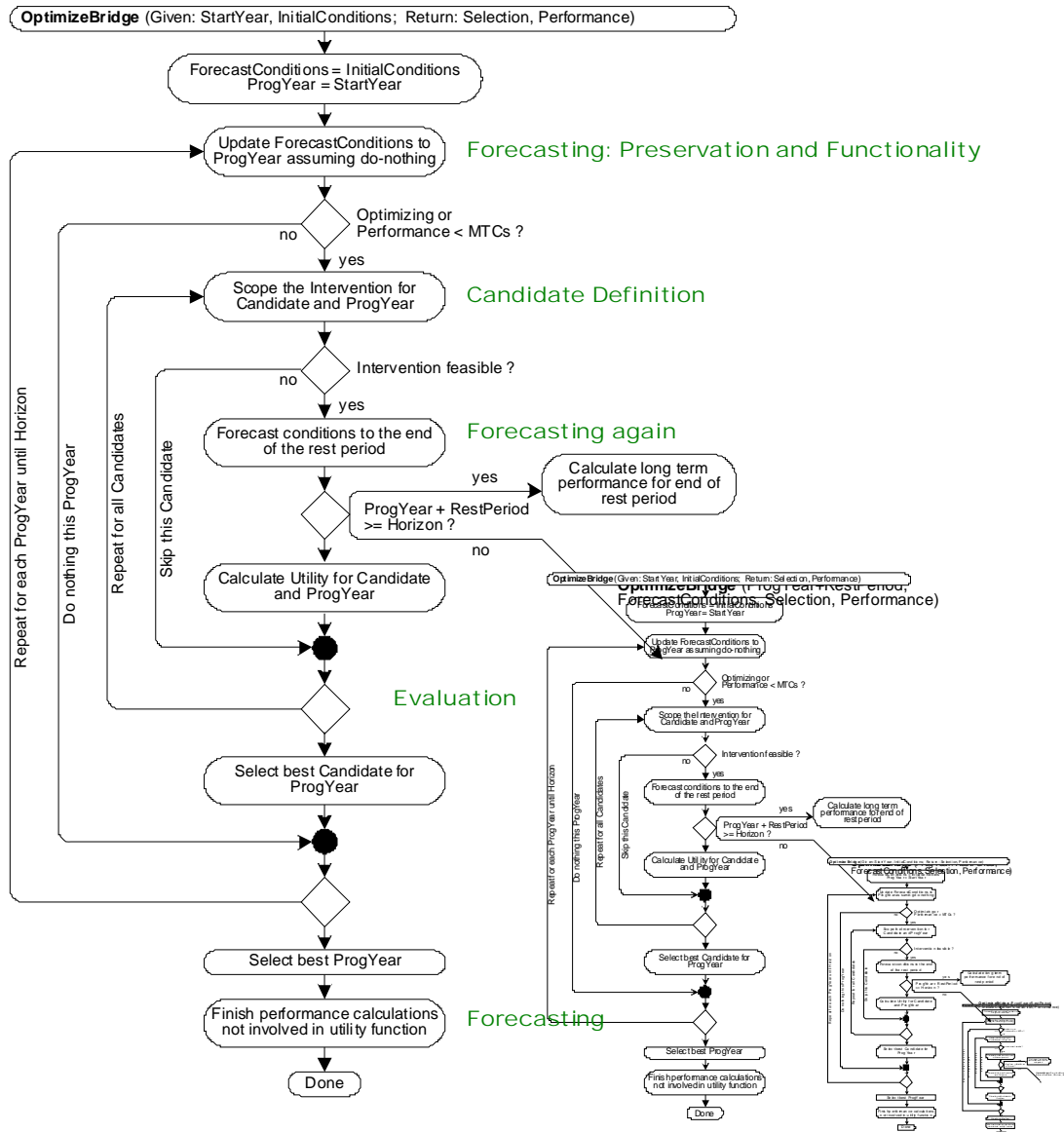


FIGURE 3. Recursive Bridge Level Optimization

PROGRAM-LEVEL ANALYSIS

The bridge level optimization produces an evaluated set of candidates which are stored in a database to be made available for programming. Each candidate has a cost and a vector of performance measures. At the program level, the challenge is to select a candidate from each bridge, in such a way that network-wide utility is maximized, and all cost and performance constraints are satisfied if possible.

This challenge can be recognized as a special case of integer programming problems known as the Knapsack problem. 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 intractable for full-size databases (9).

One of the oldest and most common applications of the Knapsack problem is the capital budgeting problem (10). The large domain of its applications also include cutting stock, investment policy for the tourism sector, allocation of databases and processors in a distributed data processing network, delivery of groceries in multi-compartment vehicles, multi-commodity network optimization and daily management of a remote sensing satellite (9). More specifically the program optimization problem is a Multi-Choice Multi-Dimensional Knapsack (MCMDKP) problem.

Accuracy vs. Responsiveness

The MCMDKP problem is considered NP-hard in 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 one minute on a test data set of 1,000 bridges might take roughly three days on an average state's inventory of 12,000 bridges and roughly 45 days on Texas' inventory of 50,000 bridges. In contrast, a polynomial algorithm's times for the same problem might be one 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 this. If the solution method could be reduced to simple sorting, it would have " $n(\log n)$ " execution time that would turn one minute for 1,000 bridges into 13 minutes for 12,000 bridges and 31 minutes for 50,000 bridges. Obviously this faster behavior is preferred since it allows the user of the system to investigate many program alternatives.

There are two classes of methods that exist to solve this problem: exact methods and heuristics. Exact methods are guaranteed to arrive at the optimal solution but are typically associated with lower computational speeds. On the other hand, heuristic methods strive to achieve "good" approximate (near optimal) solutions quickly. So there is a tradeoff between the accuracy and computational speed of the solution methods. The largest network we are dealing with is of 50,000 bridges. Each bridge could have as many as 5 possible interventions including the do-nothing alternative. 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, so approximate heuristic methods are very appropriate. The performance of a solution method can be evaluated based on:

- Accuracy, i.e. how close is the solution to the true optimal solution?
- Computational speed, i.e. how long does it take to solve?
- Robustness of the method, i.e. how sensitive is its performance to variation in inputs?
- 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 type of method (theoretical computational complexity), coding language, coding efficiency, specific problem instances (realistic data sets), network parameters and computer configuration.

Solution Method

Several heuristic methods were evaluated and tested to determine which one would give the optimal balance of accuracy and responsiveness for a program level bridge management optimization, with acceptable robustness and ease of use. The algorithms selected for detailed evaluation were the incremental utility/cost (IUC) heuristic, the Lagrangian relaxation method (11), and the pivot-and-complement method (12).

These were tested on a set of example bridge management problems, ranging from 100 bridges to 50,000 bridges, generated by the Florida PLAT (7). For evaluation of accuracy, the problems were also solved to exact solutions using an industrial-strength integer programming software system, ILOG CPLEX (13). In the largest cases with CPLEX and the Lagrangian relaxation technique, memory requirements or execution time prevented complete testing.

The IUC method proved to be fastest, and very close to the most accurate. It is based on the classic linear relaxation of the MCMDKP problem, which is much more easily solved than the integer program by using a “greedy algorithm.” Investment alternatives are arranged in decreasing order of reward to cost ratios. The algorithm simply scans down the list to fill the knapsack. Since the algorithm is based on sorting, it has $n \log n$ performance.

The IUC heuristic can be generalized to incorporate multiple constraints if we recognize an order of priority among the constraints. The strategy is to divide it into two separate single-constraint knapsack problems that are solved in tandem (Figure 4):

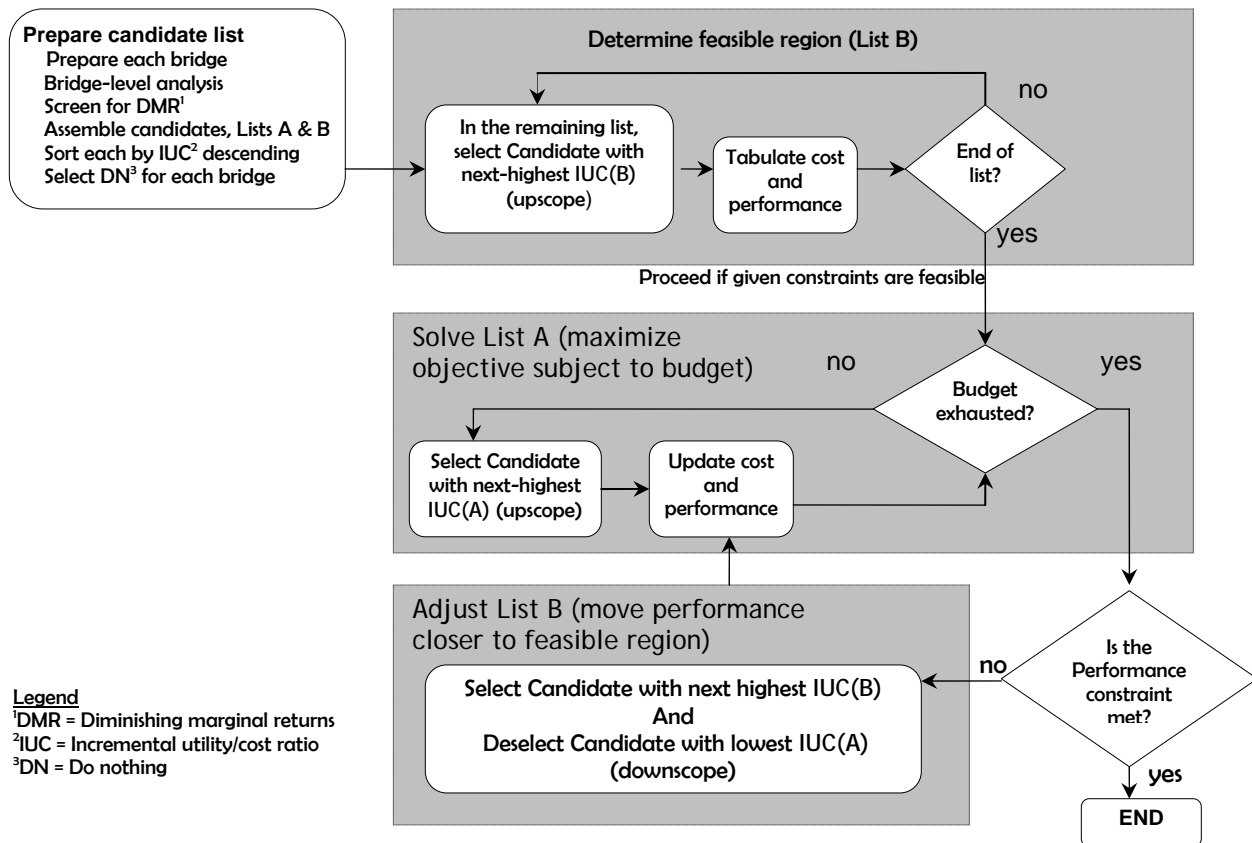


FIGURE 4 Flowchart for the IUC Heuristic to Solve the Multi-Constraint Knapsack Problem

List A – Simple budget-constrained problem: $IUC(A) = \frac{\Delta Utility}{\Delta Cost}$

List B – Feasible region problem: $IUC(B) = \frac{\Delta Performance}{\Delta Cost}$

In the figure, List B describes the tradeoff between the two performance criteria that present the constraints: performance and cost. The list represents an ordinary MCKP problem that maximizes performance subject to a budget constraint and can be solved using the ordinary IUC algorithm. The numerator of $IUC(B)$ is the utility function that comprises multiple performance criteria. List A is the underlying optimization problem that has only the budget constraint. In the event that there is no feasible solution for the given combination of budgetary and performance constraints, such lack of a solution is detected in List B before proceeding into the main iteration of the procedure.

Typically, the first pass through the List A optimization does not satisfy the performance constraint – in that case List B is used to identify the candidate most responsible for the constraint violation. Then a second pass is made through List A. The process is repeated until both constraints are satisfied. If the performance constraint is not binding, the optimal solution will be one of the candidate combinations found in List A but would not necessarily be visited by the IUC algorithm on List B.

DIGITAL DASHBOARDS

The software products of the study include two visualization tools called Digital Dashboards: one for bridge level analysis and one for network level.

The bridge level dashboard (Figure 5) presents the current and forecast condition of a bridge’s elements, and any functional or structural deficiencies that might exist. It presents the possible futures of a bridge as a table and graph of scope and timing alternatives. The graphs make it easy to see how changes in scope or timing might affect the future performance of the structure. Tools are included to help the engineer develop custom scope items and evaluate their effect on future performance.

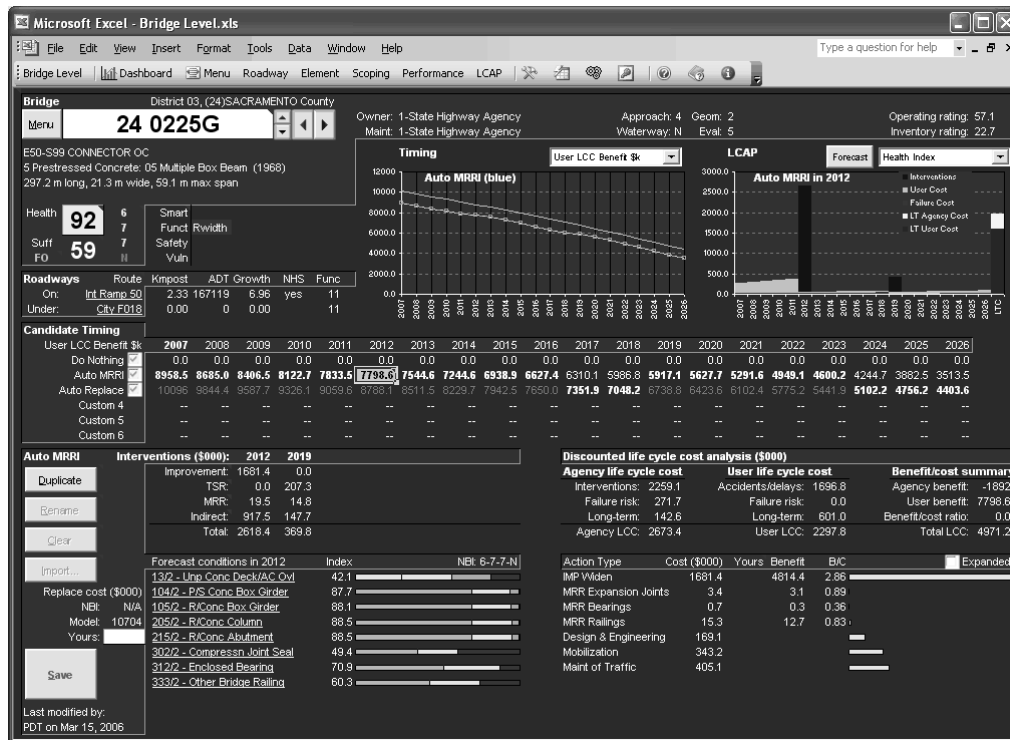


FIGURE 5 Bridge Level Dashboard

For network level analysis, the network dashboard (Figure 6) presents the key tradeoffs between performance and funding. The program manager can specify annual budgets, and the weight given to different performance measures. The graphics show how these settings affect future costs and performance. The optimization model can find the best performance achievable for any given funding level, or the funding level required to achieve any given level of performance.

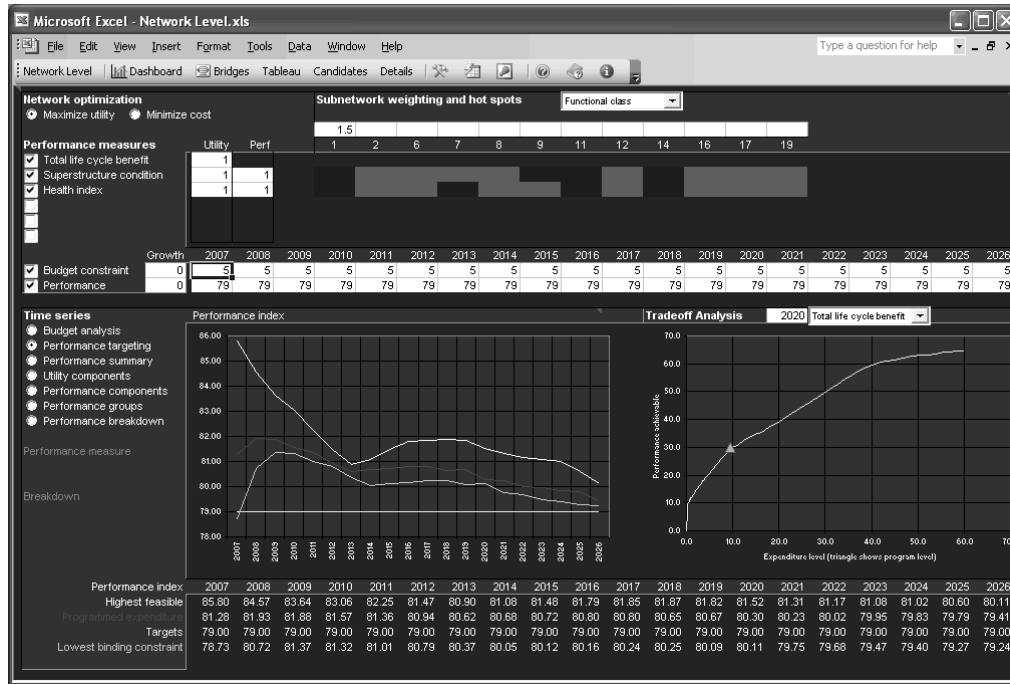


FIGURE 6 Network Level Dashboard

CONCLUSIONS AND NEXT STEPS

The study presented a novel new combination of methods that together make up a new state-of-the-art bridge management analytical framework. The various analytical tools were shown to be compatible with each other, to make effective use of readily-available data, and to be usable within a user-friendly decision support system. They accomplished the project objectives of incorporating multiple objectives effectively while increasing the usefulness of bridge management systems.

Conclusions in the Final Report of the study identified a few additional data requirements, most notably a new system for assessing the vulnerability of structures to natural and man-made hazards. The research found certain deficiencies in commonly-used methods for forecasting bridge deterioration, and for translating those methods to National Bridge Inventory condition ratings. Future research will be necessary to correct these deficiencies.

Major new analytical and presentation methods described in this paper have encountered widespread interest since completion of the project. The AASHTO BRIDGEware Task Force, which oversees development of Pontis, has decided to implement most of them in release 5.2 of that product, scheduled for completion in 2009.

ACKNOWLEDGEMENTS

The authors are grateful to the Transportation Research Board of the National Academies, for funding for this effort, and to the NCHRP Project Panel for their feedback and assistance.

The opinions and conclusions expressed or implied are those of the authors, and not necessarily those of the Transportation Research Board or the National Academies.

REFERENCES

1. Patidar, V., S. Labi, K. Sinha, and P.D. Thompson. *Multi-Objective Optimization for Bridge Management Systems*. National Cooperative Highway Research Program Report 590. Transportation Research Board of the National Academies, Washington DC, 2007.
2. Cambridge Systematics, Inc. *Pontis Technical Manual*. Release 4.0. Washington, DC, American Association of State Highway and Transportation Officials, 2001.
3. Keeney, R.L., and Raiffa, H. *Decision with Multiple Objectives – Preferences and Value Tradeoffs*, Wiley, 1976.
4. FHWA. *Recording and Coding Guide for the Structure Inventory and Appraisal of the Nation's Bridges*. U.S. Department of Transportation, Federal Highway Administration, 1995.
5. Shepard, R.W. and M.B. Johnson. California Bridge Health Index: A Diagnostic Tool to Maximize Bridge Longevity, Investment, *Transportation Research News*, pp. 6-11. Transportation Research Board of the National Academies. Washington. D.C., 2001.
6. Saaty, T.L. 1980. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw Hill, Hightstown, NJ.
7. Sobanjo, J.O. and P.D. Thompson, *Project Planning Models for Florida's Bridge Management System: Final Report*, Florida Department of Transportation, available at www.pdth.com/images/projplan.pdf, 2004.
8. Thompson, P., Merlo, T., Kerr, B., Cheetham, A. and Ellis, R. The New Ontario Bridge Management System. *8th International Bridge Management Conference*, 1999.
9. Freville, A. The multidimensional 0-1 Knapsack Problem: An Overview. *European Journal of Operational Research*, Vol. 155, pp. 1-21, 2004.
10. Lorie, J.H., and Savage, L.J. Three Problems in Rationing Capital. *Journal of Business*, XXVIII, No. 4, pp. 229-239. The University of Chicago Press, Illinois, 1955.
11. Magazine, M. J. and Oguz, O. A Heuristic Algorithm for the Multidimensional Zero-One Knapsack Problem. *European Journal of Operational Research*, Vol. 16, pp. 319-326, 1984.
12. Balas, E. and Martin, C. H. Pivot and Complement – A Heuristic for 0-1 Programming. *Management Science*, Vol. 26, No. 1, pp. 86-96, 1980.
13. ILOG CPLEX, <http://www.ilog.com/products/cplex/product/mip.cfm>.