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

Jerry G. Pigman* 
Jesse G. Mayes†

Kenneth R. Agent†
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/902
OPTIMAL HIGHWAY SAFETY IMPROVEMENT INVESTMENTS
BY DYNAMIC PROGRAMMING

by

J. G. Pigman
Research Engineer Principal

K. R. Agent
Research Engineer

J. G. Mayes
Research Engineer

C. V. Zegeer
Research Engineer

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

accepted for publication by the
Transportation Research Board

December 1974
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. The applicability of dynamic programming to the safety improvement program is demonstrated in this study.

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. The type of dynamic programming being considered here is multistage. Multistage is defined as cost optimization of several projects, each with one or more alternatives. All safety improvement costs are dealt with in terms of present worth with consideration given to construction or installation cost, yearly maintenance cost, present interest rate, and the expected life of the improvement. The option of staging installation of safety improvements over a number of years was excluded from this analysis. All possible combinations of improvements were input as alternatives for each of the 61 projects involved in this study. 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.
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. 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 cost 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. Lorrie 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 for 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**

In this study, multistage dynamic programming is evaluated as a means of assigning priorities and allocating expenditures for the spot safety improvement program in Kentucky. Multistage is defined as a process involving several projects, each with one or more alternatives.

All safety improvement costs are dealt with in terms of present worth with consideration given to construction costs, maintenance cost, and the expected life of the improvement. The option of staging installation of safety improvements over a number of years was excluded from this analysis. All possible combinations of alternatives were considered for each of the sixty-one projects involved in the analysis. As an example, a safety improvement project necessitated because of a large number of accidents on a curve may be accomplished by several alternative actions. These may involve realignment, resurfacing, signing, delineation, or any combination of these or other alternatives.

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 "0" 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):

\[
C = S + A \left[ \frac{(1 - i)^L - 1}{i(1 - i)^L} \right]
\]

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.

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

where

- \( B \) = present worth benefit,
- \( t \) = exponential growth rate factor for traffic volume = 4 percent, and
- \( \beta \) = benefit per year associated with the improvement.

\[
\beta = \frac{\sum \sum a_m N_{mn} \gamma_n}{T}
\]

where

- \( \beta \) = benefit per year associated with the improvement,
- \( T \) = time (years) of accident history,
- \( J \) = number of accident causes associated with the location,
- \( a_m \) = 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.,

\[ 0_1(j) = R_1(j) \]

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 \( D_1(j) = \) chosen alternative Location 1 for an investment of j increments.

STEP 3 (STAGE 2 through STAGE M) Repeat STEP 2 for each STAGE.

\[ 0_i(j) = \max \left\{ (R_i(k) + 0_{i-1}(j - k)) \right\} \]

where \( M = \) number of locations considered,

\( 0_i(j) = \) total optimum return after STAGE i for an investment of j increments,

\( R_i(k) = \) return from Location i for an investment of k increments \( (k \leq j) \),

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

\( D_i(j) = \) chosen alternative at Location i for an investment of j increments.

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,

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

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

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

\[ A_i = D_i(N_i + 1) \]
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, 447 improvement projects 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 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. Associated with the high accident locations were 447 improvement projects. Many of the improvement projects included a combination of the various improvements listed in Table 1. Therefore, an alternative which was input for the dynamic programming model may be a combination of several types of improvements with respective adjustments in the percentage accident reduction. In order to make the data manageable for this evaluation, 61 improvement projects were selected as input.

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. For an available budget of $50,000, the computer processing time was 38 seconds at a cost of $5.86 using an IBM 360 computer. The computer storage required for the 61 improvement projects and increments of $250.00 was 268 K.

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. The insignificant difference between benefit-cost analysis and dynamic programming can be attributed to the fact that the priority allocation of funds by benefit-cost is a very efficient method in many cases. However, there is no assurance that benefit-cost will always assign priorities which will yield the greatest return for a specified budget. Following is a comparison of dynamic programming and benefit-cost which exhibits a weakness of benefit-cost for certain situations.

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.

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.

A more detailed explanation of the logic employed in the comparative example of benefit-cost versus dynamic programming may be enlightening at this point. With reference to Table 3 and Figure 2, it can be illustrated that an available budget of $15,000 will produce a greater return by using dynamic programming as compared to benefit-cost. The benefit-cost procedure permitted a sequential selection of projects in the order of decreasing benefit-cost ratios and a corresponding accumulative costs and benefits total. Those projects whose costs would make the total exceed $15,000 were omitted and the procedure would continue until the available budget was reached or the projects were exhausted. From Table 3, it can be seen that this logic would enable the selection of Location Numbers 1, 2, 3, 4, 5, 7, 9, and 10 with an available budget of $15,000. Therefore, using benefit-cost analysis and a $15,000 budget, the improvement costs would be $15,000 and the return would be $137,000 in benefits.

The dynamic programming procedure is not constrained by the benefit-cost ratios and may search throughout the list of projects for those projects which would provide the greatest return for an available budget. In this particular case with the $15,000 budget, dynamic programming would select Location Numbers 1, 2, 4, and 6. These selections would provide a return of $145,000 for improvement costs of $15,000.

From Figure 2, it is obvious that there is a great difference between the respective returns at an available budget of $20,000. This is because for the benefit-cost procedure there were not any additional
projects added to the preceding $15,000 budget because the only remaining projects had costs of $9,000
and $10,000. An addition of either would have exceeded the available budget of $20,000. In contrast,
dynamic programming was able to use all of the available budget because it was not constrained by
limits similar to benefit-cost analysis. The respective benefits at an available budget of $20,000 were
$137,000 using benefit-cost and $190,000 with dynamic programming.

Benefits from benefit-cost techniques may sometimes equal benefits from dynamic programming.
In addition, when it is impossible to arrange the project costs such that they are an exact multiple
of the budget increment, then the benefits from benefit-cost may exceed those from dynamic programming
due to rounding errors. However, dynamic programming will always produce the optimal scheme if project
costs are expressed as multiples of the increment. For these reasons, it is suggested that both benefit-cost
and dynamic programming be tested when it is not feasible to express project costs as multiples of
the budget 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,
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.
Also, 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. The option of staging safety improvements over a number of years was excluded from this analysis. 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. All possible combinations of improvements were included as alternatives in the model for each of the 61 projects.
6. Safety improvement costs were dealt with in terms of present worth with consideration given to construction or installation cost, yearly maintenance cost, present interest rate, and expected life of improvement.
7. 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 3).
8. 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.
9. It is recommended that both benefit-cost and dynamic programming be tested when it is not possible to express project costs as multiples of the budget increment.
10. 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


<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, Maint.</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, 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, Maint.</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><strong>Total Improvements</strong></td>
<td>447</td>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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>

$34000  $271000
Figure 1. Expected Return versus Available Budget for Dynamic Programming – Benefit-Cost Analyses
Figure 2. Expected Return versus Available Budget for Dynamic Programming - Benefit Cost Analyses (Example Problem)
APPENDIX A

PROGRAM SOURCE DECK
PROGRAM MAIN

C DATE: AUGUST 5, 1974
C PROGRAMMER: THIS PROGRAM WAS WRITTEN BY JESSE MAYES, DIVISION OF
C RESEARCH, DEPT. OF TRANS., COMMONWEALTH OF KY., 533 S. LIMESTONE ST.,
C LEXINGTON, KY. PARTS OF THE PROGRAM, INCLUDING THE DYNAMIC
C PROGRAMMING ALGORITHM, HAVE BEEN ADAPTED FROM A PROGRAM WRITTEN BY
C THE STATE OF ALABAMA HIGHWAY DEPT., BUREAU OF MAINT., 1973. SEE
C REPORT "CORRECT: COST/BENEFIT OPTIMIZATION FOR THE REDUCTION OF ROAD
C ENVIRONMENT CAUSED TRAGEDIES".
C PURPOSE: THIS PROGRAM CALCULATES COSTS AND BENEFITS FOR EACH
C ALTERNATIVE AT EACH LOCATION THEN DETERMINES THE OPTIMAL SOLUTION
C SET OF ALTERNATIVES TO BE IMPLEMENTED FOR A GIVEN RANGE OF BUDGETS.
C INPUT AND OUTPUT: SEE DIVISION OF RESEARCH REPORT: "OPTIMAL HIGHWAY
C SAFETY IMPROVEMENTS BY DYNAMIC PROGRAMMING".
C DIMENSION ORET(64,401), NDE(64,401)
C DIMENSION TITL(20), XLOC(64,5), NDE(64), C(64,11), R(64,11), LOC(64)
C NINP = 401
C NINP = NUMBER OF INCREMENTS --- MAXIMUM BUDGET EQUALS NINP*XINC
C NLOC = MAXIMUM NUMBER OF LOCATIONS
C INN = 5
C IOUTPR = 6
C INN, IOUTPR = LOCAL INPUT AND OUTPUT DEVICE NUMBERS
C DIMENSION ORET(64,401), NDE(64,401)
C TITL(20), XLOC(64,5), NDE(64), C(64,11), R(64,11), LOC(64)
C NINP = 401
C NINP = NUMBER OF INCREMENTS --- MAXIMUM BUDGET EQUALS NINP*XINC
C NLOC = MAXIMUM NUMBER OF LOCATIONS
C INN = 5
C IOUTPR = 6
C INN, IOUTPR = LOCAL INPUT AND OUTPUT DEVICE NUMBERS

1000 FORMAT (20A4)
1010 FORMAT (20X,20A4) TITL
1020 FORMAT (T1,14,T5,F6.0,T11,2I4)
   CALL COSREN(C,B,XLOC,LOC,NDE,NSTG,NLOC,XINC,INN,IOUTPR,KIK)
   IF (KIK.EQ.1) GO TO 10
   CALL DYNAM(C,B,LOC,XLOC,NDE,NSTG,XINC,K1,K2,NINP,NLOC,
   + ORT1,NDE,IOUTPR)
10 CONTINUE
   CALL EXIT
END
SUBROUTINE COSBEN

SUBROUTINE COSBEN(POWC, PWI, XLNC, NDE, NSTG, NLNC, XINC, INN, IOUTPR)

+ KIK

C THIS SUBROUTINE CALCULATES PRESENT WORTH COSTS AND BENEFITS

C ASSOCIATED WITH EACH ALTERNATIVE AT EACH LOCATION.

C DIMENSION XLNC(NLNC,5),SEV(4,8),CSFF(10,11),R(A),

+ NDE(NLNC),PW(NLNC,11),PWR(NLNC,11),LOC(NLNC)

READ(IINN,1000) CFAT,CINJ,CPDQ,RATEIN,RATFGR

1000 FORMAT(AF10,01)

WRITE(IOUTPR,1010) CFAT,CINJ,CPDQ,RATEIN,RATFGR

1010 FORMAT(1H1)

C THE ABOVE READS AND PRINTS THE BASIC PARAMETERS CONSTANT FOR THE

C WHOLE PROGRAM.

KIK = 0

C BELOW IS THE INPUT WHICH IS EXECUTED FOR EACH ACCIDENT LOCATION.

10 READ(IINN,1070) NO1,XLNC(NUMBER,1),I=1,5),TIME,NMO,NYR,NCAU

1070 FORMAT(I4,3RF2,01)

C ROUTINE TO CHECK CARD SEQUENCE CODE.

IF(NMO-N01) GO TO 30

WRITE(IOUTPR,1030) N01,N02

1030 FORMAT(1H1)

C SECOND CARD INPUT FOR EACH CRITICAL LOCATION (SEVERITIES).

READ(IINN,1070) NO2,(SEV(I,J),J=1,4),I=1,NCAU)

1070 FORMAT(I4,3RF2,01)

C ROUTINE TO CHECK CARD SEQUENCE CODE.

IF(NO1-NO2) 50,60,50

50 WRITE(IOUTPR,1050) NO1,(SEV(NUMBER,1),I=1,5),TIME,NMO,NYR,NCAU

1050 FORMAT(I4,3RF2,01)

C ROUTINE TO CALCULATE TOTALS.

WRITE(IOUTPR,1090) TOT1=TOT1+SEV(I,1)

1090 FORMAT(1H1)

C OUTPUT OF SEVERITIES AND TOTALS.

WRITE(IOUTPR,1090) TOT1=TOT1+SEV(I,1)
TOT2=TOT2+ SEV(I,2)
TOT3=TOT3+ SEV(I,3)
TOT4=TOT4+ SEV(I,4)
RO CONTINUE
WRITE(IOUTPR,1110) TOT1,TOT2,TOT3,TOT4
1110 FORMAT(10I5)
C INPUT NEXT SET OF NALT CARDS, ONE FOR EACH ALTERNATIVE
NJ=0,NALT
DO 110 I=1,NALT
READ(INN,1120) NO3,(CSEF(I,J), J=1,NJ)
1120 FORMAT(3I5,10F10.0)
IF(NO3-NO1)90,100,90
90 WRITE(IOUTPR,1080) NO1, NO3
KIK = 1
GO TO 190
100 CONTINUE
110 CONTINUE

C OUTPUT OF ALTERNATIVE INFORMATION.
WRITE(IOUTPR,1130)(I,I=1,NALT)
1130 FORMAT(' ALTERNATIVE COST LIFE MAINTAINANCE TOTAL COST EFFECT ON... ',15B15)
C NUMBER COUNT CHECK OF SEVERITIES.
DO 120 I=1,NALT
WRITE(IOUTPR,1140) I,(CSEF(I,J), J=1,NJ)
1140 FORMAT(1I3,10F10.0)
120 CONTINUE
C COMPUTATION OF R(I), THE ITH ALTERNATIVE BENEFIT.
DO 130 I=1,NALT
B(I) = 0.
DO 130 J=1,NCAU
JFFT = J + 3
R(I) = R(I) + (CFAT*SEV(J,2)+LINJ*SEV(J,3)+CPDO*SEV(J,4))
+ CSEF(I,JFFT)
130 CONTINUE
140 CONTINUE
C CALCULATION OF BENEFIT/COSTS AND OUTPUT.
WRITE(IOUTPR,1150)
1150 FORMAT(' ALTERNATIVE MAINTENANCE TOTAL COST BENEFIT/COST ',15B15)
DO 150 I=1,NALT
R(I)=R(I)*CSEF(I,2)/TIME
RNCS = B(I)/CSEF(I,1)
WRITE(IOUTPR,1160) I,CSEF(I,1),R(I),RNCS
1160 FORMAT(1I3,1F10.0,1F10.0)
150 CONTINUE
WRITE(IOUTPR,1170)
1170 FORMAT(' ALTERNATIVE MAINTENANCE TOTAL COST BENEFIT/COST ',15B15)
DO 170 I=1,NALT
XMAX=CSEF(I,2)*CSEF(I,3)
RNCS = B(I)/(CSEF(I,1)*XMAX)
TCMST=CSEF(I,1)*XMAX
WRITE(IOUTPR,1190) I,XMAX,TCMST,RNCS
1190 FORMAT(1I3,1F10.0,1F10.0,1F10.0)
170 CONTINUE
WRITE(IOUTPR,1200)
1200 FORMAT(' ALTERNATIVE MAINTENANCE TOTAL COST BENEFIT ',15B15)
DO 190 I=1,NALT
X = (1.+RATEIN)**LIFF
190 CONTINUE
WRITE(IOUTPR,1210)
1210 FORMAT(' ALTERNATIVE MAINTENANCE TOTAL COST ',15B15)
PWF = (X-1.)/(RATEIN*X)  
Y = (1.+RATEGR)/(1.+RATEIN)  
PWXGR = (Y*(LIFE+1)-1.)/(Y-1.) - 1  
PWC(NUMBER,1) = 0  
PWB(NUMBER,1) = 0  
PWMMAIN = PWF*CSEF(1,3)  
PWC(NUMBER,I+1) = CSEF(I,1) + PWMAIN  
CROUND PRESENT WORTH COSTS TO NEAREST INCREMENT  
IPWC = PWC(NUMBER,I+1)/XINC + .5  
PWC(NUMBER,I+1) = IPWC*XINC  
PWB(NUMBER,I+1) = PWXGR*R(I)/LIFE  
PWB(Number,I+1) = PWR(NUMBER,1+1)/PWC(NUMBER,1+1)  
WRITE(IOUTPR,1220) I,PWMAIN,PWC(NUMBER,I+1),PWB(NUMBER,I+1),PWBC  
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('1','40(''*''),' WARNING '40(''*''))  
WRITE(IOUTPR,1240) NUMBER,NSTG  
1240 FORMAT('1','8,'NUMBER OF LOCATIONS READ = '13//,'NUMBER OF LOCATIONS EXPECTED = '13)  
190 CONTINUE  
RETURN  
END
SUBROUTINE DYNAM

SUBROUTINE DYNAM(C,B,LOC,XLOC,NDE,NSTG,XINC,K1,K2,NINP,NLOC,
+ ORET,NDR,IOUTPR)  

C THIS SUBROUTINE USES "DYNAMIC PROGRAMMING" TO FIND THE OPTIMAL
C SOLUTION SET ALTERNATIVES (NIP 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),NDR(NLOC,NINP),NDE(NLOC),
+ C(NLOC,11),R(NLOC,11),R(11),XLOC(NLOC,5),LOC(NLOC)

IST=0
VRET=0.0
WRITE(IOUTPR,1130)
WRITE(IOUTPR,1000)
1000 FORMAT(*,40('**'),'PARAMETER VALUES',40('**')///)
WRITE(IOUTPR,1010)
1010 FORMAT(*,27X,1R(*'),1R('OUTPUT',1R('')))
WRITE(IOUTPR,1020) NSTG,XINC,K1,K2
1020 FORMAT(5X,'LOCATIONS---INCREMENT---LOWER LIMIT---INCREMENT
+ DEP',3X,12.2,I9,10X )
10 CONTINUE
WRITE(IOUTPR,1030) LOC(I),NDE(I)
1030 FORMAT(7X,I5,I10)

10 CONTINUE
WRITE(IOUTPR,1040)
1040 FORMAT(*,30('**'),'LOCATIONS,ALTERNATIVES,COSTS AND BENEFITS',
+ 30('**')///)
WRITE(IOUTPR,1050)
1050 FORMAT (1H,'---LOCATION---LOCATION NAME--------ALT-NUM--------COST---RATIO')

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

DO 140 I=1,NSTG
NDE=NDE(I)+1
R(I)=0.
DO 20 IC=2,NDE
20 R(IC) = R(I,IC)
DO 50 IC=2,NDE
ICM1 = IC-1
ACRAT = R(IC)/C(I,IC)
WRITE(IOUTPR,1060) LOC(I),XMOC(I,J),J=1,5,ICM1,C(I,IC),R(IC),
+ ACRAT
1060 FORMAT(19,5X,5A4,I6,3X,F11.0,F11.0,4X,F10.2)
30 CONTINUE
1070 FORMAT(8F10.0)
DO 130 J=1,NINP
130 CONTINUE

C DETERMINE THE BEST ALTERNATIVE--NODE(1,J)--AT J-TH LOCATION GIVEN
C J-1 INCREMENTS TO SPEND ON LOCATION 1 THRU LOCATION J-----YIELDING
C A RETURN OF--ORET(I,J)--

DO 120 K=1,NDEC
CALL XNUT(1,IST,XIN,K,KICK,XINC,C,NLOC)
120 CONTINUE
30 CONTINUE
40 GO TO 120
40 CONTINUE

40 CONTINUE
50 CONTINUE
60 TEST=R(K)
GO TO 80
60 TEST=R(K)+ORET(1-1,IST)
GO TO 80
70 TEST=R(K)+ORET(1-1,IST)
GO TO 80
50 CONTINUE
40 CONTINUE
30 CONTINUE
20 CONTINUE
10 CONTINUE

C END OF--DYNAM(C,B,LOC,XLOC,NDE,NSTG,XINC,K1,K2,NINP,NLOC,'
GO TO 90
80 IF((DUM-TEST))90,100,100
90 DUM=TEST
   ORET(i,j)=DUM
   NOD(i,j)=K
100 GO TO 110
110 CONTINUE
120 CONTINUE
130 CONTINUE
140 CONTINUE

IPAGF = 0
C WRITE MAIN BUDGET OUTPUT HEADING
WRITE(IOUTPR,1080)
   1080 FORMAT('1*',90('**'),*','37('**'),*',' BUDGET OUTPUT ','37('**'),*',' + 90('**')/100/)
   DO 160 K=K1,NINP,K2
   J=M
   XIN=(J-1)*XINC
   IPAGF = IPAGE + 1
   IF(IPAGE.NE.1) WRITE(IOUTPR,1130)
C WRITE INDIVIDUAL BUDGET OUTPUT HEADING
   WRITE(IOUTPR,1090)
   1090 FORMAT('1*','15X,'BUDGET LOCATION =','4X,'LOCATION NAME + ','4X,'ALT-NUM ','5X,'COST ','4X,'RETURN ','4X,'ACCUM RETURN ')
   WRITE(IOUTPR,1100) XIN
   1100 FORMAT('1*','6X,F15.2)
   TOTCST = 0
   TOTRTN = 0
   DO 150 L=1,NSTG
   J=NSTG+1-L
   K=NOO(I,J)
   KK=NOO(I,J)-1
   TOTCST = TOTCST + C(I,K)
   TOTRTN = TOTRTN + R(I,K)
   C WRITE I-TH LOCATION INFORMATION--TOTAL BUDGET OF M INCRMNTS
   WRITE(IOUTPR,1110) LOC(I),XLOC(I,J),JJ=1,5),KK,C(I,K),
   + R(I,K),ORET(I,J)
   1110 FORMAT('1*','24X,'14,9X,5A4,5X,I4,2F12.0,4X,F12.0)
   CALL XOUT(I,IST,XIN,K,KICK,XINC,C,NLOC)
   J=IST
   XIN = XIN-C(I,K)
150 CONTINUE
C WRITE TOTALS
   WRITE(IOUTPR,1120) TOTCST,TOTRTN,ORET(NSTG,M)
   1120 FORMAT('1*','29('**'), 'TOTALS ','29('**'),2F12.0,4X,F12.0)
160 CONTINUE
170 WRITE(IOUTPR,1130)
   1130 FORMAT('1*')
180 CONTINUE
RETURN
END
SUBROUTINE XOUT

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

THIS SUBROUTINE CALCULATES THE OUTPUT STATE NUMBER RESULTING FROM THE INPUT XIN AND SAFETY MEASURE K. IT ALSO DETERMINES THE COST OF A PARTICULAR SAFETY MEASURE CORRESPONDING TO STAGE I.

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
APPENDIX B

VARIABLE LIST
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
KIK  Kickout variable — value of 1 terminates program

**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)

- ***RATEGRI**  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(1,1) — Total number of accidents
SEV(2,1) — Number of fatal accidents
SEV(3,1) — Number of nonfatal injury accidents
SEV(4,1) — 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(J,I) — Effect (percent reduction) on cause \(J\); \(J = 4, 5, \ldots\)

*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)
Pigman, Agent, Mayes, and Zegeer

*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

TOTRTN  
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
PROGRAM MAIN

/ MAIN /

{01

NIPF = 401

NIPF = NUMBER OF INCREDITS--RAIDS
BUDGET EQUALS NIPF*INC

{02

BLOC = 64

BLOC = MAXIMUM NUMBER
OF LOCATIONS

{03

IXX = 5

IOTPF = 6

IIXX, IOTPF = LOCAL
INPUT AND OUTPUT
DEVICE NUMBERS

{04

READ FROM DEV
S/ VIA FORMAT
TOO INTO THE LIST

{05

NOTE 05

* LIST = TITLE

* ************

{06

WRITE TO DEV
S/ VIA FORMAT
TOO FROM THE LIST

NOTE 07

* LIST = TITLE

* ************

{08

READ FROM DEV
S/ VIA FORMAT
TOO INTO THE LIST

NOTE 08

* LIST = AUTOC.

* TITLE, R1, R2

* ************

10

| CODEX |

1. | (C, E, BLOC, LOC, E |

0. | BLOC, AUTOC, AUTOC, |

11 | LIST, (R1, R2, |

12 | IOTPF, R10 |

13

7 D/RAX

| (C, D, AUTOC, BLOC, E |

10 | BLOC, AUTOC, AUTOC, |

11 | LIST, (R1, R2, |

12 | IOTPF, R10 |

14

CONTINUE |

15

EXIT |

CALL 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 in an integer field of two)

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);
   his 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 datadeck MUST be blank.
APPENDIX E

SAMPLE INPUT AND OUTPUT
## Sample of Input Data

<table>
<thead>
<tr>
<th>1</th>
<th>11</th>
<th>21</th>
<th>31</th>
<th>41</th>
<th>51</th>
<th>61</th>
<th>71</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>61250.0</td>
<td>41</td>
<td>40</td>
<td>45000.0</td>
<td>2700.0</td>
<td>400.0</td>
<td>0.10</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>163-25-10.9</td>
<td>14</td>
<td>12</td>
<td>21</td>
<td>1</td>
<td>100010</td>
<td>100.62</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Test Run 1** Dynamic Programming

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)

REF NO
1 63-25-10-9

ACCIDENT HISTORY 1.00 YEARS, MONTH 1, YEAR 69, 1 CAUSE

ROADWAY CAUSE\n1 4\nTOTALS 4

ALTERNATIVE\nCOST\n1 1000.00

BENEFIT/COST ANALYSIS

ALTERNATIVE\n1 1000.00

BENEFIT/COST ANALYSIS, MAINTENANCE INCLUDED

ALTERNATIVE\n1 1000.00

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

ALTERNATIVE\n1 1000.00

PARAMETER VALUES

LOCATIONS --- INCREMENT --- LOWER LIMIT --- INCREMENTS PER STEP
1 250.00 41 40
<table>
<thead>
<tr>
<th>ALTERNATIVES SELECTED FOR $100,000 BUDGET</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BUDGET LOCATION</strong></td>
</tr>
<tr>
<td>50000.00</td>
</tr>
<tr>
<td>56</td>
</tr>
<tr>
<td>59</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>57</td>
</tr>
<tr>
<td>56</td>
</tr>
<tr>
<td>55</td>
</tr>
<tr>
<td>54</td>
</tr>
<tr>
<td>53</td>
</tr>
<tr>
<td>52</td>
</tr>
<tr>
<td>51</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>49</td>
</tr>
<tr>
<td>48</td>
</tr>
<tr>
<td>47</td>
</tr>
<tr>
<td>46</td>
</tr>
<tr>
<td>45</td>
</tr>
<tr>
<td>44</td>
</tr>
<tr>
<td>43</td>
</tr>
<tr>
<td>42</td>
</tr>
<tr>
<td>41</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>39</td>
</tr>
<tr>
<td>38</td>
</tr>
<tr>
<td>37</td>
</tr>
<tr>
<td>36</td>
</tr>
<tr>
<td>35</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>33</td>
</tr>
<tr>
<td>32</td>
</tr>
<tr>
<td>31</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

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

| **50000.00** | **868692.0** | **868692.0** |