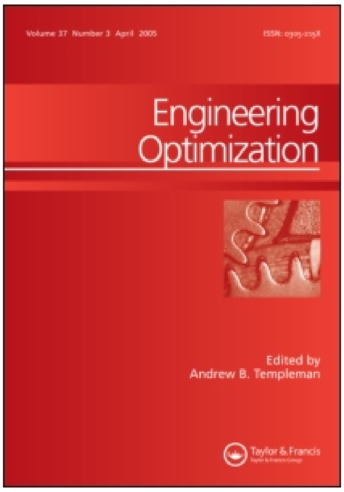

Engineering Optimization
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713641621

Cost-effective selection and multi-period scheduling of pavement maintenance and rehabilitation strategies

J. Yoo[a]; A. Garcia-Diaz[b]
[a] R&D Innovation Team, Corporate Technology Office, Suwon-City, Gyeonggi-Do, Republic of Korea [b] Department of Industrial and Information Engineering, University of Tennessee, Knoxville, Tennessee, USA

To cite this Article Yoo, J. and Garcia-Diaz, A.(2008) 'Cost-effective selection and multi-period scheduling of pavement maintenance and rehabilitation strategies', Engineering Optimization, 40: 3, 205 — 222
To link to this Article: DOI: 10.1080/03052150701686937
URL: http://dx.doi.org/10.1080/03052150701686937

Engineering Optimization
Vol. 40, No. 3, March 2008, 205–222

Taylor & Francis
Taylor & Francis Group

Cost-effective selection and multi-period scheduling of pavement maintenance and rehabilitation strategies

J. Yoo[a] and A. Garcia-Diaz[b]*

[a]*R&D Innovation Team, Corporate Technology Office, Samsung Electronics, Suwon-City, Gyeonggi-Do, Republic of Korea;* [b]*Department of Industrial and Information Engineering, University of Tennessee, Knoxville, Tennessee, USA*

(*Received 22 June 2006; final version received 26 March 2007*)

An optimization methodology is developed for determining the most cost-effective maintenance and rehabilitation (M&R) activities for each pavement section in a highway pavement network, along an extended planning horizon. A multi-dimensional 0–1 knapsack problem with M&R strategy-selection and precedence-feasibility constraints is formulated to maximize the total dollar value of benefits associated with the selected pavement improvement activities. The solution approach is a hybrid dynamic programming and branch-and-bound procedure. The imbedded-state approach is used to reduce multi-dimensional dynamic programming to a one-dimensional problem. Bounds at each stage are determined by using Lagrangian optimization to solve a relaxed problem by means of a sub-gradient optimization method. Tests for the proposed solution methodology are conducted using typical data obtained from the Texas Department of Transportation.

Keywords: multi-dimensional 0–1 knapsack problem; dynamic programming; branch-and-bound; lagrangian relaxation; sub-gradient optimization; optimization in pavement management systems

1. Introduction

As pavements deteriorate, they become structurally deficient and functionally obsolete. In the face of limited resources, selecting and scheduling maintenance and rehabilitation (M&R) programs has become a major concern for most highway agencies. The main focus of this article is the formulation of a model and development of a solution methodology for selecting and scheduling timely cost-effective M&R activities for each pavement section of a highway network, along a specified planning horizon.

The objective function of the optimization model to be considered here is defined as the total measure of effectiveness associated with the set of selected pavement improvement activities over a specified planning period horizon. It is assumed that the unused portion of the budget specified for a period can be carried over to subsequent periods. The measure of effectiveness of a given set of M&R strategies is defined as the *total area* under the corresponding *pavement quality index*

*Corresponding author. Email: agd@utk.edu

ISSN 0305-215X print/ISSN 1029-0273 online

DOI: 10.1080/03052150701686937
http://www.informaworld.com

curves. The problem of selecting and scheduling M&R activities over a specified planning time period is formulated as a multi-dimensional 0–1 knapsack problem with M&R strategy-selection and precedence-feasibility constraints.

The proposed solution methodology in this article can be viewed as a hybrid dynamic programming/branch-and-bound approach. The approach is essentially a dynamic programming approach in the sense that the problem is divided into smaller sub-problems corresponding to each single period problem. However, the idea of fathoming partial solutions that could not lead to an optimal solution is incorporated within the approach to reduce storage and computational requirements in the dynamic programming frame using the branch-and-bound approach.

The imbedded-state approach is used to reduce a multi-dimensional dynamic programming to a one-dimensional dynamic programming. For bounding at each stage, the problem is relaxed in a Lagrangian fashion so that it separates into longest-path network model sub-problems. The values of the Lagrangian multipliers are found by a sub-gradient optimization method, while the Ford–Bellman network algorithm is employed at each iteration of the sub-gradient optimization procedure to solve a longest-path network problem, as well as to obtain improved lower and upper bounds.

Relevant research work in the area of pavement management systems (PMSs) can be classified according to three well-known mathematical optimization techniques: linear programming, integer programming and dynamic programming.

Previous linear programming optimization procedures for PMSs are due to, among others, Golabi *et al.* (1982), Grivas *et al.* (1993) and Mbwana and Turnquist (1996). Li *et al.* (1998) developed a cost-effectiveness-based integer programming on a year-by-year basis for the preservation of deteriorated pavements in a road network with the constraints of budget limitations and required pavement serviceability levels. The objective of the optimization system was to select the most cost-effective M&R projects for each programming year. Fwa *et al.* (2000) developed a genetic-algorithm-based procedure for solving multi-objective network level pavement maintenance programming problems formulated with integer programming. The use of Markov chain prediction models in dynamic programming formulation began to take shape when Butt *et al.* (1987) reported application of a Markov chain to pavement performance prediction. Soon after, Feighan *et al.* (1987) showed how to use dynamic programming for optimization using this Markov model, although it was not fully implemented at the time. Since then a number of articles using dynamic programming or similar formulations have been published (Butt *et al.* 1994, Feighan *et al.* 1988).

Some of the most important limitations of current optimization procedures in most PMSs are: (i) only relatively small-sized problems are considered; (ii) multi-period budgeting and regulated use and frequency of maintenance activities are not included; (iii) state-of-the-art mathematical techniques are implemented for simple scenarios only.

The most significant contributions of this article are as follows.

(i) Development of a computationally effective procedure for multi-period planning of pavement M&R strategies by using network-flow algorithms within a sub-gradient optimization procedure.
(ii) Development of an integrated *optimization* procedure that allows for the selection and scheduling of M&R activities for each section in a highway network subject to *realistic* constraints such as budget, precedence and frequency-of-use constraints regarding M&R strategies.
(iii) Capability of the proposed methodology to allow pavement system managers to make more consistent and effective decisions regarding allocation of limited funds in each period of a planning horizon.

Section 2 presents the mathematical formulation of the problem and the overall solution approach. Section 3 describes each procedural component of the proposed solution methodology. Section 4 shows computational results and includes a short discussion of relevant aspects. Finally, Section 5 consists of a summary along with conclusions and recommendations.

2. Model and solution approach

2.1. *Model*

The pavement network (or system) to be considered consists of I pavement sections and the planning horizon includes T time periods. During each of these periods one of J available pavement improvement strategies is chosen for each pavement section. Let e_{ijt} represent the effectiveness of alternative j for pavement section i in period t, c_{ijt} the cost of alternative j for pavement section i in period t and B_t the budget available for period t. Furthermore, let N_{ij} be the maximum number of times alternative j can be used on pavement section i in the planning period horizon, PQ_{it} the *pavement quality index* of pavement section i in period t, and Δp_j the *treatment effect* of alternative j. Three reference levels of the pavement quality index are defined. First, let M be the maximum pavement quality index allowed, s an index above which no improvement is required for any section at any time and m an index below which a section must be repaired. The decision variable x_{ijt} is equal to 1 if alternative j for pavement section i in period t is selected, and it is equal to 0 otherwise.

A *pavement quality index curve* is a graphical representation of how the index increases as a result of pavement improvement strategies and decreases as a result of increasing traffic loads through periods of the planning horizon. The effectiveness of a set of M&R strategies applied on a section can be determined from the pavement quality index curves. For example, the illustration shown in Figure 1 indicates that a section is improved in period 5s and 8 of a 10-year planning horizon. The effectiveness of strategy j in period 5, e_{ij5}, is defined as the area under the pavement quality level curve in period 5 (the shaded area in Figure 1). The stochastic variation of pavement quality over time could be represented by a known transition probability matrix, which is assumed to be available in this article.

The problem of generating a sequence of interrelated M&R strategies over a fixed planning time period for each pavement section can be formulated as a multi-dimensional 0–1 knapsack

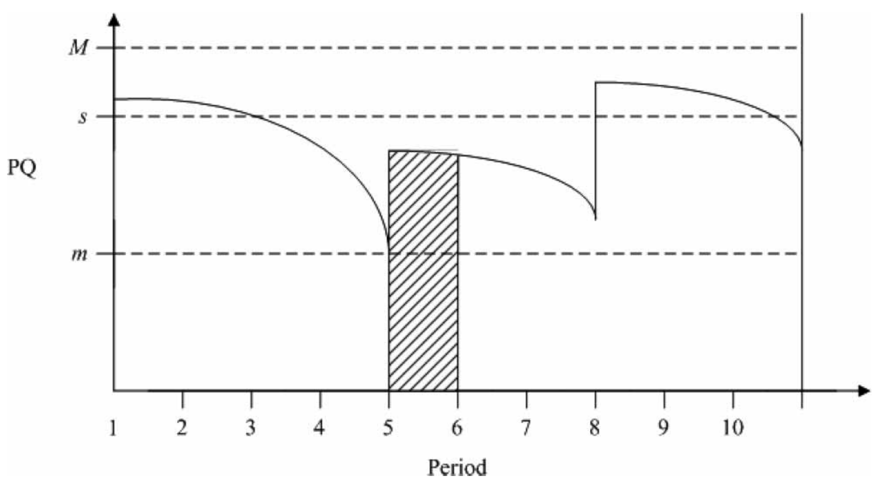

Figure 1. Pavement quality level curves for a set of M&R strategies.

problem with alternative selection and precedence-feasibility constraints. In this formulation, the overall highway pavement quality level is maximized without exceeding the budget available in each period or without exceeding the maximal frequency allowed for the chosen M&R actions over the planning periods. It is furthermore assumed that unused budget portions in a period are carried over to subsequent periods. The mathematical model is referred to as Problem (P) and is formulated in relationships (1)–(8).

Problem (P)

$$\max \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{j=1}^{J} e_{ijt} x_{ijt} \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ijt}\, x_{ijt} \leq B_t \quad \text{for all } t \tag{2}$$

$$\sum_{t=1}^{T} x_{ijt} \leq N_{ij} \quad \text{for some } i, j \tag{3}$$

$$\sum_{j=1}^{J} x_{ijt} = 1 \quad \text{for all } i, t \tag{4}$$

$$PQ_{it} \geq m \quad \text{for all } i, t \tag{5}$$

$$PQ_{it} \geq s \Longrightarrow \sum_{j=2}^{J} x_{ijt} = 0 \quad \text{for all } i, t \tag{6}$$

$$PQ_{i,t-1} + \sum_{j=1}^{J} \Delta p_j x_{ijt} \leq M \quad \text{for all } i, t \tag{7}$$

$$x_{ijt} \in \{0, 1\} \quad \text{for all } i, j, t. \tag{8}$$

In the formulation of Problem (P) the objective function (1) being maximized is defined as the total measure of effectiveness for the system. The budget constraint set (2) indicates that capital consumption by the selected alternatives cannot exceed the available budget in each period. The frequency constraint set (3) ensures that some alternatives on some pavement sections cannot be taken more than the available frequency regarding the lifetimes of the alternatives over the multi-period planning horizon. The alternative-selection constraint set (4) forces the problem to choose one and only one strategy for each pavement section in any time period; strategy 1 is defined as the 'do-nothing' option. Constraint set (5) is used to eliminate any alternative strategy that does not meet the minimum pavement quality level requirements for a pavement section in a period. Constraint set (6) ensures that a pavement section is not considered for maintenance if its condition is better than a predefined serviceable pavement quality level in a given period. Constraint set (7) limits the alternatives to be chosen from among those providing treatment effects such that the maximum level of the quality index after improvement does not exceed a specified value M. Constraint set (8) imposes integrality of the decision variables.

There are three types of constraints imposed on this problem: resource, alternative-selection and precedence-feasibility. The resource constraints consist of constraint sets (2) and (3). Constraint set (4) is an alternative-selection constraint. Constraint sets (5), (6) and (7) are associated with precedence-feasibility constraints.

2.2. *Solution approach*

Dynamic programming and branch-and-bound approaches are combined to produce a hybrid algorithm for solving the problem formulated as a multi-dimensional 0–1 knapsack problem with alternative-selection and precedence-feasibility constraint. The algorithm is essentially a dynamic programming approach in the sense that the problem is divided into smaller sub-problems corresponding to each single period problem. However, the idea of fathoming partial solutions that could not lead to an optimal solution is incorporated within the algorithm to reduce storage and computational requirements in the dynamic programming using the branch-and-bound approach. The procedure for the hybrid dynamic programming/branch-and-bound algorithm was initially proposed by Marsten and Morin (1978).

The salient feature of the proposed hybrid algorithm is its capability of reducing the state-space, which otherwise would present an obstacle in solving multi-dimensional dynamic programming problems. This is due to the use of the imbedded-state approach, which reduces a multi-dimensional dynamic programming to one-dimensional dynamic programming. Morin and Marsten (1976) applied the imbedded-state approach to develop an algorithm for the solution of nonlinear knapsack problems. Other reductions are made through fathoming the state-space and subsequent elimination of the state-space, which tends to eliminate inferior solutions compared with the predetermined lower bound or updated lower bound.

Due to alternative-selection, precedence-feasibility and an integrality constraint set, the original problem can be transformed to a resource-constrained longest-path problem (RCLPP). Lagrangian relaxation of this problem into an unconstrained longest-path problem provides initial lower and upper bounds on the objective function, as well as lower and upper bounds for bounding tests at each stage of dynamic programming approach. At each stage, the lower and upper bounds are also updated and are used as termination and fathoming criteria. The relaxed problem can be solved by using a sub-gradient optimization procedure, while a network algorithm (Ford–Bellman) is employed at each iteration of the sub-gradient optimization procedure to solve the longest-path network problem, as well as to obtain an improved lower and upper bound. If the gap of the lower and upper bounds is a predetermined parameter ε or the improved lower bound is optimal, then the procedure is terminated rather than continuing to stage T. Otherwise, the dynamic programming approach for a single-period problem is used to identify feasible solutions to the next period problem corresponding to the next stage in the multi-period dynamic programming formulation.

Feasible solutions dominated by any other feasible solutions are eliminated and *efficient solutions* are selected from the remaining set of strategies. By performing a bounding test, the efficient partial solutions that cannot lead to a solution that has a lower bound better than the best known bound are fathomed. Lagrangian relaxation and sub-gradient optimization procedures are applied to the remaining problem in order to perform the bounding process at the current stage. The survivors are then used to generate potential solutions for the next stage.

Figure 2 shows the overall conceptual approach of the proposed methodology, which can be divided into two procedures—dynamic programming and branch-and-bound. The dynamic programming procedure consists of Steps 4 and 5; the branch-and-bound procedure comprises steps 1, 2, 3, 6 and 7. A brief description of each major component of the methodology is as follows.

Step 1. Model reformulation. The set of alternative-selection, precedence-feasibility and integrality constraints allow Problem (P) to be transformed as an equivalent problem referred to as the RCLPP model.

Step 2. Lagrangian relaxation. The RCLPP model is relaxed by dualizing the budget constraint set in each period and the frequency-constraint set over a multi-period planning horizon, so that the relaxed problem is decomposable into one sub-problem for each pavement section.

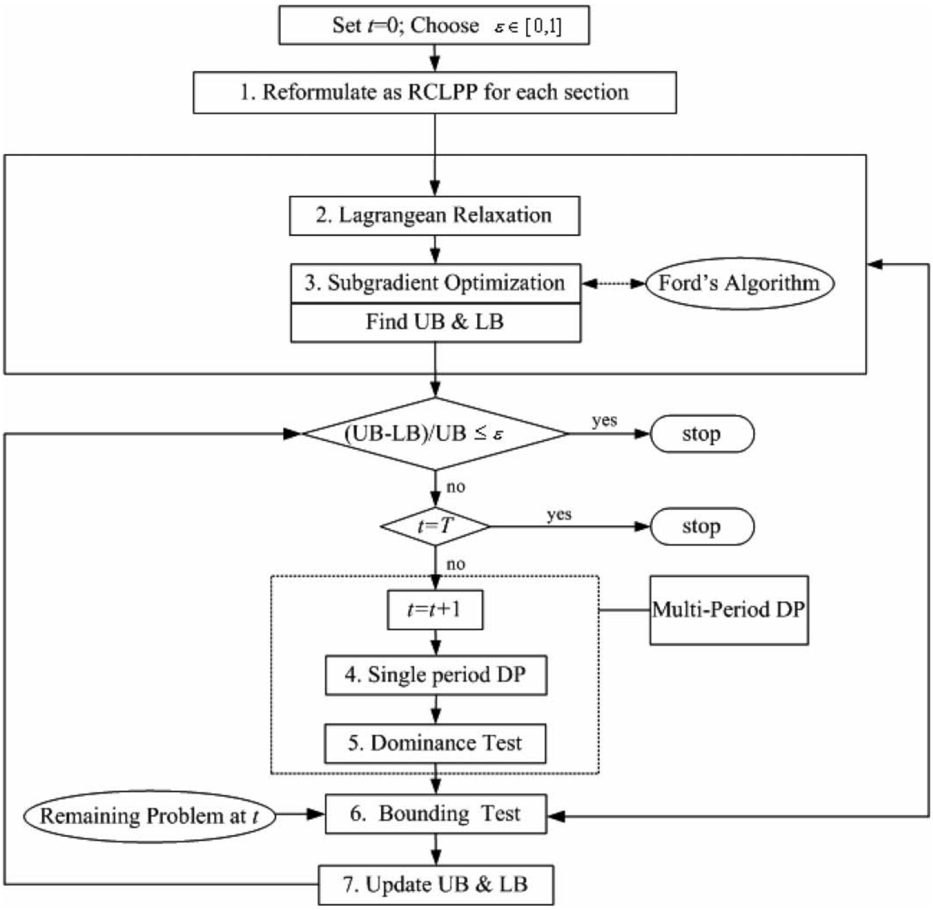

Figure 2. Overall conceptual approach.

Step 3. Sub-gradient optimization. The value of the Lagrangian multipliers, which gives the least upper bound for Problem (P), is obtained by sub-gradient optimization. At each iteration of the sub-gradient optimization procedure, a network algorithm (Ford–Bellman) is employed to solve the longest-path network problem as well as to obtain an improved lower and upper bound for Problem (P). If the improved lower bound is optimal or the gap of the lower and upper bound are a predetermined parameter ε, then the procedure is terminated; otherwise, Step 4 is performed next.

Step 4. Single-period dynamic programming. A dynamic programming decomposition methodology is conducted to identify feasible solutions to a single-period problem corresponding to a stage in multi-period dynamic programming approach. This is done by means of the imbedded-state approach.

Step 5. Dominance test. Feasible solutions that are dominated by any other feasible solutions are eliminated. The remaining set of solutions will be referred to as *efficient* solutions.

Step 6. Bounding test. Steps 2 and 3 are performed on the remaining problems. Efficient partial solutions that cannot lead to a solution that is better than the best known lower bound are fathomed. The *survivors* (those not eliminated by the bounding test) at the stage representing the current period are obtained and used to generate potential solutions for the next stage.

Step 7. Update upper bound (UB) and lower bound (LB). The values of both UB and LB at the current stage are updated if the values obtained from the remaining problems for the *survivors* in Step 6 are better than the best known ones.

3. Description of the solution method

3.1. *Resource-constrained longest-path network representation*

By means of the set of alternative-selection, precedence-feasibility and integrality constraints, the model for Problem (P) can easily be reformulated as the RCLPP model. The network model for pavement section i is illustrated in Figure 3. This model is built in a stage-wise fashion, with each stage corresponding to a time period. The total number of stages is equal to T. Variables considered for network generation at each stage t (where $t = 1, 2, \ldots, T$) are all included in the corresponding alternative-selection constraint set.

At stage t, variables $x_{i1t}, x_{i2t}, \ldots, x_{iJt}$ are considered. Node s_i is a *source node*. A feasible set of variables in an alternative-selection constraint set is determined from the precedence-feasibility constraints set with period $t = 1$, and arcs are added for each feasible strategy x_{ij1} from the source node to nodes $1, 2, \ldots, m$, where m is the number of feasible strategies in stage $1(m \leq J)$. This set of nodes is considered to represent stage 1. A feasible set of strategies at stage 2 is again determined at each of the nodes $1, 2, \ldots, m$, and more arcs and nodes are generated for each of these nodes to represent stage 2. This process is continued until stage T is reached. Arcs emanating from each node in stage T are connected to a single node e_i, defined as the *sink node*. The arc lengths are calculated from the objective function for the corresponding value of alternative j and period t on pavement section i. This calculation is possible because each e_{ijt} is a function of the strategies employed at previous stages on a path from node s_i to a particular node.

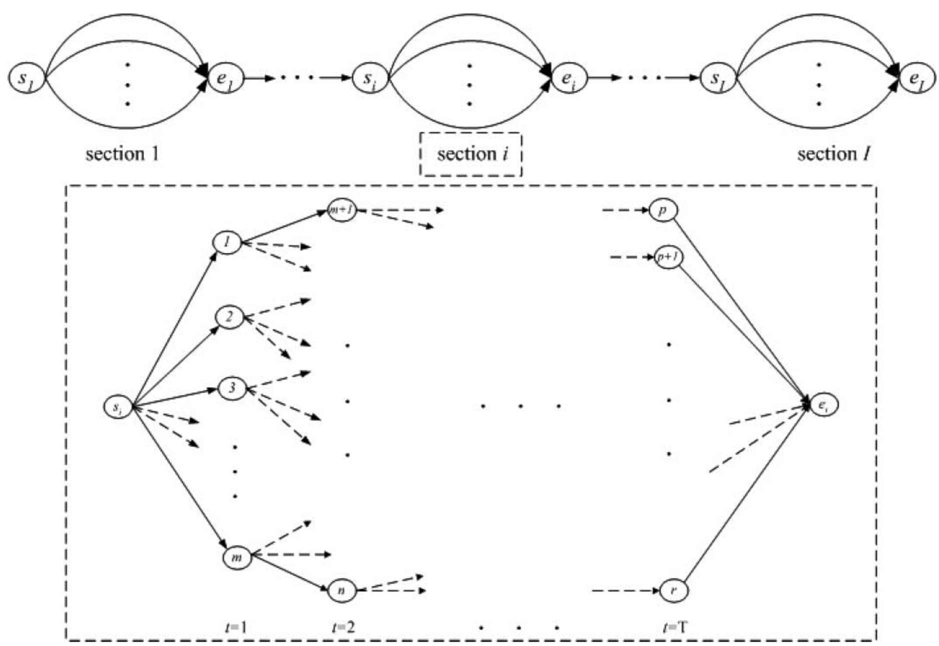

Figure 3. Network model for pavement Section i.

Even if there are only four or five strategies feasible at each stage, the number of nodes and arcs rapidly increases beyond computational limitations. However, the number of arcs and nodes can be reduced by the following method. Suppose at some node n at stage t strategy j is feasible and an arc is emanating from node n to some other node $q > n$, where, as previously described, node q is at a stage $t + 1$. If only strategy 1 (do nothing) were feasible for node q, the corresponding effective coefficient $e_{i,1,t+1}$ would be added to the length of the arc from node n to node q, and at this point node q is moved into stage $t + 2$. This process is repeated until node q reaches a stage $t\prime$ such that a strategy other than just strategy 1 is feasible or node q reaches state T. This procedure is always applicable if there are precedence-feasibility constraints in the problem.

A similar network is generated for each sub-problem corresponding to each pavement section in the highway network. The network models for each sub-problem are linked sequentially such that e_i is connected to s_{i+1} for $i = 1, 2, \ldots, I - 1$. Then the longest-path from source s_1 to sink node e_I (satisfying resource constraints) is an optimal solution to Problem (P).

3.2. *Lagrangian relaxation*

A general theory of Lagrangian relaxation can be found in the literature (Geoffrion 1974). Specifically, Beasley and Christofides (1989) applied Lagrangian relaxation to develop an algorithm for the resource-constrained shortest-path problem. The Lagrangian relaxation approach to obtaining bounds in integer programming problems is the second most widely used after longest-path relaxation. A Lagrangian relaxation of an integer programming problem is obtained by removing the complicating constraints and including them in the objective function using multipliers so that the resulting problem is much easier to solve because of the special structure of the remaining constraints, and sometimes being decomposable by itself. A Lagrangian relaxation scheme is more attractive than longest-path relaxation if the decomposition and continuity of the special structure can be achieved.

Using matrix notation, Problem (P) can be reformulated as

Problem (P)

$$\max\{RX : AX \leq b, X \in \Omega\}, \tag{9}$$

where (A, b) is for the knapsack constraint set and $\Omega = \{X$: alternative-selection constraint, precedence-feasibility constraint, integrality constraint$\}$. By dualizing the knapsack constraint set, the relaxed Problem $(LR(\lambda))$ is obtained:

Problem $(LR(\lambda))$

$$\max\{(R - \lambda A)X + \lambda b : X \in \Omega\}, \tag{10}$$

where λ is the vector of Lagrangian multipliers. This problem can be easily solved by decomposing it into I sub-problems, one sub-problem for each pavement section that represents a longest-path problem. The solution to each sub-problem is obtained by applying the Ford–Bellman algorithm.

The least upper bound for Problem (P) is obtained by solving Problem (LD).

Problem (LD)

$$\min\{LR(\lambda) : \lambda \geq 0\}. \tag{11}$$

An optimal value of the Lagrangian multipliers, λ^*, is an optimal solution to the Lagrangian dual, Problem (LD), but it need not be solved optimally. Since the convergence rate of λ is very slow in the neighborhood of the optimal value, a near-optimal value will be satisfactory. The sub-gradient optimization method will be used to solve Problem (LD).

3.3. *Sub-gradient optimization*

A review of methods for solving the Lagrangian dual, Problem (*LD*), can be found in Bazaraa and Goode (1979). The most popular method is sub-gradient optimization because it is easy to implement and has worked well on many practical problems, especially on 0–1 integer programming problems. The sub-gradient method is an adaptation of the gradient method in which gradients are replaced by sub-gradients. Given λ_0, a sequence λ_p is generated by the rule

$$\lambda_{p+1} = \lambda_p + t_p(AX_p^* - b)^T, \tag{12}$$

where:

$\lambda_p =$ the value of λ at iteration p of the sub-gradient procedure; usually $\lambda_0 = 0$ is the most natural choice, but in some cases other appropriate values (which are obtainable through experiments) can do better;
$t_p =$ the step length at iteration p, given by

$$t_p = \pi_p \frac{v(LR(\lambda_p)) - LB}{||AX_p^* - b||^2}; \tag{13}$$

$X_p^* =$ the optimal solution to Problem ($LR(\lambda_p)$);
$\pi_p =$ a scalar satisfying $0 < \pi_p \leq 2$;
$v(LR(\lambda_p)) =$ the value of the optimal solution to Problem ($LR\ (\lambda_p)$);
$LB =$ the lower bound $=$ the value of the best known solution; and
every negative element of λ_{p+1} must be replaced with zero because of the non-negativity requirement of λ.

The sub-gradient optimization algorithm used in this article is delineated as follows. Since the sub-problem structure is straightforward, the sub-gradient algorithm can be run for a large number of iterations in order to ensure convergence of the Lagrangian multipliers to a near-optimal value.

3.3.1. *Sub-gradient optimization procedure*

(i) Choose $\lambda_0 \geq 0, 0 < \pi_0 \leq 2, z^* = a$ large number, $X^* = 0$ and $\eta^* = \lambda_0$, Let $p = 0$ and $LB = 0$ (the value of the best known solution).
(ii) Solve Problem ($LR(\lambda_p)$). Let X_p^* be the solution.
(iii) If X^* is feasible, then if $v(LR(\lambda_p)) - LB < \varepsilon$, go to (xi); otherwise, $X = X_p^*$.
(iv) If $RX > LB$, then $LB = RX$ and the best known solution $= X$.
(v) If $z^* > v(LR(\lambda_p))$, then $z^* = v(LR(\lambda_p))$, $X^* = X_p^*$ and $\eta^* = \lambda_p$.
(vi) If the improvement of $v(LR(\lambda_p))$ is not more than 1 in 10 consecutive iterations, then set π_p equal to 0.5 π_p.
(vii) If the improvement of $v(LR(\lambda_p))$ is not more than 1 in 20 consecutive iterations, then go to (x).
(viii) Obtain (λ_{p+1}) using Equation (12).
(ix) Set p equal to $p + 1$. Go to (ii).
(x) Terminate with $\lambda^* = \eta^*$, $v(LD) = v(LR(\lambda^*)) = z^*$, X^*, and the best known solution and LB.
(xi) X_p^* is an optimal solution to Problem (P) and the procedure is terminated.

3.4. *Dynamic programming strategy for multiple periods*

The dynamic programming model for multiple periods is developed in a compact form of the separable nonlinear multidimensional knapsack problem (NKP) as shown in Problem (D), which is equivalent to Problem (P), and is as shown in Figure 4.

Problem (D)

$$f_T(b) = \max \sum_{t=1}^{T} R_t \hat{X}_t \tag{14}$$

$$\text{s.t.} \quad \sum_{t=1}^{T} A_{rt} \hat{X}_t \le b_r \quad 1 \le r \le T + I + J \tag{15}$$

$$\hat{X}_t \in \Omega_t \quad t = 1, 2, \ldots, T, \tag{16}$$

where $\hat{X}_t$ is an M&R strategy for a highway network in period t ($\hat{X}_t = (X_{1t}, X_{2t}, \ldots, X_{It})$, $X_{it} = (x_{i1t}, x_{i2t}, \ldots, x_{iJt})$), R_t is a return (effectiveness) of strategy $\hat{X}_t$, A_{rt} is the amount of resource (budget and frequency of actions) r taken by strategy $\hat{X}_t$ for a highway network in period t, b_r is an available resource r and $\Omega_t = \{\hat{X}_t$: alternative-selection constraint, precedence-feasibility constraint, and 0–1 integrality constraint$\}$.

Referring to Figure 4, allocation of resources to a highway network using the dynamic programming approach results in the recursive relationship:

$$f_n^*(S_n) = \max_{\hat{X}_n \in \Phi_n} \left\{ f_{n-1}^*(S_{n-1}) + R_n(S_n, \hat{X}_n) \right\}, \tag{17}$$

where $\Phi_n = \{\hat{X}_t$; budget constraint, frequency of action constraint, alternative-selection constraint, precedence-feasibility constraint, & 0–1 integrality constraint$\}$ and the state variable S_n represents the amount of resource b that is available for allocation in period n and is a $T + I \times J$ dimensional vector. The vector is divided into two groups. The first group is represented by a T-dimensional vector corresponding to the budget in each period. The second group is represented by an $I \times J$-dimensional vector corresponding to frequency availability of actions for pavement sections.

Problem (D) can be decomposed into sub-problems that can be considered as a single stage in the multi-period dynamic programming problem. In each single period, the feasible solutions are obtained by applying a single-period dynamic programming approach and the efficient solutions that are not dominated by any other feasible solutions are obtained by dominance testing.

Let $\tilde{X}_n^f$ denote the set of feasible solutions of Problem (D) until stage n. The feasible solution $\tilde{X}_n = (\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n) \in \tilde{X}_n^f$ is said to be dominated by the feasible solution $\tilde{X}_n' \in \tilde{X}_n^f$, if we

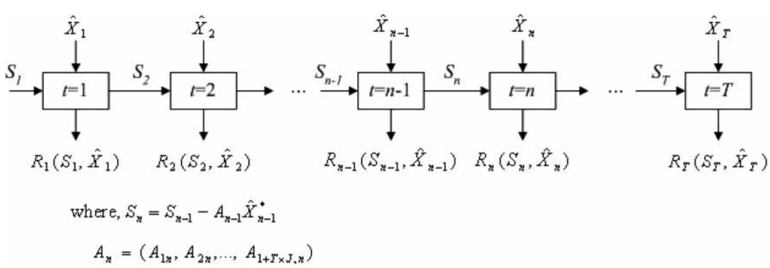

Figure 4. Dynamic programming for multiple periods.

have both

$$\sum_{t=1}^{n} A_{rt} \hat{X}'_t \leq \sum_{t=1}^{n} A_{rt} \hat{X}_t, \quad \forall r, \tag{18}$$

and

$$\sum_{t=1}^{n} R_t \hat{X}'_t \geq \sum_{t=1}^{n} R_t \hat{X}_t, \tag{19}$$

with at least one strict inequality. If $\tilde{X}_n \in \tilde{X}_n^f$ is not dominated by any other element of $\tilde{X}_n^f$, then we say that $\tilde{X}_n$ is efficient with respect to $\tilde{X}_n^f$.

3.5. *Single-period dynamic programming*

In order to obtain feasible and efficient solutions in each period, a dynamic programming approach is applied to single-period problems corresponding to single stages in multi-period dynamic programming. A single-period dynamic programming model is developed in a compact form as shown in Problem (D_S) and Figure 5.

Problem (D_S)

$$R_n(S_n, \hat{X}_n) = \max \sum_{i=1}^{I} R_{in} X_{in} \tag{20}$$

$$\text{s.t.} \sum_{i=1}^{I} A_{rin} X_{in} \leq S_{rn}, \quad 1 \leq r \leq T + I \times J \tag{21}$$

$$X_{in} \in \Omega_{in}, \quad i = 1, 2, \ldots, I, \tag{22}$$

where X_{in} is $(x_{i1n}, x_{i2n}, \ldots, x_{iJn})$, A_{rin} is the amount of resource (budget and available frequency of actions) r taken by strategy X_{in} for pavement section i in period n, $1 \leq r \leq T + I \times J$, S_{rn} is available resource r in period n, R_{in} is the return (effectiveness) of strategy X_{in}, and $\Omega_{in} = \{X_{in}$: alternative-selection constraint, precedence-feasibility constraint, and 0–1 integrality constraint$\}$.

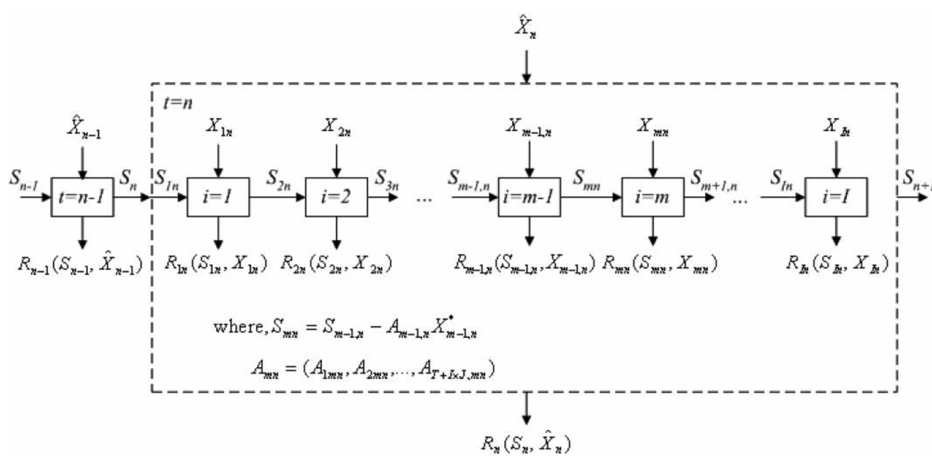

Figure 5. Single-period dynamic programming model.

Allocation of resources to a highway network using the dynamic programming approach in a single-period results in the recursive relationship:

$$f_{mn}^{*}(S_{mn}) = \max_{X_{mn}\in\Phi_{mn}} \{f_{m-1,n}^{*}(S_{m-1,n}) + R_{mn}(S_{mn}, X_{mn})\}, \tag{23}$$

where $\Phi_{mn} = \{X_{mn}$; budget constraint, frequency of action constraint, alternative-selection constraint, precedence-feasibility constraint, and 0–1 integrality constraints$\}$, and the state variable S_{mn} represents the amount of resources available for allocation on pavement section m in period n. The solution to $f_{In}^{*}(S_{In})$ in period n is equal to the solution to $R_n(S_n, \hat{X}_n)$.

3.6. *Branch-and-bound approach*

Fathoming of a partial solution by the branch-and-bound approach effectively eliminates non-promising points from the state-space and hence provides extensive savings in computational time and storage. This is done by incorporating elimination by bound into the dynamic programming framework.

Consider any $\tilde{X}_n = (\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n) \in \tilde{X}_n^e$, where $\tilde{X}_n^e$ denotes the set of efficient solutions, and let

$$\beta = \sum_{t=1}^{n} A_t \hat{X}_t. \tag{24}$$

We may interpret β as the resource consumption vector for the partial solution $\tilde{X}_n$. Given $\tilde{X}_n$, the remaining problem at stage n, Problem (R) is formulated as follows.

Problem (R)

$$\bar{f}_{n+1}^{*}(b-\beta) = \max \sum_{t=n+1}^{T} R_t \hat{X}_t \tag{25}$$

$$\text{s.t.} \sum_{t=n+1}^{T} A_{rt}\hat{X}_t \le b_r - \beta_r, \quad 1 \le r \le T + I \times J \tag{26}$$

$$\hat{X}_t \in \Omega_t, \quad n+1 \le t \le T. \tag{27}$$

Thus $\bar{f}_{n+1}^{*}(b-\beta)$ is the maximum possible return from the remaining stages, given that resources β have already been consumed. For each $0 \le n \le T-1$, let UB_{n+1} be an upper bound functional for $\bar{f}_{n+1}^{*}(b-\beta)$, *i.e.*

$$\bar{f}_{n+1}^{*}(b-\beta) \le UB_{n+1}(b-\beta) \quad \text{for all } 0 \le \beta \le b \tag{28}$$

UB_{n+1} may be taken as the optimal value of any relaxation of Problem (R).

Any known feasible solution of Problem (D) provides a lower bound on $f_T^{*}(b)$. The best of the known solutions is called the *incumbent* and its value denoted LB, so that $LB \le f_T^{*}(b)$. These upper and lower bounds can be used to eliminate efficient partial solutions, which cannot lead to a solution that is better than the incumbent. That is, if $\tilde{X}_n \in \tilde{X}_n^e$ and

$$\sum_{t=1}^{n} R_t \hat{X}_t + UB_{n+1}\left(b - \sum_{t=1}^{n} A_t \hat{X}_t\right) \le LB, \tag{29}$$

then no completion of $\tilde{X}_n$ can be better than the incumbent. The survivors at stage n will be denoted $\tilde{X}_n^s$ where

$$\tilde{X}_n^s = \left\{ \tilde{X} \in \tilde{X}_n^e \,\middle|\, \sum_{t=1}^{n} R_t \hat{X}_t + UB_{n+1}\left(b - \sum_{t=1}^{n} A_t \hat{X}_t\right) > LB \right\}. \tag{30}$$

The lower bound may be improved during the course of the algorithm by finding additional feasible solutions. Only the survivors at stage n are used to generate potential solutions at stage $n+1$.

As mentioned in Sections 3.2 and 3.3, Lagrangian relaxation and sub-gradient optimization techniques are applied to obtain tight bounds on Problem (R) at each stage in multi-period dynamic programming. The relaxed problem, Problem′ ($LR(\lambda)$), is formulated as follows.

Problem′ ($LR(\lambda)$)

$$\max \left\{ \sum_{t=n+1}^{T} (R_t - \lambda A_t)\hat{X}_t + \lambda(b - \beta) : \hat{X}_t \in \Omega_t \right\}, \tag{31}$$

where λ is the Lagrangian multiplier vector. This problem can be solved by decomposing it into I sub-problems, one sub-problem for each pavement section, each of which is a longest-path problem and is solved by the Ford–Bellman network flow algorithm.

The sub-gradient optimization method is applied to Problem′ ($LR(\lambda)$) to obtain a good (near-optimal) set of Lagrangian multipliers while improving feasible solutions. An integer feasible solution could be obtained through the solution process of Problem′ ($LR(\lambda)$). If this integer feasible solution is better than the currently known best solution,

$$\sum_{t=1}^{n} R_t \hat{X}_t + UB_{n+1}\left(b - \sum_{t=1}^{n} A_t \hat{X}_t\right) > LB, \tag{32}$$

then this solution becomes the best known solution or the incumbent.

At the end of stage n we know that $f_T^*(b)$ falls between LB and the global upper bound:

$$UB = \max \left\{ \sum_{t=1}^{n} R_t \hat{X}_t + UB_{n+1}\left(b - \sum_{t=1}^{n} A_t \hat{X}_t\right) \,\middle|\, \tilde{X}_n \in \tilde{X}_n^s \right\}. \tag{33}$$

If the gap $(UB\text{-}LB)/UB$ is sufficiently small, then we may choose to accept the incumbent as being close to optimality in value and terminate the algorithm rather than continuing to stage T.

4. Computerization and applications

All the developed procedures have been computerized and run for a range of multi-period planning PMS scenarios to test the computational performance of the proposed methodology. In order to demonstrate application to a state department of transportation, typical data for the state of Texas were used. All procedures were coded using MATLAB and executed on a PC with an Intel Pentium IV 3.06 GHz processor.

For the sample highway maintenance problem, a total of 402.5 lane-miles of different class of highways were taken from Brazos and Robertson counties in the Bryan district. The network is segmented into 40 pavement management sections. Data for the problem were obtained from the 1998 Road Inventory File. The set of pavement sections used in this problem was segmented by the column of 'highway department control/section number' in the 1998 Road Inventory File.

Table 1. Pavement network information data.

Pavement management section	Highway type	Length (miles)	Lane	Traffic volume AADT/truck (%)	Current PSI
004909	US	7.5	4	224170/7.9	3.9
004912	SH	13.6	3.8	456510/11.7	2.6
005001	SH	7.9	4	573500/1.8	2.7
005002	SH	15.5	4	192080/12.6	3.2
011604	SH	11.8	3.4	424200/14.3	2.8
011605	FM	1.3	4	55700/9.2	3.3
011701	US	9.8	2.9	162500/12.3	3.4
011702	US	6.8	2	42200/16.1	2.6
021203	FM	12.9	2.7	283200/11.5	3.8
047501	SH	8	2	5020/18.5	3.2
047502	SH	21	2	13800/16.8	2.6
050601	FM	10.8	3.8	894200/4.6	4
054003	FM	11.6	2	35700/9.6	3.8
054004	FM	21.4	3	306180/4.4	4.1
054005	FM	16.6	2	4170/8.5	3.9
059901	SH	1.4	4	46500/1.9	3.7
064802	FM	4.8	2	7850/10.7	2.6
131601	FM	14.9	3	263410/6.8	3.8
131602	FM	5.4	2	920/15.3	2.5
156001	FM	10.8	2	10030/10.4	3.3
004906	US	14.5	2.8	140700/23.4	3.5
004907	US	7.1	3.4	153300/18.7	3.3
004908	US	12.2	4	332600/14.7	3.6
004914	FM	1.3	2	4050/14.2	3.8
004915	SH	4.2	2	8950/12.6	3.5
009308	US	1.3	2	3500/12.9	3.3
020409	US	8.5	2.7	60300/18.9	3.4
020501	US	9.1	2.3	56100/22.9	2.6
020502	US	17.5	2.3	111000/25	3.8
026203	FM	6.1	2	14000/9.3	3.2
026206	SH	13.3	2	24450/6.5	2.6
038204	SH	8.9	2	20150/14.9	4
054001	FM	16.9	2	26060/9.3	3.8
054002	FM	13	2	19050/9.6	4.1
054006	FM	10.5	2	3150/13.1	3.9
064801	FM	11.8	2	8700/15.4	3.7
119105	FM	5.5	2	1240/13.7	2.6
121001	FM	10.7	2	720/12.6	3.8
121002	FM	5.1	2	1020/12.4	2.5
156301	FM	11.3	2	3120/10.8	3.2

Note: US: United States; SH: State Highway; FM: Farm-to-Market; AADT: Annual Average Daily Traffic.

Pavement network information data, including pavement section code, highway type, section length and lane, traffic volume, and the current present serviceability index (PSI) are shown in Table 1. PSI is used to measure pavement section deterioration. The Markov chain approach is used to reflect the stochastic nature of individual changes in present serviceability and service life.

There are four alternatives to be considered for managing a highway network: (1) do nothing; (2) minor maintenance; (3) major maintenance; and (4) rehabilitation. Table 2 lists three standardized M&R treatment strategies for the highway network. Each of the strategies includes treatment requirements and specifications for each M&R action, treatment effects in terms of raising the existing pavement condition state by a certain amount of PSI points and unit costs for implementing the M&R actions. It is assumed that application of M&R treatment strategies to highway pavement sections results in that highway pavement rating being set equal to the points gained by application of these strategies for highway pavements. The highway pavement quality level resulting from application of any maintenance strategy cannot be greater than the

Table 2. Treatment types and costs.

Treatment strategy	Treatment requirement and specification	Treatment effect & impact/cost ($1000/lane mile)
Minor maintenance	• Crack sealing • Joint sealing • Surface sealing	• Raise existing PSI by 0.5 • Unit cost: $6
Major maintenance	• Concrete pavement restoration • Thin asphalt overlay	• Raise existing PSI by 1.0 • Unit cost: $60
Rehabilitation	• Patching • Mill and thick asphalt overlay	• Raise existing PSI by 1.5 • cost: $125

ideal highway pavement quality or less than the minimum. If an application of any one strategy causes this to occur, the maintenance strategy is infeasible because of the constraints of desired driving requirements of highway pavement quality.

The minimum pavement quality level and the serviceable pavement quality level are defined to be 50% and 80% of the maximum quality level. These are used to determine feasible strategies in each time period. A pavement section is also not considered for maintenance scheduling if the pavement quality levels are greater than the serviceable pavement quality level. Theses constraints, along with the alternative-selection constraints, are used to construct a network model for each pavement section over multiple periods. To examine the behavior of the proposed algorithm as a function of both problem size and budget availability, the following combinations were considered

(i) number of pavement sections $\in \{20, 40\}$
(ii) number of M&R alternatives $\in \{3, 4\}$
(iii) number of periods $\in \{5, 7\}$
(iv) budget availability factor $\in \{10\%, 20\%\}$.

Each combination is referred to as a problem type. The budget availability factor is defined as a percent value θ such that the budget for period t is set equal to $B_t = \theta A_t$, A_t being the *average* cost for all M&R activities in the period.

It is assumed that the unused portion of the budget for one period can be carried over to subsequent periods and major M&R on each pavement section cannot be taken more than once over the planning periods for which the number of M&R alternatives factor is 3 and 4.

A fractional factorial design was applied to plan the experiments. In this experiment there are four factors, each with two levels: (1) number of pavement sections; (2) number of M&R alternatives; (3) number of periods; and (4) budget availability factor. The eight experimental conditions were obtained from orthogonal arrays for the experimentation. Table 3 represents the orthogonal array OA (8,4,2,3)—in this notation, 8 indicates the number of the run; 4 the number of the factor; 2 the number of the levels; and 3 the strength, which is the number of columns where seeing all the possibilities an equal number of times is guaranteed. The per-level combination factors are translated into problem types. The rows of the array represent the experimental conditions. The columns of the orthogonal array correspond to the different variables or factors whose effects are being analysed. The entries in the array specify the levels at which the factors are to be applied.

The quality of solutions is measured by a gap, defined as truncated hundredths of a percent of the difference between the lower bound (the value of an optimal solution or the best known solution) and the upper bound compared to the upper bound:

$$\text{Gap} = 100\frac{\text{Upper Bound Value–Lower Bound Value}}{\text{Upper Bound Value}}.$$

Table 3. OA (8,4,2,3).

0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Table 4. Computational results.

Problem type				Measures		
Section, i	Alternative, j	Period, t	Budget, θ (%)	CPU time (min:s)	Gap	Memory usage (MB)
20	3	5	10	02:55	0.0143	0.45
20	3	7	20	08:10	0.0116	2.69
20	4	5	20	00:28	0.0171	0.78
20	4	7	10	24:19	0.0138	3.66
40	3	5	20	02:07	0.0158	0.82
40	3	7	10	133:09	0.0050	5.33
40	4	5	10	02:11	0.0112	1.55
40	4	7	20	19:23	0.0142	8.01

In this case study, it is assumed that the gap between lower and upper bounds is within 2% for a termination rule of the proposed algorithm.

In addition to the gaps, the computational performance is measured by computation times in minutes and seconds. Computational results for the experiment are reported in Table 4, which indicates that every solution for each experimental condition is within a 2% gap and memory usage increases with increasing problem size.

In the range of this experiment, computation times seem to decrease with earlier periods finding a solution within a 2% gap and seem to increase with increasing number of efficient solutions obtained by using dynamic programming in each period. This indicates that more efficient solutions in each period take much longer to obtain survivors (or a solution within the 2% gap) because of the number of performing bounding tests. The increasing budget level reduces the fewer variables since the number of promising M&R alternative combinations is increased.

The problem size is exponentially proportional to the number of pavement sections and periods. Increasing the number of periods seems to increase computation time in obtaining bounds on efficient solutions (using Lagrangian relaxation and sub-gradient optimization methodology); increasing the number of pavement sections tends to increase computation time for obtaining efficient solutions in each period (using dynamic programming).

The dominating factor in computation time is likely to be budget availability and data structure since the computation time appears to decrease or increase according to the combination of the two. Problems with some combinations will be harder to solve than those of larger size with other combinations. For example, the computation time for a problem with 40 pavement sections, 3 alternatives, 7 periods and 10% budget level is about 133 minutes compared with about 19 minutes for a problem with 40 pavement sections, 4 alternatives, 7 periods and 20% of the budget level. Figure 6 shows the variation of computation time with problem size.

No problem was solved to optimality in the proposed algorithm. It is not possible to compare the computational results from this algorithm with those from others since no special-purpose algorithms for this type of research problem are available in the related technical literature.

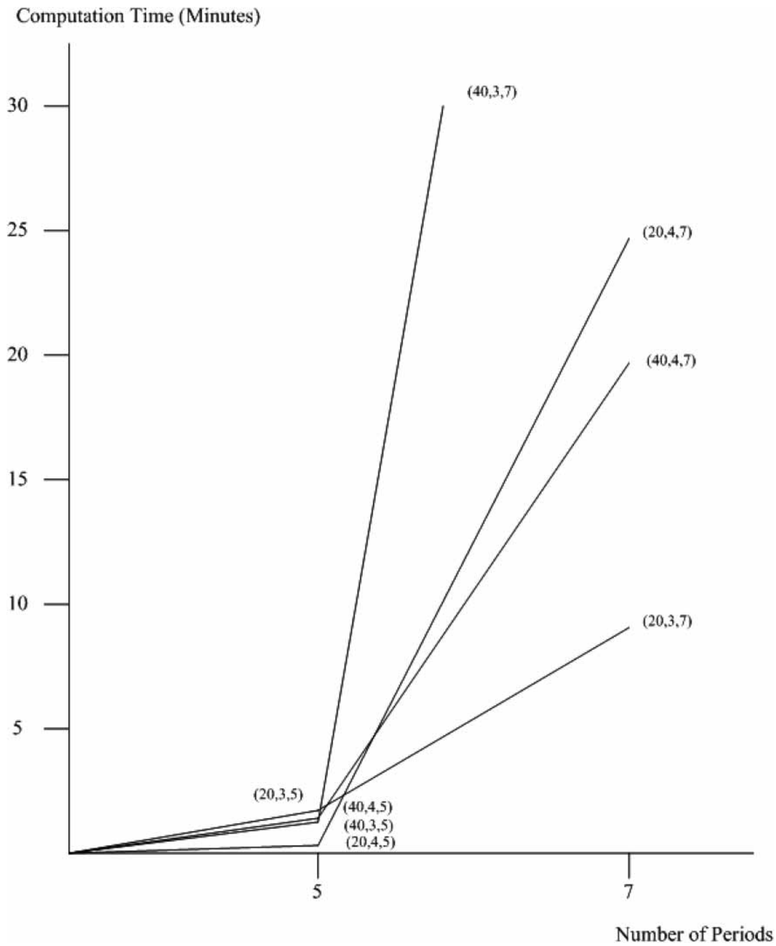

Figure 6. Computation time vs. number of periods.

5. Summary and suggestions

The multi-period optimization model of a PMS was formulated as a multi-dimensional 0–1 knapsack model with alternative-selection and precedence-feasibility constraints. The purpose of the model was to maximize the total effectiveness of pavement improvement activities in an entire set of selected strategies over a planning period horizon. Dynamic programming and branch-and-bound approaches were combined to produce a hybrid algorithm for solving the problem.

The imbedded-state approach was used to reduce multi-dimensional dynamic programming to one-dimensional dynamic programming and to obtain all promising solution points in a stage-wise fashion. The non-promising solution points that cannot lead to an optimal solution are eliminated by three schemes—feasibility tests, dominance tests and bounding tests. The feasibility test eliminates the solution space leading to an infeasible point. The dominance test is conducted to screen those solution points that consume more resources and provide lesser returns. The bounding test eliminates solutions in the state-space that result in a return worse than the best known bound.

In order to obtain initial bounds and bounds at each stage in the dynamic programming, the original problem was transformed to a resource-constrained longest-path network model. A Lagrangian optimization methodology for solving the RCLPP was developed. For bounding tests at each stage in dynamic programming, the Lagrangian optimization methodology was applied to each of the remaining problems. In Lagrangian optimization, the values of the Lagrangian multipliers are found by a sub-gradient optimization method, while the Ford–Bellman network algorithm was employed at each iteration of the sub-gradient optimization procedure to solve the longest-path network problem as well as to obtain improved lower and upper bounds.

Future work in this area could include the development of a heuristic to obtain tighter bounds for use in branch-and-bound procedures. Also, more constraints may be added to the model. For instance, other resources such as man-hours, materials, etc., could be added to explain the scope of the model.

References

Bazaraa, M.S. and Goode, J.J., 1979. A survey of various tactics for generating Lagrangian multipliers in the context of Lagrangian duality. *European journal of operational research*, 3, 322–338.

Beasley, J.E. and Christofides, N., 1989. An algorithm for the resource constrained shortest path problem. *Networks*, 19, 379–394.

Butt, A.A., Shahin, M.Y., Carpenter, S.H., and Carnahan, J.V., 1987. Pavement performance prediction model using the Markov process. *Transportation research record*, 1123, 12–19.

Butt, A.A., Shahin, M.Y., Carpenter, S.H. and Carnahan, J.V., 1994. Application of Markov process to pavement management systems at network level. In: *Third international conference on managing pavements*, vol. 2, 159–172.

Feighan, K.J., Shahin, M.Y. and Sinha, K.C., 1987. A dynamic programming approach to optimization for pavement management systems. In: *Proceedings of second North American conference on managing pavements*, vol. 2, 196–206.

Feighan, K.J., Shahin, M.Y., Sinha, K.C. and White, T.D., 1988. Application of dynamic programming and other mathematical technique to pavement management systems. *Transportation research record*, 1200, 90–98.

Fwa, T.F., Chan, W.T. and Hoque, K.Z., 2000. Multiobjective optimization for pavement maintenance programming. *Journal of transportation engineering*, 126, 367–374.

Geoffrion, A.M., 1974. Lagrangian relaxation for integer programming. *Mathematical programming study*, 2, 82–114.

Golabi, K., Kulkarni, R.B. and Way, G.B., 1982. A statewide pavement management system. *Interfaces*, 12, 5–21.

Grivas, D.A., Ravirala, V. and Schultz, B.C., 1993. State increment optimization methodology for network-level pavement management. *Transportation research record*, 1397, 25–33.

Li, N., Haas, R. and Huot, M., 1998. Integer programming of maintenance and rehabilitation treatment for pavement networks. *Transportation research record*, 1629, 242–248.

Marsten, R.E. and Morin, T.L., 1978. A hybrid approach to discrete mathematical programming. *Mathematical programming*, 14, 21–40.

Mbwana, J. and Turnquist, M.A., 1996. Optimization modeling for enhanced network-level pavement management system. *Transportation research record*, 1524, 76–85.

Morin, T.L. and Marsten, R.E., 1976. An algorithm for nonlinear knapsack problems. *Management science*, 22, 1147–1158.