Application of Dynamic Programming and Other Mathematical Techniques to Pavement Management Systems

Kieran J. Feighan, Mohamed Y. Shahin, Kumares C. Sinha, and Thomas D. White

This paper describes the application of improved mathematical techniques to the PAVER and Micro PAVER Pavement Management Systems. The use of stochastic dynamic programming to determine optimal strategies and related mean costs over specified life-cycle periods is outlined. The incorporation of simple simulation techniques to estimate the variance associated with these costs is described. The suitability of this approach for investigating the effects of deferred maintenance is presented. The use of outputs from these programs in subsequent prioritization and budget allocation modules is briefly discussed. An example that incorporates outputs from the dynamic programming and simulation programs is shown, and the validity of these outputs is discussed.

There has been increased research interest recently in prediction and optimization methods for Pavement Management Systems (PMS). This is a logical extension of earlier efforts directed primarily toward developing useful, reproducible condition survey techniques. Once confidence has been established in the survey results, the data can be used in conjunction with optimization methods to obtain more cost-effective use of management resources, which is the ultimate aim of most management systems.

This paper describes the application of a set of mathematical tools to the PAVER and Micro PAVER PMS. Dynamic programming is used in conjunction with a Markov-chain probability-based prediction model to obtain minimum cost maintenance strategies over a given life-cycle. Simulation methods, using a random number generator, are applied to determine the variance in the cost associated with these strategies.

The suitability of using dynamic programming to monitor the effects of deferred maintenance is outlined, and the use of output from the programs in establishing budget priorities and allocations is briefly discussed. Dynamic programming has a simple and efficient structure that allows rapid execution, even at the microcomputer level.

BACKGROUND

The present research is part of an ongoing effort to improve the prediction and optimization capabilities of the PAVER

K. J. Feighan, 602 E. Stoughton #33, Champaign, Ill. 61820. M. Y. Shahin, United States Army, Construction Engineering Research Laboratory, Champaign, Ill. 61820-1305. K. C. Sinha and T. D. White, Department of Civil Engineering, Purdue University, West Lafayette, Ind. 47907.

and Micro PAVER pavement management systems. The PAVER system has been well documented elsewhere (1). It is based on the Pavement Condition Index (PCI), an index with a range of 0 to 100, reflecting the current condition of a pavement section.

Several Army installation databases were used in the development of prediction models. There is a separate database for each installation. Generally, each pavement section for which information is stored in the database can be identified by location, pavement type, pavement use, and pavement rank. Large variation in the condition data is expected from section to section in a network, even when all sections are the same age. There is generally little or no specific traffic volume or structural information available on a section-by-section basis. Obviously, it would be desirable to have this information, but at the moment this system must work with the available information.

In order to reduce the variation in the data and increase confidence in the predicted performance over time, it is necessary to group the data using common variable characteristics (2). Currently, the variables used to define groups, or "families" of data, are pavement surface type, pavement use, and pavement rank. It is possible that the primary source of pavement distress may be subsequently included (3). These families are also used in the dynamic programming process.

From a search of the available literature, it appears that very few PMSs currently in use have access to, or use, relationships between pavement condition and cost to repair and maintain the pavement in arriving at the most desirable and cost-effective solutions. However, this data is of critical importance in evaluating the tradeoffs between repairing now and repairing later, both at the project level and at the network level.

Cost information for the PCI is available from the results of an ongoing study into the relationship between PCI and cost that is being carried out by Purdue University for the U.S. Army Construction Engineering Research Laboratory (USA CERL) (4). The study identifies the cost of several maintenance alternatives, including routine maintenance, on different surfaces as a function of surface condition. This information can be readily incorporated into the dynamic programming framework. It is believed that there are no user cost models currently available for the Micro PAVER system because of the lack of confidence in the accuracy of the cost predictions. Consequently, an effectiveness/cost ratio approach is applied using only agency costs in the denominator. A

FIGURE 1 Diagram of state, state vector, and duty cycle.

simple modification can incorporate user costs if realistic model relationships become available.

MARKOV PREDICTION MODEL

It is not possible to describe the functioning of the dynamic programming algorithm without first describing the prediction model, as the two function together. The following is an abbreviated description of the model that includes only the parts relevant to dynamic programming; a more detailed and comprehensive description has already been published elsewhere (3, 5). Much of the terminology used in this description also is used in dynamic programming.

All sections in a particular network are grouped into families on the basis of common characteristics. The *PCI* range of 0 to 100 is divided into 10 states, each state being 10 *PCI* points wide. A pavement is modeled to begin its life in near-perfect condition and to deteriorate as it is subjected to a sequence of duty-cycles. A duty-cycle is defined as the effects of one year's weather and traffic. A state vector indicates the probability of a pavement section being in each of the 10 states in any given year. Figure 1 shows the schematic representation of state, state vector, and duty cycle for a particular family. All of the sections of a family are categorized as belonging to one of the ten states at any age. It is assumed that all of the pavement sections are in state 1 (*PCI* = 90 to 100) at an age of zero years.

To model the way in which the pavement deteriorates with time, it is necessary to identify the Markov probability matrix. The assumption is made that the pavement condition will not drop by more than one state in a single year. Thus, the pavement will either stay in its current state or move to the next lowest state in one year if it remains in the same family. The probability transition matrix has a diagonal structure, as shown in Figure 2. The entry of 1 in the last row of the transition matrix indicates a trapping state. The pavement condition cannot transit from this state unless repair is performed.

The state vector for any duty cycle t, $S(t)$, is obtained by multiplying the initial state vector $S(0)$ by the transition matrix P raised to the power of t. Thus

$$S(1) = S(0)*P$$
$$S(2) = S(1)*P = S(0)*P^2$$
$$\vdots \quad\quad \vdots \quad\quad \vdots \quad \vdots$$
$$S(t) = S(t-1)*P = S(0)*P^t$$

If the transition matrix probabilities can be estimated, the future condition of the road at any duty cycle (time) t can be predicted.

The probabilities are estimated using a non-linear programming approach, which has as its objective function minimizing the absolute distance between the actual family *PCI* versus age data points and the expected (predicted) *PCI* for the corresponding age generated by the Markov chain using the nine probability parameters. It has been found that this approach can accurately model the pavement deterioration over time. A sample curve fit generated by this program is shown in Figure 3.

INTRODUCTION TO DYNAMIC PROGRAMMING

Dynamic programming is an approach to optimization. It is not based on a single, well-defined procedure such as the simplex algorithm of linear programming. Rather, it seeks to take a single, complicated problem and break it down into a number of simpler, more easily solvable problems. Obtaining solutions to these problems is generally faster and requires less computational effort than solving the original problem, while giving the same final result.

The principle of optimality, the foundation of dynamic programming, states that "every optimal policy consists only of optimal subpolicies" (6). Instead of examining all possible combinations of decisions, dynamic programming examines a small, carefully chosen subset of the combinations, rejecting those combinations that cannot possibly lead to an optimal solution. If performed correctly, the subset examined is certain to contain the optimal solution.

Dynamic programming has an advantage over almost all classical optimization techniques in that it will yield the abso-

$$P = \begin{bmatrix} p(1) & q(1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & p(2) & q(2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p(9) & q(9) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$q(i) = 1 - p(i)$$

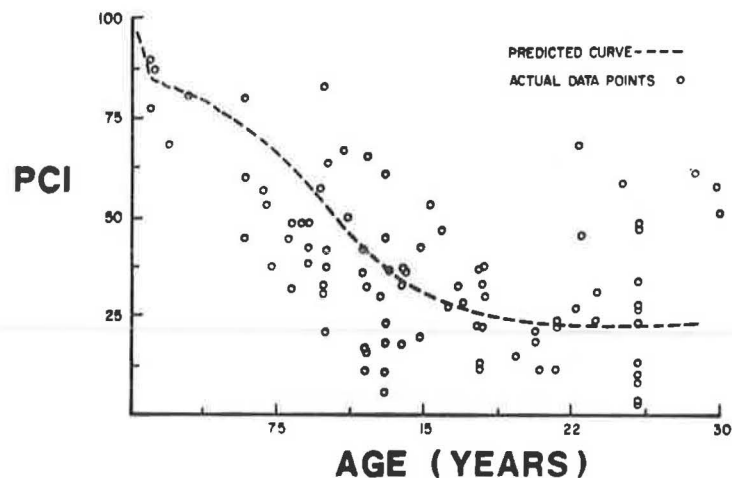FIGURE 2 Probability transition matrix structure.

FIGURE 3 Sample Markov prediction curve.

lute or global maxima or minima rather than local optima. Dynamic programming is also extremely powerful in that it can handle integrality, negativity, discreteness, and other constraints of variables. It also, by its nature, produces the optimal solution to all its constituent subproblems. These solutions may be of interest to the pavement manager, especially in time-varying problems such as pavement performance where the results of a 25 year life-cycle analysis will also contain the optimal results for all life-cycle lengths less than 25 years.

The major drawback to dynamic programming is that it may not be possible to formulate the proposed problem in such a way as to effectively use dynamic programming. However, if it is possible to express the problem in an effective form, dynamic programming provides an outstanding optimization tool.

The basic components of dynamic programming are states, stages, decision variables, returns, and transformation or transition functions (6). A physical system is considered to progress through a series of consecutive stages. In pavement performance, each year is viewed as a stage.

At each stage, the system must be fully describable by the state variables or state vector. In the present case, as described earlier, each state is a 10 PCI bracket for every pavement family, and the condition of the pavement in each section at any year (stage) can be defined as being in one of the ten states.

At each stage, for every possible state, there must be a set of allowable decisions. The decisions being made in the dynamic programming model are the repair alternatives to be implemented in each state at every stage.

Finally, there is the transformation or transition function. If a process is in a given state and a feasible decision is made, there must be a function that determines to which state the process moves. In general, dynamic programming transformation functions can be deterministic or stochastic. In this particular case, the transition function is defined by the Markov probability matrix derived in the curve-fitting process described earlier and, hence, is a stochastic process.

In summary, the problem set-up for this dynamic programming formulation is

MINIMIZE: Expected cost over a specified life cycle, subject to keeping all sections above a defined performance standard. The dynamic programming parameters are

STATES: Each bracket of 10 PCI points in a family.
STAGES: Each year in the analysis period.
DECISION VARIABLES: Which maintenance treatment to apply.
TRANSFORMATION FUNCTION: The Markov transition probability matrix defines the transformation.
RETURN: Expected cost if a particular decision is made in each state at each stage.

INPUTS REQUIRED FOR THE DYNAMIC PROGRAMMING ALGORITHM

The inputs required for the dynamic programming algorithm are

1. Markov Transition Probabilities for state i of matrix **j**

p_{ij}; $i = 1, \ldots, 10$ states

   j $= 1, \ldots, m$ families

2. Cost of applying treatment k to family j in state i

C_{ijk}; $k = 1, \ldots, n$ maintenance alternatives

Routine maintenance is always designated as $k = 1$. The cost is entered on a dollar per square yard basis.

3. Feasibility Indicator for alternative k when in state i of family j

$R_{ijk} = 1$ if maintenance alternative is feasible

   $= 0$ if maintenance alternative is infeasible

4. Number of Years in the life-cycle analysis; N
5. Interest Rate ; r
6. Inflation Rate ; f
7. Rate of Increase in Funding ; q
8. The associated benefit over one year of being in state i

$B_i = 95, 85, \ldots, 5$ for $i = 1,2, \ldots, 10$

The benefit is taken to be the area under the PCI curve over a period of 1 year.

9. The minimum allowable state for each family; in other words, the lowest state to which a particular family will be

allowed to deteriorate before major maintenance is performed. This state is designated by S_j.

10. The transformations that define the new family to move to if treatment k is applied in family j: $(j,k) - (j^1,1)$.

OPERATION OF DYNAMIC PROGRAMMING

The dynamic programming process starts at year N, the final year of the life-cycle analysis. In dynamic programming terms, this is stage 0. In effect, the life-cycle cost analysis is being performed over 0 years at this stage.

The first step is to calculate the routine maintenance cost for each state in every family in year N. Routine maintenance is not feasible if (1) $R_{ijk} = 0$, or (2) $S_j \geq i$ for family j. For all feasible states, the cost of routine maintenance is obtained from $c^o_{ij1,N} = c_{ij1}$, and these values are stored. If routine maintenance is not feasible, a very large value is added to the cost to ensure that it will not be chosen as the cheapest alternative.

All other feasible alternative costs are also calculated for all states in each family from $c_{ijk,N}$ c_{ijk}. The optimum repair strategy for each state i of each family j in year N is then given by:

$$C^*_{ij,N} = \text{Min}(c^o_{ij1,N}, c_{ijk,N}) \text{ for all } i, j, k$$

In general, the decision process can be described for year $N-n$, or equivalently for stage n. As before, the feasibility of routine maintenance is examined. If routine maintenance is found to be feasible, the following expression is used to

calculate the total present worth in the year $N-n$ of applying routine maintenance when the analysis period is n years long:

$$c^o_{ij1,N-n} = c_{ij1} + [p_{ij1}c^*_{ij,N-n+1}$$

$$\text{(i)} \qquad\qquad \text{(ii)}$$

$$+ (1 - p_{ij1})c^*_{i-1,j,N-n+1}] * 1/(1 + i^*)$$

This expression has two parts: the part indicated by (i) is the immediate cost of routine maintenance in year n, while (ii) is the total expected cost in the remaining $N-n$ years as a consequence of applying routine maintenance in year n. As shown in Figure 4, this expected cost is obtained by identifying the probability that the condition will remain in a given state, multiplying this probability by the expected cost of that state, and then finding the associated probability of dropping a state if routine maintenance is applied and multiplying this by the expected cost of the lower state. This sum is then discounted by the effective interest rate, i^*, to bring the total into present worth dollars in the year $N-n$. The effective interest rate is calculated using the interest rate, inflation rate, and rate of increase in funding inputs.

Similarly, the cost of all other feasible maintenance alternatives can be calculated. The expression used is

$$c_{ijk,N-n} = c_{ijk} + [p_{1j^11}C^*_{1j^1,N-n+1}$$

$$+ (1 - p_{1j^11})C^*_{2j^1,N-n+1}] * 1/(1 + i^*)$$

This expression differs from the routine maintenance equation in that it is known that the pavement condition will return

FIGURE 4 Calculation of expected cost in year n.

to state 1 after the repair alternative is carried out. The family that the pavement moves to, j^1, as a result of having this alternative performed is defined in the input transformation matrix. For example, if a thin overlay is performed on an AC pavement section, that section will move to the thin overlay family for performance prediction after the overlay is placed.

The optimal cost is then given by

$$C^*_{ij,N-n} = \text{Min}(c^o_{ij1,N-n}, c_{ijk,N-n})$$

with the associated optimal maintenance alternative to be performed for this (i, j) family/state combination in year $N - n$ being the choice of k that minimizes the above expression.

This backward recursion is performed for every successive year of the analysis period until the analysis for year 0, or stage N, is reached.

DYNAMIC PROGRAMMING OUTPUT

The output from the dynamic programming program consists of (1) a file containing the optimal maintenance alternative in every year for every family/state combination, (2) the discounted present-worth costs expected to accrue over the life-cycle specified if the optimal decisions are implemented, (3) the expected effectiveness accrued as a result of following the optimal decisions calculated for every family/state combination, and (4) the effectiveness/cost ratio calculated for every family/state combination.

Thus, after the family/state combination for a particular section has been defined, the optimal maintenance alternative and associated cost and effectiveness can be readily obtained. As the intention of the research is to produce updated programs for both the PAVER and Micro PAVER systems, it was imperative that all of the programs be executable on a microcomputer. The dynamic programming program executes extremely quickly at this level.

SIMULATION AND DYNAMIC PROGRAMMING

The dynamic programming algorithm calculates the least-cost strategy on a probabilistic basis by using the probabilities of being in various states multiplied by the cost of being in those states. Consequently, the minimum expected cost over an N-year life cycle is obtained. This is a valid and useful parameter to use in decision making; but, if possible, it is desirable to obtain also an estimate of the likely variation in this expected cost. This can be of use to decision makers in looking at worst case and best case scenarios, as well as in deciding how much confidence there is in the mean estimate of cost.

It is not possible to estimate this variation through dynamic programming directly, but by use of simple simulation techniques the variation in cost can be estimated. Simulation in essence uses repeated deterministic runs to simulate a continuous, probabilistic situation. In the present case, the performance of the pavement over time is simulated using the pavement state definition and a random number generator. By performing many simulation runs, a good idea is obtained of the mean and variance of the expected cost of maintaining the pavement. The mechanics of the simulation process are outlined below.

It is assumed that a pavement is in a given state in a given year, and that there are two possible states that the pavement can go to the following year for a particular maintenance action. This assumption is consistent with the Markov assumptions used earlier in performance prediction and in dynamic programming. As an example, assume that the pavement is now in state 3 and routine maintenance is applied.

There are two possible states for the pavement to be in the following year, state 3 and state 4. The probability of remaining in state 3 is given by the Markov transition probability, $p(3)$, and the probability of transiting to state 4 is given by $1 - p(3)$. For clarity, assume that $p(3) = 0.65$, thus $1 - p(3) = 0.35$.

The random number generator is used to generate a random number, RAND, between 0 and 1. The decision mechanism outlined below is then used.

If RAND < 0.65, then pavement stays in state 3.
If RAND > 0.65, then pavement moves to state 4.

In this way, the performance of the pavement (in terms of which state it is in) can be tracked over time. If the cost of being in each state is known, the total cost over time for each simulation run can be obtained. This value can be used to estimate the mean and variance of the total minimum expected cost.

RANDOM NUMBER GENERATOR

All the research (including computer programs) performed in this study was done on microcomputers, as this is the environment of choice for most pavement management systems today. Unfortunately, most random number generators have been designed for larger, mainframe computers and use the larger word length capabilities of those machines. A portable random number generator that can overcome the problem of limited integer arithmetic capabilities by using floating point arithmetic was located (1). This was the generator used for performing simulations in this study.

Most random number generators are based on the linear congruential method, which can yield problems if the parameters are chosen unwisely. The generator used here is based on the subtractive method, which does not contain the same element of risk. The random number generator returns uniform deviates in the range of zero to one. Any number in this range is as likely to be chosen as any other. The range of zero to one is used, as this is the same as the possible range of possible Markov probability values. This approach makes the algorithm programming very simple as well. A flowchart of the simulation program is shown in Figure 5.

PRIORITIZATION

By combining the simulation output with the dynamic programming output, the optimal maintenance alternatives, their associated cost and effectiveness, and a confidence limit on the expected cost are all readily obtained. Obviously, this is of critical importance in moving to the next step of network optimization, the prioritization of sections on a cost-effectiveness or weighted cost-effectiveness basis and the subse-
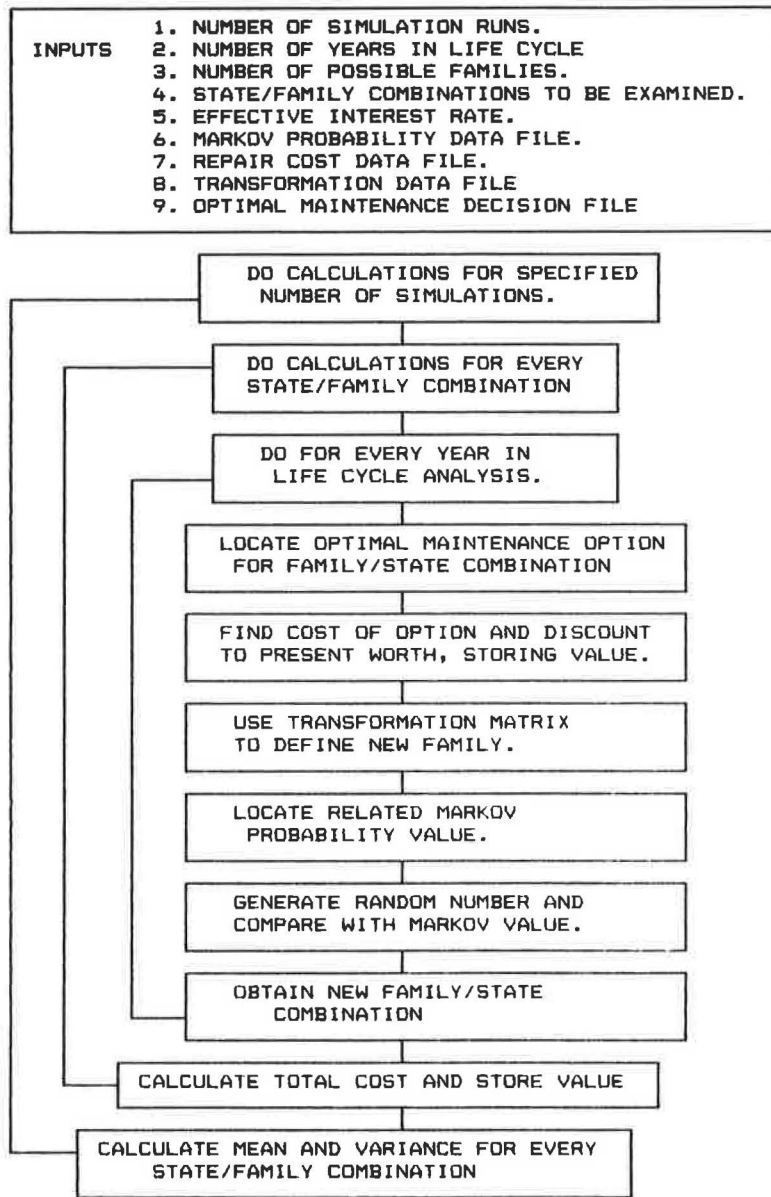
```
┌─────────────────────────────────────────────────────┐
│         1. NUMBER OF SIMULATION RUNS.                 │
│ INPUTS  2. NUMBER OF YEARS IN LIFE CYCLE              │
│         3. NUMBER OF POSSIBLE FAMILIES.               │
│         4. STATE/FAMILY COMBINATIONS TO BE EXAMINED.  │
│         5. EFFECTIVE INTEREST RATE.                   │
│         6. MARKOV PROBABILITY DATA FILE.              │
│         7. REPAIR COST DATA FILE.                     │
│         8. TRANSFORMATION DATA FILE                   │
│         9. OPTIMAL MAINTENANCE DECISION FILE          │
└─────────────────────────────────────────────────────┘

         DO CALCULATIONS FOR SPECIFIED
             NUMBER OF SIMULATIONS.

         DO CALCULATIONS FOR EVERY
            STATE/FAMILY COMBINATION

         DO FOR EVERY YEAR IN
           LIFE CYCLE ANALYSIS.

       LOCATE OPTIMAL MAINTENANCE OPTION
          FOR FAMILY/STATE COMBINATION

       FIND COST OF OPTION AND DISCOUNT
       TO PRESENT WORTH, STORING VALUE.

       USE TRANSFORMATION MATRIX
          TO DEFINE NEW FAMILY.

       LOCATE RELATED MARKOV
         PROBABILITY VALUE.

       GENERATE RANDOM NUMBER AND
       COMPARE WITH MARKOV VALUE.

       OBTAIN NEW FAMILY/STATE
            COMBINATION

     CALCULATE TOTAL COST AND STORE VALUE

    CALCULATE MEAN AND VARIANCE FOR EVERY
            STATE/FAMILY COMBINATION
```

FIGURE 5 Simulation flowchart.

quent budget allocation procedure. It is at this stage that budgetary constraints in each year are addressed. When the cost values obtained are guaranteed to be optimal, as is the case for dynamic programming, the prioritization methodology used can be made very simple and rapidly executable.

This consideration is, again, of utmost importance in the present case, where the environment in which the programs are to operate is a microcomputer one, with its attendant limited computing capabilities. The detailed discussion of prioritization and budget allocation procedures to be used are beyond the scope of this paper and will be fully discussed in the future, but it is believed that workable algorithms have been formulated that take full advantage of the dynamic programming outputs. Figure 6 shows the flowchart describing the direction of the overall research, from raw database information to annual budget allocation work plans.

DEFERRED MAINTENANCE

Another advantage of dynamic programming is the ability to identify the effects of deferred maintenance on the optimal treatment cost over a given life-cycle length. The methodology of investigating the effect of deferring the optimal treatment for D years is as follows.

A constraint is imposed in the dynamic programming algorithm that will allow only routine maintenance to be chosen in the last D years of the life cycle. This is effectively the impact that deferred maintenance has, as almost inevitably some type of routine maintenance will be needed to hold the pavement until sufficient funds for the recommended treatment are forthcoming. The dynamic programming program is then run again as before, and the optimal treatments and costs noted.
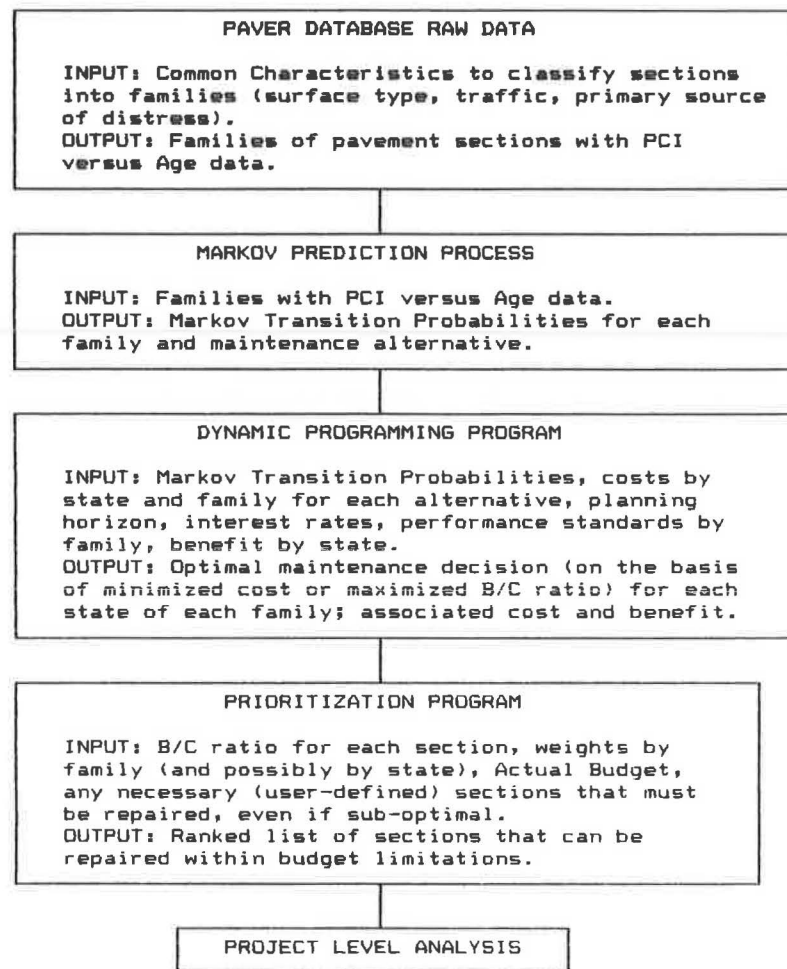
```
┌─────────────────────────────────────────────────────┐
│            PAVER DATABASE RAW DATA                   │
│                                                      │
│  INPUT: Common Characteristics to classify sections  │
│  into families (surface type, traffic, primary source│
│  of distress).                                       │
│  OUTPUT: Families of pavement sections with PCI       │
│  versus Age data.                                    │
└─────────────────────────────────────────────────────┘
                          │
┌─────────────────────────────────────────────────────┐
│            MARKOV PREDICTION PROCESS                 │
│                                                      │
│  INPUT: Families with PCI versus Age data.           │
│  OUTPUT: Markov Transition Probabilities for each     │
│  family and maintenance alternative.                 │
└─────────────────────────────────────────────────────┘
                          │
┌─────────────────────────────────────────────────────┐
│           DYNAMIC PROGRAMMING PROGRAM               │
│                                                      │
│  INPUT: Markov Transition Probabilities, costs by    │
│  state and family for each alternative, planning     │
│  horizon, interest rates, performance standards by   │
│  family, benefit by state.                           │
│  OUTPUT: Optimal maintenance decision (on the basis  │
│  of minimized cost or maximized B/C ratio) for each  │
│  state of each family; associated cost and benefit.  │
└─────────────────────────────────────────────────────┘
                          │
┌─────────────────────────────────────────────────────┐
│             PRIORITIZATION PROGRAM                  │
│                                                      │
│  INPUT: B/C ratio for each section, weights by       │
│  family (and possibly by state), Actual Budget,      │
│  any necessary (user-defined) sections that must     │
│  be repaired, even if sub-optimal.                   │
│  OUTPUT: Ranked list of sections that can be         │
│  repaired within budget limitations.                 │
└─────────────────────────────────────────────────────┘
                          │
       ┌──────────────────────────────────┐
       │     PROJECT LEVEL ANALYSIS        │
       └──────────────────────────────────┘
```

FIGURE 6 Overall research flowchart.

As expected, deferred maintenance generally has the greatest impact in the middle and lower states where the preferred options are high cost, and the effect of transiting to a lower state with the consequent increased surface preparation costs is considerable. It is extremely useful to be able to obtain deferred maintenance costs, as the effect of lowering budgets in the present on future long-term costs is readily demonstrable to decision makers and can be used to justify increased budgets in early years.

The ability to play what-if games using dynamic programming and deferred maintenance scenarios is obviously crucial in this whole area of budget justification. The ease and speed of calculation and recalculation that the dynamic programming approach offers is a major factor in its intended incorporation into the PAVER and Micro Paver systems.

ILLUSTRATIVE EXAMPLE OF DYNAMIC PROGRAMMING

A brief example follows to illustrate the operation of the program. Performance curves were developed for the Fort Eustis base in Virginia based on *PCI* condition surveys performed there. Families were defined on the basis of branch use and surface type. For the branch use of roadway, four families were defined: asphalt concrete, surface treated, thin overlay, and structural overlay.

Four maintenance alternatives were considered: routine maintenance, surface treatment, thin overlay, and structural overlay. Dollar cost as a function of *PCI* was defined for both initial repair cost and subsequent routine maintenance cost in research carried out for USA-CERL by Purdue University (4). Markov probability calculations for each family were performed, and the probability transition matrices obtained.

The dynamic programming program was then run on a Compaq 286 microcomputer with an 80287 math coprocessor. An effective interest rate of 5 percent was used, and a life-cycle analysis of 25 years was specified. A minimum allowable state of seven (*PCI* of 30 to 40) was specified. In other words, if the condition of the pavement section falls to between 30 and 40, it must be repaired. It is of course very possible that the section will be chosen for repair at a greater *PCI*. The program took 35 seconds to produce the entire output for every year of the 25 years.

The simulation program was also run using the decisions output from the dynamic programming program. For every family/state combination, a total of $4 * 7 = 28$ combinations in all, 25 simulations of a 25-year life cycle were performed,

and the cost associated with each simulation was computed. The mean and variance for each set of 25 simulations were computed.

In theory, if an infinite number of simulation runs were performed instead of the 25 actually run, the mean obtained would be the same as that given for year one of each family/state combination by dynamic programming. To verify the validity of the dynamic programming results, the dp mean and the simulated mean should be within 2.5 standard deviations of one another at the 95 percent confidence level. To perform all of the above calculations and calculate the mean and variance for each family/state combination took 28 seconds on the microcomputer equipment described earlier.

ANALYSIS OF EXAMPLE RESULTS

The dynamic programming results are shown in Table 1 for a 25-year life-cycle analysis using a discount rate of 5 percent. The optimal decisions corresponding to the numbers shown are:

1. Routine Maintenance
2. Surface Treatment
3. Thin Overlay

In this case, for 5 percent interest and the actual repair costs identified in the Purdue study (*4*), the structural overlay was not chosen as the most cost-effective solution at any *PCI*

level considered. This is in agreement with the deterministic life-cycle analysis carried out by Purdue in the same study.

The overall pattern of optimal decisions is reasonable, with routine maintenance being chosen in the upper states, surface treatment in the middle-to-upper states, and thin overlays in the middle to lower states. The cost of repair for surface treated sections (family 2) and sections with thin overlays (family 3) is the same in this case, as the optimal decisions are the same, and the costs of surface treatment or thin overlay were found to be a function of condition only, not of section surface type before treatment. This is not the case in family 1 (AC pavements), nor is it the case in most of the data obtained in the Purdue study.

Table 2 shows the results obtained from the 25 simulation runs. As can be seen, the mean values obtained are reasonably close to those found through dynamic programming and shown in Table 1. In fact, establishing a 95-percent confidence interval about the dynamic programming means (2.5 standard deviations on either side) shows that for all but one case, the family 1/state 1 combination, the simulated mean is well within the confidence limits. These bounds are also illustrated in Table 2.

To obtain the optimal treatment for any section in the network, all that is necessary is to decide what state and family the section is currently in, and look up the optimal treatment for that family/state combination in Table 1. The minimum allowable state and/or interest rates used can be varied to determine their effect on the optimal decisions reached through dynamic programming.

TABLE 1 DYNAMIC PROGRAMMING RESULTS FOR 25-YEAR ANALYSIS

FAMILY	STATE	OPTIMAL DECISION	OPTIMAL COST
FAMILY 1	1	1	0.48
	2	1	3.73
	3	2	4.59
	4	2	4.96
	5	3	6.26
	6	3	7.51
	7	3	9.43
FAMILY 2	1	1	2.39
	2	1	3.41
	3	2	4.35
	4	2	5.59
	5	3	8.61
	6	3	10.51
	7	3	12.08
FAMILY 3	1	1	1.07
	2	1	3.38
	3	2	4.35
	4	2	5.59
	5	3	8.61
	6	3	10.51
	7	3	12.08
FAMILY 4	1	1	0.58
	2	1	2.97
	3	1	3.89
	4	2	4.35
	5	3	5.47
	6	3	6.57
	7	3	8.49

TABLE 2   SIMULATION RESULTS AND CONFIDENCE
LIMITS ON EXPECTED COST OF OPTIMAL TREATMENTS

FAMILY	STATE	SIMULATION		UPPER BOUND	LOWER BOUND
		MEAN	VARIANCE		
FAMILY 1	1	0.29	0.12	0.65	0.31
	2	3.78	1.07	4.25	3.21
	3	4.10	1.26	5.15	4.03
	4	4.96	1.51	5.57	4.35
	5	6.16	1.15	6.80	5.72
	6	7.59	1.59	8.14	6.88
	7	9.52	1.41	10.02	8.83
FAMILY 2	1	2.19	1.77	3.06	1.72
	2	3.34	1.8	4.08	2.74
	3	4.42	1.75	5.01	3.69
	4	5.52	0.83	6.05	5.13
	5	8.65	1.45	9.21	8.01
	6	10.51	1.40	11.10	9.92
	7	12.31	1.32	12.65	11.51
FAMILY 3	1	0.88	0.79	1.51	0.63
	2	3.41	1.39	3.97	2.79
	3	3.89	1.46	4.95	3.75
	4	5.72	1.95	6.29	4.89
	5	8.72	1.20	9.16	8.06
	6	10.83	2.30	11.27	9.75
	7	11.99	1.57	12.71	11.45
FAMILY 4	1	0.45	0.58	0.96	0.20
	2	3.14	1.31	3.54	2.40
	3	4.08	1.31	4.46	3.32
	4	4.46	1.98	5.05	3.65
	5	5.33	1.36	6.05	4.89
	6	6.42	1.40	7.16	5.98
	7	8.50	1.11	9.02	7.96

FUTURE DEVELOPMENTS AND CONCLUSIONS

The methodology for integrating the outputs from the programs described herein into prioritization and budget allocation modules will be outlined in the future, at which stage a complete optimization and prioritization package from raw data to allocated budget at the network level will be operational. Currently, there is research ongoing to validate the dynamic programming results over a wide range of input values for several different databases by comparing results obtained with a deterministic life-cycle analysis approach. Results from this research will be available in the near future.

The application of dynamic programming as an optimization tool in PMS has great potential for obtaining outputs considerably faster without loss of accuracy. In combination with simple simulation techniques and a probability-based prediction model, it offers the decision maker a simple and rapid method of performing life-cycle analysis, obtaining expected costs and confidence limits of those costs for networks of almost any size at the microcomputer level.

REFERENCES

1. M. Y. Shahin and S. D. Kohn. *Overview of the 'PAVER' Pavement Management System*. Technical Manuscript M-310, USA-CERL, January 1982.
2. M. Y. Shahin, M. M. Nunez, M. R. Broten, S. H. Carpenter, and A. Sameh. New Techniques for Modeling Pavement Deterioration. In *Transportation Research Record 1123*, 1987.
3. A. A. Butt, M. Y. Shahin, K. J. Feighan, and S. H. Carpenter. Pavement Performance Prediction Model Using the Markov Process. In *Transportation Research Record 1123*, 1987.
4. C. L. Reichelt, K. C. Sharaf, K. C. Sinha, and M. Y. Shahin. *The Relationship of Pavement Maintenance Costs to the Pavement Condition Index*. Technical Manuscript M-87-02, USA-CERL, February 1987.
5. Keane and Wu. *An Integrated Decision-Making Methodology for Optimal Maintenance Strategies*. Master's thesis, University of Illinois, 1985.
6. *Introduction to Dynamic Programming*. Pergamon Press, 1981.
7. *Numerical Recipes—The Art of Scientific Computing*. Cambridge University Press, 1986.