Evaluating Portfolios of Multi-Stage Investment Projects With Approximate Dynamic Programming

Pinar Keles
Joseph C. Hartman
Lehigh University

Report No. 07T-002

Evaluating Portfolios of Multi-stage Investment Projects with Approximate Dynamic Programming*

Pinar Keles and Joseph C. Hartman

Abstract

We model the portfolio management problem for multi-stage investment projects, such as those routinely associated with research and development projects, with stochastic dynamic programming. As the recursion is intractable for large-scale problem instances, we present an approximation scheme which allows for the solution of long horizon problems in order to ensure good time zero decisions when maximizing the discounted, expected worth of decisions over time. Additionally, the approximation approach provides two estimates of the probability of making the best decision at time zero, providing additional information to the decision maker. Numerous examples illustrate the models ability to examine different budget levels, delay options (lengths, penalties, and costs), initial portfolios, project returns, and interaction effects of projects in the pipeline. While previous research focusing on single project analysis has highlighted the importance of the delay option, we illustrate how critical this option is when one considers a portfolio of projects over time, especially when projects late in the review process may fail and/or budgets are small.

1 Introduction

Companies often make investments in stages. For example, pharmaceutical firms develop drugs through a multi-stage process, culminating in a final review by the Food and Drug Administration (FDA) before production and distribution. Energy firms acquire land or drilling rights and perform extensive testing (i.e., seismic testing and drilling exploratory wells) before fully developing an oil field. Manufacturing firms iterate through various designs before mass producing a product. These investments are often associated with the research and development (R&D) process.

During these investments, information is collected at each stage about the potential success of the product or project. With this information, the firm must often determine whether to continue the project (to the next stage) or terminate its development. (The decision to delay an action may also be feasible.) These investment decisions are inherently difficult due to the uncertainty

*Industrial and System Engineering Technical Report 07T-002, Lehigh University, March, 2007.

in the information, timing, and returns, but the analysis is further complicated as companies generally consider a multitude of these types of projects simultaneously in a portfolio under a budget constraint.

There are numerous factors that influence whether a project is to be continued or abandoned. These include:

1. The success (or lack thereof) of the project in previous development stages.
2. The pipeline or portfolio of projects currently under development.
3. The potential of newly arriving projects to consider for investment.
4. Budgeting constraints.

In this paper, we model these sequential investment decisions for a portfolio of investments over time with stochastic dynamic programming. Specifically, we assume the following:

- New investment opportunities (projects) arrive over time according to a stochastic process. Upon arrival, the expected return (at a future time), expected development process (stages, durations and probabilities of success), and expected development costs are made known. (These are estimates which may be updated over the course of a project's development.)
- Previously accepted projects in a portfolio are reviewed at scheduled times.
- Costs are paid during each stage in which a project is accepted (or continued).
- Rewards are not received until the project is launched (after successfully completing all review stages).
- New projects may be accepted or rejected while current projects may be continued, abandoned, or delayed.

The objective of the problem is to maximize expected, discounted returns subject to a budget constraint. It is assumed that budgets are constrained over some period of time (say a year), but projects are evaluated more frequently (say each month), and the process repeats over time (every year).

This problem has many features of a dynamic, stochastic knapsack problem (DSKP). In this problem, items arrive according to a stochastic process with the size and value of the item becoming known upon arrival. If capacity exists in the knapsack, the item may be accepted, earning a reward. If rejected (possibly at some cost), space in the knapsack is reserved for future arrivals.

Researchers have examined many variants of this problem. Caraway et al. [12], Henig [33], Steinberg and Parks [75] and Sniedovich [72, 73], studied a static version of the problem with stochastic rewards. Das and Ghosh [18] studied the binary knapsack problem with a random knapsack size.

More closely related to our problem, Kleyweight and Papastavrou [39, 40] and Papastavrau et. al. [55] studied variants of the dynamic and stochastic knapsack problem differing by item size and horizon time. For certain conditions, they prove an optimal threshold acceptance policy. Lu et. al. [47] study the stochastic knapsack problem over a finite horizon where projects arrive sequentially. Instead of deriving threshold policies, they provide acceptance and rejection time intervals for each of project type based on the remaining budget and time in the acceptance interval for the finite horizon case.

Note that there are many applications which can be viewed as dynamic, stochastic knapsack models. Some of these problems include yield management (Belobaba [3], Brumelle and McGill [11], Wollmer [82], Lee and Hersh [43], Robinson [64], and Van Slyke and Young [81]); resource allocation (Prastacos [60], Mendelson et. al. [50] and Righter [62]); and communication allocation problems (Ross and Tsang [66], and Ross and Yao [67]).

The application in this paper differs from previous DSKP work because two types of items arrive over time: those according to a stochastic process and those according to a scheduled (deterministic) process. Furthermore, knapsack capacity is replenished with a new budget each cycle and returns from accepting an item are delayed as they are not received until the multi-stage investment process is complete. That is, acceptance of an item (project) may entail a number of costs (investments) before a reward (return) is received.

The literature on capital budgeting is also related and vast, with the work in R&D project selection most closely related. R&D project selection methods and techniques have been studied for years and there are numerous studies with quantitative and qualitative approaches. Early studies focus on the ranking of all available projects according to a metric, subject to resource restrictions. Ranking metrics include BC Analysis, NPV Analysis, Profitability Index, and various other scoring methods (i.e., Matheson et al. [49], Dean and Nishry [21], Moore and Baker [53], Krawiec [42], Kocaoglu and Iyigun [41], and Henriksen and Traynor [34]).

Analysis techniques for R&D investment decisions have progressed tremendously over the past decade. This has mainly been accomplished through the application of financial analysis, or real options, models. The number of papers in this area seems immeasurable, including the comparison of real options to traditional measures of worth (Faulkner [23], Mitchell and Hamilton [52], Pries et al. [61]), ability to capture the value of managerial flexibility (Benninga and Tolkowsky [4], Herath and Park [35], Mitchell [51] and Pennings and Lint [56]), or details of implementation (Angelis [1] and Bowman and Moskowitz [10]). Merck is noted as the first drug firm to use real options analysis (Nichols [54]). Numerous other pharmaceutical cases are noted in Trigeorgis [80]. Unfortunately, as pointed out recently by Trigeorgis [80], real options are designed to analyze a single investment opportunity over time. MacMillan and McGrath [48] and Trigeorgis [80] have suggested rectifying this by using rules to tie the real options analysis to portfolio decisions.

A number of researchers have analyzed R&D portfolio decisions with the use of mathematical programming models (i.e. Freeman and Gear [26], Golabi [29], Bard et al. [2], Liberatore [44, 45, 46], Graves et. [30], and Jones and Mendez [38]) or decision networks (i.e. Heidenberger [32] and Blau et. al [9, 8]). However, these models generally assume a single period problem and often ignore project interdependencies, which distorts the notion of a pipeline of projects over time. Caraway and Schmidt [13] and Schmidt [68] consider project interdependencies with the use of mathematical and dynamic programming. However, they also do not consider multiple periods of analysis.

Studies that consider multiple periods, but not interdependencies, include Prastacos [60], Ringuest and Graves [63], and Chun [17]. Choi et al. [16] considers scheduling projects considering uncertainties in phase durations, costs, and phase outcomes. Studies which include some interactions, through additive returns or mutually exclusive constraints, include Fox et al. [25], Childs et al. [14], Childs and Triantis [15], Ghasemzadeh et. al [27], Dickinson et al. [22] and Stummer and Heidenberger [76].

Two papers related to our approach include Rogers et al. [65], who use real options and mixed integer linear programming, and Subramanian et al. [77, 78], who combine mathematical programming with discrete event simulation, to make project investment decisions. However, these approaches do not consider project interdependencies and do not capture all of the dynamics (new project arrivals and budgets available over time) of the process.

Our objective is to maximize the expected, discounted return resulting from project investment decisions over time. We assume the process starts arbitrarily at time zero, where a number of possible investment decisions may be under consideration. As the process evolves dynamically over time, our goal is to make optimal investment decisions at time zero, assuming an infinite horizon of analysis. As the state space for our stochastic dynamic program requires tracking projects through development stages, large-scale problems become intractable over reasonable time horizons. Thus, we provide an approximation approach to solve large-scale instances, defined by long horizons. In our approximate dynamic programming framework, we solve the stochastic dynamic program optimally over some horizon time T while approximating the value of each reachable system state at time T for an infinite horizon problem with the use of simulation, deterministic dynamic programming, and regression. In addition to allowing for the solution of large problems, the approximation approach provides some measure of variance in returns which can be used to gauge risk.

A number of approximate dynamic programming approaches have been developed in the literature, often for specific applications. In most approaches, optimal or heuristic methods are utilized to provide estimates for a subset of state values while the remaining state values are estimated or interpolated via another process. If the (rough) shape of the value function is known, parametric approximation techniques (Bertsekas [5]) can be successful. Bertsekas and Tsitsiklis [6] utilize neural networks and simulation techniques for approximating state values. Polynomial and spline approximations have been studied by Johnson et al. [37], Foufoula-Georgiou [24], and Philbrick and Kitanidis [57].

In other applications, linear programming is used to estimate state values and the resulting dual values are used to approximate the remaining state values. Other applications include using non-linear and piecewise linear approximation methods. There are numerous approximate DP applications which have been successful in solving multidimensional knapsack (Bertsimas and Demir [7] and Hua et. al. [36]), vehicle routing (Secomandi [69, 70], Powell [58] and Spivey and Powell [74]), fleet management (Godfrey and Powell [28], Topaloglu and Powell [79], Powell and Carvalho [59]) and capacity reservation (Hartman and Perry [31]) problems.

As dynamic programs can also be solved with linear programming (resulting in an exponential number of constraints), studies have also attacked the problem by approximating the associated linear program. These methods include the work of de Farias and Van Roy [19, 20].

In addition to providing a solution procedure, we validate our decisions by computing the probability of selecting the optimal time zero decision through two sampling procedures. This provides confidence in the decision for the decision-maker. Note that while many approximation procedures provide bounds on the optimal solution value, we are more interested in validating the correct time zero decision. We also provide insight into a number of different problem parameters, including budget levels, delay options, delay penalties and interaction effects through numerous examples.

The paper proceeds as follows: Section 2 details our problem definition and modeling approach. In Section 3, we present an illustrative example which motivates the approximate dynamic programming approach presented in Section 4. Sections 5, 6 and 7 provide several examples which considered different aspects of the problem and we analyze decisions for various cases. In Section 8, we consider interaction effects and provide an new example, before concluding

2 Model Definition

In this section, we define the stochastic dynamic program (SDP) and its defining features.

2.1 System State

In order to make optimal investment decisions, the portfolio of projects must be tracked over time. Thus, define the vector $[B_t, n_1, n_2, ..., n_N]$ as the state, S_t, of the system at time t. The value B_t refers to the amount of budget remaining for investments while the values n_1 through n_N refer to the number of projects in each investment stage, of which there are N.

Note that for implementation, we must actually track the review schedule of each project more closely. However, we utilize this "shorthand" notation of tracking projects in stages for the moment.

2.2 Arrivals

Define Ψ_t as the set of potential projects to be evaluated at time t. These projects consist of:

1. New projects which arrive according to a defined stochastic process. For our model, we assume at most one arrival per period with probability p_t^A, although multiple arrivals may be modeled accordingly. Upon arrival, estimates of stage lengths, probability of success in each stage, and the expected return are made known; and

2. Projects currently in the pipeline and up for review at time t. It is assumed that project i successfully completes stage j with probability p_{ij}^S and only successful projects are reviewed.

Further define any realization of arrivals as $\psi_t \subset \Psi_t$ in period t. Note that the probability of arrival of ψ_t can be defined straightforwardly from p_t^A and p_{ij}^S assuming independence of arrivals.

2.3 Decisions, Costs, Rewards and Constraints

Newly arrived projects must be (1) accepted or (2) rejected while renewal projects are (1) continued; (2) terminated; or (3) delayed. The set of projects to accept or continue is defined by $\delta_t \subseteq \psi_t$. Defining the budget at time t as B_t and the investment costs associated with δ_t as $C(\delta_t)$, for feasibility, we require:

$$C(\delta_t) \leq B_t \quad \forall t = 0, 1, \ldots, T,$$

as decisions are made in periods 0 through T. In addition to incurring an investment cost, an expected reward $R(\delta_t)$ is received for projects which have completed all review stages.

Projects which are rejected or terminated do not incur costs or receive rewards. However, these projects are no longer considered for investment through time T.

When projects are delayed, they incur a delay cost of $C_d(\delta_t)$. This cost can be perceived as an operational cost and it does not affect the budget. We can delay projects for a defined number of periods and delays may be repeated. Furthermore, if we accept a project after a delay period, a delay penalty may be incurred which lowers the return, $R(\delta_t)$.

2.4 State Transition

Given the state vector $[B_t, n_1, n_2, ..., n_N]$, transitions are defined according to δ_t. At the end of each period, the budget is updated as:

$$B_{t+1} = B_t - C(\delta_t).$$

For newly accepted projects:

$$n_1 \rightarrow n_1 + 1$$

For projects in stage j that are terminated:

$$n_j \rightarrow n_j - 1$$

For projects in stage j that are continued:

$$n_j \rightarrow n_j - 1,$$

$$n_{j+1} \rightarrow n_{j+1} + 1$$

Finally, for projects that are delayed:

$$n_j \rightarrow n_j$$

Note that these decisions are "cumulative" such that a decision to terminate a project in stage 3 and continue a project in stage 2 results in a decrease in the number of projects in stage 2 (by one project) and no change in the number of projects in stage 3. We also update review times of projects according to each decision.

2.5 Recursion Equation

Define $V_t(S)$ as the maximum expected, discounted value when starting with initial state S at time t, defined by budget B_t and a portfolio of projects, and making optimal investment decisions through period T. The recursion is formally defined as:

$$V_t(S) = \sum_{\psi_t \subseteq \Psi_t} p_t(\psi_t) \left\{ \max_{\delta_t \subseteq \psi_t : C(\delta_t) \leq B_t} \left(R(\delta_t) - C(\delta_t) - C_d(\delta_t) + \alpha V_{t+1}(S'(\delta_t)) \right) \right\}, t = 0, 1, 2, \ldots, T. \quad (1)$$

Note that $S'(\delta_t)$ defines the transition of the state vector S given the decisions set δ_t which is constrained by the arrival set ψ_t and budget B_t. The value of α is the one period discount factor. To solve the recursion over a finite horizon, the terminal condition is defined as:

$$V_{T+1}(S) = 0, \quad \forall S. \quad (2)$$

We examine the solution of this recursion in the following example.

3 Illustrative Example

Consider an example with eleven projects in a portfolio at time zero. Table 1 defines the projects according to stages, review times and estimated returns, if launched. For example, project 10 is up for review at time zero and project 1 will finish the first stage in one period. A project must complete six review stages before launching. The final column provides current estimates of the probability of technical success in each remaining stage. If a project does not meet its technical requirements, its market value is zero. It is assumed for this portfolio that projects are similar (such as drugs that combat similar diseases), and hence have similar returns.

We solve this problem over 12 monthly periods (one year). In addition to reviews of projects 1, 3, 4, 5, 7, 9, 10 and 11 in that time frame, the probability of a new arrival in a given period is 0.5. However, we limit new arrivals to at most one per three periods. As noted earlier, new projects can be accepted or rejected whereas current projects can be terminated, continued or delayed. In this example, we can delay projects for 2-period increments. Delay costs ($1 per two

Table 1: Example data for initial project set.

Project (i)	**Stage** (j)	**Renewal Time**	$R_i(\$)$	p_{ij}^S
1	1	1	1500	0.48, 0.51, 0.65, 0.78, 0.85, 1
2	2	11	1500	0.55, 0.65, 0.80, 0.90, 1
3	2	8	1500	0.5, 0.6, 0.75, 0.88, 0.95
4	2	4	1500	0.53, 0.62, 0.76, 0.85, 0.98
5	3	9	1500	0.68, 0.75, 0.89, 1
6	4	14	1500	0.77, 0.88, 1
7	4	10	1500	0.8, 0.9, 1
8	5	12	1550	0.85, 1
9	5	5	1550	0.82, 0.98
10	6	0	1600	1
11	6	6	1600	0.95
New Project	-	-	1600	0.48, 0.56, 0.68, 0.75, 0.89, 1

Table 2: Cost to start corresponding stage.

Stage	**Length**	**Cost** (\$)
1	18	18
2	12	24
3	12	12
4	18	18
5	24	48
6	12	6
Launch	-	1

period delay) do not affect the budget. Furthermore, a delay penalty (10% reduction in return) is assessed for projects which are accepted after a delay. The investment costs and stage lengths are given in Table 2.

For this example, we assume that there is one new project arrival at time zero, thus, as shown in Figure 1, the decisions are to (1) launch (accept) project 10 and accept the new arrival (project 12), (2) terminate project 10 and accept the new arrival, (3) launch project 10 and reject the new arrival, and (4) terminate project 10 and reject the new arrival. The decisions to launch or not, and accept or not, affect the available budget for future projects, as shown in the state space. If

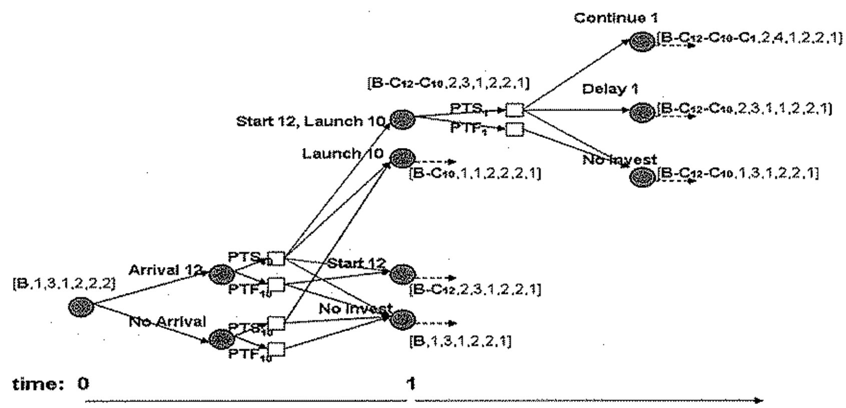

Figure 1: First two periods of decisions.

project 10 is launched and a new project is accepted, an investment cost and acceptance cost are paid with an expected return for launching project 10. At time period one, project 1 is up for review. The values of PTS_1 and PTF_1 in the figure refer to the probabilities of technical success and failure of project 1, as a result of its first stage trials. Given a success, the decisions are to continue, delay or reject while a failure leads to termination.

Table 3 displays the optimal decision in each period for the scenario where all projects are successful and there is a new arrival every three periods. As only projects 10 and 11 are launched and can achieve a return, these are the only accepted projects. As expected, the results are misleading, as the horizon is not long enough to consider all possible returns of the projects over time to ensure good decisions. Ideally, we must consider an infinite horizon problem (or sufficiently long finite horizon with discounting), but this is computationally intractable if one considers all possible trial outcomes and investment decisions over time.

The computational difficulty of this problem is due to the exponential growth in the number of states with the number of projects to be analyzed over time. Specifically, the recursion is exponential in the number of projects to be analyzed at each time period as combinations of investment decisions are analyzed. For this reason, an approximation method is presented in the following section to solve this large scale problem.

Table 3: Decisions in each period if all projects are successful.

Time	Decision
0	Launch Project **10** and Reject New Project
1	Stop Project **1**
2	Reject new project
3	Do Nothing
4	Stop Project **4**
5	Stop Project **9** and Reject New Project
6	Launch Project **11**
7	Do Nothing
8	Stop Project **3** and Reject New Project
9	Stop Project **5**
10	Stop Project **7**
11	Stop Project **2** and Reject New Project

4 An Approximate Dynamic Programming Approach

Our goal is to determine the optimal time zero decision under the assumption of an infinite horizon. However, there exists some finite horizon time which approximates an infinite horizon because discounting drives the effect of future costs and revenues to zero at time zero. (This is often referred to as a forecast horizon. See Smith and Schochetman [71] among others for further details on infinite horizon problems.) This motivates our approximation approach.

For a given initial state, we can build a network of "reachable" states (all states that can feasibly result from decisions over time) over some time horizon T. Our approximation approach assumes that we can solve the stochastic dynamic program optimally (without approximation) over these T periods and thus it is possible that T is defined by a relatively small number of periods. If this were a finite horizon problem to be solved over T periods, then we would designate the value of $V_T(S) = 0$ for all reachable states S at time T. In our approximation approach, we substitute estimates of $V_T(S)$ under the assumption that the problem continues for an infinite horizon (or considerably longer than T). Thus, the value of each state at time T must be estimated in order to solve the SDP optimally through time zero to obtain the initial decision.

This framework requires four decisions:

1. Determining the horizon T over which the SDP can be solved optimally. This is generally defined by computational limits.

2. Determining which reachable states at time T are to be directly estimated and how they are to be estimated. We define these states as the set S_{sample}.

3. Determining how the remaining states at time T (not in S_{sample}) are to be estimated.

4. Determining how the time zero decision is to be verified.

As the definition of T is constrained by the computing environment (number of states that can be retained in memory), we will address the last three questions in the following sections.

4.1 Estimating a Reachable State Value

The difficulties with solving the SDP are twofold: (1) combinations of investment options for a given time period and (2) increased number of probabilistic paths due to probabilistic arrivals and stage outcomes. As it is difficult to remove the combinatorial aspects of the problem other than limiting arrivals or the number of projects in the initial pipeline, we address (2) in our approximation approach.

If we simulate arrivals, stage outcomes, and successive budget levels over time, we would define a deterministic problem which merely requires analyzing investment decisions over time that could be solved with deterministic dynamic programming (DP) as the problem is still sequential. As an example, the multiple arcs emanating from the circular nodes in Figure 1 would be replaced with a single, deterministic arc. The remaining decisions would merely entail examining the combinations of investment choices in the same period.

Thus, for each state S at time T, we could simulate outcomes over a sufficiently long horizon and solve the corresponding deterministic DP. This could be repeated with the average of the time T state values over all runs providing an estimate of $V_T(S)$. This method could be repeated for each state at time T, with common random numbers being used to reduce the burden of producing replica data.

4.2 Defining the Set S_{sample}

As it may be computationally prohibitive to obtain an explicit estimate of $V_T(S)$ for each state S at time T, we may select a sample of states to estimate with simulation and DP, and estimate the remaining state values through interpolation. This requires selecting a subset of S at time T to estimate explicitly. We use stratified sampling to accomplish this goal, which is related to our interpolation method. In the previous example, projects were similar (returns and development processes) and thus, the portfolio was defined by the number of projects in each stage. With six stages, we have $2^6 = 64$ combinations of investments in various stages and we sample from these 64 categories, although some of the categories are infeasible due to budget limits, reducing the category selection set. Defining S_{sample} is clearly problem specific.

4.3 Estimating the Remaining Reachable State Values

Given that we have sampled a subset of states at time T and estimated the state values with simulation and DP, the remaining states are to be estimated with regression. Unfortunately, predicting the shape of the value function is not easy for a given state (portfolio and budget). Thus, we estimate a regression equation similarly to our sampling procedure. In the previous example, the state value is the dependent variable and the budget level and the number of projects in each development stage are dependent variables. Hence, we have seven independent variables. We perform a multi-linear regression on the state sample estimates. If the model is adequate and valid, it is used to estimate remaining state values.

4.4 Verifying the Time Zero Solution

Defining these estimated state values as $\widetilde{V}_T(S)$, we revise our terminal condition (2) as:

$$V_T(S) = \widetilde{V}_T(S), \quad \forall S. \tag{3}$$

and solve the SDP accordingly.

While it is common when using approximation techniques to bound the optimal solution such that one can determine whether the approximation is reasonable, we are more interested in determining whether the time zero decision is the best option. For this reason, we estimate the probability that the time zero decision returned by the approximation approach is optimal.

In our approach, the simulation and DP method provide a number of estimates of state values in S_{sample} at time T. If we sample from these numerous estimates for each state in S_{sample}, generate the regression equation and solve the SDP, we can estimate $V_0(S)$. Furthermore, we can repeat this procedure to develop a distribution on the value of $V_0(S)$, or, equivalently, a distribution on the value of each decision at time zero. With this distribution information, we can calculate the probability of selecting the optimal decision when compared to next best decision. Define P as the probability that the state value of the best decision at time zero is larger than the state value of the second best decision.

Additinally, when developing the distribution information on $V_0(S)$, we solve the SDP optimally numerous times. For each of these instances, an optimal and second best decision are identified. As the decision ordering for each instance may change, define the "best" decision as the one which occurs more frequently over these instances. Further define P' as the ratio of instances where the best decision state value is larger than second best decision state value. This provides a second estimate of the probability of making the correct decision. We argue that P and P' provide insight into the accuracy of our approximation procedure and provide some measure of risk to the decision maker.

Our final testing procedure requires solving the SDP with approximation over a number of different horizons T. Note that for longer T, the model more accurately captures actual decisions and outcomes which may not be captured in the approximation. Thus, we examine a number of horizons to see if the optimal decision at time zero is consistent over different horizon lengths. This follows the literature on infinite horizon optimization which says that if the optimal time zero decision does not change for a number of horizons, the optimal time zero decision has been identified. We cannot state this definitively here due to the complex state space (i.e., optimal rules exist for the equipment replacement problem where the state space is defined by a single parameter). However, consistent time zero decisions provide the decision-maker with greater assurance that the optimal time zero decision has been identified.

5 Revisiting Example 1 with Approximation Techniques

The example introduced in Section 3 is solved again with the approximation algorithm. We followed the implementation method described in the previous section. Specifically, we determined S_{sample} at $T = 12$ and estimated the sample state values with simulation and deterministic DP. A regression equation was generated to determine the remaining reachable state values at $T = 12$. Table 4 provides a summary of our results when solving the SDP optimally over horizons of $T = 1$ through $T = 12$. The table denotes the number of states sampled and the time required to estimate the sample state values (simulation and deterministic DP). The simulation and deterministic DP were solved over a horizon of 108 periods (9 years) to allow for returns to be realized. As an example of this analysis, we obtained the following fit equation for the 12 horizon case (n_j represents the number of projects in stage j):

$$V_T(S) = 1249 + 73n_1 + 91n_2 + 275n_3 + 464n_4 + 904n_5 + 1409n_6 \quad (4)$$

The resulting estimates of V_0 and the time zero decision are given in the Table 4 along with a confidence interval on V_0. Furthermore, estimates of P and P' are provided. These values give a good indication that the correct time zero decision has been made along with the fact that decisions do not change with T. Unfortunately, the results are not very intriguing, as the time zero decision is clearly dominated by the decision to launch project 10 and reject a new project at time zero. Thus, we analyze a number of cases in further detail with varying budget levels, delay costs, and penalties.

Table 4: Time zero decisions and values for different horizons.

T	**Reachable States**	S_{sample}	**Approx Time** (hr)	V_0 **(95% C.I.)**	**Time Zero Decision**	P	P'
1	4	4	1.37	6914 (6660,7169)	Launch 10, Reject New	0.99	0.9
2	8	8	3.32	7286 (6822,7751)	Launch 10, Reject New	0.99	0.95
3	32	32	8.53	7063 (6667,7459)	Launch 10, Reject New	0.99	1
4	32	32	8.22	7372 (6894,7850)	Launch 10, Reject New	0.99	1
5	96	16	8.6	6536 (5955,7117)	Launch 10, Reject New	0.99	1
6	1,008	32	7.26	6750 (6260,7377)	Launch 10, Reject New	0.99	1
7	3,360	32	9.2	7328 (6916,7740)	Launch 10, Reject New	0.99	0.95
8	7,200	32	8.1	6452 (6048,6856)	Launch 10, Reject New	0.99	1
9	58,104	32	8.5	7036 (6702,7370)	Launch 10, Reject New	0.99	1
10	284,364	32	7.92	7003 (6614,7392)	Launch 10, Reject New	0.99	1
11	1,396,222	102	15.6	7079 (6679,7479)	Launch 10, Reject New	0.99	0.9
12	8,979,840	102	15.43	7072 (6787,7357)	Launch 10, Reject New	0.99	1

Table 5: Periodic decisions for high ($100) and low ($60) budget levels without the delay option.

t	Budget	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
		All successful	Project 7 fails	Project 5 fails	Project 7 and 5 fail
0	High	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)
	Low	Launch 10, Reject New ($59)	Launch 10, Reject New ($59)	Launch 10, Reject New ($59)	Launch 10, Reject New ($59)
1	High	Stop 1 ($99)	Stop 1($99)	Stop 1 ($99)	Stop 1 ($99)
	Low	Stop 1 ($59)	Stop 1($59)	Stop 1 ($59)	Stop 1 ($59)
2	High	Reject New ($99)	Reject New ($99)	Reject New ($99)	Reject New ($99)
	Low	Reject New ($59)	Reject New ($59)	Reject New ($59)	Reject New ($59)
3	High	Do Nothing	Do Nothing	Do Nothing	Do Nothing
	Low	Do Nothing	Do Nothing	Do Nothing	Do Nothing
4	High	**Continue 4** ($87)	**Continue 4** ($87)	**Continue 4** ($87)	**Continue 4** ($87)
	Low	**Stop 4** ($59)	**Stop 4** ($59)	**Stop 4** ($59)	**Stop 4** ($59)
5	High	Continue 9, Reject New ($81)	Continue 9, Reject New ($81)	Continue 9, Reject New ($81)	Continue 9, Reject New ($81)
	Low	Continue 9, Reject New ($53)	Continue 9, Reject New ($53)	Continue 9, Reject New ($53)	Continue 9, Reject New ($53)
6	High	Launch 11 ($80)	Launch 11 ($80)	Launch 11 ($80)	Launch 11 ($80)
	Low	Launch 11 ($52)	Launch 11 ($52)	Launch 11 ($52)	Launch 11 ($52)
7	High	Do Nothing ($80)	Do Nothing ($80)	Do Nothing ($80)	Do Nothing ($80)
	Low	Do Nothing ($52)	Do Nothing ($52)	Do Nothing ($52)	Do Nothing ($52)
8	High	**Continue 3**, Reject New ($68)	**Continue 3**, Reject New ($68)	**Continue 3**, Reject New ($68)	**Continue 3**, Reject New ($68)
	Low	**Stop 3**, Reject New ($52)	**Stop 3**, Reject New ($52)	**Stop 3**, Reject New ($52)	**Stop 3**, Reject New ($52)
9	High	**Continue 5** ($50)	**Continue 5** ($50)	Stop 5 ($68)	Stop 5 ($68)
	Low	**Stop 5** ($52)	**Stop 5** ($52)	Stop 5 ($52)	Stop 5 ($52)
10	High	Continue 7 ($2)	Stop 7 ($50)	Continue 7 ($20)	Stop 7 ($68)
	Low	Continue 7 ($4)	Stop 7 ($52)	Continue 7 ($4)	Stop 7 ($52)
11	High	Stop 2, Reject New ($2)	Continue 2, Accept New ($20)	**Continue 2**, Reject New ($8)	Continue 2, Accept New ($38)
	Low	Stop 2, Reject New ($4)	Continue 2, Accept New ($22)	**Stop 2**, Reject New ($4)	Continue 2, Accept New ($2)

5.1 High and Low Budget Levels without the Delay Option

Table 5 compares solutions for four different scenarios under two different budget level assumptions. In this case, we delay, instead of terminate, projects 3, 4 and 5. Under scenario 2 where project 7 fails, we accept projects 3, 4 and 5 after time 10 and use most of the budget. This is a critical change compared to having no delay option. Under scenario 4, we also continue project 2 since project 5 fails. Note that the delay option did not change the previous solutions with a high budget.

5.2 Including the Delay Option

Table 6 illustrates solutions where we have a low budget and the delay option. In this case, we

Table 6: Periodic decisions for low budget level ($60) with the delay option.

t	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
	All successful	Project 7 fails	Project 5 fails	Project 7 and 5 fail
0	Launch 10, Reject New ($59)	Launch 10, Reject New ($59)	Launch 10, Reject New ($59)	Launch 10, Reject New ($59)
1	Stop 1 ($59)	Stop 1 ($59)	Stop 1 ($59)	Stop 1 ($59)
2	Reject New ($59)	Reject New ($59)	Reject New ($59)	Reject New ($59)
4	**Delay 4** ($59)	**Delay 4** ($59)	**Delay 4** ($59)	**Delay 4** ($59)
5	Continue 9, Reject New ($53)	Continue 9, Reject New ($53)	Continue 9, Reject New ($53)	Continue 9, Reject New ($53)
6	Launch 11, Delay 4 ($52)	Launch 11, Delay 4 ($52)	Launch 11, Delay 4 ($52)	Launch 11, Delay 4 ($52)
8	**Delay 3,4**, Reject New ($52)	**Delay 3,4**, Reject New ($52)	**Delay 3,4**, Reject New ($52)	**Delay 3,4**, Reject New ($52)
9	**Delay 5** ($52)	**Delay 5** ($52)	Stop 5 ($52)	Stop 5 ($52)
10	Continue 7, Stop 3,4 ($4)	Stop 7, **Continue 3,4** ($28)	Continue 7, Stop 3,4 ($4)	Stop 7, **Continue 3,4** ($28)
11	Stop 2,5, Reject New ($4)	**Continue 5**, Stop 2, Reject New ($10)	Stop 2, Reject New ($4)	Continue 2, Reject New ($16)

delay project 1 under scenario 4 and later accept it when we have available budget. This shows the importance of determining the value of the delay penalty, since it plays a significant role for delaying decisions. For higher delay penalties, project 1 is terminated.

Table 7: Periodic decisions for a high budget level ($100) with a small delay penalty.

t	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
	All successful	Project 7 fails	Project 5 fails	Project 7 and 5 fail
0	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)
1	**Delay 1** ($99)	**Delay 1** ($99)	**Delay 1** ($99)	**Delay 1** ($99)
2	Reject New ($99)	Reject New ($99)	Reject New ($99)	Reject New ($99)
3	Delay 1 ($99)	Delay 1 ($99)	Delay 1 ($99)	Delay 1 ($99)
4	Continue 4 ($87)	Continue 4 ($87)	Continue 4 ($87)	Continue 4 ($87)
5	Continue 9, Delay 1, Reject New ($81)	Continue 9, Delay 1, Reject New ($81)	Continue 9, Delay 1, Reject New ($81)	Continue 9, Delay 1, Reject New ($81)
6	Launch 11 ($80)	Launch 11 ($80)	Launch 11 ($80)	Launch 11 ($80)
7	Delay 1 ($80)	Delay 1 ($80)	Delay 1 ($80)	Delay 1 ($80)
8	Continue 3, Reject New ($68)	Continue 3, Reject New ($68)	Continue 3, Reject New ($68)	Continue 3 and Reject New ($68)
9	Continue 5 and Delay 1 ($50)	Continue 5 and Delay 1 ($50)	Stop 5, Delay 1 ($68)	Stop 5, Delay 1 ($68)
10	Continue 7 ($2)	Stop 7 ($50)	Continue 7 ($20)	Stop 7 ($68)
11	Stop 1,2, Reject New ($2)	Stop 1, Continue 2, Accept New ($20)	Stop 1, Continue 2, Reject New ($8)	**Continue 1**,2, Accept New ($14)

5.3 Including the Delay Option with a Small Delay Penalty

Including a small delay penalty does not change the decisions for the low budget case. However, Table 7 illustrates the case where we have a high budget and the delay option with a small penalty (2% reduction in the expected return). The remaining budget is indicated after each periodic decision. Under the first scenario where all projects successfully complete their stage reviews, all projects from the initial pipeline except projects 1 and 2 are accepted with a high budget. For the low budget case, we stop projects 1, 2, 3, 4 and 5, since project 7 uses most of the budget of $60. Under scenario 2 where project 7 fails, we continue project 2 and accept a new project at time 11 for both budget levels. Under scenario 3 where project 5 fails, we continue project 2 for the high budget case. On the other hand, for scenario 4 where both projects 5 and 7 fail, we again continue project 2 and accept a new project for both budget levels. However, in this case, we have a large amount of unused budget ($38) at $t = 11$. These results show that the budget is reserved for advanced development stage projects because they are closer to achieving a return. Unfortunately, having these projects at the end of the budget cycle causes problems if they fail because there is no time remaining to spend remaining funds. In order to take advantage of this scenario, we need to consider the delay option.

5.4 Including the Delay Option without a Delay Cost

In this section, the decisions under a high budget are analyzed when we have the delay option without a delay cost. Table 8 shows that we do not stop project 1 when we have the delay option, but we are indifferent between delaying and stopping project 1 until $t = 11$ where it is terminated. Without a cost, it is clear that delay decisions are not discouraged and more options remain open.

Table 8: Periodic decisions for a high budget level ($100) without a delay cost.

t	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
	All successful	Project 7 fails	Project 5 fails	Project 7 and 5 fail
0	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)
1	**Delay/Stop 1**	**Delay/Stop 1**	**Delay/Stop 1**	**Delay/Stop 1**
2	Reject New	Reject New	Reject New	Reject New
3	Delay/Stop 1	Delay/Stop 1	Delay/Stop 1	Delay/Stop 1
4	Continue 4 ($87)	Continue 4 ($87)	Continue 4 ($87)	Continue 4 ($87)
5	Continue 9, Delay/Stop 1, Reject New ($81)	Continue 9, Delay/Stop 1, Reject New ($81)	Continue 9, Delay/Stop 1, Reject New ($81)	Continue 9, Delay/Stop 1, Reject New ($81)
6	Launch 11 ($80)	Launch 11 ($80)	Launch 11 ($80)	Launch 11 ($80)
7	Delay/Stop 1 ($80)	Delay/Stop 1 ($80)	Delay/Stop 1 ($80)	Delay/Stop 1 ($80)
8	Continue 3, Reject New ($68)	Continue 3, Reject New ($68)	Continue 3, Reject New ($68)	Continue 3, Reject New ($68)
9	Continue 5, Delay/Stop 1 ($50)	Continue 5, Delay/Stop 1 ($50)	Stop 5, Delay/Stop 1 ($68)	Stop 5, Delay/Stop 1 ($68)
10	Continue 7 ($2)	Stop 7 ($50)	Continue 7 ($20)	Stop 7 ($68)
11	Stop 1,2, Reject New ($2)	Stop 1, Continue 2, Accept New ($20)	Stop 1, Continue 2, Reject New ($8)	Stop 1, Continue 2, Accept New ($38)

6 Example 2: Different Review Times

The example introduced in Section 4 is revisited while considering different project review times, hence a different initial portfolio composition. Table 9 defines the projects by stages, review times, probabilities of success, and estimated returns if launched. We assume that projects may be delayed for 4-period increments and we can leave a project delayed at the final period.

Table 9: Data for initial project set for Example 2.

Project (i)	**Stage** (j)	**Renewal Time**	R_i(\$)	p_{ij}^S
1	1	15	1500	0.48, 0.51, 0.65, 0.78, 0.85, 1
2	2	0	1500	0.55, 0.65, 0.80, 0.90, 1
3	2	3	1500	0.5, 0.6, 0.75, 0.88, 0.95
4	3	10	1500	0.62, 0.76, 0.85, 0.98
5	3	2	1500	0.68, 0.75, 0.89, 1
6	4	7	1500	0.77, 0.88, 1
7	4	5	1500	0.8, 0.9, 1
8	5	4	1550	0.85, 1
9	5	16	1550	0.82, 0.98
10	5	11	1600	0.85, 1
11	6	5	1600	0.95
New Project	-	-	1600	0.48, 0.56, 0.68, 0.75, 0.89, 1

Table 10 provides a summary of our results when solving the SDP optimally over horizons of $T = 1$ through $T = 12$. Because of the review time changes, we make funding decisions at different times. Interestingly, the time zero decision changes for $T = 1$ through $T = 4$, but remains consistent for $T = 5$ through $T = 12$. This is due to the delay option holding a project for four periods in this example and further illustrates the benefit of solving the SDP over as long a horizon as possible. When we solve it over a short horizon, we incorporate more approximation and lose key problem characteristics, so even small value changes result in decision changes.

As before, we consider three different cases for three different scenarios. Table 11 shows the case where we have no delay option. In this case, some initial projects are terminated due to the scarcity of resources. As seen in the first scenario, we choose project 7 over project 6 and project 2 over project 4. Project 7 arrives earlier so it makes sense to fund it at time 5 since we are not sure about the success of the project 6. On the other hand, we choose to fund project 2 at time zero by considering the trade-off between the low-value of project 2, which has successfully

Table 10: Time zero decisions and values for different horizons for Example 2.

T	**Reachable States**	S_{sample}	**Approx Time** (hr)	V_0 **(95% C.I.)**	**Time Zero Decision**	P	P'
1	6	6	2.99	6434 (6008-6850)	**Cont. 2**, Reject New	0.99	0.85
2	6	6	3.19	6005 (5504-6506)	**Cont. 2**, Reject New	0.99	1
3	36	36	12.3	5674 (5224-6124)	**Delay 2**, Reject New	0.99	0.8
4	108	36	10.9	5585 (5262-5898)	**Stop 2**, Reject New	0.99	0.9
5	540	72	15.4	6109 (5837-6381)	Delay 2, Reject New	0.99	1
6	5,698	106	19.2	5880 (5532-6227)	Delay 2, Reject New	0.99	1
7	9,404	106	18.74	6657 (6337-6976)	Delay 2, Reject New	0.98	0.85
8	38,564	120	17.54	6604 (6260-6948)	Delay 2, Reject New	0.99	1
9	156,208	120	17.42	6865 (6566-7165)	Delay 2, Reject New	0.99	0.9
10	246,142	120	17.5	6214 (5931-6497)	Delay 2, Reject New	0.99	0.95
11	924,502	130	18.3	6040.6 (5711-6369)	Delay 2, Reject New	0.99	1
12	9,675,884	150	20.59	6441 (6089-6794)	Delay 2, Reject New	0.99	1

completed its stage, and the high-value of project 4 which arrives at time 10 and we are not sure about its stage success. This is an interesting yet important result which shows the importance of capturing the sensitive relation between decision options. When we consider scenario 2 where project 7 fails at time 5, we fund project 6 at time 7. On the other hand, if both of these projects fail as in scenario 3, we now fund project 4 at time 10 and accept two new projects at periods 8 and 11.

Table 12 shows the second case where we consider the delay option with a large delay penalty.

Table 11: Periodic decisions (budget of $100) without the delay option.

t	**Scenario 1**	**Scenario 2**	**Scenario 3**
	All successful	Project 7 fails	Projects 6 and 7 fail
0	Continue 2, Reject New ($88)	Continue 2, Reject New ($88)	Continue 2, Reject New ($88)
1	Do Nothing ($88)	Do Nothing ($88)	Do Nothing ($88)
2	Continue 5, Reject New ($70)	Continue 5, Reject New ($70)	Continue 5, Reject New ($70)
3	Stop 3 ($70)	Stop 3 ($70)	Stop 3 ($70)
4	Continue 8 ($64)	Continue 8 ($64)	Continue 8 ($64)
5	Launch 11, Continue 7, Reject New ($15)	Launch 11, **Stop 7**, Reject New ($63)	Launch 11, **Stop 7**, Reject New ($63)
6	Do Nothing ($15)	Do Nothing ($63)	Do Nothing ($63)
7	Stop 6 ($15)	**Continue 6** ($15)	**Stop 6** ($63)
8	Reject New ($15)	Reject New ($15)	**Accept New** ($45)
9	Do Nothing ($15)	Do Nothing ($15)	Do Nothing ($45)
10	Stop 4 ($15)	Stop 4 ($15)	**Continue 4** ($27)
11	Continue 10, Reject New ($9)	Continue 10, Reject New ($9)	Continue 10, **Accept New** ($3)

Table 12: Periodic decisions (budget of $100) with a large delay penalty.

t	**Scenario 1**	**Scenario 2**	**Scenario 3**
	All successful	Project 7 fails	Projects 6 and 7 fail
0	Continue 2, Reject New ($88)	Continue 2, Reject New ($88)	Continue 2, Reject New ($88)
1	Do Nothing ($88)	Do Nothing ($88)	Do Nothing ($88)
2	Continue 5, Reject New ($70)	Continue 5, Reject New ($70)	Continue 5, Reject New ($70)
3	**Delay 3** ($70)	**Delay 3** ($70)	**Delay 3** ($70)
4	Continue 8 ($64)	Continue 8 ($64)	Continue 8 ($64)
5	Launch 11, Continue 7, Reject New ($15)	Launch 11, Stop 7, Reject New ($63)	Launch 11, Stop 7, Reject New ($63)
6	Do Nothing ($15)	Do Nothing ($63)	Do Nothing ($63)
7	**Delay 3,6** ($15)	**Delay 3**, Continue 6 ($15)	**Continue 3**, Stop 6 ($51)
8	Reject New ($15)	Reject New ($15)	Accept New ($33)
9	Do Nothing ($15)	Do Nothing ($15)	Do Nothing ($33)
10	**Delay 4** ($15)	**Delay 4** ($15)	Continue 4 ($15)
11	**Delay 3,6,** Continue 10, Reject New ($9)	**Delay 3,** Continue 10, Reject New ($9)	Continue 10, Reject New ($9)

Table 13: Periodic decisions (budget of $100) with the delay option and a low delay penalty.

t	**Scenario 1**	**Scenario 2**	**Scenario 3**
	All successful	Project 7 fails	Project 6 and 7 fail
0	**Delay 2**, Reject New ($100)	**Delay 2**, Reject New ($100)	**Delay 2**, Reject New ($100)
1	Do Nothing ($100)	Do Nothing ($100)	Do Nothing ($100)
2	**Delay 5**, Reject New ($100)	**Delay 5**, Reject New ($100)	**Delay 5**, Reject New ($100)
3	**Delay 3** ($100)	**Delay 3** ($100)	**Delay 3** ($100)
4	Continue 8, Delay 2 ($94)	Continue 8, Delay 2 ($94)	Continue 8, Delay 2 ($94)
5	Launch 11, **Delay 7**, Reject New ($93)	Launch 11, Stop 7, Reject New ($93)	Launch 11, Stop 7, Reject New ($93)
6	Continue 5 ($75)	Continue 5 ($75)	Continue 5 ($75)
7	**Delay 3**, **Continue 6** ($27)	**Delay 3**, Continue 6 ($27)	**Continue 3**, Stop 6 ($63)
8	**Delay 2**, Reject New($27)	**Delay 2**, Reject New ($27)	**Continue 2**, Accept New ($33)
9	Delay 7 ($27)	Do Nothing ($27)	Do Nothing ($33)
10	Continue 4 ($9)	Continue 4 ($9)	Continue 4 ($15)
11	**Delay 3**, Continue 10, Reject New ($3)	**Delay 3**, Continue 10, Reject New ($3)	Continue 10, Reject New ($9)

Here, we delay projects 3, 6 and 4 and leave them delayed at the final period under the first scenario. Under the second scenario, we continue project 6 at time 7 when we see project 7 fails at time 5. For the third scenario, we continue project 3 at time 7 and project 4 at time 10 and we accept a new project at time 8 after we see both projects 6 and 7 have failed.

Table 13 shows the third case which considers the delay option with a small delay penalty. In this case, we delay projects 2 and 7 in addition to delaying project 3. Now, we delay project 2 at time zero and wait to see what happens to project 4. When we see project 4 is successful at time 10, we continue project 4 and leave project 2 delayed. We delay project 7 at time 5 because we know it is successful and if we delay it, we may be able to continue project 6. Here, the decision is interesting because we delay the early arrival project and fund the late arrival when these two projects are at similar stages. Under scenario 3 where projects 6 and 7 fail, we fund the delayed project 2 at time 8. This further validates the need to solve the SDP over a long horizon.

7 Example 3: Different Expected Returns

The previous example considered similar projects and returns in the portfolio. In this example, we assume heterogenous projects with different expected returns. For this reason, we can no longer define the state by the number of projects in each stage and we need to consider each project individually. This also results in using an individual project based regression equation to estimate end period state values. As expected, this increases the number of states sampled which will be estimated directly by using simulation and solving the deterministic DP.

Table 14: Data for initial project set for Example 3.

Project (i)	**Stage** (j)	**Renewal Time**	$R_i(\$)$	p_{ij}^S
1	1	0	1800	0.48, 0.51, 0.65, 0.78, 0.85, 1
2	1	4	700	0.47, 0.55, 0.65, 0.80, 0.90, 1
3	2	2	450	0.5, 0.6, 0.75, 0.88, 0.95
4	2	6	950	0.53, 0.62, 0.76, 0.85, 0.98
5	3	11	1000	0.68, 0.75, 0.89, 1
6	4	9	1300	0.77, 0.88, 1
7	4	14	2050	0.8, 0.9, 1
8	4	0	600	0.75, 0.85, 1
9	5	10	1750	0.82, 0.98
10	5	16	1020	0.85, 1
11	6	5	500	0.95

Table 15: Data for possible new arrival project set for Example 3.

Project (i)	**Arrival Time**	$R_i(\$)$	p_{ij}^S
12	2	700	0.48, 0.56, 0.68, 0.75, 0.89, 0.95
13	5	2100	0.48, 0.56, 0.68, 0.75, 0.89, 0.95
14	8	1000	0.48, 0.56, 0.68, 0.75, 0.89, 0.95
15	11	1350	0.48, 0.56, 0.68, 0.75, 0.89, 0.95

Table 14 defines the projects by stages, review times and estimated returns if launched. Table 15 shows the data of potentially new projects. Again in this example, we assume that projects may be delayed for four period increments and we can leave a project delayed at the final period.

We again followed the approximation method described in Section 4. Table 16 provides a summary of our results when solving the SDP optimally over horizons of $T = 1$ through $T = 12$. As a result of this analysis, we obtained the following fit equation for the 12 horizon case:

$$\begin{aligned} V_T(S) = {} & 2224 + 88P_1 + 82P_2 + 100P_3 + 68P_4 + 256P_5 + 754P_6 + 535P_8 + 1520P_9 - 71P_{12} \\ & +72P_{13} - 128P_{14} + 85P_{15} + 75P_{1d} + 31P_{2d} + 3P_{3d} + 110P_{4d} + 383P_{5d} + 161P_{6d} \\ & +426P_{8d} + 1508P_{9d} \end{aligned} \tag{5}$$

Here, P_i refers to project i with "d" referring to a delayed project.

As seen in Table 16, the best decision does not change from $T = 1$ to $T = 12$, but the second best decision does change. Additionally, the values of P and P' increase with T, providing more information to the decision maker.

For this example, we considered three different cases for two different scenarios. Table 17 shows the case where we have no delay option. In this case, we stop projects 1, 2 and 4. We continue project 3, which has a lower expected return then project 4. The possible reason for this decision is the early arrival of project 3 and we do not take the risk and wait for the higher return project 4, which may fail when we reach time 6. Under scenario 2 where project 6 fails, our only option is to accept a new project at time 11 since we learn of the failure at time 9.

Table 18 shows the second case where we consider the delay option. With this option and a delay penalty, we delay projects 1, 2, 4 and 5 and leave them delayed at the final period. The interesting decision is accepting the new arrival at time 5 which has a high return value. We accept it and delay other projects which have higher probabilities of a return. Another interesting result is that delaying projects 4 and 5 has a higher value than continuing. As seen in the second scenario, even if project 6 fails, we still keep projects 4 and 5 delayed and accept a new project at time 11. Without a delay penalty, there is no change in the decisions for these two scenarios because we do not continue any project that was delayed before the horizon.

8 Considering Project Interactions

Our SDP model does not consider any interaction effects that may result from product releases over time. Clearly, if this were a pipeline of similar products, there may be concerns of hurting one's own market share. For example, "next generation" products should not be released immediately after each other.

To allow better control of product releases, we introduce an interaction factor. This may be viewed similarly to a covariance matrix in that returns are increased or decreased due to

Table 16: Time zero decisions and values for different horizons for Example 3.

T	**Reachable States**	S_{sample}	**Approx Time** (hr)	V_0 **(95% C.I.)**	**Time Zero Decision**	P	P'
1	9	9	3.27	4696 (4342-5050)	(1) Delay 1, Cont 8 (2) **Stop 1, Cont. 8**	0.9	0.65
2	9	9	3.18	4580 (4206-4954)	(1)Delay 1, Cont. 8 (2) **Stop 1, Cont. 8**	0.99	1
3	54	54	8.4	4802 (4464-5140)	(1) Delay 1, Cont. 8 (2) **Stop 1, Delay 8**	0.99	0.85
4	54	54	8.8	5105 (4683-5527)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	1
5	270	60	9.2	4475 (4120-4830)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	1
6	1,792	60	10.3	5283 (4795-5750)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	0.8
7	8,904	60	10.9	4578 (4071-5085)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	1
8	8,904	60	12.3	4981 (4561-5401)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	0.85
9	39,896	60	10.6	4558 (4296-4820)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	1
10	134,666	128	19.2	4882 (4504-5260)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	1
11	909,966	140	21.69	4815 (4495-5135)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	1
12	4,357,272	287	46.47	4263 (3933-4590)	(1) Delay 1, Cont. 8 (2) Delay 1, Delay 8	0.99	1

Table 17: Periodic decisions (budget of $100) without the delay option.

t	**Scenario 1**	**Scenario 2**
	All successful	Project 6 fails
0	Continue 8, Stop 1 ($94)	Continue 8, Stop 1 ($94)
1	Do Nothing ($94)	Do Nothing ($94)
2	Continue 3, Reject New ($82)	Continue 3, Reject New ($82)
3	Do Nothing ($82)	Do Nothing ($82)
4	Stop 2 ($82)	Stop 2 ($82)
5	Launch 11, Reject New ($81)	Launch 11, Reject New ($81)
6	Stop 4 ($81)	Stop 4 ($81)
7	Do Nothing ($81)	Do Nothing ($81)
8	Reject New ($81)	Reject New ($81)
9	Continue 6 ($33)	**Stop 6** ($81)
10	Continue 9 ($27)	Continue 9 ($75)
11	Continue 5, Reject New ($9)	Continue 5, **Accept New** ($39)

Table 18: Periodic decisions (budget of $100) with the delay option.

t	**Scenario 1**	**Scenario 2**
	All successful	Project 6 fails
0	Continue 8, **Delay 1** ($94)	Continue 8, **Delay 1** ($94)
1	Do Nothing ($94)	Do Nothing ($94)
2	Continue 3, Reject New Project ($82)	Continue 3, Reject New Project ($82)
3	Do Nothing ($82)	Do Nothing ($82)
4	**Delay 1,2** ($82)	**Delay 1,2** ($82)
5	Launch 11, Accept New ($63)	Launch 11, Accept New ($63)
6	**Delay 4** ($63)	**Delay 4** ($63)
7	Do Nothing ($63)	Do Nothing ($63)
8	**Delay 1,2** and Reject New ($63)	**Delay 1,2** and Reject New ($63)
9	Continue 6 ($15)	Stop 6 ($63)
10	**Delay 4** and Continue 9 ($9)	**Delay 4** and Continue 9 ($57)
11	**Delay 5** and Reject New ($9)	**Delay 5** and **Accept New** ($39)

interaction effects. Thus, we penalize releases that are close together. This is accomplished with a factor, ρ_{ij}, which decreases the expected return of project i based on its "distance" in time from project j.

We update $V_t(S)$ as follows:

$$V_t(S) = \sum_{\psi_t \subseteq \Psi_t} p_t(\psi_t) \left\{ \max_{\delta_t \subseteq \psi_t : C(\delta_t) \leq B_t} \left(R'(\delta_t) - C(\delta_t) - C_d(\delta_t) + \alpha V_{t+1}(S'(\delta_t)) \right) \right\}, t = 0, 1, 2, \ldots, T. \quad (6)$$

$R'(\delta_t)$ is the new expected return which incorporates the interaction penalty. The expected return of projects in the pipeline are updated whenever we make a launch decision. For example, if we launch project k, we update the return of all other projects i in the pipeline as follows:

$$R'(i) = \rho_{ij} R'(i), \forall i \in A_t/k \quad (7)$$

A_t is the set consisting of all projects in the pipeline at time t and k is the project which is launched at time t. Here, we apply a penalty to the return of the other projects according to the interaction factors determined by considering the release time difference from the launched project. For example, if project 1 is at the final stage but has 6 months before launch and we are releasing project 2 now, then the expected return of project 1 is reduced by multiplying it with the interaction factor (value between 0 and 1). $R'(\delta_t)$ is merely the sum of returns from launched projects in δ_t. Note that these returns are previously adjusted due to earlier releases. We illustrate in the following examples.

8.1 Revisiting Example 1 with Interaction Effects

We revisit Example 1 and assign the following interaction factors:

Release Time Difference	0-2	3-4	5-6	7-8	9-10	11-12	13-14	15-16	17-22	>23
ρ	0.7	0.73	0.76	0.79	0.8	0.85	0.9	0.95	0.99	1

To interpret this data assume a project is launched. If we have a project which is two months away from a launched project, its expected return will be reduced by 0.3. The expected return of projects which have a distance greater than 23 periods are not affected. We solved this example over 12 periods with our approximation technique.

Table 19 presents the solutions. In contrast to Example 1, the interaction effects lead to stopping project 5 in period 9 while continuing project 2 in order to spread out the pipeline.

8.2 Example 4 with Interaction Effects

In this example, we use a new portfolio composition which is less diversified over time as compared to Example 1. There are three projects in stage 1, three projects in stage 2, three projects in

Table 19: Periodic decisions ($100), no delay option, and interactions.

t	**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
	All successful	Project 7 fails	Project 5 fails	Project 7 and 5 fail
0	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)	Launch 10, Reject New ($99)
1	Stop1 ($99)	Stop 1 ($99)	Stop 1 ($99)	Stop1 ($99)
2	Reject New ($99)	Reject New ($99)	Reject New ($99)	Reject New ($99)
3	Do Nothing ($99)	Do Nothing ($99)	Do Nothing ($99)	Do Nothing ($99)
4	Continue 4 ($87)	Continue 4 ($87)	Continue 4 ($87)	Continue 4 ($87)
5	Continue 9, Reject New ($81)	Continue 9, Reject New ($81)	Continue 9, Reject New ($81)	Continue 9, Reject New ($81)
6	Launch 11 ($80)	Launch 11 ($80)	Launch 11 ($80)	Launch 11 ($80)
7	Do Nothing ($80)	Do Nothing ($80)	Do Nothing ($80)	Do Nothing ($80)
8	Continue 3, Reject New ($68)	Continue 3, Reject New ($68)	Continue 3, Reject New ($68)	Continue 3, Reject New ($68)
9	**Stop 5** ($68)	**Stop 5** ($68)	**Stop 5** ($68)	**Stop 5** ($68)
10	Continue 7 ($20)	Stop 7 ($68)	Continue 7 ($20)	Stop 7 ($68)
11	**Continue 2**, Reject New ($8)	Continue 2, Accept New ($38)	Continue 2, Reject New ($8)	Continue 2, Accept New ($38)

stage 3 and two projects in stage 4. Table 20 provides the data for this example.

We solve this example both with and without interaction effect over 12 periods for an initial budget of $75. We assume the following factors:

Release Time Difference	0-2	3-4	5-6	7-8	9-10	11-12	13-14	15-16	17-18	19-23	23-31	>31
ρ	0.6	0.63	0.69	0.72	0.76	0.79	0.83	0.89	0.94	0.97	0.99	1

From the approximation, we obtain two different regression equations for no-interaction and interaction cases.

(1) No-interaction:

$$V_T(S) = 2457 + 25.3n_1 + 118n_2 + 186.1n_3 + 369.8n_4 + 0n_5 + 0n_6 \tag{8}$$

In order to capture the effect of interaction between same stage projects, we update our regression equation and use individual project based regression equation.

Table 20: Data for initial project set for Example 4

Project (i)	**Stage** (j)	**Renewal Time**	$R_i(\$)$	p_{ij}
1	1	10	1500	0.48, 0.51, 0.65, 0.78, 0.85, 1
2	1	6	1500	0.42, 0.55, 0.65, 0.80, 0.90, 1
3	1	2	1500	0.4, 0.5, 0.6, 0.75, 0.88, 0.95
4	2	4	1500	0.53, 0.62, 0.76, 0.85, 0.98
5	2	2	1500	0.6, 0.68, 0.75, 0.89, 1
6	2	0	1500	0.6, 0.68, 0.77, 0.88, 1
7	3	8	1500	0.7, 0.8, 0.9, 1
8	3	2	1550	0.64, 0.75, 0.85, 1
9	3	1	1550	0.6, 0.7, 0.82, 0.98
10	4	23	1600	0.75, 0.85, 1
11	4	19	1600	0.74, 0.82, 0.95

(2) Interaction:

$$\begin{aligned} V_T(S) &= 1604 + 45P_1 + 30P_2 - 21P_3 + 96P_4 + 239P_5 + 89P_6 - 45P_7 + 214P_8 + 89P_9 \\ &+30P_{12} + 35P_{13} + 111P_{14} + 43P_{15} - 11P_{16} + 0.8P_{1d} + 5.6P_{2d} - 4.775P_{3d} + 13P_{4d} \\ &+28P_{5d} + 19P_{6d} + 367P_{7d} + 367P_{8d} + 286P_{9d} \end{aligned} \tag{9}$$

Note that projects in similar stages have drastically different values, despite having similar returns (although different review times and probabilities of success).

Table 21 presents the results with and without delay. When there is no delay option, we stop projects 1, 2, 3 and 7 because we do not have enough funds. As a result, we continue five projects (three from stage 2 and two from stage 3). We have this same result for both no-interaction and interaction cases under this scenario when there is no delay option. On the other hand, with the delay option, the interaction factor changes the decisions. When there is no interaction effect, we delay projects 1, 2, 3, 4, and 5, but we continue projects 6, 7, 8, and 9. As seen, we keep our budget for late stage projects. However, when we have interaction effects, we accept two new projects at periods 2 and 5, delay project 2, delay stage 4 projects while funding three stage 3 projects. This is to diversify the portfolio. The critical differences are accepting new projects into the pipeline and delaying the late stage projects with interaction factors.

Table 21: Periodic decisions with and without interaction effects.

t	**No Interaction**	**Interaction**	**No Interaction**	**Interaction**
	No Delay	No Delay	With Delay	With Delay
0	Continue 6, Reject New ($63)	Continue 6, Reject New ($63)	Continue 6, Reject New ($63)	Continue 6, Reject New ($63)
1	Continue 9 ($45)	Continue 9 ($45)	Continue 9 ($45)	**Delay 9** ($63)
2	Stop 3, Continue 5,8, Reject New ($15)	Stop 3, Continue 5,8, Reject New ($15)	**Stop 3, Delay 5,** Continue 8, Reject New ($27)	Continue 5, **Delay 3,8, Accept New** ($33)
3	Do Nothing ($15)	Do Nothing ($15)	Do Nothing ($27)	Do Nothing ($33)
4	Continue 4 ($3)	Continue 4 ($3)	Delay 4 ($27)	**Continue 4** ($21)
5	Reject New ($3)	Reject New ($3)	Reject New ($27)	Accept New ($3)
6	Stop 2 ($3)	Stop 2 ($3)	Delay 2 ($27)	Delay 2 ($3)
7	Do Nothing ($3)	Do Nothing ($3)	Do Nothing ($27)	Do Nothing ($3)
8	Stop 7, Reject New ($3)	Stop 7, Reject New ($3)	Continue 7, Delay 3,5, Reject New ($9)	**Delay 7,8,** Reject New ($3)
9	Do Nothing ($3)	Delay 9 ($3)	Do Nothing ($9)	Do Nothing ($3)
10	Stop 1 ($3)	Stop 1 ($3)	Delay 1 ($9)	Stop 1 ($3)
11	Reject New ($3)	Reject New ($3)	Reject New ($9)	Reject New ($3)

9 Conclusions

We model the portfolio management problem for multi-stage investments, in which new projects must be accepted or rejected and projects in the pipeline must be continued, delayed, or abandoned, with stochastic dynamic programming. These decisions are often associated with R&D projects. The model becomes intractable for large-scale problems, especially those over long horizon times. However, a long solution horizon is required to ensure good time zero decisions, as our goal is to maximize the present value of expected returns – which often occur in a distant future.

To increase the horizon of analysis, we present an approximate solution method in which the value function is estimated for all states at some horizon time T using simulation, deterministic dynamic programming, and regression analysis. These estimates are then used to solve the SDP optimally over the T-period horizon. In addition to the time zero solution (value function estimate), our approach provides two estimates of the probability of selecting the optimal decision at time zero.

We provide numerous examples to illustrate insights provided by the model, including how decisions change with differing budgets, delay options (lengths, costs, and penalties), project returns, initial portfolios, and an inclusion of interaction effects which penalize projects that are "too close to each other" in the pipeline. Our results suggest that budget dollars tend to be saved for late-stage projects, as they are more likely to achieve a return. Thus, the timing of reviews is critical because if projects fail late in the budget cycle, the opportunity to invest in additional projects over time may be lost.

It is clear from our results that the value of the delay option increases dramatically if late-stage investment projects are reviewed late in the budget cycle. The real options literature has clearly illustrated the worth of the delay option. However, this work has further illustrated the increased worth of the delay option in the context of making portfolio investment decisions over time, for which real options are not designed to analyze. It should be noted that the delay option becomes even more critical in light of lower budgets and/or the incorporation of interaction effects.

We believe this is a significant step in the dynamic analysis of portfolios of multi-stage investment projects. However, there is much research to be explored.

In this study, we were concerned with an appropriate model of the decision process with a focus on solving large-scale instances in order to ensure good time zero decisions. Thus, we assumed that the data even though probabilistic was known with certainty. As we are concerned with time zero decisions, we assumed that we could update the data each period and re-run the model. However, to more appropriately model the decision process, additional scenarios of testing outcomes may need to be incorporated into the model. Furthermore, systematic ways in which to update these outcomes should lead to better decisions.

Computationally, it may be necessary to solve larger portfolio problems. We believe the current approximation technique is well suited for changes in the review process (numbers of stages and possible outcome scenarios), as the simulation approach reduces the number of paths to analyze. However, the approximation technique still relies on solving the combinatorial problem of investment decisions in each period. Thus, in order to solve larger instances (more projects in a portfolio, more arrivals, or problems with larger budgets), further approximations which reduce this decision set are required. The use of linear programming approximations, as opposed to dynamic programming, for these instances may prove useful.

References

[1] D.I. Angelis. Capturing the option value of R&D. *Research Technology Management*, 43(4):31–34, 2000.

[2] J.F. Bard, R. Balachandre, and P.E Kaufmann. An interactive approach to R&D project selection and termination. *IEEE Transactions on Engineering Management*, 35:139–146, 1988.

[3] P.P. Belobaba. Application of a probabilistic decision model to airline seat inventory control. *Operations Research*, 37:183–197, 1989.

[4] S. Benninga and E. Tolkowsky. Real options – an introduuction and an application to R&D valuation. *The Engineering Economist*, 47(2):151–168, 2002.

[5] D.P. Bertsekas. Dynamic programming and suboptimal control: A survey from adp to mpc. *European Journal of Control,*, 11(4).

[6] D.P. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996.

[7] D. Bertsimas and R. Demir. An approximate dynamic programming approach to multidimensional knapsack problems. *Management Science*, 48(4):550–565, 2002.

[8] G. Blau. A system engineering approach to new product development. *CAST Communications*, 21(2), 1997.

[9] G. Blau, B. Mehta, S. Bose, J. Pekny, G. Sinclair, K. Keunker, and K. Bunch. Risk management in the development of new products in highly regulated industries. *Computers and Chemical Engineering*, 24:659–664, 2000.

[10] E.H. Bowman and G.T. Moskowitz. Real options analysis and strategic decision making. *Organizational Science*, 12(6):772–777, 2001.

[11] S.L. Brumelle and J.I. McGill. Air seat allocation with multiple fare nested fare classes. *Operations Research*, 41:127–137, 1993.

[12] R.L. Caraway and L.R. Schmidt, R.L.and Weatherford. An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns. *Naval Research Logistics*, 40(2):161–173, 1993.

[13] R.L. Caraway and R.L. Schmidt. An improved discrete dynamic programming algorithm for allocating resources among interdependent projects. *Management Science*, 37(9):1195–1199, 1991.

[14] P.D. Childs, S.H. Ott, and A.J. Triantis. Capital budgeting for interrelated projects: A real options approach. *The Journal of Financial and Quantitative Analysis*, 33(3):305–334, 1998.

[15] P.D. Childs and A.J. Triantis. Dynamic R&D investment policies. *Management Science*, 45(10):1359–1377, 1999.

[16] J. Choi, M.J. Realff, and J.H. Lee. Dynamic programming in a heuristically confined state space: a stochastic resource-constrained project scheduling application. *Computers and Chemical Engineering*, 28:1039–1058, 2004.

[17] Y.H. Chun. Sequential decisions under uncertainty in the R&D project selection problem. *IEEE Transaction on Engineering Management*, 41:404–413, 1994.

[18] S. Das and D. Ghosh. . 0-1 binary knapsack with random budgets. *Technical report*, 2001.

[19] D.F. de Farias and B. Van Roy. On consraint sampling in linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

[20] D.P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51:850–865, 2003.

[21] B.V. Dean and M.J. Nishry. Scoring and profitability models for evaluating and selecting engineering projects. *Operations Research*, 13(4):550–569, 1965.

[22] M.W. Dickinson, A.C. Thornton, and S. Graves. Technology portfolio management: optimizing interdependent projects over multiple time periods. *IEEE Transaction on Engineering Management*, 48(4):518–527, 2001.

[23] T.W. Faulkner. Apply options thinking to R&D valuation. *Research Technology Management*, 39(3):50–56, 1996.

[24] E. Foufoula-Georgiou. Conves interpolation for gradient dynamic programming. *Water Resource Res.*, 27(1):31–36, 1991.

[25] G.E. Fox, N.R. Baker, and J.L. Bryant. Economic models for R&D project selection in presence of project interactions. *Management Science*, 30(7):890–902, 1984.

[26] P. Freeman and A.E. Gear. A probabilistic objective functions for R&D portfolio selection. *Operations Research Quarterly*, 22(3):263–265, 1971.

[27] F. Ghasemzadeh, N.P. Archer, and P. Iyogun. A zero-one model for project portfolio selection and scheduling. *Journal of Operational Research Society*, 50(7):745–755, 1999.

[28] G. Godfrey and W.B. Powell. An adaptive, dynamic programming algorithm for stochastic resource allocation problems i:single period travel times. *Transportation Science*, 36:21–39, 2002.

[29] K. Golabi. Selecting a group dissimilar projects for funding. *IEEE Transaction on Engineering Management*, 34:138–145, 1987.

[30] S. M. Graves, J.L. Ringuest, and R.H. Case. Formulating optimal R&D portfolios. *Research Technology Management*, 43(3):47–51, 2000.

[31] J.C. Hartman and T.C. Perry. Approximating the solution of a dynamic, stochastic, multiple knapsack problem. *Journal of Control and Cybernetics*, in press.

[32] K. Heidenberger. Dynamic project selection and funding under risk: A decision tree based milp approach. *European Journal of Operations Research*, 95:284–298, 1996.

[33] M. Henig. Risk criteria in a stochastic knapsack problem. *Operations Research*, 38:820–825, 1990.

[34] A.D. Henriksen and A.J. Traynor. A practical R&D project-selection scoring tool. *IEEE Transaction on Engineering Management*, 46(2):158–170, 1999.

[35] H.S.B. Herath and C.S. Park. Economic analysis of R&D projects: An options approach. *The Engineering Economist*, 44(1):1–35, 1999.

[36] Z.S. Hua, B. Zhang, and L. Liang. An approximate dynamic programming approach to convex quadratic knapsack problems. *Computers and Operations Research*, 33:660–673, 2005.

[37] S.A. Johnson, C.A. Shoemaker J.R. Stedinger, Y. Li, and J.A. Tejada-Guibert. Numerical solutions of continuous-state dynamic progams using linear and spline interpolation. *Operations Research*, 41:484–500, 1993.

[38] D.H. Jones and F.A. Mendez. Strategic resource-allocation decision in pharmaceutical research and development. In *Proceedings of the North Eastern Decision Sciences Institute Conference*, pages 1–5, Rhode Island, 2003.

[39] A.J. Kleywegt and J.D. Papastavrou. The dynamic and stochastic knapsack problem. *Operations Research*, 46:17–35, 1998.

[40] A.J. Kleywegt and J.D. Papastavrou. The dynamic and stochastic knapsack problem with random sized items. *Operations Research*, 49(1):26–41, 2001.

[41] D.F. Kocaoglu and M.G. Iyigun. Strategic R&D project selection and resource allocation with decision support system application. *In Proceedings 1994 IEEE international Engineering Management Conferences*, pages 225–232, 1994.

[42] F. Krawiec. Evaluating and selecting research projects by scoring. *Research Management*, 27(2):22–26, 1984.

[43] T.S. Lee and M. Hersh. A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Science*, 27:252–265, 1993.

[44] M.J. Liberatore. An extension of analytical hierarchy process for industrial R&D project selection and resource allocation. *IEEE Transaction on Engineering Management*, 34:12–18, 1987.

[45] M.J. Liberatore. A decision support system linking research and development project selection with business strategy. *Project Management Journal*, 49(5):14–21, 1988.

[46] M.J. Liberatore. An expert system for R&D project selection. *Mathematical Computation Modeling*, 11:260–265, 1988.

[47] L.L. Lu, S.Y. Chiu, and L.A. Cox, Jr. Optimal project selection: Stochastic knapsack with finite time horizon. *Journal of Operations Research Society*, 50:645–650, 1999.

[48] I.C. MacMillian and R.G. McGrath. Crafting R&D project portfolios. *Research Technology Management*, 45(5):48–59, 2002.

[49] J. Matheson, M. Menke, and S. Derby. Managing R&D portfolios for improved profitability and productivity. *Journal of Science Policy & Research Management*, 4(4), 1989.

[50] H. Mendelson, J. S. Pliskin, and U. Yechiali. A stochastic allocation problem. *Operations Research*, 28:687–693, 1980.

[51] G.R. Mitchell. Alternative frameworks for technology strategy. *European Journal of Operational Research*, 47:153–161, 1990.

[52] G.R. Mitchell and W.F. Hamilton. Managing R&D as a strategic option. *Research Technology Management*, 31(3):15–22, 1988.

[53] J.R. Moore, Jr. and N.R. Baker. An analytical approach to scoring model design-application to research and development project selection. *IEEE Transaction on Engineering Management*, 16:90–98, 1969.

[54] N.A. Nichols. The new pharmaceuutical paradigm. *Harvard Business Review*, 72:88–105, 1994.

[55] J.D. Papastavrou, S. Rajagopalan, and A.J. Kleywegt. The dynamic and stochastic knapsack problem with deadlines. *Management Science*, 42(12):1706–1718, 1996.

[56] E. Pennings and O. Lint. The option value of advanced R&D. *European Journal of Operational Research*, 103:83–94, 1997.

[57] R. Philbrick and P.K. Kitanidis. Improved dynamic programming methods for optimal control of lumped parameter stochastic systems. *Operations Research*, 49:398–412, 2001.

[58] W.B. Powell. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Science*, 30(3):195–219, 1996.

[59] W.B. Powell and T. Carvalho. Dynamic control of logistics queueing networks for large scale fleet management. *Transportation Science*, 32(2):90–109, 1998.

[60] G.P. Prastacos. Optimal sequential investment decisions under condition of uncertainty. *Management Science*, 29(1):118–134, 1983.

[61] F. Pries, T. Astebro, and A. Obeidi. Economic analysis of R&D projects: Real option versus npv valuation revisited. Working paper, Department of Managmenet Sciences, University of Waterloo, Waterloo, Ontario, Canada, 2001.

[62] R. Righter. Resource allocation in a random environment. *Operations Research*, 37:329–338, 1989.

[63] J.L. Ringuest and S.B. Graves. The linear R&D project selection problem: An alternative to net present value. *IEEE Transaction on Engineering Management*, 37:143–146, 1990.

[64] L.W. Robinson. Optimal approximate control policies for airline booking with sequential nonmonotonic fare classes. *Operations Research*, 43:252–263, 1995.

[65] M.J. Rogers, A. Gupta, and C.D. Maranas. Real options for optimal pharmaceutical research and development portfolios. *Ind. Eng. Chem. Res.*, 41:6607–6620, 2002.

[66] K.W. Ross and D.H.K. Tsang. A stochastic knapsack problem. *IEEE Transaction on Communications*, 28:687–693, 1980.

[67] K.W. Ross and D.D. Yao. Monotonicity probabilities of stochastic knapsack. *IEEE Transaction on Information Theory*, 36:1173–1179, 1990.

[68] R.L. Schmidt. A model for project selection with combined benefit, outcome and resource interactions. *IEEE Transaction on Engineering Management*, 40:403–410, 1993.

[69] N. Secomandi. Comparing neuro-dynamic dynamic programming approaches for the vehicle routing problem with stochastic demands. *Computers and Operations Research*, 47:1201–1225, 2000.

[70] N. Secomandi. A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research*, 49(5):796–802, 2001.

[71] R.L. Smith and I.E. Schochetman. Infinite horizon optimization. *Mathematics of Operations Research*, 14:559–574, 1989.

[72] M. Sniedovich. Preference order stochastic knapsack problem: a counter example. *RC 7766 IBM Thomas J. Watson Research Center Yorktown Heights, New York*, 1979.

[73] M. Sniedovich. Preference order stochastic knapsack problem: methodological issues. *Journal of Operations Research Society*, 31(11):1025–1032, 1980.

[74] M.Z. Spivey and W.B. Powell. The dynamic assignment problem. *Transportation Science*, 38(4):399–419, 2004.

[75] E. Steinberg and M.S. Parks. A preference order dynamic program for a knapsack problem with stochastic rewards. *Journal of Operations Research Society*, 30:141–147, 1979.

[76] C. Stummer and K. Heidenberger. Interactive R&D portfolio analysis with project interdependencies and time profiles of multiple objectives. *IEEE Transaction on Engineering Management*, 50(2):175–183, 2003.

[77] D. Subramanian, J. Penky, and G. Reklaitis. A simulation-optimization framework for addressing combinatorial and stochastic aspects of R&D pipeline management problem. *Computers and Chemical Engineering*, 24:1005–1011, 2000.

[78] D. Subramanian, J. Penky, and G. Reklaitis. A simulation-optimization framework for stochastic optimization of R&D pipeline management problem. *AIChE Journal*, 47(10):2226–2241, 2003.

[79] H. Topaloglu and W.B. Powell. Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems. *INFORMS Journal on Computing*, 18(1):31–42, 2006.

[80] L. Trigeorgis. Making use of real options simple: An overview and applications in flexible/modular decision making. *The Engineering Economist*, 50(1):to appear, 2005.

[81] R. Van Slyke and Y. Young. Finit horizon stochastic knapsack with applicatopns to yield management. *Operations Research*, 48:155–172, 2000.

[82] R.D. Wollmer. An airline seat management model with a single leg route when lower fare classes book first. *Operations Research*, 40:26–37, 1992.