Optimal Highway Safety Improvement Investments by Dynamic Programming
[Aug. 1974]

Jerry G. Pigman∗  Kenneth R. Agent†
Jesse G. Mayes‡  Charles V. Zegeer**

∗Kentucky Department of Highways, jerry.pigman@uky.edu
†Kentucky Department of Highways, ken.agent@uky.edu
‡Kentucky Department of Highways
**Kentucky Department of Highways
This paper is posted at UKnowledge.
https://uknowledge.uky.edu/ktc_researchreports/901
August 28, 1974

MEMORANDUM TO: J. R. Harbison
State Highway Engineer
Chairman, Research Committee


The impersonal and objective aspects of many decision-making processes consist of orderly quantification of situational descriptors, alternatives, and consequences and a ranking of the consequences. The selection or choice of descriptors remains somewhat subjective or intuitive; unnecessary factors null-out in the quantification process. The principal difficulties arise in quantifying the factors.

The report enclosed herewith demonstrates a specific application; more extensive applications are implied.

Respectfully submitted,

Jas. H. Havens
Director of Research

JHH:gd
Attachment
cc's: Research Committee
OPTIMAL HIGHWAY SAFETY IMPROVEMENT INVESTMENTS
BY DYNAMIC PROGRAMMING

by
J. G. Pigman, K. R. Agent, J. G. Mayes, and C. V. Zegeer

ABSTRACT

The process of determining which projects to implement under a given budget, and which to defer until later, is central to the planning and management of highway systems. With a limited budget for construction, maintenance, and safety improvements, investments which will produce the optimal benefits must be chosen. This is often impossible to accomplish without the aid of a computer because of the complexity of the problem. Dynamic programming has been tested and verified as an efficient method for selecting priority projects to derive maximum benefits.

There are several approaches to priority programming as it is related to the capital allocation problem. Benefit-cost, present worth, and rate-of-return calculations have traditionally been used as an integral part of the transportation planning process. Construction and maintenance programs continually face the task of having to assign priorities when insufficient funds are available to complete all projects. Safety improvement programs, which were initially funded through the Highway Safety Act of 1966 and expanded through the Federal-Aid Highway Act of 1973, have become so large that they are unmanageable without a clear and concise means of priority allocation.

A dynamic programming procedure was developed in this study which selects the optimal combination of safety improvement projects for a given budget. Sixty-one projects, each with one or more alternatives, were evaluated. The input consisted of the designated budget for the safety improvement program, the improvement cost, and the benefits derived from each improvement. The accuracy and reliability of dynamic programming is dependent upon the accuracy of benefits and costs used as input.

In a comparison with benefit-cost analyses, it was shown that dynamic programming can yield a higher return for a given budget. An optimal allocation of funds will always be obtained if the individual project costs are multiples of the increment used in dynamic programming.

Applicability of dynamic programming to budget allocation in transportation planning is practically unlimited. In addition to the various highway programs, dynamic programming can be used to optimize investments for entire transportation departments.
OPTIMAL HIGHWAY SAFETY IMPROVEMENT INVESTMENTS
BY DYNAMIC PROGRAMMING

by
J. G. Pigman, K. R. Agent, J. G. Mayes, and C. V. Zegeer

ABSTRACT

The process of determining which projects to implement under a given budget, and which to defer until later, is central to the planning and management of highway systems. With a limited budget for construction, maintenance, and safety improvements, investments which will produce the optimal benefits must be chosen. This is often impossible to accomplish without the aid of a computer because of the complexity of the problem. Dynamic programming has been tested and verified as an efficient method for selecting priority projects to derive maximum benefits.

There are several approaches to priority programming as it is related to the capital allocation problem. Benefit-cost, present worth, and rate-of-return calculations have traditionally been used as an integral part of the transportation planning process. Construction and maintenance programs continually face the task of having to assign priorities when insufficient funds are available to complete all projects. Safety improvement programs, which were initially funded through the Highway Safety Act of 1966 and expanded through the Federal-Aid Highway Act of 1973, have become so large that they are unmanageable without a clear and concise means of priority allocation.

A dynamic programming procedure was developed in this study which selects the optimal combination of safety improvement projects for a given budget. Sixty-one projects, each with one or more alternatives, were evaluated. The input consisted of the designated budget for the safety improvement program, the improvement cost, and the benefits derived from each improvement. The accuracy and reliability of dynamic programming is dependent upon the accuracy of benefits and costs used as input.

In a comparison with benefit-cost analyses, it was shown that dynamic programming can yield a higher return for a given budget. An optimal allocation of funds will always be obtained if the individual project costs are multiples of the increment used in dynamic programming.

Applicability of dynamic programming to budget allocation in transportation planning is practically unlimited. In addition to the various highway programs, dynamic programming can be used to optimize investments for entire transportation departments.
OPTIMAL HIGHWAY SAFETY IMPROVEMENT INVESTMENTS
BY DYNAMIC PROGRAMMING

KYP-73-47; HPR-PL-1(10), Part III

by

J. G. Pigman
Research Engineer Senior

K. R. Agent
Research Engineer

J. G. Mayes
Research Engineer

C. V. Zegeer
Research Engineer Associate

Division of Research
Bureau of Highways
DEPARTMENT OF TRANSPORTATION
Commonwealth of Kentucky

The contents of this report reflect the views
of the authors who are responsible for the
facts and the accuracy of the data presented
herein. The contents do not reflect the official
views or policies of the Kentucky Bureau
of Highways. This report does not constitute
a standard, specification, or regulation.

August 1974
INTRODUCTION

The process of determining which projects to implement under a given budget, and which to defer until later, is central to the planning and management of the highway system. With a limited budget for construction, maintenance, and safety improvements, investments which will produce the optimal benefits must be chosen. This is often impossible to accomplish without the aid of a computer because of the complexity of the problem. Dynamic programming has been tested and verified by others as an efficient method for selecting priority projects to derive maximum benefits.

Dynamic programming is an optimization technique which transforms a multistage decision problem into a series of one-stage decision problems. The decision at each stage depends on the input to that stage, the feasible set of decisions at that stage, and the conditional set of decisions from the preceding stages.

There are three main reasons why dynamic programming is needed for transportation planning. First, dynamic programming is designed to provide the best plan over a period of time inasmuch as the scheduling of a project is a critical variable. Secondly, dynamic programming makes it possible to obtain the best combination of projects where some approaches are inaccurate and trial and error methods can become an impossible task. Thirdly, dynamic programming can determine the optimal investment plan when the usual benefit-cost, present worth, or maximum rate of return approaches are not practical. When the amount of money required for a single project is a large portion of the budget, the best set of projects does not necessarily consist of those which would be chosen by the conventional means of priority selection. Benefit-cost and rate of return methods may not provide the best overall use of resources because an efficient implementation of results may not be possible. In addition, the benefit-cost method of selecting optimal alternatives does not always produce the best results because it focuses narrowly on immediate benefits and often precludes some future combinations of alternatives which are more desirable.

Many programs do not require detailed knowledge of the mechanics of dynamic programming. The input consists only of the costs and benefits anticipated for any project along with the time required for completion. Dynamic programming, by taking all possible combinations into account, avoids the possibility of missing an optimal plan which will guarantee the best economic investment.

There are several approaches to priority programming as it is related to the capital allocation problem. Benefit-cost, present worth, and rate of return calculations have traditionally been used as an integral part of the transportation planning process. Performance budgeting has been proposed as a means of highway maintenance management (1). Construction and maintenance programs continually face the task
of having to assign priorities when insufficient funds are available to complete all projects. Safety improvement programs, which were initially funded through the Highway Safety Act of 1966 and expanded through the Federal-Aid Highway Act of 1973, have become so large that they are unmanageable without a clear and concise means of priority allocation. Possibly the most comprehensive and accurate method of cost allocation for a constrained budget is dynamic programming. The term was coined by Bellman (2) in an attempt to simplify the phrase definition previously used – mathematical theory of multistage decision processes. He has summarized dynamic programming applicability into three types of projects: single-stage, multistage, and multistage incorporating a time factor.

Single-stage dynamic programming is the evaluation of a single project with several alternatives as compared to multistage where several projects with several alternatives are evaluated. Multistage with a time factor involves the allocation of funds by dynamic programming where several projects with several alternatives are subject to implementation over a period of time.

Johnson, Dare, and Skinner (3) presented dynamic programming as a means of selecting highway improvement projects to eliminate hazardous locations and therefore maximize the annual cost reduction benefit. They suggested an optimal solution is assured when several projects are being considered and construction funds are limited. De Neufville and Mori (4) have dealt with a simplified procedure for determining the optimal construction schedule for additions over time to a highway or similar transportation network. Only costs and benefits for each project are required as input to determine the optimum schedule. Funk and Tillman (5) used the systems approach to emphasize that the costs and benefits occurring to all parts of the system must be evaluated to establish the effect upon a specific route under consideration. Dynamic programming was used to analyze the entire system such that optimal stages of construction were implemented.

Jorgensen (6) has done extensive work in the identification of high-accident locations and the development of methods for selecting improvements from among various projects. Benefit-cost, present worth, or rate of return calculations were recommended by Jorgensen as methods for determining which project yields the maximum difference between the annual investment cost and the annual expected safety benefit. Determining priorities with these methods is restrictive because they will not assure the optimal combination of projects when operating with a limited budget. Lorie and Savage (7) have shown that, under a constrained budget, the selection of a large initial cost project with a high ratio of present worth to cost may preclude the selection of several smaller projects which together yield a greater present worth. Another disadvantage is the inability of previously used methods to evaluate the relative merit of competing alternatives at varying investment levels.
Previous studies have dealt with Kentucky highway budgeting (8, 9). Agent (10) evaluated the high-accident location spot-improvement program in Kentucky and it was determined that the small investment in the program had returned significant dividends. It was felt that further study was warranted and Zegeer (11) recently completed an investigation of the various methods for selecting high-accident locations. Favorable results from the studies by Agent and Zegeer, combined with an expansion of the spot-improvement program as a result of appropriations through the Federal-Aid Highway Act of 1973, have stimulated the development of an optimal method for allocating funds within the safety improvement program. Dynamic programming, as an optimal investment plan with a constrained budget, is presented here in a rather simplified but effective form for the particular problem.

The State of Alabama Highway Department has done considerable work in the application of dynamic programming to the optimization of budget allocation for the spot safety improvement program (12). Significant modifications have been incorporated into the Alabama program to evaluate the data which were available for the spot-improvement program in Kentucky. The authors wish to acknowledge the cooperation of the Alabama Highway Department in providing information used to determine the applicability of dynamic programming to the spot safety improvement program.

PROCEDURE

The problem of optimum utilization of improvement funds can be divided into two distinct steps. First, the benefits associated with each proposed improvement must be determined. Second, given the costs and benefits for a set of improvements and given a specific budget, the optimum combination of improvements to be implemented must be chosen. The computer program presented in APPENDIX A was used to calculate the costs and benefits in the subroutine COSBEN. These results are printed out and passed into the subroutine DYNAM along with the budget and output information. DYNAM then determines and prints out the optimum combination of improvements for the desired budgets. If no alternative emerges at a particular location, alternative "O" is printed. A range of budgets including the maximum budget available are considered. In this manner, an optimum budget may be determined. A list of variables and flow chart for the computer program are presented in APPENDIX B and APPENDIX C, respectively. Coding instructions are presented in APPENDIX D, and APPENDIX E contains sample program input and output.

Calculation of Costs and Benefits Using the Present Worth Method

The following equations were used to calculate costs and benefits (13):

\[ C = S + A[(1 - i)^L - 1]/i(1 - i)^L \]  

(1)
where \( C \) = present worth cost of improvement, 
\( S \) = construction cost, 
\( A \) = yearly maintenance cost, 
\( i \) = present interest rate = 10 percent, and 
\( L \) = life of improvement.

\[
B = \frac{\left[\left\{\left[(1 + t)^{(L + 1)}/(1 + i)\right] - 1\right\}/\left[\left((1 + t)/(1 + i)\right) - 1\right]\right]\beta}{1 - \frac{1}{1 + i}}
\]

(2)

where \( B \) = present worth benefit, 
\( t \) = exponential growth rate factor for traffic volume = 4 percent, and

\[
\beta = \left( \sum_{m=1}^{J} \sum_{n=1}^{J} a_{mn} N_{mn} \gamma_{n} \right) / T
\]

(3)

where \( \beta \) = benefit per year associated with the improvement, 
\( T \) = time (years) of accident history, 
\( J \) = number of accident causes associated with the location, 
\( a_{mn} \) = percent reduction of \( m \)-th cause affected by the improvement, 
\( N_{mn} \) = number of accidents associated with \( m \)-th cause, and 
\( \gamma_{n} \) = average cost of an accident: 
\( n = 1 \) fatality, 
\( n = 2 \) nonfatal injury, and 
\( n = 3 \) property damage only.

**Dynamic Programming Algorithm**

**STEP 1** Divide budget into \( N \) equal intervals.

**STEP 2** (STAGE 1) Determine the best alternative at Location 1 to maximize the return using \( j \) increments, \( j = 1, 2, ..., N \); i.e.,

\[
0_{1}(j) = R_{1}(j)
\]

(4)

where \( 0_{1}(j) \) = total optimum return after STAGE 1 for an investment of \( j \) increments, 
\( R_{1}(j) \) = return from Location 1 for an investment of \( j \) increments, and
STEP 3 (STAGE 2 through STAGE M) Repeat STEP 2 for each STAGE.

\[ O_i(j) = \text{chosen alternative for Location } i \text{ for an investment of } j \text{ increments.} \]

\[ D_j(i) = \text{chosen alternative for Location } 1 \text{ for an investment of } j \text{ increments.} \]

\[ 0_i(j) = \text{total optimum return after STAGE } i \text{ for an investment of } j \text{ increments,} \]

\[ R_i(k) = \text{return from Location } i \text{ for an investment of } k \text{ increments (} k \leq j \text{),} \]

\[ 0_i - 1(j - k) = \text{total optimum return after STAGE } (i - 1) \text{ for an investment of } (j - k) \text{ increments, and} \]

\[ M = \text{number of locations considered,} \]

\[ j = 1, 2, \ldots, N \text{ and} \]

\[ k = 1, 2, \ldots, j \]

STEP 4 The optimum alternative at each location can now be obtained by determining the best alternative for Location M at STAGE M with N increments. The remaining increments can now be used at STAGE (M - 1), etc. Therefore,

\[ A_M = D_M(N), \text{ leaving } N_M \text{ increments,} \]

\[ A_{M-1} = D_{M-1}(N_M), \text{ leaving } N_{M-1} \text{ increments,} \]

\[ A_{M-2} = D_{M-2}(N_{M-1}), \text{ leaving } N_{M-2} \text{ increments, etc.} \]

\[ \vdots \]

\[ A_i = D_i(N_i + 1) \]

where \( A_i \) = alternative chosen at the i-th location.

Development of Benefit and Cost Values

Some of the major inputs into the dynamic programming model are the benefits assigned to each improvement at a location. For example, upgrading a traffic signal at an intersection will affect accident patterns differently than will installing channelization. To quantify the effect of various improvements on accidents, approximately 300 spot locations improved in Kentucky since 1968 were studied to determine the accident reduction (or increase) associated with each at various location types.

Various improvements on curves, intersections, and other (general) locations are given in Table 1 along with corresponding number of projects included, total accident reduction, service life of
<table>
<thead>
<tr>
<th>TYPE OF IMPROVEMENT</th>
<th>NUMBER OF PROJECTS</th>
<th>TOTAL ACCIDENT REDUCTION (PERCENT)</th>
<th>SERVICE LIFE (YEARS)</th>
<th>ANNUAL MAINTENANCE COST ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signs and Markings</td>
<td>9</td>
<td>36</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>Warning Signs</td>
<td>23</td>
<td>35</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Regulatory Signs</td>
<td>16</td>
<td>22</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Guidance Signs</td>
<td>10</td>
<td>14</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Sign Combinations</td>
<td>16</td>
<td>20</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Markings</td>
<td>8</td>
<td>16</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Sight Distance Imp.</td>
<td>9</td>
<td>28</td>
<td>2</td>
<td>50</td>
</tr>
<tr>
<td>Post Delineators</td>
<td>3</td>
<td>25</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>Comb. Delineators, Markings, Signs, Maintenance</td>
<td>11</td>
<td>22</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Shoulder Improvements</td>
<td>7</td>
<td>23</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Comb. Resurface, Patching, Drainage, Deslick, Culvert</td>
<td>22</td>
<td>16</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Rumble Strips</td>
<td>8</td>
<td>29</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Remove Median Crossovers</td>
<td>2</td>
<td>29</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Lighting</td>
<td>1</td>
<td>-58</td>
<td>10</td>
<td>500</td>
</tr>
<tr>
<td>Lighting &amp; Rumble Strips</td>
<td>1</td>
<td>17</td>
<td>7</td>
<td>300</td>
</tr>
<tr>
<td>Rumble Strips &amp; Beacon</td>
<td>2</td>
<td>32</td>
<td>7</td>
<td>50</td>
</tr>
<tr>
<td>Side Road Sign Only</td>
<td>31</td>
<td>19</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Prepare for Sudden Stop Sign Only</td>
<td>19</td>
<td>25</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Side Road Sign &amp;Warning Sign</td>
<td>15</td>
<td>27</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Signing</td>
<td>34</td>
<td>30</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Post Delineators</td>
<td>4</td>
<td>32</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Signs &amp; Delineators</td>
<td>16</td>
<td>28</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Signs &amp; Maintenance</td>
<td>6</td>
<td>47</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>Comb. Delineators, Markings, Signs, Maintenance</td>
<td>16</td>
<td>24</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Resurfacing, Patch, Drainage, Deslick, Super, Culvert, Guardrail</td>
<td>22</td>
<td>33</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Re-alignment (Relocate)</td>
<td>3</td>
<td>32</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>Signs &amp; Markings</td>
<td>21</td>
<td>24</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>Warning Signs</td>
<td>11</td>
<td>27</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Regulatory Signs</td>
<td>5</td>
<td>48</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Regulatory &amp; Warning Signs</td>
<td>20</td>
<td>16</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Markings</td>
<td>17</td>
<td>16</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Marking, Maintenance, &amp; Signing</td>
<td>9</td>
<td>35</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>Channelization - Storage Lane</td>
<td>13</td>
<td>15</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Channelization &amp; Signs</td>
<td>2</td>
<td>37</td>
<td>7</td>
<td>75</td>
</tr>
<tr>
<td>Install Beacons</td>
<td>13</td>
<td>2</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Upgrade Beacons</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Install Signals</td>
<td>10</td>
<td>23</td>
<td>10</td>
<td>300</td>
</tr>
<tr>
<td>Upgrade Signals</td>
<td>2</td>
<td>18</td>
<td>10</td>
<td>250</td>
</tr>
<tr>
<td>Total Improvements</td>
<td>447</td>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
improvement, and annual maintenance cost. Using the total accident reduction value (in percent reduction) at each location under consideration, an approximate benefit was calculated. Accidents unrelated to the location such as brake failures, drunk driving, or tire blowouts were disregarded in the calculation of expected benefits after improvement.

The subroutine COSBEN was used to compute monetary benefits from expected accident reductions. Accident costs used were recent National Safety Council values (14):

Fatality = $45,000,
Injury = 2,700, and
Property Damage Only (PDO) = 400.

The accident occurrence at each location is multiplied by the expected percent reduction for the improvement alternative. The cost of accidents are then multiplied by the expected accident reduction to give annual benefits. These annual benefits are then multiplied by an exponential growth, present-worth factor (Equation 2) to obtain the benefits for the entire service life of the improvement.

The costs used in the calculations are the sum of the improvement cost for each project and the maintenance cost. A present-worth factor (Equation 1) was used to adjust the maintenance cost from a future date to the present.

It should be understood that the process of accurately estimating benefits and costs can be very difficult. Even with a large sample of before-and-after data for locations improved by various alternatives, accident reduction estimates may be inaccurate. This is partially attributable to the varying characteristics of specific highway locations. Spuriousness in accident occurrence makes it impossible to accurately predict future accidents. Predictions of expected accidents after a particular improvement should be based on large samples combined with careful engineering judgement. Dynamic programming can give near-perfect results if all input is exactly correct. However, if benefit and cost input is carelessly or incorrectly estimated, results of dynamic programming will be equally in error.

RESULTS

A group of 61 "high-accident" locations previously improved under the Kentucky spot-improvement program were selected as test data for the dynamic programming model. Accident reports at each location were reviewed, and improvement alternatives were actual improvements made at the locations. Input into the computer program for each alternative at each location consisted of accident data, expected accident reduction, project costs, service life of improvement, maintenance costs, and interest rate.

The dynamic programming model computed benefits for each alternative. Then, as the available budget was varied from $10,000 to $80,000, an optimal scheme of alternatives was generated for each
budget.

A similar calculation of return and benefit-cost ratio was made using a benefit-cost analysis. There was very little difference in the benefit-cost analysis and the dynamic programming analysis for the test locations. This is shown in Figure 1 where expected return versus available budget is plotted for both dynamic programming and benefit-cost analyses. Details of the data used to plot Figure 1 are presented in Table 2.

Comparison of Dynamic Programming and Benefit-Cost Ratio

Theoretically, dynamic programming computer techniques will produce a scheme for allocating funds under a fixed budget such that the optimal return is obtained. After testing the computer model, it was found that this is true as long as each project cost is an exact multiple of the budget increment. For example, if computer storage constraints permit an increment of $250 with a budget of $100,000, then the cost of each improvement should be a multiple of $250 in order to obtain an optimal improvement scheme. An increment was defined as some fraction of the budget used in the computer analysis for weighing benefits against costs. In general, the smaller the increment, the better the solution obtained. The number of increments into which the maximum budget may be divided, however, is largely governed by the computer storage capacity as well as computer time required. Practically, then, the increment cannot be made as small as desired. If the majority of costs are at least twice the increment, the results seem to be reasonably good.

A simplified example (Table 3) was developed to demonstrate how the monetary return using dynamic programming techniques will exceed the return from a benefit-cost analysis if project costs are multiples of the increment. As shown in Figure 2, the dynamic programming return is the best at nearly every budget level from $5000 to $34,000. Although the two are fairly close at some points, the return from the benefit-cost curve is inferior to the dynamic programming curve by about $50,000 at a budget of $20,000 and by $40,000 at a budget of $30,000. The two curves are equal at budgets of $25,000 and $34,000. In this example, the $34,000-budget was divided into 34 increments of $1,000 each. Each project cost is a multiple of $1,000.

Benefits from benefit-cost techniques may sometimes equal benefits obtained from dynamic programming techniques and, in some cases, will produce undesirable results. Dynamic programming, however, will always produce the optimal scheme if project costs are expressed as multiples of the increment.

Use of Dynamic Programming

Application of dynamic programming techniques to the highway safety improvement program in Kentucky involves several steps. First, a list of potentially hazardous locations, based on accident data,
Figure 1. Expected Return versus Available Budget for Dynamic Programming – Benefit-Cost Analyses
<table>
<thead>
<tr>
<th>LOCATION</th>
<th>LOCATION</th>
<th>ALTERNATIVE</th>
<th>NO.</th>
<th>NAME</th>
<th>NO.</th>
<th>COST</th>
<th>RETURN</th>
<th>BC RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62-13-15-1</td>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>2</td>
<td>62-13-15-1</td>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>3</td>
<td>62-13-15-1</td>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>4</td>
<td>62-13-15-1</td>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>5</td>
<td>62-13-15-1</td>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>6</td>
<td>62-13-15-1</td>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>7</td>
<td>62-13-15-1</td>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>8</td>
<td>62-13-15-1</td>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
</tbody>
</table>

**TABLE 1**

INPUT DATA FOR DYNAMIC PROGRAMMING
BENEFIT-COST ANALYSIS COMPARISON
### Table 3

**Input Data for Dynamic Programming Benefit-Cost Analysis Comparison**

*(Example Problem)*

<table>
<thead>
<tr>
<th>Location Number</th>
<th>Alternate Number</th>
<th>Costs</th>
<th>Benefits</th>
<th>Benefit-Cost Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$1000</td>
<td>$20000</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1000</td>
<td>15000</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1000</td>
<td>12000</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3000</td>
<td>30000</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5000</td>
<td>45000</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>10000</td>
<td>80000</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1000</td>
<td>7000</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>9000</td>
<td>54000</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2000</td>
<td>6000</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1000</td>
<td>2000</td>
<td>2</td>
</tr>
</tbody>
</table>

Total: $34000 $271000
Figure 2. Expected Return versus Available Budget for Dynamic Programming – Benefit Cost Analyses (Example Problem)
must be identified. A recommended location-identification procedure for Kentucky identifies hazardous 0.3-mile (0.48-km) spots and 3-mile (4.8-km) sections based on fatal accidents, total number of accidents, accident severity rating (the "equivalent-property-damage-only" number), and accident rate (applying quality control techniques). Locations should be identified based on dual 1-year and 2-year time intervals. Locations identified by citizens, engineering personnel, and state police should be considered. All locations identified as possibly hazardous should then be reviewed. Locations considered worthy of a field inspection should be investigated for possible corrective measures.

The proposed program requires that all warranted minor improvements such as signs, paint striping, flashing beacons, and delineators be implemented without dynamic programming considerations. Major improvements such as resurfacing, bridge widening, realignment, and intersection channelization should be selected by dynamic programming techniques.

Project costs, expected benefits, maintenance costs, and expected service life of the improvement should be determined for each alternative at every location to be considered under dynamic programming. After the warranted minor improvements are considered, the remaining money should be budgeted for use in other projects where the dynamic programming may apply. An optimal set of improvement alternatives would then be generated.

SUMMARY AND CONCLUSIONS

The objective of this study was to develop or adopt appropriate dynamic programming methods that would assist in establishing optimal budgeting procedures for various highway programs. Dynamic programming is a multistage operation which involves the evaluation of several projects with several alternatives. A dynamic programming procedure was developed to select the optimal combination of safety improvement projects for a given budget. The following major findings may be cited:

1. Use of dynamic programming is relatively simple. Input consists of the budget, costs, and benefits. Estimating the benefits derived from a particular improvement presents the most difficulty.

2. Table I, which lists accident reduction by type of improvement for past safety improvements, was developed from past accident experience for use in estimating savings.

3. The accuracy and reliability of dynamic programming is dependent upon the accuracy of benefits and costs used as input.

4. A prerequisite in the use of dynamic programming for the safety improvement program is an efficient method of systematically identifying locations based on accident data. In-depth field investigations are also needed so that only necessary improvements are recommended as input for the dynamic programming model.
5. It was shown that improvements selected by dynamic programming can yield a higher return for a given budget than those chosen entirely on the basis of benefit-cost ratios (Figure 2).

6. If individual project costs are multiples of the increment used in the dynamic programming, the optimum allocation of funds will always be obtained. In general, the smaller the increment, the better the solution obtained. However, the attractiveness of a smaller increment is restricted by available computer storage.

7. Applicability of dynamic programming to budget allocation in transportation planning is practically unlimited. In addition to the highway safety improvement program, dynamic programming can be used to optimize investments in maintenance and construction programs and eventually the entire transportation department.

LIST OF REFERENCES


APPENDIX A

PROGRAM SOURCE DECK
PROGRAM MAIN

C DATE: AUGUST 5, 1974
C PROGRAMMER: THIS PROGRAM WAS WRITTEN BY JESSE MAYES, DIVISION OF MAIN0010
C RESEARCH, DEPT. OF TRANS., COMMONWEALTH OF KY., 533 S. LIMESTONE ST., MAIN0020
C LEXINGTON, KY. PARTS OF THE PROGRAM, INCLUDING THE DYNAMIC MAIN0030
C PROGRAMMING ALGORITHM, HAVE BEEN ADAPTED FROM A PROGRAM WRITTEN BY MAIN0040
C THE STATE OF ALABAMA HIGHWAY DEPT., BUREAU OF MAINT., 1973. SEE MAIN0050
C REPORT "CORRECT COST/BENEFIT OPTIMIZATION FOR THE REDUCTION OF ROAD MAIN0060
C ENVIRONMENT CAUSED TRAGEDIES".
C PURPOSE: THIS PROGRAM CALCULATES COSTS AND BENEFITS FOR EACH MAIN0080
C ALTERNATIVE AT EACH LOCATION THEN DETERMINES THE OPTIMAL SOLUTION MAIN0100
C SET OF ALTERNATIVES TO BE IMPLEMENTED FOR A GIVEN RANGE OF BUDGETS. MAIN0110
C INPUT AND OUTPUT: SEE DIVISION OF RESEARCH REPORT: "OPTIMAL HIGHWAY MAIN0120
C SAFETY IMPROVEMENTS BY DYNAMIC PROGRAMMING".
C
C DIMENSION ORET(64,401), NOD(64,401)
C DIMENSION TITL(20), XLOC(64,5), NDE(64), C(64,11), P(64,11), LOC(64)
C NINP = 401 MAIN0130
C NLOC = MAXIMUM NUMBER OF INCREMENTS---MAXIMUM BUDGET EQUALS NINP*XINC
C NLOC = 64 MAIN0140
C INN, IOUTPR = LOCAL INPUT AND OUTPUT DEVICE NUMBERS MAIN0150
C INN, IOUTPR = LOCAL INPUT AND OUTPUT DEVICE NUMBERS
C READ(INN,1000) TITL MAIN0160
C 1000 FORMAT(20A4) MAIN0170
C WRITE(IOUTPR,1010) TITL MAIN0180
C 1010 FORMAT(20X,20A4,///) MAIN0190
C READ(INN,1020) NSTG, XINC, K1, K2 MAIN0200
C 1020 FORMAT(I4, I5, F6.0, I1, I2, I4) MAIN0210
C CALL COSREN(C, K1, K2, XINC, XLOC, LOC, NDE, NSTG, NLOC, P, 401)
C CALL DYNAM(C, NSTG, XLOC, XINC, K1, K2, NLOC, P, ORET, NINP, IOUTPR)
C IF(K1.EQ.1) GO TO 10
C 10 CONTINUE MAIN0220
C CALL EXIT END MAIN0230
SUBROUTINE COSBEN

SUBROUTINE COSBEN(PWC,PWR,XLOC,LOC,NDE,NSTG,NLOC,XINC,INN,IOUTPR, COSRN0010
+ KIK)
C THIS SUBROUTINE CALCULATES PRESENT WORTH COSTS AND BENEFITS
C ASSOCIATED WITH EACH ALTERNATIVE AT EACH LOCATION
C DIMENSION XLOC(NLOC,5),SEV(4,8),CSFF(10,11),R(8),
+ NDE(NLOC),PWC(NLOC,11),PWR(NLOC,11),LOC(NLOC)
C READ(INN,1000) CFAT,CINJ,CPDO,RATEIN,RAGEGR
1000 FORMAT (F10.0)
C 1010 FORMAT (11H1)
1010 FORMAT (1H1)
C DIMENSION XLOC(11),SEV(4,8),CSFI(11),A(8), NOE(INLOC),PWC(INLOC,11),PWR(INLOC,11),LOC(INLOC)
C REAO(INN,1DD0) CFAT,CINJ,CPDO,RATEIN, RATEGR
1000 FORMAT (F10.0)
C THIS SUBROUTINE CALCULATES PRESENT WORTH COSTS AND BENEFITS
C ASSOCIATED WITH EACH ALTERNATIVE AT EACH LOCATION
C DIMENSION XLOC(NLOC,5),SEV(4,8),CSFI(11),A(8), NOE(INLOC),PWC(INLOC,11),PWR(INLOC,11),LOC(INLOC)
C REAO(INN,1DD0) CFAT,CINJ,CPDO,RATEIN, RATEGR
1000 FORMAT (F10.0)
C THE ABOVE READS AND PRINTS THE BASIC PARAMETERS CONSTANT FOR THE
C ENTIRE PROGRAM
NBR = 1
KIK = 0
C BELOW IS THE INPUT WHICH IS EXECUTED FOR EACH ACCIDENT LOCATION.
10 READ(INN,1020) NO1,(XLOC(NUMBER,I),I=1,5),TIME,NMO,NXR,NCAU
C OFF)(N01=NO2,50,60,150
50 WRITE(IOUTPR,1030) NO1,NO2
1030 FORMAT (/80C
1030 FORMAT (/80C
C SECOND CARD INPUT FOR EACH CRITICAL LOCATION (SEVERITIES).
READ(INN,1070) NO2,(SEV(J,J),J=1,4),ALT
1070 FORMAT(14,3RF2.0)
C ROUTINE TO CHECK CARD SEQUENCE CODE.
IF(NO1-NO2 .LT.50.60.50
50 WRITE(IOUTPR,1080) NO1,NO2
1080 FORMAT(14,3RF2.0)
C ROUTINE TO CHECK CARD SEQUENCE CODE.
IF(NO1-NO2 .LT.50.60.50
50 WRITE(IOUTPR,1080) NO1,NO2
1080 FORMAT(14,3RF2.0)
C OUTPUT OF SEVERITIES AND TOTALS.
WRITE(IOUTPR,1090)
1090 FORMAT(14,3RF2.0)
TOT2=TOT2+ SEVI(I,2)  
TOT3=TOT3+ SEVI(I,3)  
TOT4=TOT4+ SEVI(I,4)  
CONTINUE
WRITE(1OUTPR,1110) TOT1, TOT2, TOT3, TOT4
1110 FORMAT(/' TOTALS', F12.0, 3F6.0)

C INPUT NEXT SET OF NALT CARDS, ONE FOR EACH ALTERNATIVE
NJ=3+NCNAU
DO 110 I=1,NALT
READ(INN,1120) NO3,(CSEF(I,J), J=1,NJ)
1120 FORMAT(I4,F7.0,F2.0,F5.0,F5.0,F3.2)
IF(NO3=NO1)90,100,90
90 WRITE(1OUTPR,1080) NO1, NO3
100 CONTINUE
110 CONTINUE

C OUTPUT OF ALTERNATIVE INFORMATION.
WRITE(1OUTPR,1130)(1,1=1,NALT)
1130 FORMAT(/' ALTERNATIVE COST  LIFE  MAINTENANCE  COST  EFFECT ON...', 
+8B15)
C NUMBER COUNT CHECK OF SEVERITIES.
DO 120 I=1,NALT
WRITE(1OUTPR,1140) I,(CSEF(I,J), J=1,NJ)
1140 FORMAT(I4,F13.2,F8.0,F9.2,F24.2,F7F5.2)
120 CONTINUE

C COMPUTATION OF B(I), THE ITH ALTERNATIVE BENEFIT.
DO 140 I=1,NALT
B(I) = 0.
DO 130 J=1,NCNAU
JFET = J +3
R(I) = B(I) + (CFAT*SEVI(J,2)+CINJ*SEVI(J,3)+CPDO*SEVI(J,4))*
+ CSEF(I,JFET)
130 CONTINUE
140 CONTINUE

C CALCULATION OF BENEFIT/COSTS AND OUTPUT.
WRITE(1OUTPR,1150)
1150 FORMAT(/' BENEFIT/COST ANALYSIS'/' ALTERNATIVE COST 
+ BENEFIT/COST')
DO 150 I=1,NALT
B(I)=B(I)+CSEF(I,2)/TIME
BNCM = B(I)/CSEF(I,1)
WRITE(1OUTPR,1160) I,CSEF(I,1),B(I),BNCM
1160 FORMAT(I4,F14.2,F14.2,F18.4)
150 CONTINUE

WRITE(1OUTPR,1170)
1170 FORMAT(/' BENEFIT/COST ANALYSIS, MAINTENANCE INCLUDED'/) 
WRITE(1OUTPR,1180)
1180 FORMAT(/' ALTERNATIVE MAINTENANCE TOTAL COST BENEFIT/COST') 
1190 FORMAT(/' ALTERNATIVE MAINTENANCE TOTAL COST BENEFIT/COST')
DO 160 I=1,NALT
LIFE = CSEF(I,2)
X = (1+RATEN)**LIFE
160 CONTINUE
$PWF = \frac{(X-1.1)}{RATEIN*X}$

$Y = \frac{(1.1+RATGER)/(1.1+RATEIN)}{}$

$PWEXGR = \frac{Y^*(LIFE+1)-1.1}{(Y-1.1)} - 1$

$PW(NUMBER,1) = 0$

$PWB(NUMBER+1) = 0$

$PWMAIN = PWF*CSEF(1.3)$

$PW(NUMBER,I+1) = CSEF(I+1) + PWMAIN$

ROUND PRESENT WORTH COSTS TO NEAREST INCREMENT

$IPWC = PW(NUMBER,I+1)/XINC + .5$

$PW(NUMBER,I+1) = IPWC*XINC$

$PWB(NUMBER,I+1) = PWEXGR*A(I)/LIFE$

$PWB = PWB(NUMBER,I+1)/PW(NUMBER,I+1)$

$WRITE(IOUTPR,1220) I,PWMAIN,PWB(NUMBER,I+1),PWB(NUMBER,I+1),PWB C$


170 CONTINUE

$NUMBER = NUMBER + 1$

GO TO 10

180 CONTINUE

$NUMBER = NUMBER - 1$

IF(NUMBER.EQ.NSTG) GO TO 190

WRITE(IOUTPR,1230)

1230 FORMAT(1140(*),', WARNING ',40(*),/)]

WRITE(IOUTPR,1240) NUMBER,NSTG

1240 FORMAT(1140(*), 'NUMBER OF LOCATIONS READ = ',I3/ 'NUMBER OF LOCATIONS'

+NS EXPECTED = ',I3)

190 CONTINUE

RETURN

END
SUBROUTINE DYNAM

SUBROUTINE DYNAM(C,R,LOC,XLOC,NDE,NSTG,XINC,K1,K2,NINP,NLOC,
+ ORET,NOD,XOUTPR)  

C THIS SUBROUTINE USES "DYNAMIC PROGRAMMING" TO FIND THE OPTIMAL
C SOLUTION SET ALTERNATIVES (ONE AT EACH LOCATION) GIVEN COSTS,
C BENEFITS AND A RANGE OF BUDGETS. THE ALGORITHM IS BASED ON WORK BY
C RICHARD BELLMAN (DYNAMIC PROGRAMMING, 1957)

DIMENSION ORET(NLOC,NINP),NOD(NLOC,NINP),NDE(NLOC),
+ C(NLOC,11),R(NLOC,11),XLOC(NLOC,5),LOC(NLOC)

IST=0
VRET=0.0
WRITE(IOUTPR,1130)
WRITE(IOUTPR,1000)
1000 FORMAT(' ',40(*'I,'PARAMETER VALUES',40(*'I//')//)
WRITE(IOUTPR,1010)
1010 FORMAT(' ',27X,18('I-',18('I'))
WRITE(IOUTPR,1020) NSTG,XINC,K1,K2

1020 FORMAT(10,3X,'LOCATIONS---INCREMENT---LOWER LIMIT---INCREMENTS PER STD')
1 0 CONTINUE
WRITE(IOUTPR,1030) LOC(I),NDE(I)
1030 FORMAT(' ',7X,'LOCATION---LOCATION NAME--------ALT-NUM--------COST---RETURN---B/C RATIO')
1 0 CONTINUE
WRITE(IOUTPR,1040)
1040 FORMAT(' ',30(*'I,'LOCATIONS,ALTERNATIVES,COSTS AND BENEFITS',
+ 30('I'))//)
WRITE(IOUTPR,1050)
1050 FORMAT(' ',1H,'LOCATION---LOCATION NAME---------ALT-NUM-----------COST')
+ 'RETURN--B/C RATIO')

C FIND THE OPTIMAL ALTERNATIVE AT THE I-TH LOCATION WITH J INCREMENTS
C AVAILABLE

DO 160 I=1,NSTG
NDEC=NDE(I)+1
R(I)=0.
DO 20 IC=2,NDEC
20 R(IC) = R(I,IC)
DO 30 IC=2,NDEC
30 ICM1 = IC-1
BCRAT = R(ICM1)/C(IC,IC)

WRITE(IOUTPR,1060) LOC(I),XLOC(I,J),J=1,5),ICM1,C(IC,IC),R(IC),
+ BCRAT
1060 FORMAT(19,5X,5A4,16,3X,F11.0,F11.0,4X,F10.2)
30 CONTINUE
1070 FORMAT(8F10.0)
DO 130 J=1,NINP

C INCREMENT BUDGET
XIN=J-1)*XINC
NUM=-1000000000000.
NDE=NDE(J)+1

C DETERMINE THE BEST ALTERNATIVE--NOD(I,J)--AT I-TH LOCATION GIVEN
J-1 INCREMENTS TO SPEND ON LOCATION 1 THRU LOCATION I------YIELDING
A RETURN OF--ORET(I,J)--
C
CALL XOUT(I,IST,XIN,K,KICK,XINC,C,NLOC)
IF(KICK)50,50,40
40 GO TO 120
50 CONTINUE
IF(I-1)60,60,70
60 TEST=R(K)
GO TO 80
70 TEST=R(K)+ORET(I-1,IST)
GO TO 80
80 IF((DUM-TEST))90,100,100
90 DUM=TEST
  ORE(T(I,J))=DUM
  NOD(I,J)=K
GO TO 110
110 CONTINUE
120 CONTINUE
130 CONTINUE
140 CONTINUE
  IPAGE = 0
C WRITE MAIN BUDGET OUTPUT HEADING
  WRITE(IOUTPR,1080)
1080 FORMAT('13',90('**'),',37('**')', 'BUDGET OUTPUT',',37('**')',
  + 90('**')',////)
  DO 160 M=K1,NINP,K2
    J=M
    XIN=(J-1)*XINC
    IPAGE = IPAGE + 1
    IF(IPAGE.NE.1) WRITE(IOUTPR,1130)
C WRITE INDIVIDUAL BUDGET OUTPUT HEADING
  WRITE(IOUTPR,1090)
1090 FORMAT('15X','BUDGET LOCATION = ',4X,'LOCATION NAME
  + ',4X,'ALT-NAME',5X,'COST',6X,'RETURN',4X,'ACCUM RETURN')
  WRITE(IOUTPR,1100) XIN
1100 FORMAT('15X,6X,F15.2')
  TOTCST = 0
  TOTRTN = 0
  DO 150 L=1,NSTG
    I=NSTG+1-L
    K=NOD(I,J)
    KK=NOD(1,J)-1
    TOTCST = TOTCST + C(I,K)
    TOTRTN = TOTRTN + R(I,K)
C WRITE I-TH LOCATION INFORMATION---TOTAL BUDGET OF M INCREMENTS
  WRITE(IOUTPR,1110) LOC(I),XINC,JJ, JJ=1,5),KK,C(I,K),
  + R(I,K),ORET(I,J)
1110 FORMAT('15X,9X,F4.4,5X,4X,F12.4,4X,F12.4)
  CALL XOUT(I,IST,XIN,K,KICK,XINC,C,XINC)
  J=IST
  XIN = XIN-C(I,K)
150 CONTINUE
C WRITE TOTALS
  WRITE(IOUTPR,1120) TOTCST,TOTRTN,ORET(NSTG,M)
1120 FORMAT('15X,9X,F12.4,4X,F12.4')
  CONTINUE
C WRITE TOTALS
  WRITE(IOUTPR,1130)
1130 FORMAT('15X')
  CONTINUE
  RETURN
END
SUBROUTINE XOUT

SUBROUTINE XOUT(I, IST, XIN, K, KICK, XINC, C, NLOC)

C
C THIS SUBROUTINE CALCULATES THE OUTPUT STATE NUMBER
C RESULTING FROM THE INPUT XIN AND SAFETY MEASURE K. IT
C ALSO DETERMINES THE COST OF A PARTICULAR SAFETY MEASURE
C CORRESPONDING TO STAGE I.
C
DIMENSION C(NLOC, 11)
OUT = XIN - C(1, K)
IF (OUT) 10, 20, 20
10 KICK = 1
   IST = 1
   GO TO 30
20 KICK = 0
   IST = (OUT/XINC) + 1.5
30 RETURN
END
VARIABLE LIST

The following is a description of the variables used in the main program and in subroutines COSBEN, DYNAM, and XOUT. Variables preceded by * are part of the input data.

VARIABLE LIST FOR MAIN

INN       Device number for local card reader (specify in MAIN)

IOUTPR    Device number for local printer (specify in MAIN)

*TITL(K)  Title of run

*XLOC(N,K) Alphanumeric array containing location name for N-th location

*LOC(N,K) Integer array containing reference number for N-th location

*NDE(N)   Integer array containing number of alternatives at N-th location

NLOC      Maximum number of locations to be considered.

*NSTG     Number of locations

*XINC     Increment size

*XINP     Number of increments into which budget is divided

*K1       Starting budget for printout (in number of increments + 1)

*K2       Budget printout intervals (in number of increments)

C(N,I)    Cost of I-th alternative at N-th location

B(N,I)    Benefit of I-th alternative at N-th location
VARIABLE LIST FOR COSBEN

The following variables are stored and kept throughout the entire program execution:

NUMBER Number of locations

XLOC(N,K) See variable list for MAIN

INN See variable list for MAIN

IOUTPR See variable list for MAIN

KIK See variable list for MAIN

LOC(N,K) See variable list for MAIN

NDE(N) See variable list for MAIN

*RATEIN Present interest rate (decimal)

*RATEGR Present traffic volume growth rate (decimal)

PWC(N,I) Present worth cost (including exponential growth factor) for I-th alternative at N-th location

PWB(N,I) Present worth benefit (including exponential growth factor) for I-th alternative at N-th location.

The following variables pertain to each location and the values are destroyed after cost-benefit calculations are made:

*SEV(J,I) Real array containing the following accident history for the I-th alternative:
SEV(1,I) -- Total number of accidents
SEV(2,I) -- Number of fatal accidents
SEV(3,I) -- Number of nonfatal injury accidents
SEV(4,I) -- Number of property damage only accidents

*CSEF(J,I) Real array containing the following cost and effect data for I-th alternative:
CSEF(1,I) -- Initial cost
CSEF(2,I) -- Life (years)
CSEF(3,I) -- Maintenance cost per year
CSEF(4,I) -- Effect (percent reduction) on cause (J - 3); J = 4, 5, ...

B(I) Real array containing total benefit for the I-th alternative (calculated neglecting economic and volume growth factors.)

XMAIN Total maintenance cost for the I-th alternative (calculated neglecting economic and volume growth factors)

BNCS Benefit-cost ratio for the I-th alternative (calculated neglecting economic and volume growth factors) excluding maintenance.

BNCM Benefit-cost ratio for the I-th alternative (calculated neglecting economic and volume growth factors) including maintenance.

FWBC Benefit-cost ratio for the I-th alternative (calculated neglecting economic and volume growth factors) including maintenance and using an exponential growth rate factor and the present worth method of calculating costs and benefits

*ALT Number of alternatives

NALT Number of alternatives

*NCAU Number of accident causes

*TIME Time period of accident history (years)

*NMO Month of investigation
*NYR  Year of investigation

LIFE  Life (years)

**VARIABLE LIST FOR DYNAM**

The following variables are described in the variable list for MAIN. All are passed as arguments into DYNAM:

*C
*B
*LOC
*XLOC
*NDE
*NSTG
*XINC
*K1
*K2
*NINP
*NLOC

The following variables are used for calculations at the I-th stage:

I  Stage of investigation

K  Alternative at Location I being considered

IST  Number of increments that would remain if K-th alternative, Location I, were chosen at Stage I

NDEC  Number of alternatives + 1 (Location I)

R(K)  Return from K-th alternative (Location I)

BCRAT  Benefit-cost ratio for K-th alternative (Location I)
XIN Variable budget \((J - 1)\) increments

NOD(I,J) Integer array containing best alternative from \(I\)-th Location given \((J - 1)\) increments to spend at \(I\)-th stage

ORET(I,J) Real array containing optimum return for spending \((J - 1)\) increments at \(I\)-th stage

TEST Return at \(I\)-th stage from \(K\)-th alternative plus optimum return for remaining budget at \((I - 1)\)-th stage

DUM Maximum value of TEST

KICK Integer containing "O" if there is insufficient budget left to do \(K\)-th alternative (Location \(I\))

TOTCST Total cost of chosen improvements

TOTRIN Total return from chosen improvements

VARIABLE LIST FOR XOUT

The following variables are described in DYNAM. All are passed as arguments into XOUT:

I

IST

XIN

K

KICK

*XINC

*C

*NINP

OUT Budget that would remain if \(K\)-th alternative, Location \(I\), were chosen at Stage \(I\)
APPENDIX C

PROGRAM FLOW CHART
SUBROUTINE CMHRN

02.16---02

THIS SUBROUTINE CALCULATES DISTANCE WHERE COST AND
SHIFTS ARE ASSOCIATED WITH EACH
LOCATION.

* 01
READ FROM DEY
100
VIA FORMAT
1000
FROM THE LIST

01
WRITE TO DEV
100
VIA FORMAT
100
FROM THE LIST

02
WRITE TO DEV
100
VIA FORMAT
100
FROM THE LIST

6.01

NOTE 07
LIST = NU11, T = 1.50, TYPE, RND, NPS, NCXU

NOTE 08
LOC(NU11) = 001

NOTE 21
LIST = NUMBER,

NOTE 22
CONTINUE

NOTE 23
EXIT

NOTE 24
EXIT

NOTE 25
EXIT

NOTE 26
EXIT

NOTE 27
EXIT

NOTE 30
EXIT

NOTE 31
EXIT

NOTE 32
EXIT

NOTE 99
EXIT
APPENDIX D

INPUT CODING INSTRUCTIONS
INPUT CODING INSTRUCTIONS

The following is a description of the input required to use the program presented in APPENDIX A. It should be pointed out that the input and output device numbers, INN and IOUTPR, respectively, must be defined in MAIN. Also in MAIN, the following dimensions must be specified: the dimension of NDE and LOC must be the same as the first dimension of XLOC, C, B, ORET and NOD; all of these dimensions are equal to NLOC. The second dimension of ORET and NOD must be equal to NINP. The variables NLOC and NINP correspond to the maximum number of locations and budget increments, respectively, and must be defined in MAIN. Any capitalized term refers to the variable exactly as found in the program. All integer quantities must be right-adjusted. Real numbers should be punched with a decimal or right-adjusted.

CARD 1 (Type A) Title Card

I. Title of run: TITL
   In Columns 1-80 place any alphanumeric symbols desired

CARD 2 (Type B) Printout Card

I. Number of locations: NSTG
   In Columns 1-4 place the number of locations actually being considered (integer number)
II. Size of increment: XINC
    In Columns 5-10 place the size of the increments into which the budget is divided (real number)
III. Starting budget increment: K1
     In Columns 11-14 place the number of increments (+1) corresponding to the first budget desired printed out (integer number)
IV. Budget increments: K2
    In Columns 15-18 place the number of increments between successive budgets desired printed out (integer number)

CARD 3 (Type C) Accident Cost Card

I. Cost of fatality accident: CFAT
In Columns 1-10 place the average cost of a fatal accident (real number)

II. Cost of nonfatal injury accident: CINJ
   In Columns 11-20 place the average cost of a nonfatal injury accident (real number)

III. Cost of property damage only accident: CPDO
   In Columns 21-30 place the average cost of a property damage only accident (real number)

IV. Interest rate: RATEIN
   In Columns 31-40 place the present available interest rate (real number)

V. Exponential growth rate: RATEGR
   In Columns 41-50 place the expected traffic volume growth rate (real number)

Note: Card types D, E, and F are repeated for each location.

CARD 4 (Type D) Location Card

I. Location reference number: LOC
   In Columns 1-4 place location reference number (integer number)

II. Location name: XLOC
   In Columns 5-68 place the alphanumeric name associated with the location

III. Leave Columns 69-71 blank

IV. Time period of accident history
   In Columns 72-75 place the time period (in years) of the accident history (real number)

V. Present date
   In Columns 76-77 place month (integer number)
   In Columns 78-79 place two last digits of year (integer number)

VI. Number of causes: NCAU
   In Column 80 place the number of accident causes (integer number)

CARD 5 (Type E) Severity Card

I. Location reference number: XLOC
   In Columns 1-4 place location reference number (integer number); this should be the same
   as on Card 4
II. Severities for CAUSE 1 (real number, right-adjusted)
   In Columns 5-6 place number of accidents attributed to CAUSE 1
   In Columns 7-8 place number of fatal accidents attributed to CAUSE 1
   In Columns 9-10 place number of nonfatal injury accidents attributed to CAUSE 1
   In Columns 11-12 place number of property damage only accidents attributed to CAUSE 1

III. Repeat II for CAUSE 2, CAUSE 3, etc., continuing on same card; use integer fields of two,
     i.e., Columns 13-14, Columns 15-16, etc.

   Note: Maximum of eight causes

IV. Number of alternates: ALTR
   In Columns immediately following last CAUSE place the number of alternatives (Real number,
     right-adjusted)

   Note: Maximum of ten alternatives

CARD 6 (Type F) Alternative Description Card

I. Location reference number: LOC
   In Columns 1-4 place location reference number (integer number);
     this number should be the same as on Cards 4 and 5

II. Cost
   In Columns 5-11 place initial cost of alternative (real number)

III. Life
   In Columns 12-13 place estimated life (in years) of alternative (integer number)

IV. Maintenance cost
   In Columns 14-18 place estimated maintenance cost per year of alternative (real number)

V. Effect on CAUSE 1
   In Columns 19-21 place the fractional reduction of CAUSE 1 by implementation of alternative
     (real number)

VI. Repeat V for CAUSE 1, CAUSE 2, etc., continuing on the same card using Columns 22-24,
    Columns 25-27, etc.

   Repeat Card type F for each alternative at given location. Last card of data deck MUST be blank.
SAMPLE OF INPUT DATA

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>21</th>
<th>31</th>
<th>41</th>
<th>51</th>
<th>61</th>
<th>71</th>
<th>80</th>
<th>COLUMN NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>61250.0</td>
<td>41</td>
<td>40</td>
<td>45000.</td>
<td>2700.</td>
<td>400.</td>
<td>0.10</td>
<td>0.04</td>
<td>1.0</td>
<td>1691</td>
</tr>
<tr>
<td>163-25-10.9</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>21</td>
<td>100000</td>
<td>100.02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REPEAT CARD TYPES D, E, AND F FOR EACH LOCATION
LAST CARD IS BLANK
SAMPLE OF OUTPUT

TEST RUN 1 DYNAMIC PROGRAMMING

NEG UTILITY FATALITY = 45000 INJURY = 2700. PRP DM = 400.
INTEREST RATE = 0.100
EXPONENTIAL GROWTH RATE = 0.040

COST-BENEFIT OUTPUT FOR LOCATION 1
(see Table 2 for summary of all 61 locations)

<table>
<thead>
<tr>
<th>REF NO</th>
<th>63-25-10.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCIDENT HISTORY</td>
<td>1.00 YEARS. MONTH 1, YEAR 69, 1 CAUSE.</td>
</tr>
<tr>
<td>ROADWAY CAUSE</td>
<td>TACC NFAT NINJ NPRO</td>
</tr>
<tr>
<td>1</td>
<td>4. 1. 2. 2.</td>
</tr>
<tr>
<td>TOTALS</td>
<td>4. 1. 2. 2.</td>
</tr>
<tr>
<td>ALTERNATIVE</td>
<td>COST LIFE MAIN COST EFFECT ON...</td>
</tr>
<tr>
<td>1</td>
<td>1000.00 10. 100.00 0.02</td>
</tr>
</tbody>
</table>

BENEFIT/COST ANALYSIS

<table>
<thead>
<tr>
<th>ALTERNATIVE</th>
<th>COST</th>
<th>BENEFIT</th>
<th>BENEFIT/COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000.00</td>
<td>10240.00</td>
<td>10.2400</td>
</tr>
</tbody>
</table>

BENEFIT/COST ANALYSIS, MAINTENANCE INCLUDED

<table>
<thead>
<tr>
<th>ALTERNATIVE</th>
<th>MAINTENANCE</th>
<th>TOTAL COST</th>
<th>BENEFIT/COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000.00</td>
<td>2000.00</td>
<td>5.1200</td>
</tr>
</tbody>
</table>

BENEFIT/COST ANALYSIS, MAINTENANCE INCLUDED ***PRESENT WORTH METHOD***

<table>
<thead>
<tr>
<th>ALTERNATIVE</th>
<th>MAINTENANCE</th>
<th>TOTAL COST</th>
<th>BENEFIT</th>
<th>BENEFIT/COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>614.45</td>
<td>1500.00</td>
<td>7614.84</td>
<td>5.08</td>
</tr>
</tbody>
</table>

***************PARAMETER VALUES***********************

-------------OUTPUT-------------

LOCATIONS---INCREMENT---LOWER LIMIT---INCREMENTS PER STEP
01 250.00 41 40
<table>
<thead>
<tr>
<th>LOCATION NAME</th>
<th>ALT-NUM</th>
<th>COST</th>
<th>RETURN</th>
<th>ACCUM RETURN</th>
</tr>
</thead>
<tbody>
<tr>
<td>63-25-10.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>868692.0</td>
</tr>
<tr>
<td>22-60-26.2</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>868692.0</td>
</tr>
<tr>
<td>30-60-16.4</td>
<td>2</td>
<td>500</td>
<td>333.8</td>
<td>868692.0</td>
</tr>
<tr>
<td>82-31W-1.2</td>
<td>1</td>
<td>500</td>
<td>192.1</td>
<td>868692.0</td>
</tr>
<tr>
<td>21-41-20.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>868692.0</td>
</tr>
<tr>
<td>82-31W-2.2</td>
<td>1</td>
<td>500</td>
<td>10792</td>
<td>868692.0</td>
</tr>
<tr>
<td>102-25-11.9</td>
<td>7</td>
<td>2250</td>
<td>1656.5</td>
<td>868692.0</td>
</tr>
<tr>
<td>63-29-16.4</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>852640.0</td>
</tr>
<tr>
<td>47-31W-23.3</td>
<td>2</td>
<td>1750</td>
<td>802.1</td>
<td>852640.0</td>
</tr>
<tr>
<td>70-1560-11.3</td>
<td>9</td>
<td>1000</td>
<td>21532</td>
<td>852640.0</td>
</tr>
<tr>
<td>79-61-12.5</td>
<td>3</td>
<td>750</td>
<td>5860</td>
<td>777356.0</td>
</tr>
<tr>
<td>10-60-9.2</td>
<td>3</td>
<td>750</td>
<td>5860</td>
<td>777356.0</td>
</tr>
<tr>
<td>30-60-15.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>771496.0</td>
</tr>
<tr>
<td>82-31W-1.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>771496.0</td>
</tr>
<tr>
<td>82-31W-12.7</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>771496.0</td>
</tr>
<tr>
<td>82-31W-12.7</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>771496.0</td>
</tr>
<tr>
<td>70-60-11.4</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>762399.0</td>
</tr>
<tr>
<td>51-60-20.3</td>
<td>1</td>
<td>750</td>
<td>86641</td>
<td>762399.0</td>
</tr>
<tr>
<td>79-61-13.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>675988.0</td>
</tr>
<tr>
<td>17-60-8.3</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>675988.0</td>
</tr>
<tr>
<td>51-60-12.3</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>675988.0</td>
</tr>
<tr>
<td>120-60-12.6</td>
<td>1</td>
<td>500</td>
<td>11078</td>
<td>675988.0</td>
</tr>
<tr>
<td>84-60-18.6</td>
<td>1</td>
<td>500</td>
<td>9008</td>
<td>664910.0</td>
</tr>
<tr>
<td>82-31W-1.1</td>
<td>1</td>
<td>500</td>
<td>17168</td>
<td>621416.0</td>
</tr>
<tr>
<td>51-41-20.6</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>604246.0</td>
</tr>
<tr>
<td>41-175-155.6</td>
<td>1</td>
<td>4000</td>
<td>152094</td>
<td>599549.0</td>
</tr>
<tr>
<td>63-25-10.9</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>599549.0</td>
</tr>
<tr>
<td>36-19-1651P-130.1</td>
<td>1</td>
<td>1500</td>
<td>11711</td>
<td>435743.0</td>
</tr>
<tr>
<td>32-26-200</td>
<td>3</td>
<td>3750</td>
<td>8513</td>
<td>379644.0</td>
</tr>
<tr>
<td>54-41A-12.2</td>
<td>3</td>
<td>750</td>
<td>8513</td>
<td>379644.0</td>
</tr>
<tr>
<td>23-32-6-16.9</td>
<td>3</td>
<td>3000</td>
<td>12280</td>
<td>371131.0</td>
</tr>
<tr>
<td>73-62-16.9</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>371131.0</td>
</tr>
<tr>
<td>73-45-6.5</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>371131.0</td>
</tr>
<tr>
<td>73-62-17.8</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>371131.0</td>
</tr>
<tr>
<td>41-22-11.3</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>371131.0</td>
</tr>
<tr>
<td>11-431W-14.1</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>371131.0</td>
</tr>
<tr>
<td>30-54-12.6</td>
<td>1</td>
<td>750</td>
<td>4290</td>
<td>347316.0</td>
</tr>
<tr>
<td>47-31W-20.6</td>
<td>1</td>
<td>2250</td>
<td>26564</td>
<td>347316.0</td>
</tr>
<tr>
<td>37-12-9.7</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>347316.0</td>
</tr>
<tr>
<td>106-629.2</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>347316.0</td>
</tr>
<tr>
<td>63-25-16.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>347316.0</td>
</tr>
<tr>
<td>11-431W-16.2</td>
<td>3</td>
<td>750</td>
<td>4071</td>
<td>318372.0</td>
</tr>
<tr>
<td>24-66-9.1</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>318372.0</td>
</tr>
<tr>
<td>56-41A-12.4</td>
<td>3</td>
<td>750</td>
<td>18460</td>
<td>318372.0</td>
</tr>
<tr>
<td>79-61-10.9</td>
<td>1</td>
<td>750</td>
<td>3548</td>
<td>292713.0</td>
</tr>
<tr>
<td>39-11-6.3</td>
<td>1</td>
<td>6500</td>
<td>26292</td>
<td>292713.0</td>
</tr>
<tr>
<td>30-60-4.1</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>318372.0</td>
</tr>
<tr>
<td>72-61-8.5</td>
<td>1</td>
<td>750</td>
<td>40588</td>
<td>266684.0</td>
</tr>
<tr>
<td>30-60-15.2</td>
<td>3</td>
<td>5000</td>
<td>31866</td>
<td>226095.0</td>
</tr>
<tr>
<td>102-25-9.2</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>194229.0</td>
</tr>
<tr>
<td>21-54-1.1</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>194229.0</td>
</tr>
<tr>
<td>46-421-14.4</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>194229.0</td>
</tr>
<tr>
<td>73-45-7.2</td>
<td>3</td>
<td>750</td>
<td>6726</td>
<td>362875.0</td>
</tr>
<tr>
<td>73-45-6.5</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>187504.0</td>
</tr>
<tr>
<td>54-41A-12.0</td>
<td>1</td>
<td>500</td>
<td>9792</td>
<td>177712.0</td>
</tr>
<tr>
<td>30-66-15.2</td>
<td>1</td>
<td>2000</td>
<td>11385</td>
<td>166327.0</td>
</tr>
<tr>
<td>30-60-15.1</td>
<td>1</td>
<td>4250</td>
<td>15870</td>
<td>150000.0</td>
</tr>
<tr>
<td>63-25-10.4</td>
<td>1</td>
<td>1500</td>
<td>7620</td>
<td>7620.0</td>
</tr>
</tbody>
</table>

*************** TOTALS **********************

<table>
<thead>
<tr>
<th>ALT-NUM</th>
<th>COST</th>
<th>RETURN</th>
<th>ACCUM RETURN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>868692.0</td>
</tr>
</tbody>
</table>