

Research Report
412

**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 "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):

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

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.

$$B = \left[\frac{[(1 + t)^{L+1} - 1]}{(1 + i)} - 1 \right] / \left[\frac{[(1 + t) - 1]}{(1 + i)} - 1 \right] \beta \quad 2$$

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

$$\beta = \left(\sum_{m=1}^J \sum_{n=1}^3 a_m N_{mn} \gamma_n \right) / T \quad 3$$

where β = 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
 γ_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, $j = 1, 2, \dots, N$; i.e.,

$$O_1(j) = R_1(j) \quad 4$$

where $O_1(j)$ = 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.

$$O_i(j) = \text{Max} [(R_i(k) + O_{i-1}(j - k)], \quad 5$$

$$j = 1, 2, \dots, N \text{ and}$$

$$k = 1, 2, \dots, j$$

where M = number of locations considered,

$O_i(j)$ = 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 ($k \leq j$),
 $O_{i-1}(j - k)$ = 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 1, 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

1. *Performance Budgeting System for Highway Maintenance Management*, NCHRP Report 131, Highway Research Board, 1972.
2. Bellman, R., *Dynamic Programming*, Princeton University Press, 1957.
3. Johnson, M. M., Dare, C. E., and Shinner, H. B., *Dynamic Programming of Highway Safety Projects*, *Transportation Engineering Journal*, ASCE, November 1971.
4. de Neufville, R., and Mori, Y., *Optimal Highway Staging by Dynamic Programming*, *Transportation Engineering Journal*, ASCE, February 1970.
5. Funk, M. L. and Tillman, F. A., *Optimal Construction Staging by Dynamic Programming*, *Journal of the Highway Division*, ASCE, November 1968.
6. *Evaluation of Criteria for Safety Improvements on the Highway*, Roy Jorzensen and Associates, Westat Research Analysts, Inc., 1966.
7. Lorie, J. H., and Savage, L. J., *Three Problems in Rationing Capital*, *Journal of Business*, Vol 28, No. 4, October 1955.
8. *Kentucky Highway Budgeting*, Bureau of Business Research, College of Commerce, University of Kentucky, 1963.
9. *Budgeting Support for Kentucky Highways by Systems*, Bureau of Business Research, College of Commerce, University of Kentucky, 1963.
10. Agent, K. R., *Evaluation of the High-Accident Location Spot-Improvement Program in Kentucky*, Kentucky Bureau of Highways, 1973.
11. Zegeer, C. V., *Identification of Hazardous Locations on Rural Highways in Kentucky*, Kentucky Bureau of Highways, 1974.
12. *Correct: Cost/Benefit Optimization for the Reduction of Roadway Environment Caused Tragedies*, Phase II, State of Alabama Highway Department, 1973.
13. Winfrey, R., *Economic Analysis for Highways*, 1969.
14. *Traffic Safety Memorandum No. 113*, National Safety Council, July 1971.

TABLE 1
SUMMARY OF IMPROVEMENT
COSTS AND BENEFITS

	TYPE OF IMPROVEMENT	NUMBER OF PROJECTS	TOTAL ACCIDENT REDUCTION (PERCENT)	SERVICE LIFE (YEARS)	ANNUAL MAINTENANCE COST (\$)
General	Signs and Markings	9	36	3	25
	Warning Signs	23	35	5	25
	Regulatory Signs	16	22	5	25
	Guidance Signs	10	14	5	25
	Sign Combinations	16	20	5	25
	Markings	8	16	2	0
	Sight Distance Imp.	9	28	2	50
	Post Delineators	3	25	5	20
	Comb. Delineators, Markings, Signs, Maintenance	11	22	5	25
	Shoulder Improvements	7	23	10	100
	Comb. Resurface, Patching, Drainage, Deslick, Culvert	22	16	10	100
	Rumble Strips	8	29	5	0
	Remove Median Crossovers	2	29	20	0
	Lighting	1	-58	10	500
	Lighting & Rumble Strips	1	17	7	300
	Rumble Strips & Beacon	2	32	7	50
	Side Road Sign Only	31	19	5	25
	Prepare for Sudden Stop Sign Only	19	25	5	25
	Side Road Sign & Warning Sign	15	27	5	25
	Curves	Signing	34	30	5
Post Delineators		4	32	5	25
Signs & Delineators		16	28	5	25
Signs & Maintenance		6	47	3	25
Comb. Delineators, Markings, Signs, Maintenance		16	24	5	25
Resurfacing, Patch, Drainage, Deslick, Super, Culvert, Guardrail		22	33	10	100
Re-alignment (Relocate)		3	32	20	100
Intersections	Signs & Markings	21	24	3	25
	Warning Signs	11	27	5	25
	Regulatory Signs	5	48	5	25
	Regulatory & Warning Signs	20	16	5	25
	Markings	17	16	2	0
	Marking, Maintenance, & Signing	9	35	5	25
	Channelization - Storage Lane	13	15	10	100
	Channelization & Signs	2	37	7	75
	Install Beacons	13	2	10	100
	Upgrade Beacons	10	5	10	100
	Install Signals	10	23	10	300
Upgrade Signals	2	18	10	250	
	Total Improvements	447	24		

TABLE 2
BENEFIT DATA FOR DYNAMIC PROGRAMMING
BENEFIT-COST ANALYSIS PROGRAM

LOCATION	LOCATION	NAME	ALTERNATIVE	NO.	COST	RETURN	B/C
1	63-75-10.9	1500.	1	1	1500.	7620.	5.08
2	30-60-15.1	4250.	1	1	1507.7	37.34	0.24
3	30-60-15.2	2000.	1	1	11385.	5.69	0.27
4	54-41-12.0	500.	1	1	4792.	19.56	0.40
5	73-45-6.5	500.	1	1	973.	1.95	0.40
6	73-45-7.2	500.	1	1	0.	0.0	0.0
7	73-45-7.2	750.	1	1	0.	0.0	0.0
8	48-42-14.4	500.	2	2	544.	1.18	0.24
9	20-51-1.1	750.	1	1	2487.	3.28	0.44
10	102-25-9.7	1000.	1	1	224.	0.45	0.45
11	30-60-15.2	500.	1	1	10145.	32.92	65.7
12	30-60-15.2	4500.	2	2	3524.	0.74	0.16
13	30-60-15.2	5000.	1	1	3166.	0.37	0.08
14	72-64-16.5	750.	1	1	4054.	5.42	7.3
15	30-60-4.1	1250.	1	1	814.	0.65	0.16
16	34-71-6.5	4500.	1	1	2623.	0.78	0.17
17	70-61-13.5	750.	1	1	2448.	3.21	4.3
18	114-31-16.2	500.	1	1	1422.	1.86	2.4
19	114-31-16.2	500.	1	1	1422.	1.86	2.4
20	114-31-16.2	750.	1	1	4071.	5.43	7.3
21	63-75-10.9	1500.	1	1	7620.	1.93	2.5
22	63-75-10.9	1500.	1	1	7620.	1.93	2.5
23	63-75-10.9	1500.	1	1	7620.	1.93	2.5
24	63-75-10.9	1500.	1	1	7620.	1.93	2.5
25	63-75-10.9	1500.	1	1	7620.	1.93	2.5
26	63-75-10.9	1500.	1	1	7620.	1.93	2.5
27	63-75-10.9	1500.	1	1	7620.	1.93	2.5
28	63-75-10.9	1500.	1	1	7620.	1.93	2.5
29	63-75-10.9	1500.	1	1	7620.	1.93	2.5
30	63-75-10.9	1500.	1	1	7620.	1.93	2.5
31	63-75-10.9	1500.	1	1	7620.	1.93	2.5
32	63-75-10.9	1500.	1	1	7620.	1.93	2.5
33	63-75-10.9	1500.	1	1	7620.	1.93	2.5
34	63-75-10.9	1500.	1	1	7620.	1.93	2.5
35	63-75-10.9	1500.	1	1	7620.	1.93	2.5
36	63-75-10.9	1500.	1	1	7620.	1.93	2.5
37	63-75-10.9	1500.	1	1	7620.	1.93	2.5
38	63-75-10.9	1500.	1	1	7620.	1.93	2.5
39	63-75-10.9	1500.	1	1	7620.	1.93	2.5
40	63-75-10.9	1500.	1	1	7620.	1.93	2.5
41	63-75-10.9	1500.	1	1	7620.	1.93	2.5
42	63-75-10.9	1500.	1	1	7620.	1.93	2.5
43	63-75-10.9	1500.	1	1	7620.	1.93	2.5
44	63-75-10.9	1500.	1	1	7620.	1.93	2.5
45	63-75-10.9	1500.	1	1	7620.	1.93	2.5
46	63-75-10.9	1500.	1	1	7620.	1.93	2.5
47	63-75-10.9	1500.	1	1	7620.	1.93	2.5
48	63-75-10.9	1500.	1	1	7620.	1.93	2.5
49	63-75-10.9	1500.	1	1	7620.	1.93	2.5
50	63-75-10.9	1500.	1	1	7620.	1.93	2.5
51	63-75-10.9	1500.	1	1	7620.	1.93	2.5
52	63-75-10.9	1500.	1	1	7620.	1.93	2.5
53	63-75-10.9	1500.	1	1	7620.	1.93	2.5
54	63-75-10.9	1500.	1	1	7620.	1.93	2.5
55	63-75-10.9	1500.	1	1	7620.	1.93	2.5
56	63-75-10.9	1500.	1	1	7620.	1.93	2.5
57	63-75-10.9	1500.	1	1	7620.	1.93	2.5
58	63-75-10.9	1500.	1	1	7620.	1.93	2.5
59	63-75-10.9	1500.	1	1	7620.	1.93	2.5
60	63-75-10.9	1500.	1	1	7620.	1.93	2.5
61	63-75-10.9	1500.	1	1	7620.	1.93	2.5

TABLE 3
INPUT DATA FOR DYNAMIC PROGRAMMING
BENEFIT-COST ANALYSIS COMPARISON
(EXAMPLE PROBLEM)

LOCATION NUMBER	ALTERNATE NUMBER	COSTS	BENEFITS	BENEFIT-COST RATIO
1	1	\$1000	\$20000	20
2	1	1000	15000	15
3	1	1000	12000	12
4	1	3000	30000	10
5	1	5000	45000	9
6	1	10000	80000	8
7	1	1000	7000	7
8	1	9000	54000	6
9	1	2000	6000	3
10	1	1000	2000	2
		\$34000	\$271000	

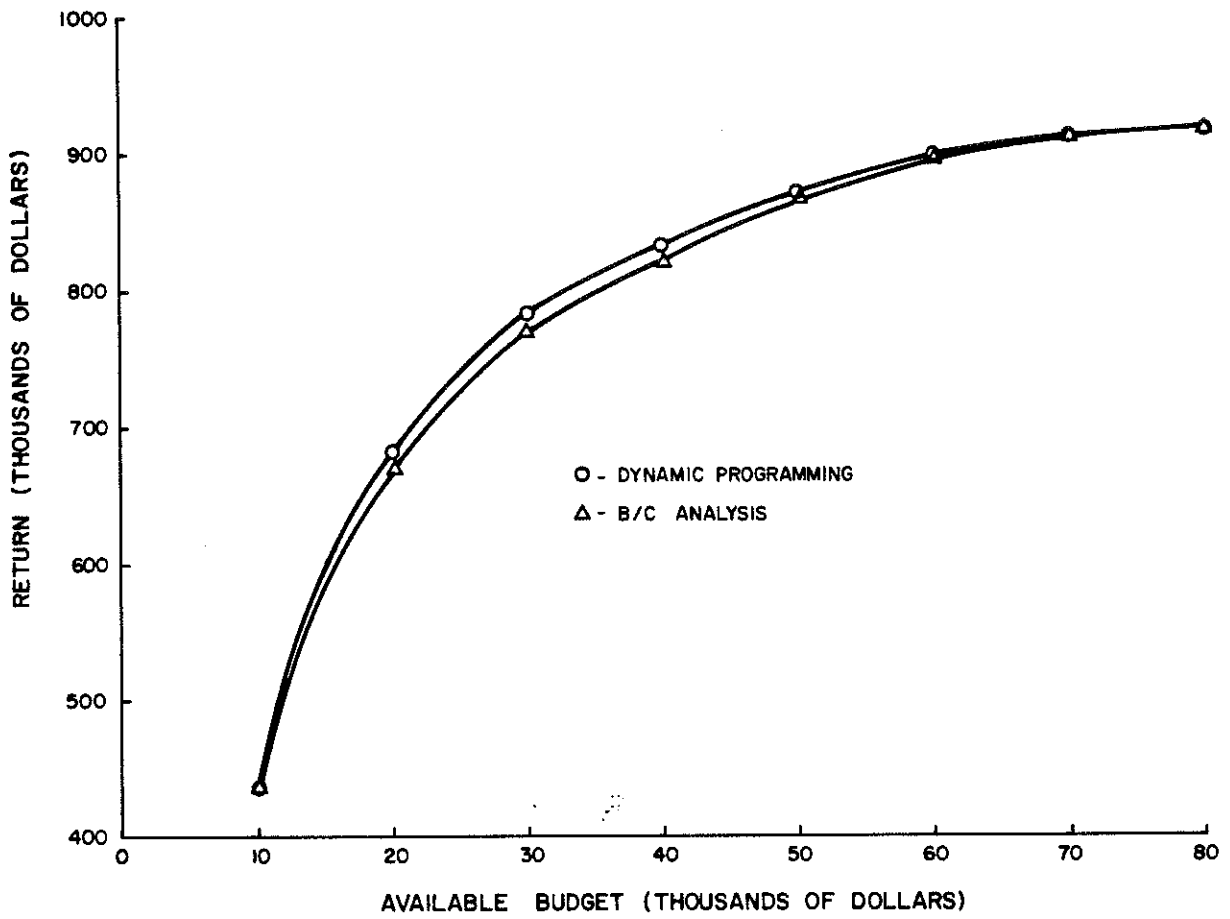


Figure 1. Expected Return versus Available Budget for Dynamic Programming -- Benefit-Cost Analyses

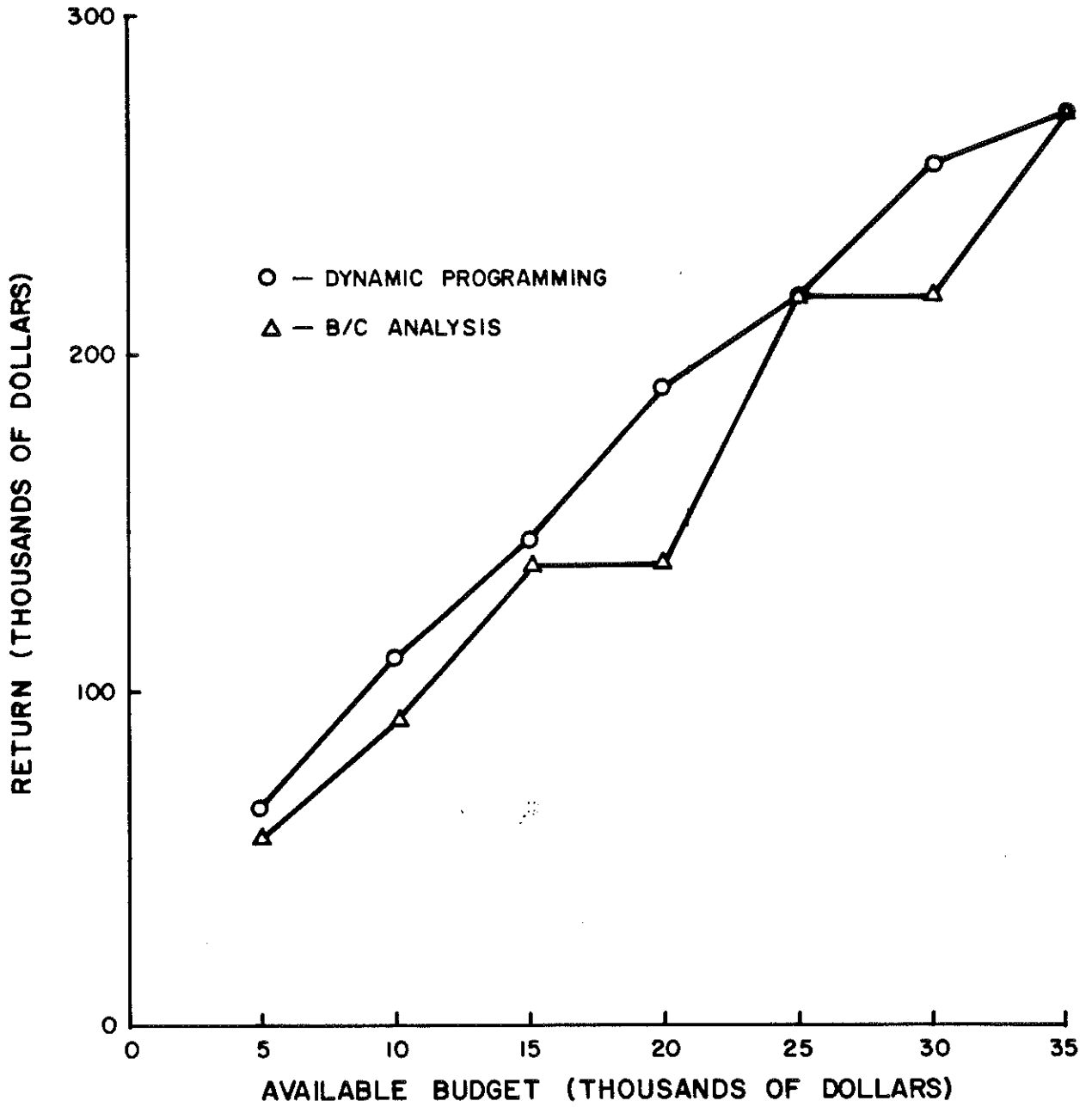


Figure 2. Expected Return versus Available Budget for Dynamic Programming -- Benefit Cost Analyses (Example Problem)

APPENDIX A

PROGRAM SOURCE DECK

SUBROUTINE COSBEN

```

SUBROUTINE COSBEN(PWC,PWR,XLOC,LOC,NDE,NSTG,NLOC,XINC,INN,IOUTPR, COSB0010
+ KIK) COSB0020
C THIS SUBROUTINE CALCULATES PRESENT WORTH COSTS AND BENEFITS COSB0030
C ASSOCIATED WITH EACH ALTERNATIVE AT EACH LOCATION COSB0040
  DIMENSION XLOC(NLOC,5),SEV(4,8),CSFF(10,11),R(8), COSB0050
  + NDE(NLOC),PWC(NLOC,11),PWR(NLOC,11),LOC(NLOC) COSB0060
  READ(INN,1000) CFAT,CINJ,CPDD,RATEIN,RATEGR COSB0070
1000 FORMAT (8F10.0) COSB0080
  WRITE(IOUTPR,1010) CFAT,CINJ,CPDD,RATEIN,RATEGR COSB0090
1010 FORMAT(' NEG UTILITY FATALITY=',F7.0,' INJURY=',F6.0,' PRP DM=' COSB0100
+,F5.0/' INTEREST RATE = ',F5.3/' EXPONENTIAL GROWTH RATE = ',F5.3/ COSB0110
+////) COSB0120
C THE ABOVE READS AND PRINTS THE BASIC PARAMETERS CONSTANT FOR THE COSB0130
C ENTIRE PROGRAM COSB0140
  NUMBER = 1 COSB0150
  KIK = 0 COSB0160
C BELOW IS THE INPUT WHICH IS EXECUTED FOR EACH ACCIDENT LOCATION. COSB0170
10 READ(INN,1020) NO1,(XLOC(NUMBER,I),I=1,5),TIME,NMO,NYR,NCAU COSB0180
1020 FORMAT(I4,5A4,47X, F4.0,I2,I2,I1) COSB0190
  LOC(NUMBER) = NO1 COSB0200
  IF(NO1)20,180,20 COSB0210
20 CONTINUE COSB0220
  WRITE(IOUTPR,1030) COSB0230
1030 FORMAT(1H1) COSB0240
  WRITE(IOUTPR,1040) COSB0250
1040 FORMAT(' REF NO') COSB0260
  IF(NCAU.EQ.1) GO TO 30 COSB0270
  WRITE(IOUTPR,1050) NO1,(XLOC(NUMBER,I),I=1,5),TIME,NMO,NYR,NCAU COSB0280
1050 FORMAT(3X,I4,8X,5A4, //9X,'ACCIDENT HISTORY ',F4.2, COSB0290
+' YEARS. MONTH ',I2,', YEAR ',I2,', ',I1,' CAUSE. ') COSB0300
  GO TO 40 COSB0310
30 WRITE(IOUTPR,1060) NO1,(XLOC(NUMBER,I),I=1,5),TIME,NMO,NYR,NCAU COSB0320
1060 FORMAT(3X,I4,8X,5A4, //9X,'ACCIDENT HISTORY ',F4.2, COSB0330
+' YEARS. MONTH ',I2,', YEAR ',I2,', ',I1,' CAUSE. ') COSB0340
40 CONTINUE COSB0350
C SECOND CARD INPUT FOR EACH CRITICAL LOCATION (SEVERITIES). COSB0360
  READ(INN,1070)NO2,((SEV(I,J),J=1,4),I=1,NCAU),ALT COSB0370
1070 FORMAT(I4,38F2.0) COSB0380
  NALT=ALT/10.+1 COSB0390
  NDE(NUMBER) = NALT COSB0400
C ROUTINE TO CHECK CARD SEQUENCE CODE. COSB0410
  IF(NO1-NO2) 50,60,50 COSB0420
50 WRITE(IOUTPR,1080)NO1,NO2 COSB0430
1080 FORMAT(' SEQUENCE/CODE NO. ERROR. CHECK ',I5,' AND',I5, COSB0440
+' **EXECUTION TERMINATED') COSB0450
  KIK = 1 COSB0460
  GO TO 190 COSB0470
60 CONTINUE COSB0480
C OUTPUT OF SEVERITIES AND TOTALS. COSB0490
  WRITE(IOUTPR,1090) COSB0500
1090 FORMAT(/ ' ROADWAY CAUSE TACC NFAT NINJ NPRO') COSB0510
  TOT1=0 COSB0520
  TOT2=0 COSB0530
  TOT3=0 COSB0540
  TOT4=0 COSB0550
  DO 80 I=1,NCAU COSB0560
  WRITE(IOUTPR,1100) I,(SEV(I,J),J=1,4) COSB0570
1100 FORMAT (1X,I7,F12.0,3F6.0) COSB0580
70 CONTINUE COSB0590
  TOT1=TOT1+ SEV(I,1) COSB0600

```

```

      TOT2=TOT2+ SEV(I,2)
      TOT3=TOT3+ SEV(I,3)
      TOT4=TOT4+ SEV(I,4)
80  CONTINUE
      WRITE(IOUTPR,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+NCAU
      DO 110 I=1,NALT
      READ(INN,1120) NO3,(CSEF(I,J), J=1,NJ)
1120 FORMAT(I4,F7.0,F2.0,F5.0,8F3.2)
      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,NCAU)
1130 FORMAT('/' ALTERNATIVE COST LIFE MAIN COST EFFECT ON...',
+8I5)
C   NUMBER COUNT CHECK OF SEVERITIES.
      DO 120 I=1,NALT
      WRITE(IOUTPR,1140) I,(CSEF(I,J),J=1,NJ)
1140 FORMAT(I7,F13.2,F8.0,F9.2,F24.2,7F5.2)
      120 CONTINUE
C   COMPUTATION OF B(I), THE ITH ALTERNATIVE BENEFIT.
      DO 140 I=1,NALT
      B(I) = 0.
      DO 130 J=1,NCAU
      JFFT = J +3
      B(I) = B(I) + (CFAT*SEV(J,2)+CINJ*SEV(J,3)+CPDN*SEV(J,4))*
+ CSEF(I,JFFT)
      130 CONTINUE
      140 CONTINUE
C   CALCULATION OF BENEFIT/COSTS AND OUTPUT.
      WRITE(IOUTPR,1150)
1150 FORMAT('///5X,' BENEFIT/COST ANALYSIS'///' ALTERNATIVE COST
+ BENEFIT BENEFIT/COST')
      DO 150 I=1,NALT
      B(I)=B(I)*CSEF(I,2)/TIME
      BNCS = B(I)/CSEF(I,1)
      WRITE(IOUTPR,1160)I,CSEF(I,1),B(I),BNCS
1160 FORMAT(I7,F14.2,F14.2,F18.4)
      150 CONTINUE
      WRITE(IOUTPR,1170)
1170 FORMAT(' ///' BENEFIT/COST ANALYSIS, MAINTENANCE INCLUDED')
      WRITE(IOUTPR,1180)
1180 FORMAT(' ALTERNATIVE MAINTENANCE TOTAL COST BENEFIT/COST
+ST' )
      DO 160 I=1,NALT
      XMAIN=CSEF(I,2)*CSEF(I,3)
      BNCM = B(I)/(CSEF(I,1)+XMAIN)
      TMCST=CSEF(I,1)+XMAIN
      WRITE(IOUTPR,1190) I,XMAIN,TMCST,BNCM
1190 FORMAT(I7,F23.2,F14.2,F16.4)
      160 CONTINUE
      WRITE(IOUTPR,1200)
1200 FORMAT(' ///' BENEFIT/COST ANALYSIS, MAINTENANCE INCLUDED ***PREFCOST
+SENT WORTH METHOD***')
      WRITE(IOUTPR,1210)
1210 FORMAT(' ALTERNATIVE MAINTENANCE TOTAL COST BENEFIT
+ BENEFIT/COST')
      DO 170 I=1,NALT
      LIFE = CSEF(I,2)
      X = (1.+RATEIN)**LIFE

```

COSR0610
 COSR0620
 COSR0630
 COSR0640
 COSR0650
 COSR0660
 COSR0670
 COSR0680
 COSR0690
 COSR0700
 COSR0710
 COSR0720
 COSR0730
 COSR0740
 COSR0750
 COSR0760
 COSR0770
 COSR0780
 COSR0790
 COSR0800
 COSR0810
 COSR0820
 COSR0830
 COSR0840
 COSR0850
 COSR0860
 COSR0870
 COSR0880
 COSR0890
 COSR0900
 COSR0910
 COSR0920
 COSR0930
 COSR0940
 COSR0950
 COSR0960
 COSR0970
 COSR0980
 COSR0990
 COSR1000
 COSR1010
 COSR1020
 COSR1030
 COSR1040
 COSR1050
 COSR1060
 COSR1070
 COSR1080
 COSR1090
 COSR1100
 COSR1110
 COSR1120
 COSR1130
 COSR1140
 COSR1150
 COSR1160
 COSR1170
 COSR1180
 COSR1190
 COSR1200
 COSR1210
 COSR1220
 COSR1230
 COSR1240
 COSR1250
 COSR1260

```

PWF = (X-1.)/(RATEIN*X)
Y = (1.+RATEGR)/(1.+RATEIN)
PWEXGR = (Y**LIFE-1.)/(Y-1.) - 1
PWC(NUMBER,1) = 0
PWB(NUMBER,1) = 0
PWMAIN = PWF*CSEF(I,3)
PWC(NUMBER,I+1) = CSEF(I,1) + PWMAIN
C ROUND PRESENT WORTH COSTS TO NEAREST INCREMENT
IPWC = PWC(NUMBER,I+1)/XINC + .5
PWC(NUMBER,I+1) = IPWC*XINC
PWB(NUMBER,I+1) = PWEXGR*B(I)/LIFE
PWBC = PWB(NUMBER,I+1)/PWC(NUMBER,I+1)
WRITE(IOUTPR,1220) I,PWMAIN,PWC(NUMBER,I+1),PWB(NUMBER,I+1),PWBC
1220 FORMAT(I7,F23.2,F14.2,F11.2,5X,F11.2)
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(' ',NUMBER OF LOCATIONS READ = ',13/ ' ',NUMBER OF LOCATIONS
+NS EXPECTED = ',13)
190 CONTINUE
RETURN
END

```

```

COSB1270
COSB1280
COSB1290
COSB1300
COSB1310
COSB1320
COSB1330
COSB1340
COSB1350
COSB1360
COSB1370
COSB1380
COSB1390
COSB1400
COSB1410
COSB1420
COSB1430
COSB1440
COSB1450
COSB1460
COSB1470
COSB1480
COSB1490
COSB1500
COSB1510
COSB1520
COSB1530
COSB1540

```


SUBROUTINE DYNAM

```

SUBROUTINE DYNAM(C,B,LOC,XLOC,NDE,NSTG,XINC,K1,K2,MINP,NLOC,
+ ORET,NOD,IOUTPR)
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,MINP),NOD(NLOC,MINP),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,18('-'),'OUTPUT',18('-'))
WRITE(IOUTPR,1020) NSTG,XINC,K1,K2
1020 FORMAT(5X,'LOCATIONS---INCREMENT---LOWER LIMIT---INCREMENTS PER STDYNA0170
+EP',/,3X,I9,3X,F12.2,I9,10X,I9,/, '-----LOCATION---ALTERNATIVES')
DO 10 I=1,NSTG
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-----COSTDYN0270
+-----RETURN-----B/C RATIO')
C FIND THE OPTIMAL ALTERNATIVE AT THE I-TH LOCATION WITH J INCREMENTS
C AVAILABLE
DO 140 I=1,NSTG
NDEC=NDE(I)+1
R(1)=0.
DO 20 IC=2,NDEC
20 R(IC) = R(I,IC)
DO 30 IC=2,NDEC
ICM1 = IC-1
BCRAT = R(IC)/C(I,IC)
WRITE(IOUTPR,1060) LOC(I),(XLOC(I,J),J=1,5),ICM1,C(I,IC),R(IC),
+ BCRAT
1060 FORMAT(I9,5X,5A4,I6,3X,F11.0,F11.0,4X,F10.2)
30 CONTINUE
1070 FORMAT(8F10.0)
DO 130 J=1,MINP
C INCREMENT BUDGET
XIN=(J-1)*XINC
DUM=-1000000000000.
NDEC=NDE(I)+1
C DETERMINE THE BEST ALTERNATIVE--NOD(I,J)--AT I-TH LOCATION GIVEN
C J-1 INCREMENTS TO SPEND ON LOCATION 1 THRU LOCATION I-----YIELDING
C A RETURN OF--ORET(I,J)--
DO 120 K=1,NDEC
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)
DYN00010
DYN00020
DYN00030
DYN00040
DYN00050
DYN00060
DYN00070
DYN00080
DYN00090
DYN00100
DYN00110
DYN00120
DYN00130
DYN00140
DYN00150
DYN00160
DYN00170
DYN00180
DYN00190
DYN00200
DYN00210
DYN00220
DYN00230
DYN00240
DYN00250
DYN00260
DYN00270
DYN00280
DYN00290
DYN00300
DYN00310
DYN00320
DYN00330
DYN00340
DYN00350
DYN00360
DYN00370
DYN00380
DYN00390
DYN00400
DYN00410
DYN00420
DYN00430
DYN00440
DYN00450
DYN00460
DYN00470
DYN00480
DYN00490
DYN00500
DYN00510
DYN00520
DYN00530
DYN00540
DYN00550
DYN00560
DYN00570
DYN00580
DYN00590
DYN00600

```

```

      GO TO 80
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
      IPAGE = 0
C   WRITE MAIN BUDGET OUTPUT HEADING
      WRITE(IOUTPR,1080)
1080 FORMAT('1',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-NUM',5X,'COST',6X,'RETURN',4X,'ACCUM RETURN')
      WRITE(IOUTPR,1100) XIN
1100 FORMAT('0',6X,F15.2)
      TOTCST = 0
      TOTRTN = 0
      DO 150 L=1,NSTG
      I=NSTG+1-L
      K=NOD(I,J)
      KK=NOD(I,J)-1
      TOTCST = TOTCST + C(I,K)
      TOTRTN = TOTRTN + B(I,K)
C   WRITE I-TH LOCATION INFORMATION---TOTAL BUDGET OF M INCREMENTS
      WRITE(IOUTPR,1110) LOC(I),(XLOC(I,JJ),JJ=1,5),KK,C(I,K),
      + R(I,K),ORET(I,J)
1110 FORMAT(' ',24X,I4,9X,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('0',29('*'),' TOTALS ',29('*'),2F12.0,4X,F12.0)
160 CONTINUE
170 WRITE(IOUTPR,1130)
1130 FORMAT('1')
180 CONTINUE
      RETURN
      END
DYN00610
DYN00620
DYN00630
DYN00640
DYN00650
DYN00660
DYN00670
DYN00680
DYN00690
DYN00700
DYN00710
DYN00720
DYN00730
DYN00740
DYN00750
DYN00760
DYN00770
DYN00780
DYN00790
DYN00800
DYN00810
DYN00820
DYN00830
DYN00840
DYN00850
DYN00860
DYN00870
DYN00880
DYN00890
DYN00900
DYN00910
DYN00920
DYN00930
DYN00940
DYN00950
DYN00960
DYN00970
DYN00980
DYN00990
DYN01000
DYN01010
DYN01020
DYN01030
DYN01040
DYN01050
DYN01060
DYN01070
DYN01080
DYN01090
DYN01100
DYN01110

```

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(I,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
```

XOUT0010
XOUT0020
XOUT0030
XOUT0040
XOUT0050
XOUT0060
XOUT0070
XOUT0080
XOUT0090
XOUT0100
XOUT0110
XOUT0120
XOUT0130
XOUT0140
XOUT0150
XOUT0160
XOUT0170

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)

*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

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(J,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
TOTRTN	Total return from chosen improvements

VARIABLE LIST FOR XOUT

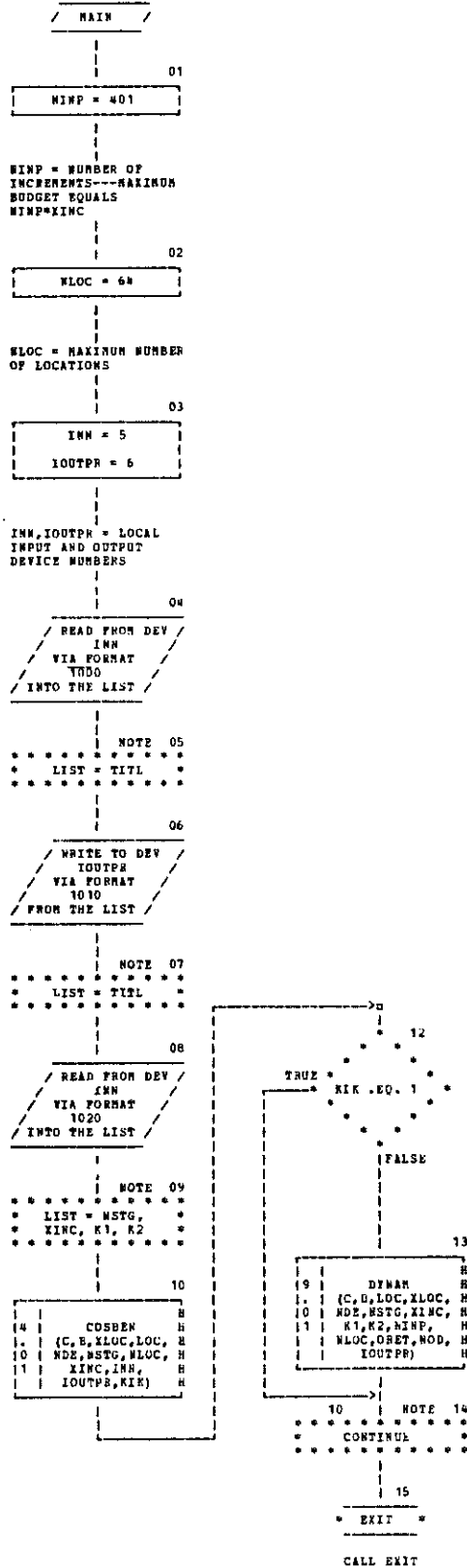
The following variables are described in DYNAM. All are passed as arguments into XOUT:

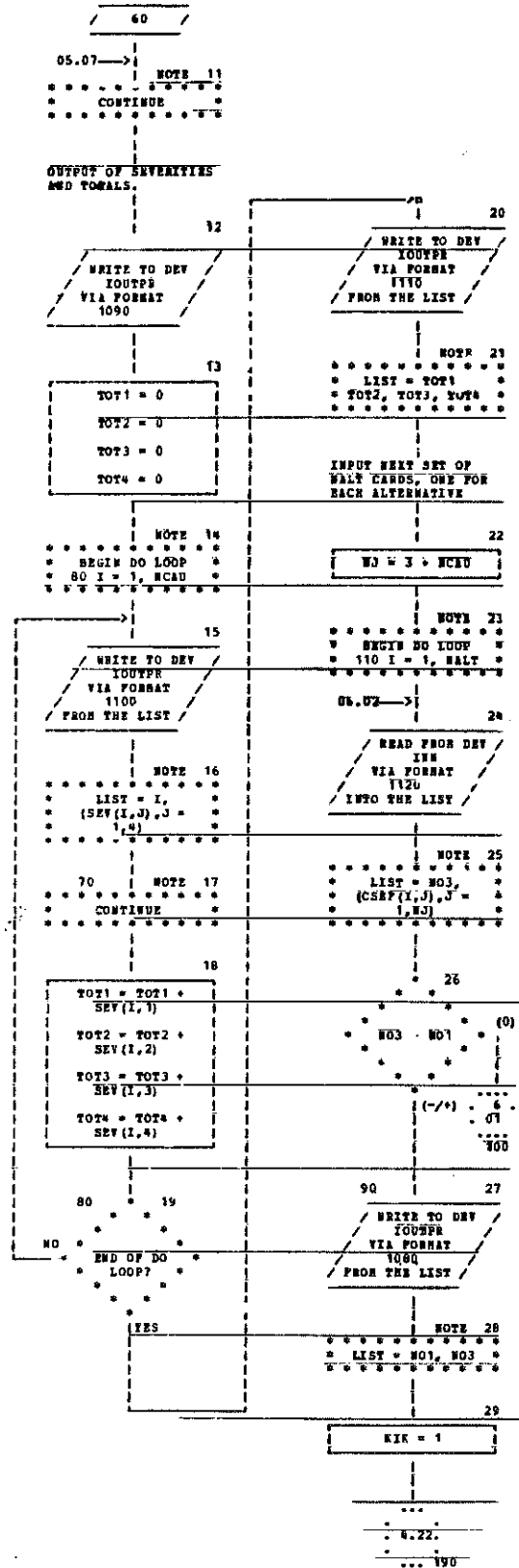
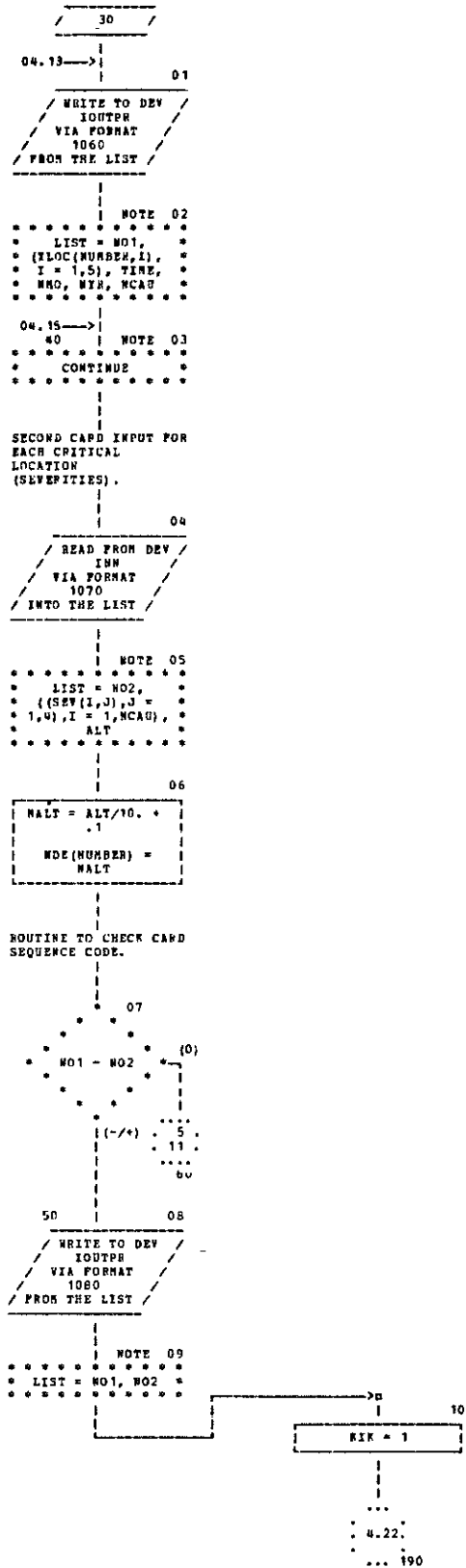
I	
IST	
XIN	
K	
KICK	
*XINC	
*C	
*NINP	
OUT	Budget that would remain if K-th alternative, Location I, were chosen at Stage I

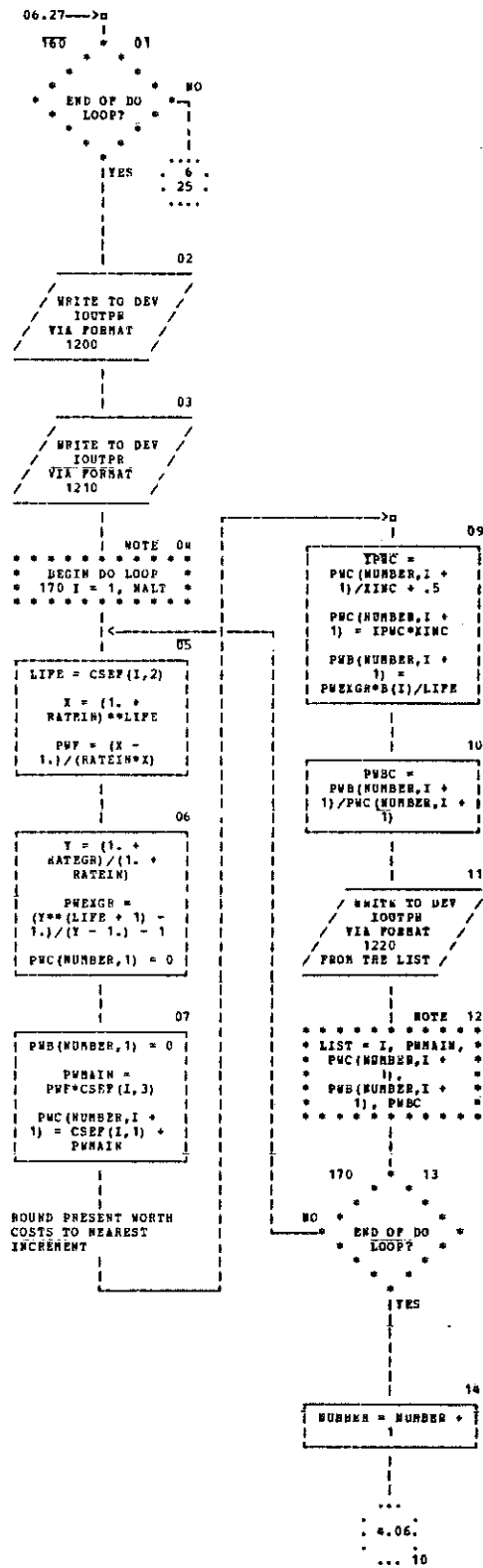
APPENDIX C

PROGRAM FLOW CHART

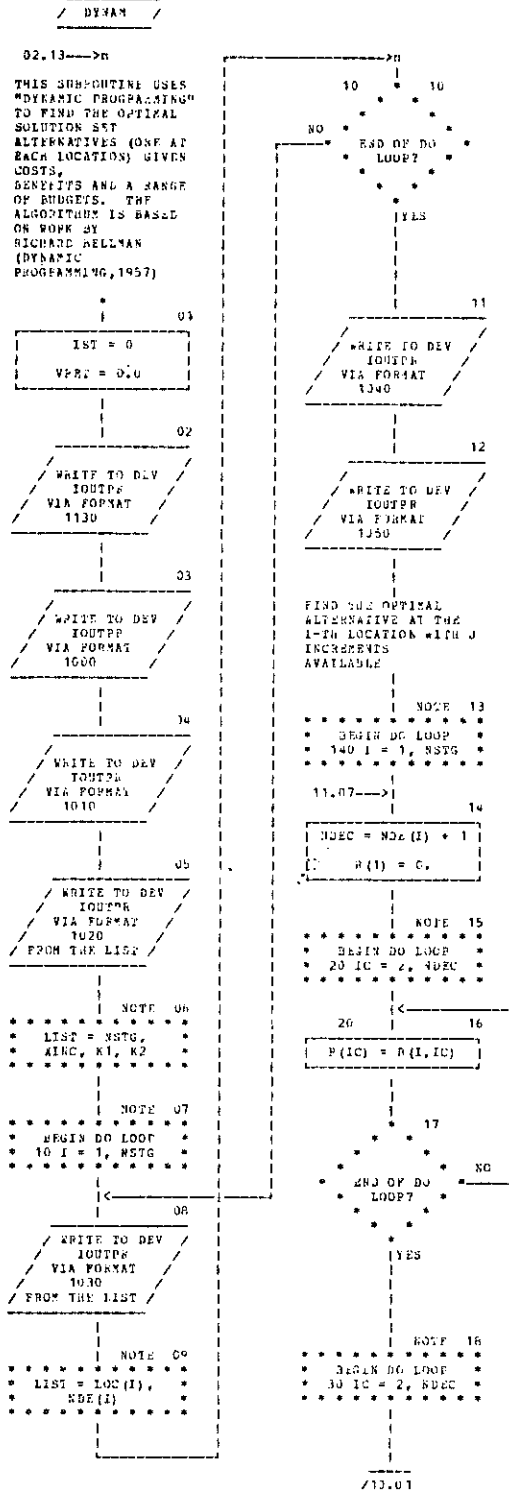
PROGRAM MAIN

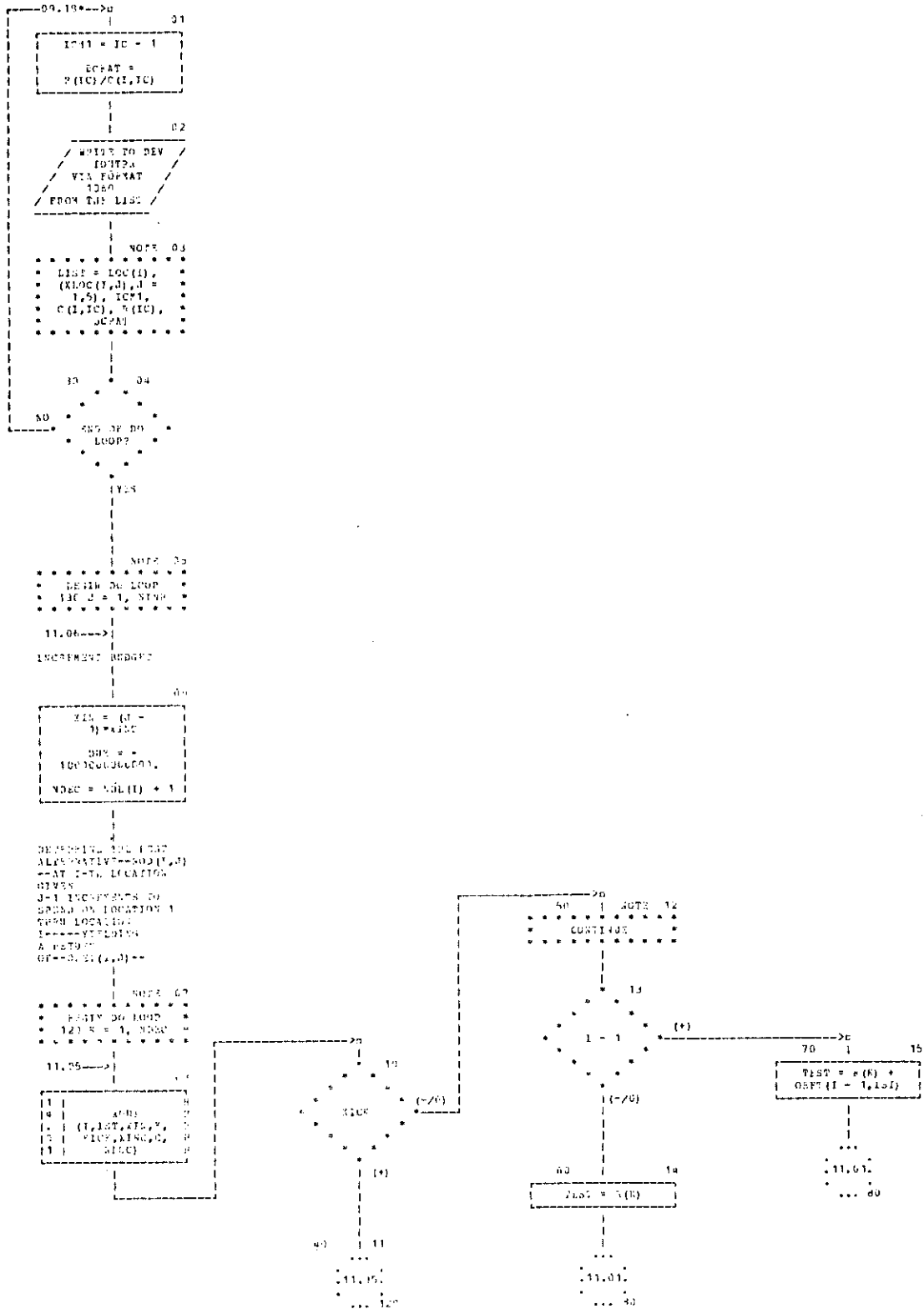


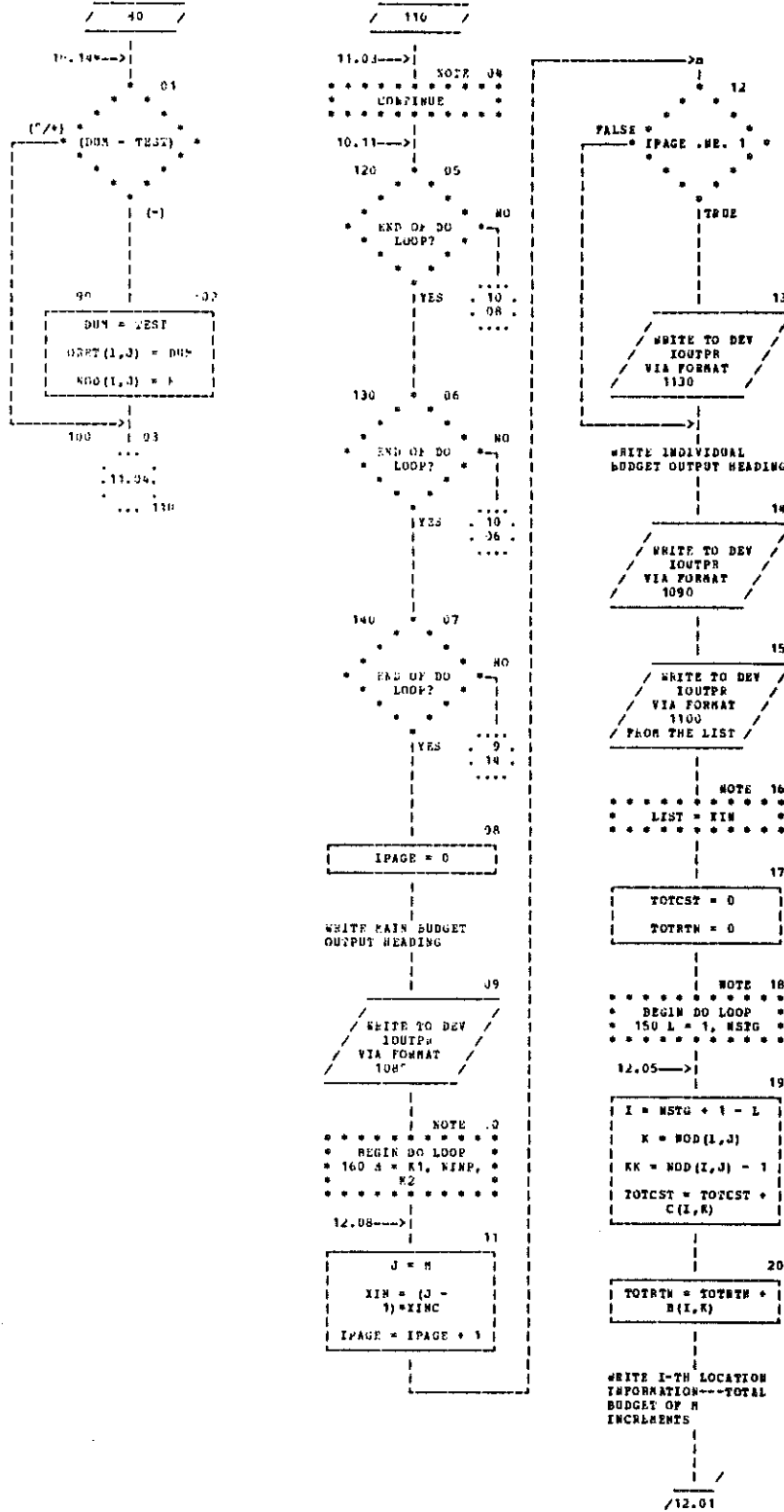


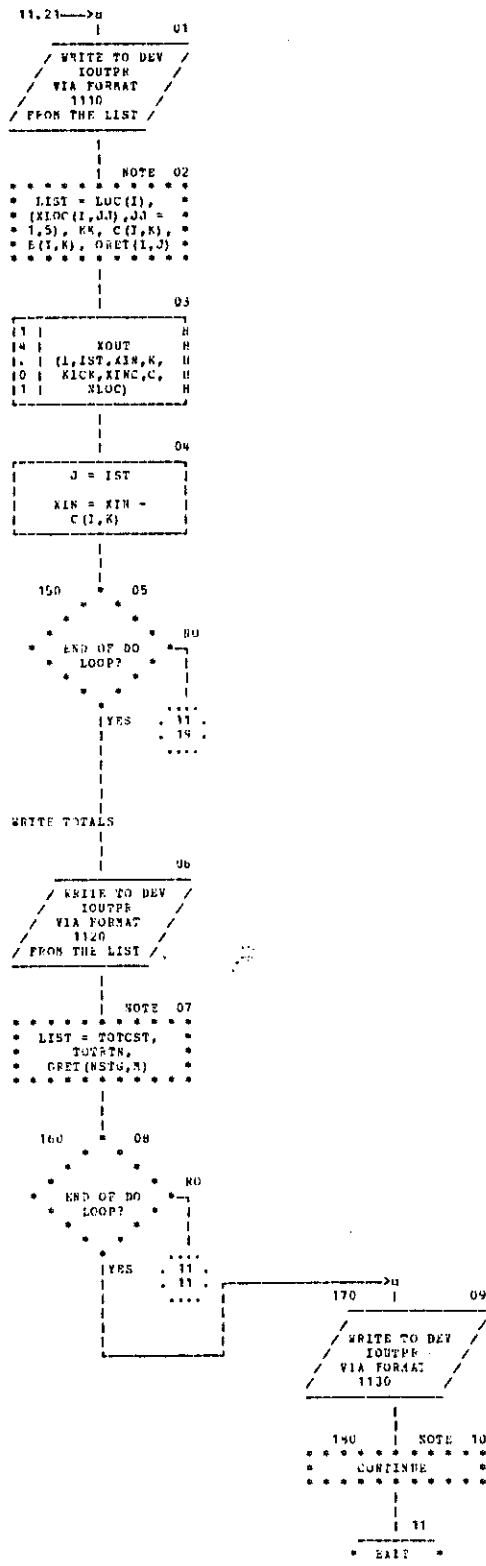


SUBROUTINE DYNAM







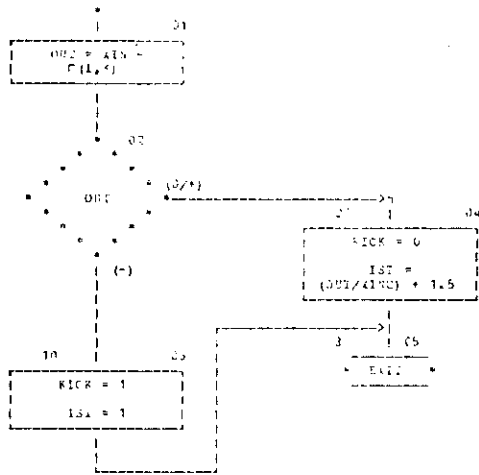


SUBROUTINE XOUT

 / XOUT /

10,000-->01

THIS SUBROUTINE
 CALCULATES AND OUTPUTS
 STATE LOSSER
 RESULTING FROM THE
 INPUT AND ASSUMED
 MEASURE Y. IT
 ALSO DETERMINES THE
 COST OF A PARTICULAR
 SAFETY MEASURE
 CORRESPONDING TO
 STAGE 1.



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); 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 datadeck MUST be blank.

APPENDIX E

SAMPLE INPUT AND OUTPUT

SAMPLE OF INPUT DATA

1	11	21	31	41	51	61	71	80	COLUMN NO.
	TEST	RUN	1	DYNAMIC	PROGRAMMING				
61250.0	41 40								CARD TYPE A
45000.	2700.	400.	0.10	0.04					CARD TYPE B
163-25-	10.9								CARD TYPE C
1 4 1 2	21						1.0	1691	CARD TYPE D
1	100010	100.02							CARD TYPE E } LOCATION
									CARD TYPE F } NO. 1

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 CAUST	TACC	NFAT	NINJ	NPRD
1	4.	1.	2.	2.
TOTALS	4.	1.	2.	2.

ALTERNATIVE	COST	LIFE	MAIN COST	EFFECT ON...	1
1	1000.00	10.	100.00		0.02

BENEFIT/COST ANALYSIS

ALTERNATIVE	COST	BENEFIT	BENEFIT/COST
1	1000.00	10240.00	10.2400

BENEFIT/COST ANALYSIS, MAINTENANCE INCLUDED

ALTERNATIVE	MAINTENANCE	TOTAL COST	BENEFIT/COST
1	1000.00	2000.00	5.1200

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

ALTERNATIVE	MAINTENANCE	TOTAL COST	BENEFIT	BENEFIT/COST
1	614.45	1500.00	7614.64	5.08

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

-----OUTPUT-----

LOCATIONS	INCREMENT	LOWER LIMIT	INCREMENTS PER STEP
61	250.00	41	40

ALTERNATIVES SELECTED
FOR \$50,000 BUDGET

BUDGET	LOCATION =	LOCATION NAME	ALT-NUM	COST	RETURN	ACCUM RETURN
50000.00						
	61	63-25-10.0	0	0.	0.	868692.
	60	22-60-26.2	0	0.	0.	868692.
	59	30-60-16.4	2	500.	3338.	868692.
	56	82-31W-1.2	1	500.	1921.	865354.
	57	51-41-20.0	1	500.	10792.	863433.
	56	82-31W-2.2	0	0.	0.	852640.
	55	102-25-11.9	1	500.	3934.	852640.
	54	63-25-10.4	7	2250.	19645.	848707.
	53	47-31W-23.3	0	0.	0.	829062.
	52	70-US60-11.3	5	1750.	8021.	829062.
	51	79-641-12.5	1	1000.	26532.	821040.
	50	79-641-12.5	3	1000.	17152.	794508.
	49	10-60-4.2	3	750.	5860.	777356.
	46	30-60-15.0	0	0.	0.	771496.
	47	82-31W-1.0	0	0.	0.	771496.
	46	82-US60-12.7	0	0.	0.	771496.
	45	82-US60-12.7	1	750.	9097.	771496.
	44	70-60-11.4	0	0.	0.	762399.
	43	51-60-20.3	1	750.	86411.	762399.
	42	79-641-13.0	0	0.	0.	675988.
	41	10-60-8.3	0	0.	0.	675988.
	40	82-60-12.3	1	500.	11078.	675988.
	39	120-60-12.6	1	500.	9008.	664910.
	38	84-68-18.5	1	1000.	34486.	655902.
	37	82-31W-1.1	1	500.	17168.	621416.
	36	051-41-20.0	1	500.	4699.	604248.
	35	51-41-20.0	0	0.	0.	599549.
	34	41-175-155.6	1	4000.	152094.	599549.
	33	63-25-10.9	0	0.	0.	447455.
	32	56-165KTP-130.1	1	1500.	11711.	447455.
	31	72-641-8.5	3	2000.	56100.	435743.
	30	54-41A-12.2	3	750.	8513.	379644.
	29	63-25-10.8	0	0.	0.	371131.
	28	73-62-16.9	3	3000.	12286.	371131.
	27	73-45-6.5	0	0.	0.	358846.
	26	73-62-17.8	0	0.	0.	358846.
	25	41-22-11.3	4	1000.	11530.	358846.
	24	114-31W-16.1	0	0.	0.	347316.
	23	30-54-12.6	1	750.	4290.	347316.
	22	47-31W-26.0	1	2250.	24654.	343026.
	21	37-127-8.7	0	0.	0.	318372.
	20	106-6-3.2	0	0.	0.	318372.
	19	63-25-16.0	0	0.	0.	318372.
	18	114-31W-16.2	3	750.	4071.	318372.
	17	24-68-9.1	0	0.	0.	314301.
	16	54-41A-12.4	3	750.	18040.	314301.
	15	79-641-18.9	1	750.	3548.	296261.
	14	39-171-63.7	1	4500.	26029.	292713.
	13	30-60-4.1	0	0.	0.	266684.
	12	72-641-8.5	1	750.	40588.	266684.
	11	30-60-15.2	3	5000.	31866.	226095.
	10	142-25-54	0	0.	0.	194229.
	9	102-25-9.2	0	0.	0.	194229.
	8	20-51-1.1	0	0.	0.	194229.
	7	48-421-14.4	0	0.	0.	194229.
	6	73-45-7.2	3	750.	6726.	194229.
	5	73-45-6.5	0	0.	0.	187504.
	4	54-41A-12.0	1	500.	9792.	187504.
	3	30-60-15.2	1	2000.	11385.	177712.
	2	30-60-15.1	1	4250.	158707.	166327.
	1	63-25 10.9	1	1500.	7620.	7620.
*****	TOTALS	*****		50000.	868692.	868692.