SOLVING LARGE REPLACEMENT PROBLEMS WITH BUDGET CONSTRAINTS

Nejat Karabakal
Faculty of Busmess Administration
Bilkent University
06533 Ankara, Turkey

James C. Bean
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117

and
Jack R. Lohmann
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Technical Report 94-2

January 1994
Revised January 1996

Solving Large Replacement Problems With Budget Constraints

Nejat Karabakal and James C. Bean
The University of Michigan

Jack R. Lohmann
Georgia Institute of Technology

November 1995

Abstract

Traditional replacement models assume unlimited capital. In practice, however, firms frequently use budgets to control their expenditures. Budget constraints necessitate that all replacement decisions be considered as a portfolio, creating a difficult combinatorial problem. In previous research, a branch-and-bound algorithm was developed for solving moderately-sized problems optimally. In this paper, we propose a dual heuristic for dealing with large, realistically-sized problems. First, the individual replacement problems are solved ignoring the budget constraints. Then, we reduce (or eliminate, if possible) budget violations by solving a Lagrangian dual problem. The computational tests suggest that the effectiveness of the approach increases with problem size.

Introduction

Traditional replacement models assume unlimited capital. In practice, however, firms frequently use budgets to control their expenditures. In this paper, we present a practical approach for studying the effects of budget constraints on replacement decisions.

Suppose we need to keep n parallel assets to provide a required service over H periods. For each asset, the decision maker (DM) is faced with a current choice between keeping an existing asset (defender) and replacing it immediately with one of several new assets (current challengers). The decision must take into account the chain of future challengers since anticipated future challengers directly affect the economy of the defender and current challengers. All asset replacements are subject to a common budget in each period.

There is a rich literature for single asset replacement. Early replacement models assume any current challenger would be replaced repeatedly with identical assets into an indefinite future. Later attempts to relax the repeatability assumption include 1) extensions to the early infinite horizon models (Terborgh [23]; Oakford [21]; Bean, Lohmann, and Smith [2]), and 2) finite-horizon dynamic programming (DP) models (Wagner [24]; Oakford, Lohmann, and Salazar [22]).

Introducing budget constraints links each single asset replacement problem. The economic interdependence of assets caused by rationing capital makes it necessary to consider all replacement decisions in each time period as a portfolio, creating a difficult combinatorial problem.

In Karabakal, Lohmann, and Bean [18], we added budget constraints to Oakford-Lohmann-Salazar's dynamic replacement economy decision model and formulated a zero-one integer program (IP). The proposed branch-and-bound algorithm involves relaxing single asset replacement constraints and solves moderately-sized problems optimally. As a continuation of this line of research, we propose a heuristic in this paper for dealing with large, realistically-sized problems. We start by solving individual replacement problems while ignoring budget constraints entirely. The heuristic then reduces (or eliminates, if possible) budget violations by solving a Lagrangian dual problem. Both approaches employ new multiplier adjustment methods (MAMs) for solving the Lagrangian duals. The concept of total floats from critical path methods is used to substantially enhance the performance. This approach was suggested in [19]. In this paper, we develop a new MAM, based on total floats, for relaxation of the budget constraints. The computational tests suggest that the effectiveness of the approach increases with problem size.

We first review the integer programming formulation in [18]. Next, we present a Lagrangian relaxation followed by a heuristic multiplier adjustment method for solving the Lagrangian dual. Results of computational tests on random problems are also reported.

Formulation

We assume that all replacements and cash transactions occur at the end of discrete time periods and that the end of period zero refers to the current time.

The decision variable $x(a, c, i, j)$ is set to one if challenger c of asset a is installed from period i to period j, zero otherwise. Here,

$$
\begin{aligned}
a &\in \{1,\ldots,n\} \\
c &\in \{0,\ldots,m(a)\} \\
i &\in \{0,\ldots,H-1\} \\
j &\in \{i+1,\ldots,\min\{H,\, i+N(a,c)\}\}
\end{aligned}
$$

where $m(a)$ is the number of challengers considered for asset a, and $N(a,c)$ is the maximum service life of challenger c of asset a. The notation $c = 0$ refers to the defender, a special challenger that cannot be repeated and is only available in period zero. For ease of exposition, for each asset a we define a dummy decision variable $x(a,c',H,0) = 1$, where the value of c' is arbitrary.

Let $v(a,c,i,j)$ be the marginal net present value (NPV) benefit of the replacement decision corresponding to $x(a,c,i,j) = 1$. A convenient way to compute these NPV benefits is as follows. For each challenger c of asset a acquired in period i, define

$$
\begin{aligned}
P(a,c,i) &= \begin{cases} \text{installation (purchase) cost,} & \text{if } c > 0 \\ \text{current salvage value,} & \text{if } c = 0 \text{ and } i = 0 \end{cases} \\
r(a,c,i,t) &= \text{revenue generated in relative time period } t,\ t = 1,\ldots,j-i, \\
e(a,c,i,t) &= \text{expense in relative time period } t, \\
S(a,c,i,k) &= \text{salvage value after } k \text{ periods of usage, } k = 1,\ldots,N(a,c).
\end{aligned}
$$

Letting α be the interest rate per period, and $y = 1/(1+\alpha)$,

$$
v(a,c,i,j) = \left(-P(a,c,i) + \sum_{t=1}^{j-i} \left[r(a,c,i,t) - e(a,c,i,t) \right] y^t + S(a,c,i,j-i)\, y^{(j-i)} \right) y^i.
$$

The above computations suppose before-tax cash flows to avoid computational complexity. Unless tax consequences are a function of the replacement decisions, the role of taxes can be incorporated by calculating after-tax cash flows and adjusting other parameters accordingly. Also, the NPVs for future challengers acquired at $i > 0$ are estimated from the current challenger's NPVs using functional relationships to account for the effects of technological improvements and inflation [22].

To keep notation simple, we suppose $v(a,c,i,j) = -\infty$ whenever $c = 0$ and $i > 0$. Furthermore, we define two index sets, $J(a,c,i)$ and $I(a,c,i)$, as the time periods such that, for a given (a,c) pair, replacements are possible to and from period i, respectively, without violating maximum service life restrictions.

The following is a classical IP formulation assuming hard budget constraints. The objective is to maximize the overall NPV of cash flows resulting from replacement decisions of n assets over H periods.

$$\text{maximize} \quad \sum_{a=1}^{n}\sum_{c=0}^{m(a)}\sum_{i=0}^{H-1}\sum_{j\in J(a,c,i)} v(a,c,i,j)\,x(a,c,i,j)$$

subject to the following constraints:

1. Serial replacement: Each asset may be replaced each time period by exactly one asset. So, for $a = 1,\ldots,n$ and $i = 0,\ldots,H$:

$$\sum_{c=0}^{m(a)}\sum_{j\in J(a,c,i)} x(a,c,i,j) = \sum_{c=0}^{m(a)}\sum_{j\in I(a,c,i)} x(a,c,j,i).$$

2. Budget: Expenditures should be within budgets. So, for $i = 0,\ldots,H-1$

$$\sum_{a=1}^{n}\sum_{c=1}^{m(a)} P(a,c,i) \sum_{j\in J(a,c,i)} x(a,c,i,j) \le B(i)$$

where $B(i)$ is the budget available in period i.

3. Integrality: $x(a,c,i,j) \in \{0,1\}$.

The above formulation has a network characterization which is exploited for the rest of the paper. Consider a directed graph where nodes represent the end of time periods and arcs represent replacement decisions. Associated with each arc are two parameters, length (NPV benefit of the replacement) and resource consumption (installation cost). We have as many such graphs as the number of assets. An example graph is shown in Figure 1 for a certain asset a with one challenger and a defender. The challenger's maximum life is three periods. The defender may be used for at most two more periods. We seek the longest path from node 0 to node H in each asset's graph such that no resource (budget) constraint is violated.

The problem of finding longest paths under resource constraints is NP-complete (Handler and Zang [14]). Approaches to resource constrained path finding problems include Lagrangian relaxation in a path ranking approach [11, 14], network reduction techniques followed by implicit enumeration [1, 15], and branch and bound with Lagrangian relaxation bounds [4]. Most applications reported in the literature deal with general constraints, for example, traversal time limit, energy consumption limit, etc., along the path

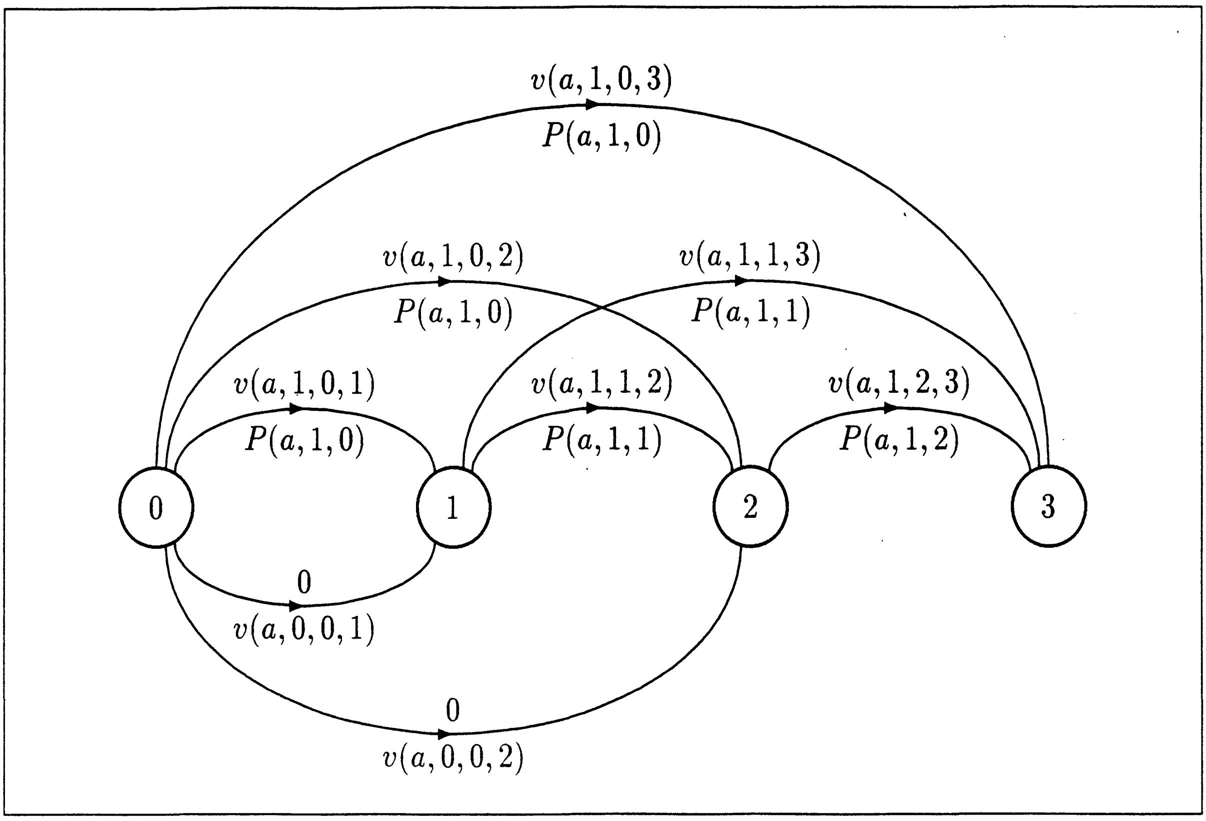

Figure 1: A sample directed graph for a particular asset with one challenger and a defender.

in a single network. In contrast, the replacement problem we describe in this paper has resource constraints across individual networks, rendering a unique structure. Next, we discuss how this structure is exploited.

Lagrangian Relaxation

We propose a method based on Lagrangian relaxation. It relaxes constraints into the objective function producing a Lagrangian problem that yields an upper bound (for maximization problems) on the optimal value of the original problem (Fisher [8]; Geoffrion [10]).

Two alternative Lagrangian relaxations can be considered for the previous IP. If we relax the budget constraints (REL-1), the relaxed problem is solved by longest path algorithms. However, the relaxation has the integrality property so that the best upper bound is the linear programming (LP) relaxation bound. Alternatively, if we relax the serial replacement constraints (REL-2), the relaxed problem is solved by knapsack algorithms. REL-2 is likely to produce tighter upper bounds since it lacks the integrality property.

In [18], an exact solution procedure is developed by using REL-2 as a bounding tool in a branch-and-bound algorithm. Although the exact procedure is capable of solving moderately-sized problems (up to ten parallel assets over eight periods) in reasonable times, it is computationally too expensive for larger problems. In this paper, we present an approximate solution method using REL-1 for the following reasons:

1. The relaxed problem is solved finding longest paths on acyclic graphs, which are much easier to solve than the knapsack problems from REL-2.
2. Any Lagrangian solution x gives a valid replacement schedule that may only violate the budget constraints. Some of the violations could be insignificant and an acceptable solution can be obtained. In contrast, no valid replacement schedule is possible in the second relaxation until an optimum solution is reached.

We consider a solution that satisfies the replacement constraints but violates budget constraints more useful than a solution that violates the replacement constraints but satisfies the budget constraints. The justification lies behind the logic of imposing budget constraints. In practice, budgets do not represent hard limits on finance, rather they are provisional limitations for the purposes of expenditure control (Weingartner [25, 26, 27]). The survey results summarized by Gurnani [13] about the reasons of capital rationing support this viewpoint. Hence, the purpose of our proposed method is not bounding the IP optimum per se, but producing an approximate solution with acceptable budget violations. In addition, estimating the losses which result from imposing tight budgets in certain periods is valuable in budget planning. The extent of budget violations and loss of optimality are discussed in computational experiments section.

Let $\lambda(i) \geq 0$ be the multiplier associated with the budget constraint of period i. Dualizing budget constraints using λ, we get

$$L(\lambda) = \max_{x} \sum_{a} \sum_{c} \sum_{i} \sum_{j} \left[v(a,c,i,j) - P(a,c,k)\,\lambda(i) \right] x(a,c,i,j) + \sum_{i} B(i)\,\lambda(i) \qquad (1)$$

where the maximization is over all x satisfying serial replacement and integrality constraints. Any solution, x, obtained in evaluating $L(\lambda)$ for a $\lambda \geq 0$ is referred to as a Lagrangian solution.

Relaxing the budget constraints removes the interdependency of replacement decisions across assets. Hence, the problem can now be decomposed into n independent serial replacement problems, each of which is solved by finding a longest path on an acyclic graph.

We use a dynamic program (DP) to solve each longest path problem efficiently. For a given $\lambda \geq 0$, define

$$\overline{v}(a,c,i,j) = v(a,c,i,j) - P(a,c,i)\,\lambda(i)$$

for all a, c, i, j. Let $f(a,j)$ be the longest path from node 0 to node j in asset a's graph. For every a, initialize $f(a,0) = 0$ and solve the following forward recursive equations:

$$f(a,j) = \max_{c,i} \{\overline{v}(a,c,i,j) + f(a,i)\} \qquad \text{for } j = 1,\ldots,H \tag{2}$$

where the maximization is over all c and $i = j-1,\ldots,\max\{0,\, j - N(a,c)\}$.

The Lagrangian value is then computed from

$$L(\lambda) = \sum_i B(i)\,\lambda(i) + \sum_a f(a,H).$$

For future reference, let us define $b(a,i)$ as the longest path from node H to node i. After setting $b(a,H) = 0$ and $b(a,0) = f(a,H)$, we can use the following backward recursion to compute the rest of the $b(a,i)$ quantities:

$$b(a,i) = \max_{c,j} \{\overline{v}(a,c,i,j) + b(a,j)\} \qquad \text{for } i = H-1,\ldots,1 \tag{3}$$

where the maximization is over all c and $j = i+1,\ldots,\min\{H,\, i + N(a,c)\}$.

The associated Lagrangian dual problem ([5, 10, 8]) is formulated as

$$\min_{\lambda \geq 0} L(\lambda). \tag{4}$$

Given the discussion on acceptable violations of the budget constraints, this Lagrangian dual problem could be interpreted as the appropriate formulation. The multiplier $\lambda(i)$ can be interpreted as the penalty of violating the budget constraint of period i. If the total replacement expenditure in period i exceeds the budget, its penalty $\lambda(i)$ should be increased to discourage some replacements.

Next, we present a multiplier adjustment method (MAM) for solving (4).

A Multiplier Adjustment Method

MAMs are heuristic algorithms for solving Lagrangian dual problems exploiting the special structure of a particular application. Previous successful integer programming applications include the uncapacitated facility location problem (Erlenkotter [7]), generalized assignment problem (Fisher, Jaikumar, and Van Wassenhove [9]; Guignard and Rosenwein [12]; Karabakal, Bean, and Lohmann [17]), and set partitioning problem (Chan and Yano [6]). A MAM usually guarantees monotonic bound improvement, however, it does not guarantee bounds better than those obtained by more common subgradient algorithms [8].

The following development considers a directed graph of asset a and assumes that the longest path $f(a, H)$ is associated with the Lagrangian solution x.

DEFINITION 1 *Given a Lagrangian solution x, expenditure in period i is defined as*

$$E(x,i) = \sum_a \sum_c \sum_j P(a,c,i)\, x(a,c,i,j)$$

for $i = 0, \ldots, H-1$.

When $E(x,i) > B(i)$, the central idea in our MAM is to increase $\lambda(i)$ until one of the replacements in period i is canceled. The exact amount of this increase is determined using Theorem 1 which, in turn, uses the following sensitivity results.

LEMMA 1 *If $x(a,c,i,j) = 0$ in an optimal solution to (1), the minimum δ such that $\overline{v}(a,c,i,j)+\delta$ would cause $x(a,c,i,j) = 1$ in an alternative optimum is given by $s(a,c,i,j)$, where*

$$s(a,c,i,j) = f(a,H) - b(a,j) - f(a,i) - \overline{v}(a,c,i,j).$$

PROOF. Let $\tilde{f}(a,H)$ be the longest path from node 0 to H constrained to include the arc from i to j using challenger c. Then

$$\tilde{f}(a,H) = f(a,i) + \overline{v}(a,c,i,j) + b(a,j).$$

In order to set $x(a,c,i,j) = 1$ in an alternative optimum, $\tilde{f}(a,H)$ should be at least as large as $f(a,H)$. To enforce this selection we have to increase $\overline{v}(a,c,i,j)$ by at least $f(a,H) - \tilde{f}(a,H)$, where

$$\begin{aligned} f(a,H) - \tilde{f}(a,H) &= f(a,H) - [f(a,i) + \overline{v}(a,c,i,j) + b(a,j)] \\ &= s(a,c,i,j). \blacksquare \qquad (5) \end{aligned}$$

LEMMA 2 *If c is the only challenger of asset a that can possibly be installed in period i and $x(a,c,i,j) = 1$, then the minimum δ such that $\overline{v}(a,c,i,j) - \delta$ would cause no replacement in period i is given by δ^*:*

$$\delta^* = f(a,H) - f(a,H \mid \sim i)$$

where $f(a,H \mid \sim i)$ is the longest path from node 0 to H constrained to exclude any replacement in period i.

PROOF. Omitted. ■

THEOREM 1 *If asset a has a replacement at $i^* > 0$ so that $x(a,c,i^*,j) = 1$ for some c and j, then the minimum Δ such that $\lambda(i) + \Delta$ would cause $x(a,c,i^*,j) = 0$ in an optimum is given by*

$$\Delta(a) = \min\{\Delta_1(a),\, \Delta_2(a)\} \tag{6}$$

where

$$\Delta_1(a) = \min_{j', \{c' \mid P(a,c',i^*) < P(a,c,i^*)\}} \left\{ \frac{s(a,c',i^*,j')}{P(a,c,i^*) - P(a,c',i^*)} \right\}$$

and

$$\Delta_2(a) = \frac{f(a,H) - f(a,H \mid \sim i^*)}{P(a,c,i^*)}.$$

PROOF. The current longest path, 0-i^*-j-H in Figure 2, has length $f(a,H)$. For each Δ increase of $\lambda(i^*)$, the value of $\overline{v}(a,c,i^*,j)$ is decreased by $\Delta\, P(a,c,i^*)$; as is $f(a,H)$. When the reduction in $f(a,H)$ reaches a certain threshold, another path emerges as an alternative to the current longest path. The new path will either 1) pass through node i^*, or 2) skip node i^*.

1. If the value of $\overline{v}(a,c,i^*,j)$ is reduced sufficiently, a new path from 0 to H passing through node i^*, but installing another (cheaper) challenger c' from node i^* to any node j', becomes as long as the current one. In Figure 2, this new path is 0-i^*-j'-H. As in Lemma 1, let $\tilde{f}(a,H)$ be the length of the longest path constrained to include the arc from i^* to j' using challenger c'. To enforce the lengths of these two paths, 0-i^*-j-H and 0-i^*-j'-H, to be equal, $\lambda(i^*)$ should be increased by at least $\Delta \geq 0$, satisfying

$$\begin{aligned} f(a,H) - \Delta\, P(a,c,i^*) &= \tilde{f}(a,H) - \Delta\, P(a,c',i^*) && (7) \\ f(a,H) - \tilde{f}(a,H) &= \Delta\, [P(a,c,i^*) - P(a,c',i^*)] && (8) \end{aligned}$$

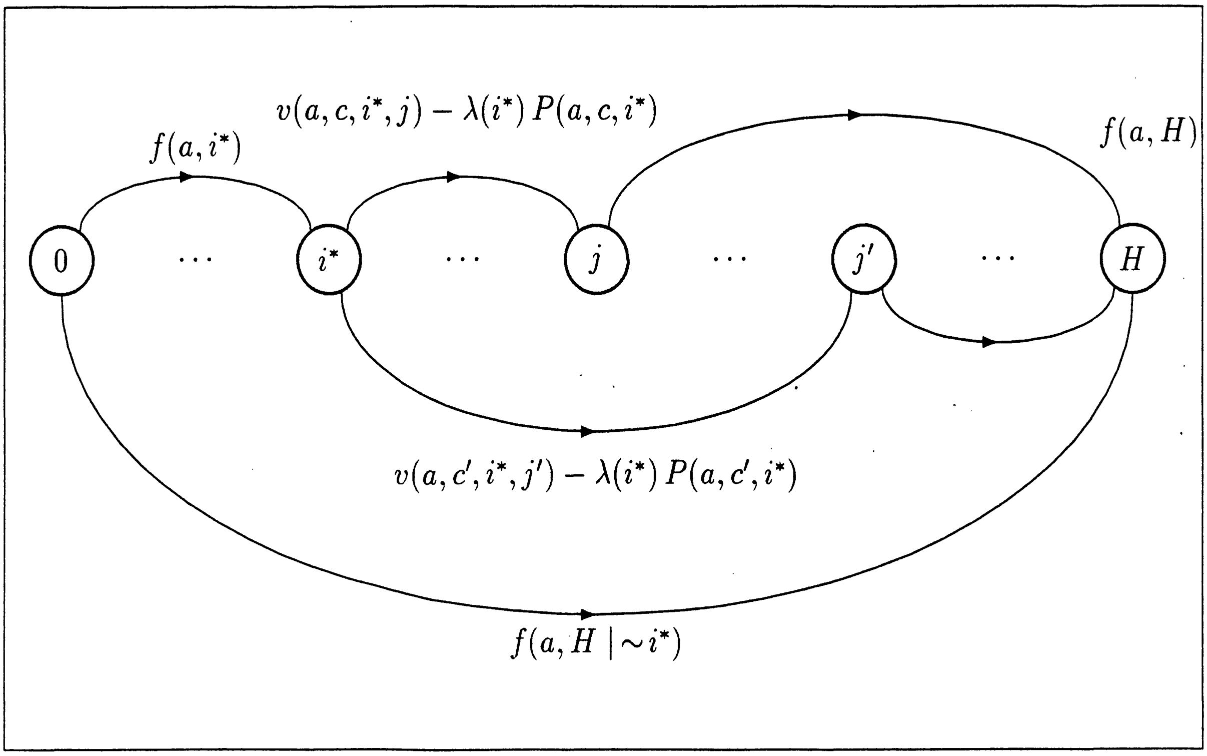
$v(a, c, i^*, j) - \lambda(i^*)\, P(a, c, i^*)$
$f(a, i^*)$
$f(a, H)$
0
i^*
j
j'
H
$v(a, c', i^*, j') - \lambda(i^*)\, P(a, c', i^*)$
$f(a, H \mid \sim i^*)$

Figure 2: Two possible ways of creating an alternative optimum by increasing $\lambda(i^*)$ in order to set $x(a, c, i^*, j) = 0$.

provided

$$P(a, c', i^*) < P(a, c, i^*). \tag{9}$$

If $P(a, c', i^*) \geq P(a, c, i^*)$, an alternative longest path passing through i^* and installing c' does not exist. An illustration of equation (7) is given in Figure 3. The relation (9) between the slopes must hold for the intersection of the two lines, hence for the existence of Δ. From (5), $f(a, H) - \tilde{f}(a, H) = s(a, c', i^*, j')$. Since there can be many arcs emanating from i^* and installing challenger $c' \neq c$, $\Delta_1(a)$ is the minimum Δ over appropriate challengers and their service lives satisfying (8). Note that if there is no challenger c' satisfying (9), $\Delta_1(a)$ is set to $+\infty$.

2. If the value of $\overline{v}(a, c, i^*, j)$ is reduced sufficiently, a new path from 0 to H not passing through node i^* becomes as long as the current one. This is the path 0–H in Figure 2. From Lemma 2, $\lambda(i^*)$ should be increased by at least $\Delta = \Delta_2(a)$, satisfying

$$f(a, H) - \Delta\, P(a, c, i^*) = f(a, H \mid \sim i^*).$$

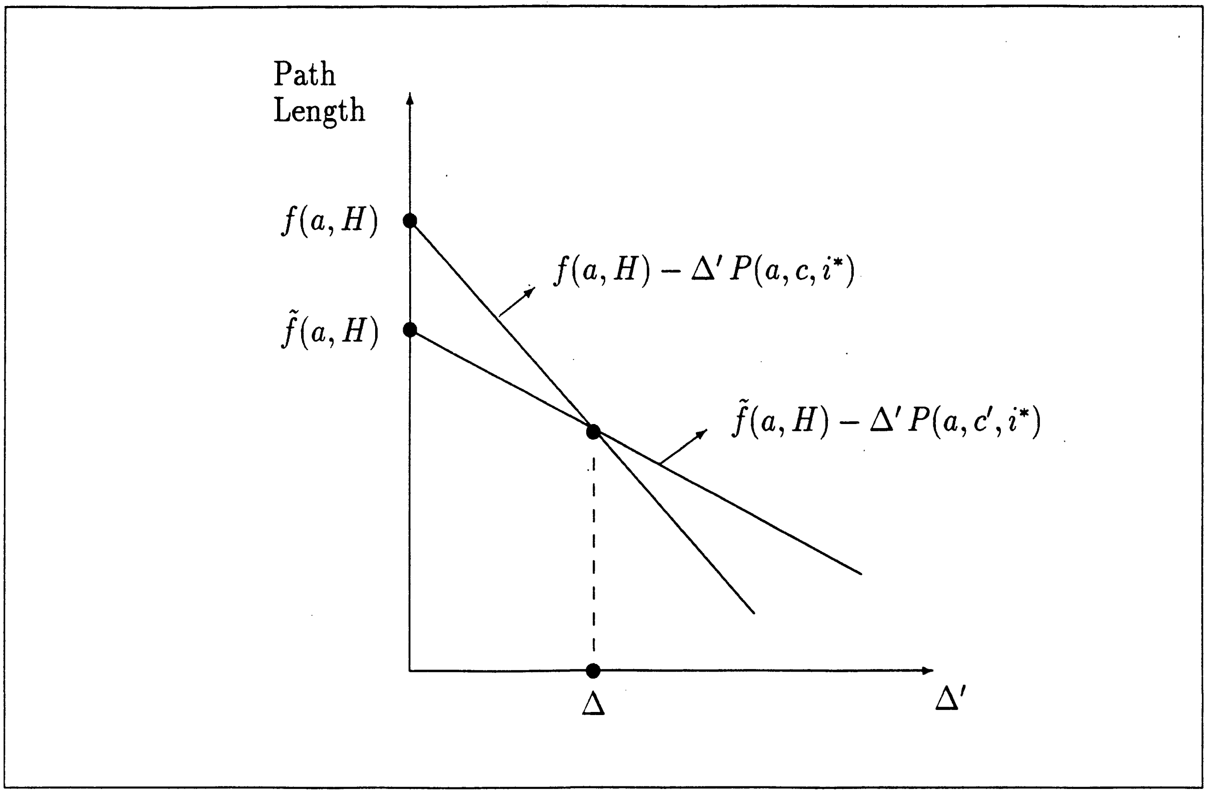
Path Length
$f(a,H)$
$\tilde{f}(a,H)$
$f(a,H) - \Delta' P(a,c,i^*)$
$\tilde{f}(a,H) - \Delta' P(a,c',i^*)$
Δ
Δ'

Figure 3: When $\lambda(i^*)$ is increased by Δ, there will be two alternative longest paths.

The value of $\overline{v}(a,c,i^*,j)$ is reduced until either case occurs. Hence, (6). ∎

COROLLARY 1 *If $i^* = 0$ so that $x(a,c,0,j) = 1$ for some c and j, then*

$$\Delta(a) = \begin{cases} \Delta_1(a), & \textit{if } c > 0 \\ +\infty, & \textit{if } c = 0 \end{cases} \tag{10}$$

PROOF. By problem definition, node 0 cannot be skipped. Thus, $\Delta_2(a) = +\infty$. If $c = 0$, $\Delta_1(a)$ is also $+\infty$ because the replacement cost of a defender is zero. ∎

The following MAM is a systematic way of adjusting the multipliers (penalties) of violated constraints.

The Multiplier Adjustment Algorithm

Step 0: Initialize $\lambda(i) = 0$ for all i (i.e., ignore all budget constraints). Use DP recursions (2) and (3) for solving n independent longest path problems and obtaining x^0, f^0, and b^0. Set the iteration counter $k \leftarrow 0$.

Step 1: Let $I = \{i \mid E(x^k, i) > B(i)\}$. If $I = \phi$, stop; current x^k is feasible.

Step 2: Select a period $i^* \in I$ for which the present value of the budget deficit is the largest, i.e.,

$$i^* = \operatorname*{argmax}_{i \in I} \left\{ \frac{E(x^k, i) - B(i)}{(1+\alpha)^i} \right\}.$$

Step 3: Let $A(i^*) = \{a \mid x^k(a, c, i^*, j) = 1 \text{ for some } c \text{ and } j\}$. Compute $\Delta(a)$ for each $a \in A(i^*)$ from either (6) or (10), depending on whether $i^* = 0$ or not. Let

$$a^* = \operatorname*{argmin}_{a \in A(i^*)} \{\Delta(a)\} \quad \text{and} \quad \Delta^* = \Delta(a^*).$$

If $\Delta^* = 0$, update $I \leftarrow I \backslash \{i^*\}$. If $I = \phi$, stop; no further improvement is possible. Otherwise, go to step 2.

If $\Delta^* > 0$, update multipliers:

$$\lambda^{k+1}(i^*) \leftarrow \lambda^k(i^*) + (\Delta^*)^+; \qquad \lambda^{k+1}(i) \leftarrow \lambda^k(i) \quad \text{for } i \neq i^*$$

where $(\Delta^*)^+$ is infinitesimally greater than Δ^* to ensure that asset a^* would not have the same replacement in period i^* again.

Solve n independent longest path problems with new multipliers to obtain x^{k+1}, f^{k+1}, and b^{k+1}. Set $k \leftarrow k+1$, go to step 1.

The monotonicity of the MAM is established by the following theorem.

THEOREM 2 *Let i^* be a period in which the budget constraint is violated at the k-th iteration of the MAM. That is,*

$$E(x^k, i^*) > B(i^*).$$

Increasing $\lambda^k(i^*)$ by $\Delta^* > 0$ found in step 3 reduces $L(\lambda^k)$ by $\Delta^* [E(x^k, i^*) - B(i^*)] > 0$.

PROOF. In the k-th iteration

$$L(\lambda^k) = \sum_i B(i)\, \lambda^k(i) + \sum_a f^k(a, H).$$

Increasing $\lambda^k(i^*)$ by Δ^* reduces the longest path of each asset $a \in A(i^*)$ by $\Delta^* P(a, c, i^*)$, where c denotes the challenger of asset a installed at i^* by x^k. Hence,

$$L(\lambda^{k+1}) \;=\; \sum_i B(i)\, \lambda^k(i) + \Delta^* B(i^*) + \sum_a f^k(a, H)$$

$$
\begin{aligned}
& - \Delta^* \sum_{a \in A(i^*)} P(a, c, i^*) \\
= \; & L(\lambda^k) - \Delta^* \left(\sum_{a \in A(i^*)} P(a, c, i^*) - B(i^*) \right) \\
= \; & L(\lambda^k) - \Delta^* \left[E(x^k, i^*) - B(i^*) \right]. \; \blacksquare
\end{aligned}
$$

The MAM reduces the Lagrangian value by attempting to decrease budget violations in certain periods. Therefore, the monotonic improvement of the Lagrangian value should lower the budget violations. Computational tests verify this effect.

DEFINITION 2 *Given a Lagrangian solution, x, the present value of budget deficits is defined as*

$$D(x) = \sum_{i=0}^{H-1} \left(\frac{\max\{0,\ E(x,i) - B(i)\}}{(1+\alpha)^i} \right).$$

Unfortunately, there is no guarantee that $D(x)$ is reduced monotonically. We performed the following computational experiments to evaluate the overall reduction in $D(x)$ by the MAM.

COMPUTATIONAL EXPERIMENTS

The MAM was coded in C and run with random test problems. The following discrete uniform distributions were used to generate challenger parameters for each asset a. Time horizon was fixed at $H = 20$.

Number of challengers	$U(1, 4)$
Defender life	$U(1, 3)$
Challenger life	$U(2, 5)$
Challenger's purchase cost	$100 \times a + U(100, 999)$
Defender's salvage value	$100 \times a + U(10, 99)$
Revenue	80% of purchase cost each period
Expense	10% of purchase cost in the first period
	$U(10\%, 30\%)$ increase each period
Salvage value	$U(5\%, 50\%)$ decrease each period.

Distributions are obtained after scaling and simplifying the data collected for a vehicle fleet replacement study in Michigan [3]. The study involved a major utility company with a fleet of over one thousand vehicles that are different in price, size, and service objective. Even after various strategies of clustering to decompose the problem, they must solve a problem that includes hundreds of vehicles.

Note that a challenger's cash flows are correlated with the purchase cost. Although such correlations increase the computational difficulty, they are usually observed in real applications. See [9, 17, 20] for discussions of this issue.

We conducted three computational experiments to determine the effectiveness of the MAM. First two experiments fixed the budget level and varied the problem size, first using moderately large problems (up to 30 assets), second very large problems (up to 500 assets). We set the budgets to $(n/2)$ × the average challenger purchase cost in each period. The third experiment set the problem size at 100 assets and varied the budget level. All experiments were done on an IBM RS/6000–320H.

In the first experiment, and compared the best solution the MAM produced with the optimal solution. Let

$x^* =$ optimal solution of the IP,

$x' =$ Lagrangian solution with the smallest present value of budget deficits, or $D(x') = \min_k\{D(x^k)\}$, where k is the iteration counter of the MAM,

$V(x) =$ IP objective function value associated with solution x.

Clearly, x' may have plus or minus objective value error. To determine the average magnitude of minus errors (loss of optimality), we solved problems with $n = 5, 10, 15, 20, 25$ and 30 assets using both our MAM code and IBM's Optimization Subroutine Library (OSL) [16] for optimal solutions. The results are summarized in Table 1. The average loss of optimality, the maximum of zero and the deviation from the optimal objective value, is usually insignificant. Moreover, the average $D(x')$ statistics verify our conjecture that the MAM is more effective in reducing budget violations as the problem size increases. The rate of reduction in average budget violation is exponential as the number of assets increases linearly from 5 to 30. Naturally, large problems offer more freedom across assets for the MAM to fix budget violations. Given the fact that, in many real applications, the asset data and budgets are not known to a high precision, the extra computer time to locate the true optimum may be difficult to justify.

Table 1: Comparison of the MAM solutions with optimal solutions. Each line shows statistics for 10 problems.

	Average loss	Average	Max	MAM CPU time		OSL CPU time	
	of optimality*	$D(x')$	$D(x')$	Median	Max	Median	Max
n	(%)	(%)	(%)	(secs)	(secs)	(secs)	(secs)
5	3.34	6.69	13.42	0.08	0.17	30.54	4579.62
10	0.51	3.06	13.95	0.24	0.28	45.07	103.07
15	0.20	1.21	2.93	0.48	0.64	52.33	490.43
20	0.12	1.07	2.46	0.74	1.09	163.81	1069.12
25	0.15	1.06	2.11	0.87	1.22	372.33	4324.91
30	0.08	0.44	1.23	1.33	2.20	257.57	6545.9

* $\frac{\max\{0, V(x^*)-V(x')\}}{V(x^*)}$

Table 2: Results for larger problems. Each line shows statistics for 10 problems.

	Average	Maximum	MAM CPU time	
	$D(x')$	$D(x')$	Median	Max
n	(%)	(%)	(secs)	(secs)
100	0.03	0.13	13.12	19.28
200	0.08	0.55	37.71	67.42
300	0.16	0.68	73.34	188.78
400	0.11	0.37	128.97	1310.41
500	0.04	0.23	178.40	737.24

In the second experiment, we generated very large problems that are either impossible or computationally too expensive to solve optimally. Table 2 shows the performance of the MAM for problems with 100 to 500 assets. The average budget violations are usually less than 0.1% and, for all practical purposes, can be considered negligible.

In the third experiment, we fixed the number of assets at 100 and focused on the performance with respect to budget tightness. We solved problems with budget levels ranging from 30% more to 30% less than those levels in the first two experiments, which were derived from a real case. The results are shown in Table 3. As budgets are made looser, it becomes more likely to obtain alternative feasible and improving paths, and problems become easier to solve. Similarly, as budgets are made tighter, it becomes less

Table 3: Performance of the MAM with varying budget levels. Number of assets is fixed at 100. Each line shows statistics for 10 problems.

	Average	Maximum	MAM CPU time	
	$D(x')$	$D(x')$	Median	Max
Budget	(%)	(%)	(secs)	(secs)
30% more	0.00	0.03	0.44	4.87
20% more	0.01	0.06	2.15	11.16
10% more	0.03	0.19	8.35	18.48
Original	0.03	0.13	13.12	19.28
10% less	0.37	2.03	25.84	1182.29
20% less	0.71	2.05	35.30	327.48
30% less	3.27	4.94	62.50	1247.19

likely to find alternative feasible and improving paths, and problems become more difficult to solve. Thus, final budget deficits are lower in the former case, higher in the latter case. Moreover, the performance is more sensitive to tightening budgets than loosening budgets.

Conclusions

We describe a dual heuristic for solving realistically-sized replacement problems under budget constraints. Our approach is to reduce the budget violations of the unconstrained solution progressively so that a feasible or acceptable solution is obtained. We use an IP described in [18] as a starting point and develop a MAM to solve a Lagrangian dual. Computational experiments show that budget violations of the selected MAM solutions average around 1% for 15 to 30 asset problems and around 0.1% for 100 to 500 asset problems. Moreover, the average loss of optimality for problems with more than 10 assets is insignificant (less than 0.5%). The performance of the heuristic is somewhat sensitive to the budget levels, but not overly so. Therefore, the MAM appears to be a viable tool in solving large problems in practice.

Acknowledgments

This research was supported in part by National Science Foundation grant DDM-9202849 to The University of Michigan. We thank James Pappas, III for running some of the test problems.

References

[1] Aneja, Y. P., Aggarwal V., and Nair P. K., "Shortest Chain Subject to Side Constraints," *Networks*, **13**, 295-302 (1983).

[2] Bean, J. C., Lohmann J. R., and Smith R. L., "A Dynamic Infinite Horizon Replacement Economy Decision Model," *The Engineering Economist*, **30**, 99-120 (1985).

[3] Bean, J. C., Lohmann J. R., and Smith R. L., "Equipment replacement Under Technological Change," *Naval Research Logistics*, **41**, 117-128 (1994).

[4] Beasley, J. E. and Christofides N., "An Algorithm for the Resource Constrained Shortest Path Problem," *Networks*, **19**, 379-394 (1989).

[5] Bellmore, M., Greenberg H. J., and Jarvis J. J., "Generalized Penalty Function Concepts in Mathematical Optimization," *Operations Research*, **18**, 229-252 (1970).

[6] Chan, T. J. and Yano C. A., "A Multiplier Adjustment Approach for the Set Partitioning Problem," *Operations Research*, **40**, S40-S47 (1992).

[7] Erlenkotter, D., "A Dual-Based Procedure for Uncapacitated Facility Location," *Operations Research*, **26**, 992-1009 (1978).

[8] Fisher, M. L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Science*, **27**, 1-18 (1981).

[9] Fisher, M. L., Jaikumar R., and Van Wassenhove L. N., "A Multiplier Adjustment Method for the Generalized Assignment Problem," *Management Science*, **32**, 1095-1103 (1986).

[10] Geoffrion, A. M., "Lagrangian Relaxation for Integer Programming," *Mathematical Programming Study*, **2**, 82-114 (1974).

[11] Gopalan, R., Batta R., and Karwan M. H., "The Equity Constrained Shortest Path Problem," *Computers and Operations Research*, **17**, 297-307 (1990).

[12] Guignard, M. and Rosenwein M. B., "An Improved Dual Based Algorithm for the Generalized Assignment Problem," *Operations Research*, **37**, 658-663 (1989).

[13] Gurnani, C., "Capital Budgeting: Theory and Practice," *The Engineering Economist*, **30**, 19-46 (1984).

[14] Handler, G. Y. and Zang I., "A Dual Algorithm for the Constrained Shortest Path Problem," *Networks*, **10**, 293-310 (1980).

[15] Hassan, M. M. D., "Network Reduction for the Acyclic Constrained Shortest Path Problem," *European Journal of Operational Research*, **63**, 124-132 (1992).

[16] International Business Machines, *Optimization Subroutine Library, Guide and Reference*, International Business Machines, Mechanicsburg, PA, 1990.

[17] Karabakal, N., J. C. Bean, and J. R. Lohmann, "A Steepest Descent Multiplier Adjustment Method for the Generalized Assignment Problem," Technical Report 92-11, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1992.

[18] Karabakal, N., J. R. Lohmann and J. C. Bean, "Parallel Replacement Under Capital Rationing," *Management Science*, **40**, 305-319 (1994).

[19] Lee, I.-S., "On Sensitivity Analysis for Shortest Path Dynamic Programming," Working Paper, University of California, Los Angeles, (1986).

[20] Martello, S. and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley and Sons, 1990.

[21] Oakford, R. V., *Capital Budgeting: A Quantitative Evaluation of Investment Alternatives*, The Ronald Press Co., New York, 1970.

[22] Oakford, R. V., Lohmann J. R., and Salazar A., "A Dynamic Replacement Economy Decision Model," *IIE Transactions*, **16**, 65-72 (1984).

[23] Terborgh, G., *Dynamic Equipment Policy*, Machinery and Allied Products Institute, Washington D.C, 1949.

[24] Wagner, H. M., *Principles of Operations Research*, Prentice Hall Inc., 1975.

[25] Weingartner, H. M., *Mathematical Programming and the Analysis of Capital Budgeting Problems*, Prentice Hall, 1963.

[26] Weingartner, H. M., "Capital Budgeting of Interrelated Projects: Survey and Synthesis," *Management Science*, **12**, 485-516 (1966).

[27] Weingartner, H. M., "Capital Rationing: n Authors in Search of a Plot," *The Journal of Finance*, **32**, 1403-1431 (1977).