

**A MEAN-VARIANCE SERIAL REPLACEMENT
DECISION MODEL: THE INDEPENDENT CASE**

Matt Brown

Department of
Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 91-38

December 1991

**A Mean-Variance Serial Replacement Decision Model:
The Independent Case**

Matt Brown
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

December 1991

A Mean-Variance Serial Replacement Decision Model:

The Independent Case

Statement of Contribution

We develop a utility-based replacement model that explicitly considers risk. Previous replacement models have either assumed deterministic cash flows or a risk-neutral decision maker. Our contributions include:

1. A dynamic programming procedure that finds all Pareto-optimal solutions.
2. A heuristic that is based, in part, on our observations that the Pareto-optimal solutions tend to cluster.
3. Computational results that indicate we can solve optimally a wide range of problems. For problems that can not be solved optimally, the average performance of the heuristic is within one percent of an upper bound.

The research implications are that risk-averse decision makers now have a flexible, yet tractable utility-based approach which can be used to find optimal or near-optimal solutions for serial replacement problems.

A Mean-Variance Serial Replacement Decision Model:

The Independent Case

Abstract

This paper considers a serial replacement problem for a risk-averse decision maker whose objective is to maximize expected utility, but whose utility function is not necessarily known in advance. We assume the reward received from installing and operating an asset over a period of time is normally distributed, and develop a dynamic programming procedure that finds all Pareto-optimal solutions. This allows the decision maker to select the preferred alternative without specifying a utility function. We also develop a heuristic based on an observed tendency of the Pareto-optimal solutions to cluster. Finally, we report computational results that indicate we can solve optimally a wide range of problems, and that for problems that cannot be solved optimally, the average performance of the heuristic is within one percent of an upper bound.

KEY WORDS: MACHINE INVESTMENT; STOCHASTIC NETWORKS; LONGEST PATH PROBLEMS; DECISION MAKING UNDER UNCERTAINTY.

A Mean-Variance Serial Replacement Decision Model:

The Independent Case

1 Introduction

Many types of assets that provide a service or produce a product are replaced over time. Some examples include machines, tooling, buildings, roads, and bridges. Replacement occurs when an asset fails and cannot be repaired, or when the cost of keeping an asset operational is prohibitive, or when changes in technology make an asset inferior or obsolete, or simply when a change is desired. From a monetary perspective, the objective is to provide the required service over some predetermined planning horizon in the most economical manner. A solution to this problem specifies the *sequence* of assets that are replaced over time, including the installation and replacement times of each asset. The *service life* of an asset is the time that elapses between installation and replacement.

We investigate a serial replacement problem for a risk-averse decision maker whose objective is to maximize expected utility but whose utility function is not necessarily known in advance. Serial replacement models assume a single asset replaces another single asset, or a set of assets replaces another set. The simplest model (see, for instance, Newnan 1991), assumes an infinite horizon, deterministic cash flows, and future assets that are identical monetarily to the assets available currently. These assumptions, while making the problem easy to solve, frequently do not model reality well. A dynamic programming approach which allows future assets to differ mone-

tarily from the assets currently available, but requires a finite horizon, was developed by Wagner (1975). A more general dynamic programming approach was introduced by Oakford, Lohmann, and Salazar (1984). While still dealing only with finite horizons, the model allows more than one asset type to be available at each point in time. This model was extended to the infinite horizon case by Bean, Lohmann, and Smith (1985). Lohmann (1986) extended the Bean, Lohmann, and Smith model to allow cash flows to be stochastic by combining simulation and dynamic programming. However, Lohmann's model assumes the decision maker desires to maximize expected value. Thus, despite surveys (see Swalm 1966; Laughunn, Payne, and Crum 1980; and MacCrimmon and Wehrung 1990) to indicate the pervasiveness of risk-averse decision makers, no prior replacement model has directly considered risk.

Our investigation has three objectives. First, we want to develop a solution procedure. In order to develop a tractable procedure, we assume the reward received from installing and operating an asset over a period of time to be normally distributed. We note that the solution procedure can also be used directly for a restricted class of two-parameter distributions such as the lognormal (for a complete list see Hanoch and Levy 1969; and Kira and Ziemba 1977), and in principle with other distributions. However, in the latter case the computational burden increases dramatically because all of the convolutions must somehow be stored. Given the assumption of normality, a decision maker needs only to specify the mean and variance of the reward. Hence, we refer to our model as the mean-variance (MV) decision model. Our second objective is to identify what types of problems can be solved optimally. We use dynamic

programming to find all Pareto-optimal solutions, and the resulting state space can become quite large. Consequently, our third objective is to develop a *good* heuristic.

The remainder of this paper is organized as follows. We state the assumptions of the MV decision model and formally define the problem in §2. In §3, we summarize the shortcomings of some existing solution approaches, and then describe both our optimal solution approach and a heuristic. In §4, we discuss our computational study, followed by our conclusions in §5.

2 Assumptions of the MV Decision Model

The major assumptions of the MV decision model are as follows.

- Assets are replaced serially.
- The decision maker is uncertain about the monetary reward that would result from installing and operating an asset.
- The reward for the service provided by an asset is summarized by its after-tax net present value (NPV) distribution. These rewards can be either positive or negative.
- The NPV of an asset is statistically independent of the NPVs of all other assets in a sequence. This assumption does not imply that the NPV of an asset is independent across its possible service lives. For example, the NPV for an asset providing three periods of service can be correlated with the NPV for the same asset providing four periods of service. Instead, the assumption applies only

across replacements.

- The NPV of an asset is normally distributed, and hence the NPV of any sequence of assets will also be normally distributed.
- Each asset has a known, maximum service life beyond which it must be replaced.
- There is a finite number of alternative assets from which to choose at any point in time.
- The decision maker's utility function is a continuous, concave, monotonically increasing function of money (w).
- The decision maker's objective is to maximize expected utility. The problem then is find the sequence of assets of highest expected utility ($EU = \int U(w)f(w)dw$, where $U(w)$ is the decision maker's utility function and $f(w)$ is the normal density function of the NPV for a sequence of assets) over all possible sequences that provide service from time zero to the finite horizon (H).

3 Solution Approaches

Having now defined the problem, in this section we examine some existing approaches that are designed to solve related problems. After identifying the shortcomings of these approaches in solving our problem, we present an optimal solution approach. Then, we develop a heuristic that can be used when the state space for the optimal approach becomes prohibitively large. Finally, we describe an upper bounding

procedure that we use in our computational study to compute error bounds for the heuristic.

3.1 Existing Approaches

A variety of solution approaches for replacement problems exist in the literature. We group these approaches into three classes: traditional, expected value, and certain monetary equivalent. All are designed to solve problems that are related to, but not the same, as our problem. Thus, applying any of these approaches to our problem can result in suboptimal solutions. We evaluate the performance of these approaches in our computational study.

The traditional approach assumes an infinite horizon and that all future assets are monetarily identical to those available currently (see Newnan 1991). Such an approach can be used to model decision makers who treat a replacement problem as if it were a one-time decision. This may occur when the level of uncertainty about the NPVs for assets available in the future is so great, that a decision maker decides to only use the NPVs for assets available currently and simply ignore the future. The traditional approach replaces each random NPV by its mean and then selects the asset type and service life pair with the highest expected annual equivalent value (NPV amortized over the service life). This decision is then repeated forever. Unless the assumptions are satisfied, which implies a risk-neutral decision maker and no monetary changes over time (e.g., due to inflation or technological change), the resulting solution is suboptimal. Bean, Lohmann, and Smith (1985) offer a slightly

more realistic version. In their method, updated NPV forecasts are used at each replacement epoch to select the asset type and service life pair with the highest annual equivalent value. However, this method still fails to fully consider monetary changes over time (since by comparing the expected NPVs of assets with unequal service lives there is an implied assumption that future assets are monetarily identical to assets available currently) and continues to assume a risk-neutral decision maker.

While the traditional approach is limited in its ability to model monetary changes over time, the expected value approach is not. However, by definition, the expected value approach assumes a risk-neutral decision maker. This approach can be used to represent decision makers who treat equally monetary gains and losses, such as a large corporation may do for small investments. The expected value approach substitutes the mean NPV in place of the NPV distribution, and then finds the sequence of assets with the highest mean NPV. Oakford, Lohmann, and Salazar (1984) present a dynamic programming algorithm that can be used to find such a sequence.

The certain monetary equivalent approach goes one step further. This approach can model both monetary changes over time and a risk-averse decision maker, but the latter only in a limited manner. Each random NPV is replaced by its certain monetary equivalent. The certain monetary equivalent (CME) is the constant yielding the same utility as the random NPV (i.e., $U(CME) = EU$) The solution is then obtained by dynamic programming in exactly the same manner as for the expected value approach. Loui (1983) proved that unless the decision maker's utility function is linear or exponential (i.e., the decision maker has constant risk aversion) the dynamic

programming principle of optimality does not hold. Thus, for decision makers with decreasing risk aversion, the resulting solution may be suboptimal.

In summary, none of the existing approaches is guaranteed to find optimal solutions for our problem. The traditional approach fails to model both risk and monetary changes over time. The expected value approach, while accounting for monetary changes over time, still fails to model risk. The certain monetary equivalent approach partially models risk, but will only be guaranteed to find optimal solutions for decision makers with constant risk aversion. We now turn to developing an optimal solution approach that can model all types of risk-averse decision makers.

3.2 An Optimal Approach

We use a dynamic programming procedure to find all Pareto-optimal solutions (sequences of assets). This makes the approach quite general because it requires only that the decision maker is risk-averse, and not the exact form of the decision maker's utility function. The dynamic program differs from those used by the expected value and certain monetary equivalent approaches in that at each time period (stage), a Pareto-optimal set of sequences is identified instead of a single best sequence. The Pareto-optimal set of sequences also is a MV-efficient set of sequences, under the assumption that the NPV of each asset in a sequence is normally distributed. We define a MV-efficient sequence in the following proposition.

PROPOSITION 1. Let S_t be the set of all sequences providing service from time zero to time t . Then a sequence $s \in S_t$ with mean $\mu(s)$ and variance $\sigma^2(s)$ is MV-

efficient if for every $s' \in S_t$, $s' \neq s$

$$\mu(s) > \mu(s') \text{ and } \sigma^2(s) = \sigma^2(s'), \text{ or}$$

$$\mu(s) \geq \mu(s') \text{ and } \sigma^2(s) < \sigma^2(s').$$

MV-dominance is used to solve problems in finance (see Markowitz 1952, Ziemba and Vickson 1975, and Levy and Sarnat 1977) as well as in capital budgeting (see Park and Sharp 1990).

We note that the optimal approach uses a multi-objective dynamic program. (The objectives are to maximize the mean and to minimize the variance). A review of the multi-objective optimization literature reveals no approach that finds all Pareto-optimal solutions. Instead, either a specific criterion (less general than utility maximization) is selected a priori (see, for example, Daellenbach and De Kluyver 1980; Hansen 1980; Sigal, Pritsker, and Solberg 1980; and Henig 1990) or only a subset of all the Pareto-optimal solutions is identified (see, for example, Henig 1985; and Bard and Bennett 1991). The most frequently cited reason for using these methods is the computational burden required to find all Pareto-optimal solutions. The only exceptions are for small illustrative problems. Consequently, no existing multi-objective optimization procedure applied to our problem will be guaranteed to find the optimal solution. We now turn to the details of the dynamic program used by our optimal approach.

The dynamic programming algorithm for the optimal approach proceeds in a forward direction, with the stages being the time periods and the states being the asset type (j) and time of installation (t). The objective function values are the NPV

mean and variance for the sequence. At each point in time, the decisions are what asset type to install and for what length of service. Let $S_t^* \subset S_t$ be the set of all MV-efficient sequences providing service from time zero to time t , J_t be the set of assets available at time t , and $N_t(j)$ be the maximum service life for asset j available at time t . We can form the set \hat{S}_t as follows

$$\hat{S}_t = \{s \cup (j, t', t - t') : s \in S_{t'}^*, j \in J_{t'}, t' \geq 0, t - t' = 1, 2, \dots, N_{t'}(j)\}$$

where $(j, t', t - t')$ represents asset type j , installed at time t' , for $t - t'$ periods of service. Expressed differently, \hat{S}_t is the set of all sequences formed by appending one additional asset, $j \in J_{t'}$, to the end of a MV-efficient sequence providing t' , $t' < t$, periods of service, such that the resulting sequence provides t periods of service. S_t^* is then given by

$$S_t^* = \{s \in \hat{S}_t : s \text{ is MV-efficient}\}.$$

The boundary condition is $S_0^* = \emptyset$. Once S_H^* has been formed, the sequence $s \in S_H^*$ of highest expected utility is the optimal sequence (given the decision maker's utility function).

3.3 The Cluster Heuristic

Preliminary computational results indicated that some problems could not be solved optimally due the state space becoming prohibitively large. However, we observed that the NPVs of the MV-efficient sequences tend to cluster. Figures 1 and 2 are plots of the NPVs for two example problems. The data for Figure 1 was collected using a discount rate of 15 percent. Figure 2 plots data for the same problem solved

using a discount rate of 30 percent. Note that the breaks in the clusters occur in the variance and that the clustering is more pronounced when the discount rate is 30 percent. We use this clustering phenomenon as the basis for our heuristic approach, and refer to it as the *cluster heuristic*.

Figure 1

Figure 2

The cluster heuristic partitions sequences into clusters and identifies one representative sequence from each cluster. Sequences are selected for elimination using a measure based on first order stochastic dominance for truncated normal distributions. As Levy (1982) notes, for two truncated normal distributions with cumulative distribution functions $F_1(x)$ and $F_2(x)$, if $\mu_1 > \mu_2$ and $\sigma_1 > \sigma_2$, $F_1(x)$ dominates $F_2(x)$ by first order stochastic dominance if and only if $\gamma = (\mu_1 - \mu_2) / (\sigma_1 - \sigma_2) > \delta$, where δ denotes the number of standard deviations from the mean at which the distributions are truncated. (Here, the truncation is assumed to be symmetric, although rules also exist for nonsymmetric truncation.) The rationale for eliminating nodes using stochastic dominance is to identify the breaks in the variance. Qualitatively, for two adjacent sequences within the same cluster, γ will tend to be *large* because the denominator will be close to zero. On the other hand, for two adjacent sequences in different clusters, γ will be relatively smaller since the denominator will be much larger while the numerator stays about the same.

Once invoked, the cluster heuristic proceeds from the MV-efficient sequence with the highest NPV mean and variance to the MV-efficient sequence with lowest NPV mean and variance. The heuristic compares sequentially the NPVs for each sequence in the MV-efficient set with the next sequence in the set. The comparison involves computing a value for γ . If $\gamma > \delta$, the latter sequence is eliminated. When $\gamma < \delta$, a new cluster has been found. The single representative sequence selected by the cluster heuristic will always be the that with the highest NPV mean and variance within each cluster. Once the set is processed, if the number of sequences is greater than an upper limit, δ is reduced by half and the process repeated until the resulting set is smaller than the upper limit. For our computational study, we set the upper limit at 200 and also set the initial value of δ at 10 (which usually reduced the number of sequences below the upper limit in one pass for our problem sets). Thus, the cluster heuristic keeps only those sequences that could not be eliminated if the NPV distributions were truncated δ standard deviations from the mean. The cluster heuristic may eliminate either all but one sequence or only eliminate enough to remain within the upper limit.

The cluster heuristic uses a dynamic programming procedure that is very similar to the procedure used by our optimal approach. The key difference is that if the number of MV-efficient sequences at a given stage exceeds the upper limit, sequences are eliminated as described above. At a given stage, all the MV-efficient sequences are found before any sequences are eliminated. In this manner, the best subset can hopefully be identified. Let $S_t^{ch} \subset S_t^*$ be the subset of all MV-efficient sequences providing service from time zero to time t found by the cluster heuristic. We can

form the set \hat{S}_t^{ch} as follows

$$\hat{S}_t^{ch} = \{s \cup (j, t', t - t') : s \in S_{t'}^{ch}, j \in J_{t'}, t' \geq 0, t - t' = 1, 2, \dots, N_{t'}(j)\}.$$

S_t^{ch} is then given by

$$S_t^{ch} = \{s \in \hat{S}_t^{ch} : s \text{ is identified by the cluster heuristic}\}.$$

Defining \bar{t} as the time (stage) at which S_t^* first exceeds the upper limit, the boundary condition is $S_t^{ch} = S_t^*$ for all $t < \bar{t}$.

3.4 An Upper Bounding Procedure

To compute error bounds for the cluster heuristic in our computational study, we develop an upper bounding procedure. The procedure uses a dynamic programming algorithm closely related to the procedure used by the cluster heuristic. There are two key differences, however. First, when two nodes are compared and one is eliminated, the NPV variance of the remaining node is set equal to the variance of the eliminated node (which will never be greater) to create a pseudo-node. This pseudo-node MV-dominates both and hence provides a bound. Second, the value used for δ (the number of standard deviations from the mean to truncate the NPV distributions) is larger. The rationale for a higher value is to get a tighter bound. At the horizon, if the path (sequence) of highest expected utility does not pass through any pseudo-nodes, then the corresponding sequence is optimal. As more nodes are eliminated, the less likely this is to occur. Let S_t^{ub} be a set of MV-efficient sequences providing service from time zero to time t found by the upper bounding procedure. Note that some $s \in S_t^{ub}$

may be a pseudo-sequence. We can form the set \hat{S}_t^{ub} as follows

$$\hat{S}_t^{ub} = \{s \cup (j, t', t - t') : s \in S_{t'}^{ub}, j \in J_{t'}, t' \geq 0, t - t' = 1, 2, \dots, N_{t'}(j)\}.$$

S_t^{ub} is then given by

$$S_t^{ub} = \{s \in \hat{S}_t^{ub} : s \text{ is identified by the upper bounding procedure}\}.$$

The boundary condition is $S_t^{ub} = S_t^*$ for all $t < \bar{t}$. We now turn to our computational study.

4 Computational Study

The goals of our computational study are (i) to determine what types of problems can be solved optimally, and (ii) to evaluate the error bound performance of the cluster heuristic. We use the cluster heuristic only for problems we cannot solve optimally. Consequently, the results should reflect the maximum error performance of the cluster heuristic.

A secondary goal of our computational study is to compare the sequences identified by the traditional (TRAD), expected value (EV), and certain monetary equivalent (CME) approaches to the sequence identified by either the optimal approach or the cluster heuristic. This should provide us with an order-of-magnitude estimate of how frequently each approach finds optimal solutions. We distinguish among the different sequences by using superscripts (e.g., s^{TRAD} for the sequence identified by the traditional approach). The sequence identified by the optimal approach or cluster heuristic is denoted by s^{UTIL} . For purposes of our study, two sequences *match* if they

have equal expected utility, since a decision maker is indifferent between the two.

In the remainder of this section, we first define the performance measures for our study. Then, we discuss how our problem sets are generated. Finally we report computational results that indicate we can solve optimally a wide range of problems, and that for problems that cannot be solved optimally, the average performance of the cluster heuristic is within one percent of an upper bound.

4.1 Performance Measures

We collected four performance measures: (i) the CPU time required by each approach, (ii) the maximum cardinality of any MV-efficient set of sequences at any stage required by the optimal approach or cluster heuristic, (iii) the percentage of sequences found by a given approach that match s^{UTIL} , which we will refer to as the *matching performance*, and (iv) the utility performance of a sequence compared to s^{UTIL} . When computing the last two measures for the UTIL sequence, the values are set to one unless the cluster heuristic is used, in which case the values are computed by comparing s^{UTIL} to an upper bound.

We note that the most obvious utility performance measure, $EU(s) / EU(s^{UTIL})$, cannot be used. At best, this fraction is difficult to interpret, and it may even be undefined. To understand why, recall that our model does not restrict rewards (the NPVs) to positive values. Consequently, both the NPV and expected utility of a sequence may be negative, zero, or positive. The resulting range for the $EU(s) / EU(s^{UTIL})$ fraction is from negative infinity to positive infinity, rather than the more

conventional zero to one. To avoid a similar difficulty, Bean, Lohmann, and Smith (1985) use a random sequence as the benchmark of performance to which all other approaches are compared. This sequence is identified by randomly selecting the asset type and service life at each replacement epoch. Adopting a similar strategy for our computational study, we use

$$\frac{EU(s) - EU(s^{RAND})}{EU(s^{UTIL}) - EU(s^{RAND})}$$

as the utility performance measure, where s^{RAND} denotes the sequence of highest expected utility out of 100 random sequences. The fraction is set to one if $EU(s^{UTIL}) = EU(s^{RAND})$, and to zero if $EU(s) < EU(s^{RAND})$. We also considered generating 1000 random sequences, but found no significant statistical difference in the results of a preliminary experiment. (Details appear in Brown 1991). Consequently, we used a value of 100 to reduce the computational burden.

When we are unable to solve a problem optimally, we use

$$\frac{EU(s^{UTIL}) - EU(s^{RAND})}{EU(s^{UB}) - EU(s^{RAND})}$$

as the utility performance measure where $s^{UB} \in S_H^{ub}$ is the sequence of highest expected utility found by the upper bounding procedure. The value is set to zero if $EU(s^{UTIL}) < EU(s^{RAND})$.

4.2 Test Problem Generation

To generate a broad range of problems, we use a 2^6 factorial design. The six factors and their ranges are listed in Table 1. We first draw values for the discount rate,

Table 1: Ranges of the Factors

Factor	Low	High
Discount Rate (m)	U(0.1,0.2)	U(0.2,0.3)
Horizon (H)	DU(10,25)	DU(25,40)
Risk Aversion (z)	U(1.5,5)	U(16.5,20)
Service life (N)	DU(2,8)	DU(8,14)
Coefficient of Variation	U(0.1,0.4)	U(0.6,1.2)
Difference	U(0.0,0.05)	U(0.05,0.1)

U \equiv uniform distribution, DU \equiv discrete uniform distribution

the horizon, the number of asset types available at any point in time (which ranges from two to seven for our problem sets), and the maximum service life for each asset type. Then, we generate the NPV means and variances for every possible service life for each asset type. Finally, we set the decision maker's risk aversion. We briefly describe these last two steps below. Details are provided by Brown (1991).

We generate the NPV means and variances by perturbing a base set of values by up to 10 percent. The base set of values is calculated using three cash flow components: the initial capital outlay, the annual operating cost (or revenue), and the salvage value. We draw the mean for each component and then set the variance using the coefficient of variation factor shown in Table 1. We generate only the NPV means and variances for the assets available initially (time 0). For assets available in the future, we model technological improvement by allowing the NPVs to increase geometrically. Let (j, t, n) represent asset type j , installed at time t , for n periods of service. Then for assets available in the future

$$NPV(j, t, n) = NPV(j, 0, n) \times ((1 + g)/(1 + m))^t$$

where g is the rate of technological improvement and m is the discount rate. For

our problem sets, the base value for g ranges from 0.0 to 0.3. (When the mean NPV is negative the range for g is -0.3 to 0.0.) This base value is then perturbed by the difference factor shown in Table 1 to generate g values for each asset type.

We need a specific utility function in order to compute the utility performance measure. We use three different mathematical forms: (i) the exponential, $U(w) = (1 - e^{-cw})/c$, reflecting constant risk aversion, $r(w) = c$, (ii) the logarithmic, $U(w) = \ln(w + b)$, $(w + b) > 0$, capturing decreasing risk aversion, $r(w) = 1/(w + b)$, and (iii) the power, $U(w) = (w - w_0)^\beta$, $w_0 < w$ for all w , $0 < \beta < 1$, also reflecting decreasing risk aversion, $r(w) = (1 - \beta)/(w - w_0)$, where $r(w)$ is the risk aversion function. Brockett and Golden (1987) provide a summary of many utility functions commonly used for modeling purposes.

To ensure a reasonable basis for comparison, we selected parameters so that all utility functions would have the same level of risk aversion at the mean NPV of the sequence identified by the EV approach. The exponential utility function requires only one parameter, c . Thus, we use it as the basis for setting the parameters for the other utility functions. To set c , we first estimate the range of possible monetary outcomes for each problem. This is done by truncating the NPV distribution for the sequence found using the EV approach at ± 3.5 standard deviations. (The sequence found by the EV approach always has the greatest mean and variance.) With this range, say $[a, b]$, we set $c = \ln(z)/\max\{|a|, |b|\}$ where z ranges from 1.5 to 20 as shown in Table 1. The range for z is based on data collected from graduate students assessing their personal utility (Brown 1991). The resulting c values are consistent

both with examples in Thompson and Thuesen (1987) and with results in Kallberg and Ziemba (1983). The parameters for the logarithmic and power utility functions are then set so that the level of risk aversion is equal across all three functions at the NPV mean of the sequence identified by the EV approach. This point is selected because, broadly defined, it represents the middle of range of possible outcomes—and thus over half the range the decision maker will have higher risk aversion as compared to c , and over the other half, lower risk aversion.

4.3 Results

We randomly generated 320 problems by replicating our 2^6 design five times. We were able to find all the Pareto-optimal solutions (S_H^*) for 228 of the 320 problems, and used the cluster heuristic to solve the remaining 92 problems. Summary statistics for the matching and utility performance measures appear in Table 2. The third column shows the average matching performance. The last four columns show the average, minimum, maximum, and standard deviation of the utility performance.

For the exponential utility function, the CME approach is optimal and hence the matching performance and utility performance both equal one. The cluster heuristic is optimal in all one problem. (That is, the expected utility of the sequence equals the optimal expected utility as found by the CME approach. Note that since the cluster heuristic is used, there is no guarantee that S_H^{ch} contains all the sequences in S_H^* .) The result is an average utility performance of 0.9997. The EV approach identifies the optimal sequence about 84 percent of the time, with an average utility performance

Table 2: Summary Results

Utility Function	Approach	Percent Matching	Utility Performance			
			Average	Minimum	Maximum	Std. Dev.
Exponential	TRAD	46.25%	0.6553	0.00	1.00	0.44
	EV	83.75%	0.9181	0.00	1.00	0.26
	CME	100.00%	1.0000	1.00	1.00	0.00
	UTIL	99.68%	0.9997	0.92	1.00	0.01
Logarithmic	TRAD	46.25%	0.6607	0.00	1.00	0.44
	EV	85.93%	0.9420	0.00	1.00	0.22
	CME	49.06%	0.7884	0.00	1.00	0.35
	UTIL	78.12%	0.9977	0.75	1.00	0.02
Power	TRAD	45.93%	0.6614	0.00	1.00	0.50
	EV	85.00%	0.9414	0.00	1.00	0.22
	CME	40.31%	0.6854	0.00	1.00	0.44
	UTIL	78.43%	0.9977	0.75	1.00	0.02

of 0.9181. The lowest matching and utility performance are for the TRAD approach.

For the logarithmic and power utility functions, the matching performance and utility performance for the TRAD and EV approaches are similar to those for the exponential case. However, the CME approach's performance declines rather dramatically. This indicates the CME approach performs poorly for decision makers with decreasing risk aversion. While the utility performance for the UTIL sequence remains close to one, the matching performance declines to about 78 percent. This decline may be due to the fact that in this case, the UTIL sequences are compared to an upper bound, rather than to the optimal solution as was done for the exponential case.

The CPU times required by each approach are summarized in Table 3. (Problems were solved on an Apollo DN4000 workstation.) Both the TRAD and EV approaches require very little CPU time. The CPU time to identify the random sequence is

Table 3: CPU Times in Seconds by Approach

Approach	CPU Time			
	Average	Minimum	Maximum	Std. Dev.
RAND	7.64	1.61	25.02	6.90
TRAD	0.03	0.02	0.27	0.03
EV	0.03	0.02	0.06	0.01
CME	2.51	0.27	7.95	1.62
UTIL	28.01	0.96	661.22	67.96

higher because 100 different sequences must be generated and the expected utility for each calculated using numerical integration. Similarly, since the expected utility for each asset available at each point in time has to be found using numerical integration, the average CPU time for the CME approach is higher than for the TRAD and EV decision procedures. The average CPU time to identify the UTIL sequence (by using either the optimal approach or the cluster heuristic) is about 28 CPU seconds, and no problem required more than about 11 CPU minutes. The average CPU time for the optimal approach for the 228 problems is 3.99 seconds, while the cluster heuristic requires an average of 89.77 seconds for the remaining 92 problems (in which the state space was too large to solve the problem optimally).

We performed an analysis of variance of the CPU times required to obtain the UTIL sequence to characterize which problems are the most difficult to solve optimally. The analysis shows that of the six factors, the horizon and the maximum service life are significant using an α of 0.025. Neither of these results is surprising, since for longer horizons and longer service lives, a greater number of possible sequences must be considered. Somewhat more interesting is that the CPU times are asymptotically linear for the most difficult problems (those we cannot solve optimally). This occurs

because the number of sequences maintained by the cluster heuristic is at most the upper limit. In general, the greater the maximum service life, the greater the slope of this linear relationship since the dynamic program must *reach back* further at each stage.

For the cluster heuristic, we also studied how the limit on the maximum number of MV-efficient sequences affects the CPU times and utility performance. Using limits of 50 and 100, we resolved the 92 problems that required the cluster heuristic. The utility performance results are almost identical to those obtained using an upper limit of 200. Thus, even using an upper limit of 50 results in an average utility performance within one percent of optimal. By reducing the upper limit from 200 to 50, the average CPU time falls from 89.77 to 15.32 seconds—only a little more than twice that required to find the random sequence. The maximum number of MV-efficient sequences at any stage falls from 459 to 172, on the average. These results suggest that the upper limit used for the cluster heuristic can be rather small without adversely affecting performance.

5 Conclusion

Our computational study represents the first instance in the multi-objective optimization literature of solving problems of realistic size by finding all the Pareto-optimal solutions. The computational results indicate we can solve a wide range of problems optimally. For those problems we cannot solve optimally, the performance of the cluster heuristic is within one percent of an upper bound, and the performance is not

sensitive to the upper limit on the number of MV-efficient sequences maintained at each stage. CPU times are very reasonable—averaging 3.99 seconds for the optimal approach and 15.32 seconds for the cluster heuristic (for an upper limit of 50).

One extension to our model would be to relax the assumption that the NPV of an asset is statistically independent of the NPVs of all other assets in the sequence. Incorporating correlation may be helpful in modeling the effects of the external environment on the NPVs. Another extension would be to modify the optimal approach to solve other longest path problems, such as a knapsack problem. We are currently studying these extensions.

References

- Bard, J.F., and J.E. Bennett, “Arc Reduction and Path Preference in Stochastic Acyclic Networks,” *Management Science*, 37, 3 (1991), 198–215.
- Bean, J.C., J.R. Lohmann, and R.L. Smith, “A Dynamic Infinite Horizon Replacement Economy Decision Model,” *The Engineering Economist*, 30, 2 (1985), 99–120.
- Brockett, P.L., and L.L. Golden, “A Class of Utility Functions Containing all the Common Utility Functions,” *Management Science*, 33, 8 (1987), 955–964.
- Brown, M.J., *A Mean-Variance Serial Replacement Decision Model*, PhD Thesis, The University of Michigan, 1991.
- Daellenbach, H.G., and C.A. De Kluyver, “Note on Multiple Objective Dynamic

- Programming," *Journal of the Operational Research Society*, 31 (1980), 591–594.
- Hanoch, G., and H. Levy, "The Efficiency Analysis of Choices Involving Risk," *The Review of Economic Studies*, 36 (1969), 335–346.
- Hansen, P., "Bicriterion Path Problems," in *Multiple Criteria Decision Making Theory and Application*, Randel, G., and T. Gal, editors, Springer-Verlag, 1980.
- Henig, M.I., "The Shortest Path Problem with Two Objective Functions," *European Journal of Operational Research*, 25 (1985), 281–291.
- Henig, M.I., "Risk Criteria in a Stochastic Knapsack Problem," *Operations Research*, 38, 5 (1990), 820–825.
- Kallberg, J.G., and W.T. Ziemba, "Comparison of Alternative Utility Functions in Portfolio Selection Problems," *Management Science*, 29, 11 (1983), 1257–1276.
- Kira, D., and W.T. Ziemba, "Equivalence Among Alternative Portfolio Selection Criteria," in *Financial Decision Making Under Uncertainty*, Levy, H., and M. Sarnat, editors, Academic Press, 1977.
- Laughunn, D.J., J.W. Payne, and R. Crum, "Managerial Risk Preferences for Below-Target Returns," *Management Science*, 26, 12 (1980), 1238–1249.
- Levy, H., "Stochastic Dominance Rules for Truncated Normal Distributions: A Note," *The Journal of Finance*, 37, 5 (1982), 1299–1303.
- Levy, H., and M. Sarnat, *Financial Decision Making Under Uncertainty*, Academic Press, 1977.

- Lohmann, J.R., "A Stochastic Replacement Economic Decision Model," *IIE Transactions*, 18, 2 (1986), 182–194.
- Loui, R.P., "Optimal Paths in Graphs with Stochastic or Multidimensional Weights," *Communications of the ACM*, 26, 9 (1983), 670–676.
- MacCrimmon, K.R., and D.A. Wehrung, "Characteristics of Risk Taking Executives," *Management Science*, 36, 4 (1990), 422–435.
- Markowitz, H.M., "Portfolio Selection," *The Journal of Finance*, 6, 1 (1952), 77–91.
- Newnan, D.G., *Engineering Economic Analysis*, Engineering Press, Fourth Edition, 1991.
- Oakford, R.V., J.R. Lohmann, and A. Salazar, "A Dynamic Replacement Economy Decision Model," *IIE Transactions*, 16, 1 (1984), 65–72.
- Park, C.S., and G.P. Sharp, *Advanced Engineering Economics*, John Wiley, 1990.
- Sigal, C.E., A.B. Pritsker, and J.J. Solberg, "The Stochastic Shortest Route Problem," *Operations Research*, 28, 5 (1980), 1122–1129.
- Swalm, R.O., "Utility Theory—Insights Into Risk Taking," *Harvard Business Review*, (1966), 123–138.
- Thompson, R.A., and G.J. Thuesen, "Applications of Dynamic Investment Criteria for Capital Budgeting Decisions," *The Engineering Economist*, 33, 1 (1987), 59–86.

Wagner, H.M., *Principles of Operations Research*, Prentice-Hall, Inc., Englewood Cliffs, Second Edition, 1975. 349–356.

Ziemba, W.T., and R.G. Vickson, *Stochastic Optimization Models in Finance*, Academic Press, 1975.

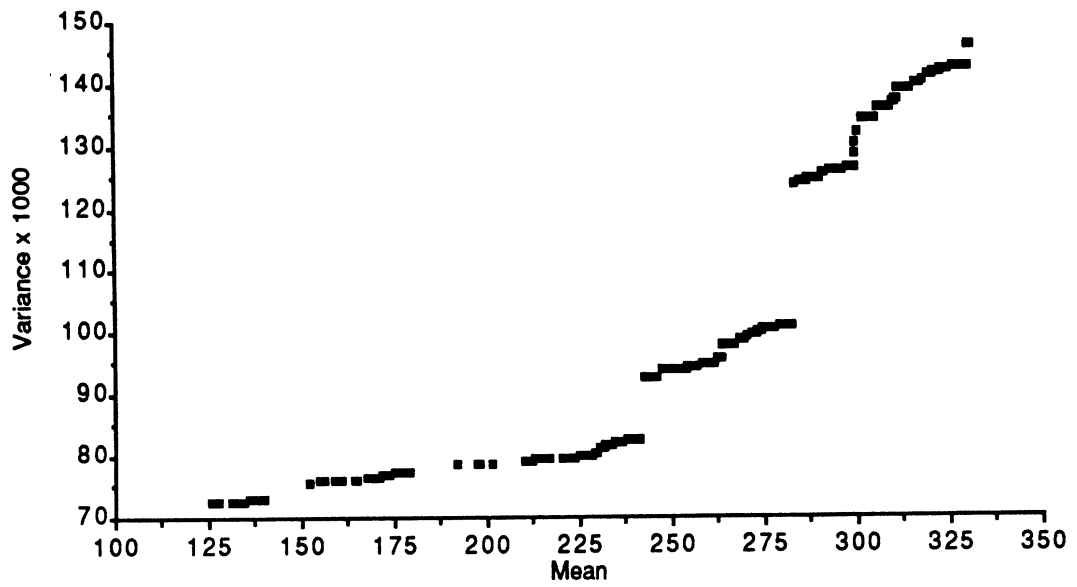


Figure 1: MV-Efficient Sequences, Discount Rate = 15%

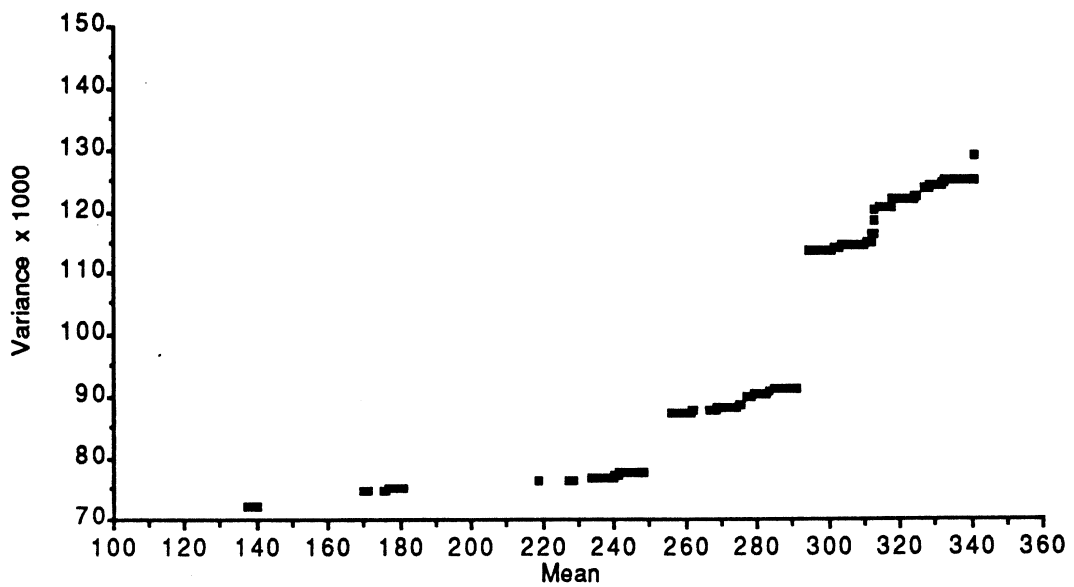


Figure 2: MV-Efficient Sequences, Discount Rate = 30%