ABSTRACT

Title of Document: METHODS FOR EMPLOYING REAL OPTIONS MODELS TO MITIGATE RISK IN R&D FUNDING DECISIONS

Jeremy Michael Eckhause, Ph.D., 2011

Directed By: Associate Professor Steven A. Gabriel, Department of Civil and Environmental Engineering

Government acquisitions requiring research and development (R&D) efforts are fraught with uncertainty. The risks are often mitigated by employing a multi-stage competition, with multiple projects funded initially until a single successful project is selected. While decision-makers recognize they are using a real options approach, analytical tools are often unavailable to evaluate optimal decisions. The use of these techniques for R&D project selection to reduce the uncertainties has been shown to increase overall project value.

This dissertation first presents an efficient stochastic dynamic programming (SDP) approach that managers can use to determine optimal project selection strategies and apply the proposed approach on illustrative numerical examples. While the SDP approach produces optimal solutions for many applications, this approach does not easily accommodate the inclusion of a budget-optimal allocation or side constraints, since its formulation is scenario specific. Thus, we then formulate an

integer program (IP), whose solution set is equivalent to the SDP model, but facilitates the incorporation of these features and can be solved using available commercial IP solvers. The one-level IP formulation can solve what is otherwise a nested two-level problem when solved as an SDP. We then compare the performance of both models on differently sized problems. For larger problems, where the IP approach appears to be untenable, we provide heuristics for the two-level SDP formulation to solve problems efficiently.

Finally, we apply these methods to carbon capture and storage (CCS) projects in the European Union currently under development that may be subject to public funding. Taking the perspective of a funding agency, we employ the real options models presented in this dissertation for determining optimal funding strategies for CCS project selection. The models demonstrate the improved risk reduction by employing a multi-stage competition and explicitly consider the benefits of knowledge spillover generated by competing projects. We then extend the model to consider two sensitivities: 1) the flexibility to spend the budget among the time periods and 2) optimizing the budget, but specifying each time period's allocation *a priori*. State size, scenario reduction heuristics and run-times of the models are provided.

METHODS FOR EMPLOYING REAL OPTIONS MODELS TO MITIGATE RISK IN R&D FUNDING DECISIONS

By

Jeremy Michael Eckhause

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Associate Professor Steven A. Gabriel, Chairman
Professor Shapour Azarm
Assistant Professor Danny R. Hughes
Dr. Peter F. Kostiuk
Associate Professor Elise Miller-Hooks

Dedication

To Marion.

Acknowledgements

First and foremost, I would like to thank my advisor, Professor Steven Gabriel, for the direction, support and encouragement he has supplied me over my years of study at Maryland. In addition to guiding me through my research, Professor Gabriel has provided me a wealth of opportunities in research and teaching. I truly appreciate the camaraderie we have developed over the years, especially while travelling abroad together, and hope to continue our collaboration in the years to come.

I want to thank the other members of my dissertation committee: Professor Shapour Azarm for serving as the dean's representative; Professor Danny Hughes, of the University of Oklahoma, for his assistance, novel research ideas, and ceaseless encouragement; Dr. Peter Kostiuk for his years of support and guidance in this work's practical applications; and Professor Elise Miller-Hooks for her suggestions on extending the applications of the models developed in this thesis.

I would like to thank Johannes Herold and Professor Christian von Hirschhausen of the Berlin University of Technology for their assistance in the development of a relevant and meaningful application of the models presented in this dissertation. Our joint work has been a highlight of my graduate work, and I hope this dissertation reflects positively on that collaboration.

My thanks also go to my long-time friend, Professor Huseyin Topaloglu of Cornell University, for his assistance with several model formulations. I am grateful to my colleagues at the Logistics Management Institute for their encouragement, backing and understanding during my graduate studies.

I would like to thank my parents, Sarah and Morton Eckhause, for their years of moral support and instilling in me the importance of education and research.

Finally, I am most indebted to my wife, Marion Eckhause, for her years of patience, encouragement and love. I am overjoyed at the prospect of spending more time with her, after years of divided attention during my graduate studies.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background and General Real Options Literature

Virtually all government acquisition activities possess some elements of risk and uncertainty. However, the acquisition of new capabilities is particularly perilous, especially when the desired capabilities are significant advances beyond current levels of technology, as is often the case in many modern public-sector and defense acquisitions. These acquisitions frequently require significant research and development (R&D) programs to provide the basic research or technology development and maturation required to produce operational products that deliver the desired capability. In addition to the various cost, schedule, and programmatic risks all government acquisitions face, R&D intensive acquisitions must contend with a higher degree of technical risk. This additional risk is due to broadly defined initial capability or threshold performance levels, changing performance targets during the course of the acquisition as requirements change, insufficient technological maturity to produce the desired capability, or uncertainty regarding the feasibility of any given technological approach. The successful management of technical risk in such long duration, one-of-a-kind R&D acquisitions is crucial for these projects' success.

Real options approaches for managing R&D activities have been shown to increase project value while mitigating the risks associated with the uncertainties inherent in R&D. Dixit and Pindyck's (1994) seminal work outlines the transition from traditional (financial) options to real options with a private sector focus, such as an oil extraction project. Using real option techniques, such as sequential decision making to manage risk, has been well-established in the literature (Trigeorgis, 1996). Perdue, *et al.* (1996), for example, provide a solid analytical framework in which to value real options with sequential decision points in an R&D setting.

However, this dissertation addresses two areas largely ignored in the real options literature. First, we consider how real options can mitigate risk and uncertainty due to variability in project performance and schedule. Most studies of real options valuation techniques in R&D projects have considered risk and uncertainty to occur in the project's market payoff. Second, we consider the value of increased managerial flexibility in a multi-stage, project source selection model for a non-traded public good, which is often difficult to value and often does not permit a program abandonment option. Considering multi-stage development projects where managers can consider continuing, improving, or abandoning development at each decision point, Huchzermeier and Loch (2001) evaluate changes in option values in the presence of five types of operational uncertainty: market payoff variability, budget variability, performance variability, market requirement variability, and schedule variability. They conclude that the value of increased managerial flexibility through the use of real options increases with increased variability in market payoffs and budgets but may actually decrease in the presence of the other types of uncertainty discussed. Building off of the same formulation, Santiago and Vakili (2005) find different results, with uncertainties beyond market payoff providing ambiguous results for the value of increased managerial flexibility. However, in the case of market payoff, they find increasing variability increases either the project value or the project option value. Santiago and Bifano (2005) consider the application of a multidimensional decision tree real options model that considers multiple types of operational uncertainty toward the development of a specific product. They demonstrate how the project could be managed by estimating its value and determining optimal managerial actions taken at each review stage. While they claim their model has applications beyond their case study project, their decision tree model contains the typical "abandon-continue-improve" decisions within one project.

These approaches are complicated when R&D is contracted outside of the firm or governmental agency, either directly or through the acquisition of an R&D intensive item, by reducing the firm's ability to directly address technical risks as they occur. If the firm or agency outsources this work, we typically refer to those groups undertaking the initiative as a "vendor." More generically, we refer to this outsourcing process as a "project," recognizing that each of these projects is not necessarily occurring within the same firm, but rather represents separate projects quite possibly working towards the same technological objective. Moreover, selecting among projects with unverifiable performance outcomes further increases the uncertainty of the technical success of an R&D effort. One approach for managing this additional uncertainty is to employ a multi-stage contract where the first stage serves as a pilot program to: (i) verify project capabilities and (ii) reduce technical risk by assessing the realized outcomes, thereby providing information to the firm regarding the likely success of the project. Snir and Hitt (2004) present a model that helps establish quality vendors by setting a two-stage project in which the pilot project stage's compensation is small enough to only attract quality vendors. Errais and Sadowsky (2008) present a model that values the outcomes at each stage of an N

period pilot project. After this initial screening stage the firm decides whether to continue the project. Often, particularly in public sector applications, such as weapon systems procurement (Rogerson, 1995), the first stage serves as a tournament, with contracts awarded to multiple vendors competing to continue the project into the next stage(s).

While reducing the technical risks associated with R&D projects, multi-stage multi-project competitions pose different challenges to a firm. Specifically, how should projects be selected at each stage? Which project should be funded at each decision point? How many stages before a single winning project is selected? How should funding be spread between stages?

Cao and Wang (2007) present a vendor selection model for two-stage multi-vendor competitions where the first stage reveals the final level of performance to be achieved by each of the competing vendors. Given a fixed budget on the part of the client firm, this model selects the optimal portfolio of vendors to fund in the first stage as well as the amount of resources to dedicate to each stage to maximize the expected benefit to the client firm upon project completion. Their approach is to use an integer programming model that resembles a knapsack problem, where the outcomes are listed in very basic terms (“poor”, “fair”, good” and “excellent”), and by this single criterion of expected benefit. They conclude that the selection in the first stage is more about creating a good portfolio of vendors than simply selecting a few “frontrunners.” In this way, public sector R&D acquisitions are much like the one-sided sequential development process in Roberts and Weitzman’s (1981) seminal work, in which benefits are received after all stages are completed. Like the examples provided by Cao and Wang (2007) and Snir and Hitt (2004) show, many large-scale information technology (IT) projects that are outsourced to third party vendors also follow this development process.

1.2 Differences between Public and Private Sector R&D Acquisitions

Unfortunately, while there exists a robust literature on the use of real options to manage uncertainty in R&D projects, this literature largely focuses on private sector R&D and does not consider the peculiarities of public sector R&D acquisitions. This is not to imply that the technical risks in public sector R&D projects are somehow different than those encountered in private sector R&D efforts. For example, the likelihood that a specific, scientific breakthrough occurs or whether developmental subassemblies can be successfully integrated according to the system's initial design is common to both the public and private sectors. Rather, it is the relative rigidities of the public sector acquisition process that influence the available approaches for mitigating the various technical risks that may occur during an R&D project. Commercial R&D projects are largely internal to the firm with direct management oversight to guide and direct as technical issues arise. While a portion of public sector R&D is performed in government facilities, a majority of the R&D required for new capabilities is either sourced to private vendors or simply embedded within the contracts issued for the completed capability. Embedding occurs when government pays for R&D through the contracts for finished products (such as satellites) with the understanding that some of the "cost" of the product is actually covering R&D expenses rather than just the cost of producing the product itself. In short, through this mechanism the government is indirectly funding R&D performed by private firms rather than directly (Lichtenberg, 1988). This significantly reduces the public sector's ability to directly mitigate technical risks as they occur, subject to the provisions incorporated and relative completeness of the vendor's contracts. Therefore, in a public sector context the question at hand is rarely how to mitigate specific technical risks that may occur in an R&D effort, but alternatively, how to mitigate the likelihood of technical risks preventing a successful project conclusion.

In addition to the rigidities present in public sector R&D efforts, there are often specific uncertainties that private sector efforts do not face. Public sector acquisitions are frequently non-market traded goods which are often difficult to value. One method is to use contingent valuation (Carson, 2007), a survey-based technique used for estimating the economic value of non-market traded goods, such as environmental quality and conservation, public services like parks or defense, and the value of human life. While techniques such as contingent valuation have been developed to meet this challenge (Carson, 2007), public decision makers still must reconcile multiple, divergent valuations as both proponents and opponents of a given acquisition submit their respective estimates. Regardless of the valuation method employed, the selection of an appropriate discount rate, and whether this rate should vary over the period of performance for lengthy acquisitions, continues to provide spirited debate among policy makers.

This is not to imply that public sector R&D acquisitions have been ignored by the real options literature. Vonortas and Hertzfeld (1998) describe at length some of the unique issues related to public sector R&D acquisitions. They claim that the valuation of options is difficult, since the goal of the public sector is to essentially enable the private sector to make better investment choices. As a result, more traditional economic valuations of these options have not been applied to public sector R&D options. Thus, they use a real options approach to attribute social benefits to traditional net present value (NPV) calculations of public sector R&D investments and apply it to several examples of interest for the National Aeronautics and Space Administration (NASA). Post, *et al.* (2004) demonstrate the increased value of real options in the implementation of Controller Pilot Data Link Communications, a Federal Aviation Administration (FAA) program. They argue that the "now or never" approach implied in NPV calculations tend to underestimate the value of the actual project, since it does not explicitly value managerial flexibility. While they admit that complications make certain valuations difficult, they argue that such an approach makes government managers think of their projects as options. Nevertheless, these models do not directly incorporate the technical risk inherent in such projects. Golabi, *et al*. (1981) provide a procedure for the U.S. Department of Energy to select a portfolio of R&D projects in solar energy. Their index does incorporate a multi-attribute utility function (where multiple evaluation measures are combined into a single measure for the purposes of government acquisition), along with budgetary restrictions, and then solves an integer programming problem. Their model does not, however, incorporate the multi-stage competition aspects of many R&D projects.

Previous option studies evaluating multi-stage development processes allow flexibility through the use of continuation, improvement, delay, or abandonment options at each stage of the development depending upon program value at each stage. Yeo and Qiu (2003) give examples of using real options for technology investments and show how it has been accepted among several industries (e.g., mining, petroleum, pharmaceuticals). Wang and Hwang (2008) use a fuzzy compound options model to evaluate the value of each R&D project that value. Meier, *et al*. (2001) present a model that combines contingent claims analysis (valuing assets by replicating return and risk characteristics through an existing portfolio of assets (Dixit and Pindyck, 1994)) and integer programming. Costa and Paixao (2009) propose a heuristic approach based involving fewer variables, which allows the model to obtain good solutions for a reasonable number of projects. Panayi and Trigeorgis (1998) demonstrate their multi-stage model on two case studies and show how the valuations can differ from the traditional NPV analysis.

For the purposes of this dissertation, however, we define a public R&D project as one that will deliver a *non-market* traded good or service upon completion and has been deemed sufficiently necessary that project completion will be funded. While all government R&D acquisitions possess some cost or schedule limit through which program abandonment becomes an option, because there are typically no directly observed market payoffs, these limits are not easily definable. Ceylan and Ford (2002) point out that the rigid planning tools for abandonment in public acquisitions have been proven inadequate. Further, it is not infrequent for programs to continue in the face of tremendous cost and schedule overruns compared to those in the private sector since government investment decisions are often determined by political or other reasons (Post, *et al*., 2004). For example, Drezner, *et al*. (1993) find that major defense acquisitions between 1960 and 1990 experienced an average of 20% cost growth from their initial cost estimates and with a substantial percentage of programs exceeding their initial estimates by as much as 50%. Therefore, we choose to evaluate the likelihood of a given R&D program successfully developing a desired capability subject to the total budget available to the acquisition manager, while assuming that the manager has no incentive to either conserve his budget to any point below his inflexible funding constraint or abandon development until the budget is exhausted. The abandonment option can be readily incorporated into our models, but as it has been well studied by the real options literature, we find it adds no further qualitative insights.

1.3 Contributions and Organization of Dissertation

1.3.1 Contributions of Dissertation

There are four distinct, but related, contributions of this dissertation. Each contribution constitutes a chapter of this dissertation. This dissertation considers a multi-stage real options problem under three budget allocation schemes. Before discussing the specific real options problem itself, consider the following two-level problem (see also (4.1) and related Figure 4.1):

$$\begin{array}{ll} \max & a_1B_1 + a_2B_2 + a_3B_3 + \alpha(B_1, B_2, B_3) \\ \text{s.t.} & B_1 + B_2 + B_3 \leq 30 \\ & x \in SOL(B_1, B_2, B_3) \\ & B_1, B_2, B_3 \geq 0 \end{array} \tag{1.1}$$

where B_1, B_2, B_3 represent the budgets available for three time periods specified in a lower-level problem. A solution x to (1.1) depends on the values of the budgets, so the decision as to how those budgets are determined greatly affects the solution set to the lower-level problem. In this dissertation, we define three possible budget allocation methods (sometimes referred to as "Models"):

1. *Fixed ("Model 1")*: The budget B_t for each time period, t, is specified without explicit regard to its effect on the objective function of the lower-level problem. It is, in effect, exogenous, as the values for each budget are given. For the above case, one example would be specifying that: $(B_1, B_2, B_3) = (10,10,10)$. We first solve the fixed-budget problem using both stochastic dynamic programming and integer programming in Chapter 2 and Chapter 3, respectively.
2. *Flexible ("Model 2")*: This method provides the greatest flexibility for the budget allocation, but does so optimally. In some sense it could be called flexible budget-optimal, but for distinction, we refer to it as simply "flexible." The budgets are determined in a manner that optimizes the lower-level problem, but their precise values are not specified until that time period, when the state of the system is known. Thus, at the beginning of the multi-stage competition, the value for B_1 is provided, but the optimal B_2 is determined for every possible outcome that occurs at the end of the first time period. We first solve the flexible-budget problem only using stochastic dynamic programming in Chapter 2.
3. *Budget-Optimal ("Model 3")*: This case optimizes the two-level problem in (1.1) by determining endogenously an optimal (B_1, B_2, B_3). The optimal values for (B_1, B_2, B_3) correspond to those that optimize $\alpha(B_1, B_2, B_3)$ if

$a_1 = a_2 = a_3 = 0$. However, these budgets cannot change after each time period has been realized. In this sense, these budgets are optimal, but are specified *a priori* (i.e., at the beginning of the first time period). As such, it can be considered optimal "fixed" budget allocation. We first solve the budget-optimal problem both as a two-level stochastic dynamic program and a one-level integer program in Chapters 4.

The first contribution of this dissertation, examined in Chapter 2, is the formulation of a stochastic dynamic program that public sector acquisition managers can use to determine optimal project selection strategies in multi-stage, multi-project competitions. Though stochastic dynamic programming (SDP) is a standard method of evaluating decisions under uncertainty, this thesis is unique in the kind of decisions that it considers. Real options models typically demonstrate the increased benefits of managerial flexibility that can be achieved through the inclusion of additional options. This paradigm makes these models an ideal approach for evaluating the dynamic investment decisions in R&D portfolios, where the numbers of distinct options grow over time as R&D projects progress. However, the project selection problem is quite different from typical real options problems in that the acquisition manager starts with many different options and then chooses to potentially reduce the number of options as the project progresses. This decreasing options problem has been largely ignored by the literature and the suggested solution methodology constitutes a useful, practical approach for devising optimal project selection strategies. Moreover, using this approach, acquisition managers may find optimal strategies that would not likely have been considered without formally modeling the acquisition's options. Our numerical examples in Chapter 2 illustrate that *ad hoc* solutions, such as "one expensive project and one cheap one" or "more options in the early stages is always better," can be significantly suboptimal to the objective of maximizing that at least one project succeeds, a special case of optimizing the probability of achieving a certain technological maturity.

The second contribution of this thesis, presented in Chapter 3, is the reformulation of the real options problem modeled in Chapter 2 as a stochastic dynamic program into an integer program. Exploring the equivalence of specific dynamic programs to an integer programming problem is an active field of research. For instance, Newman, *et al*. (2010) formulate a dynamic program into an integer program and compare the run-times and efficiencies of both approaches. Moreover, the construction of the integer programming model yields several interesting insights, such as the linearization of binary variables, the relaxation of certain binary constraints, and a generalized formulation of the time period constraints. A more practical value of the integer programming formulation lies in its ability to handle side constraints more easily. While SDPs tend to have specific applications and solution methods, an additional side constraint can be easily incorporated into an integer programming solver. For example, we might wish to consider certain funding restrictions, such as: the funding of two projects may be mutually exclusive; the funding of a certain project implies necessarily the funding of another; certain projects can commence funding in the initial time period only, while others are not so restricted, etc. While it is true these examples of side constraints can be incorporated into an SDP model, the addition of such constraints for sensitivity analysis can require some significant changes to the structure of the SDP. On the other hand, these constraints can more easily be incorporated into an integer program with a simple algebraic expression of the relationship. Thus, the integer programming formulation provided in this dissertation could be used in operational, real options integer programming decision models, as the speed of computing resources and solution efficiencies of solvers continue to improve.

The third contribution, as detailed in Chapter 4, is the examination of two-level solutions and optimal budget search techniques. In most real options problems, there is one level of decision-making in which some suitable objective function is optimized. However, in a number of settings two levels are possibly more appropriate. Consider for example a government agency which first must decide its anticipated budget levels $B_1, \ldots, B_T$, in each of T time periods. Having fixed these annual budgets, a real options problem can then be used to determine how to best allocate the various projects in support of an overall public sector goal or goals. Such a two-level problem can be written as funding the budget levels $\vec{B} = (B_1, \ldots, B_T)$ and binary funding decisions x_{it} for projects $i \in N$ and time periods $T = (1, \ldots, T)$ to solve:

$$\begin{aligned} &\min_{B,x} \ f(\vec{B}, x) \\ &s.t. \ \ \vec{B} \in S_B \\ &\qquad (\vec{B}, x) \in S_J \\ &\qquad x \in SOL(\vec{B}) \end{aligned} \tag{1.1}$$

where $f(\vec{B},x)$, S_B, S_J and $SOL(\vec{B})$ represent respectively, the overall objective function, the feasible set for just the $\vec{B}$ variables, the joint feasible region S_J for $(\vec{B},x)$ and the solution set of the lower-level real options problem when the budget levels are fixed ($SOL(\vec{B})$). One might ask whether this top layer to determine optimal budget levels by year is really needed. As it turns out, the timing of how much is allocated (the $\vec{B}$ variables) can greatly affect the overall objective. As one example in Chapter 4 illustrates, the budget-optimal problem can increase the objective function even when the basic funding strategies (x variables) are nearly the same as the fixed-budget problem. This increase results from an optimal allocation of the budget variables, $\vec{B}$. We explore solution techniques for this critical two-level problem. To solve this two-level problem using the SDP approach, one needs to solve a series of SDPs using intelligent search techniques and budget discretization to obtain an optimal budget-optimal allocation. However, using the approach outlined in Chapter 4 of this dissertation, we can easily convert the above two-level problem into a "one-level" IP by finding the optimal budget variables, B, while simultaneously solving for the optimal funding decisions variables, x, by simply converting the each time period's budget into a continuous decision variable. As we will show in this dissertation, for some problems the IP formulation solved with robust commercial solvers may solve more quickly than the equivalent SDP formulation.

The fourth and final contribution in Chapter 5 is the application of these methods to an actual series of projects in which such a real options modeling approach is valuable. While we outline the need for these real options techniques in this chapter, and apply both the SDP and integer programming approaches to illustrative examples in the next two chapters, we actually apply these methods to carbon capture and storage (CCS) projects currently being considered and developed in the European Union (but is certainly applicable in regions outside of Europe). To our knowledge, no model has been built to solve this type of multi-stage, multi-project real options problem with such a large number of variables and constraints for an actual set of R&D projects. In this dissertation, we gather survey data from experts on CCS technologies, synthesize the results into input data for the real options model, and demonstrate several key advantages to the real options approach. Taking the perspective of a funding agency, we employ a real options framework for determining an optimal funding strategy for project selection for the development of full-scale CCS plants. Specifically, we formulate and solve a SDP for obtaining optimal funding solutions in order to achieve success by a target year. The model demonstrates the improved risk reduction by employing such a multi-stage competition and explicitly considers the benefits of knowledge spillover among competing projects. We then extend the model to consider two sensitivities: 1) changing funding decisions based on the available budget and 2) flexibility to spend that budget among the time periods. This study also makes use of the two-level problem discussed previously in that it suggests an optimal allocation of the budget, which can be a necessary step for certain funding initiatives.

Throughout this dissertation, there are essentially two general solution approaches: integer programming and stochastic dynamic programming. We formulate real options models under three possible budget allocation approaches:

fixed, flexible, and budget-optimal. Table 1.1 summarizes in which chapters the approaches are applied to the real options models.

Table 1.1: Chapter Locations of Solution Approaches by Problem Type

	Fixed-Budget Problem	Flexible-Budget Problem	Budget-Optimal Problem
Integer Programming (IP)	Chapter 3	N/A	Chapter 4
Stochastic Dynamic Programming (SDP)	Chapter 2 Chapter 3 Chapter 5	Chapter 2 Chapter 5	Chapter 4 Chapter 5

1.3.2 Organization of Dissertation

Chapter 2 introduces the real options problem we outline in this chapter in greater detail. A description of the multi-stage competition model is provided. We identify the technology progression metric for our numerical example, the Technology Readiness Levels. The explicit Markov decision process (Puterman, 1994) is defined for two problems. We then solve these real options problems for two numerical examples, the latter example being a more generalized version of the first. The final section of the chapter provides a detailed overview of the algorithm implementation, state size and run-time statistics. Chapter 2 is primarily based on the work of Eckhause, *et al*. (2009).

Chapter 3 provides an equivalent integer programming formulation for the real options problem presented in Chapter 2. We present the general formulation for the integer programming version of the problem solved as a stochastic dynamic program in Chapter 2. A small two-project, two-time period problem is described in order to demonstrate the approach. The following section demonstrates how to linearize the products of binary variables, and shows that those linear variables need not themselves be binary. We then extend it to a three-project, three-time period example. Numerical examples are provided for both a two-project, two-time period problem and a three-project, three-time period problem. The integer programming model code for the three-project, three-time period problem is in Appendix B. Run-times, number of variables and iterations are calculated for two optimization solvers, and are compared with the performance of an equivalent SDP formulation, similar to the ones presented in Chapter 2. Chapter 3 is primarily based on the work of Eckhause, *et al*. (2011).

Chapter 4 presents perhaps the most significant advantage of the IP formulation vis-à-vis the SDP, when each time period's budget may be optimized, but still needs to be specified at the outset of the real options problem. The first part of the chapter describes this as a two-level problem, where the upper level is the optimization of the budget allocation. We then show the equivalence of this two-level problem to the one-level problem, if the lower-level problem is formulated as an IP. We compare the run-times for sample problems, and demonstrate the advantage of the IP approach for problems of a certain size, but the SDP is superior in some cases. However, both approaches have limits on the size of the problems they can solve for computational reasons, as we describe in Chapter 4. The third section of Chapter 4 then proves several properties about the lower-level objective function, which then outlines techniques for intelligent search heuristics for the upper –level budgets while solving a series of lower-level SDPs. The third section also discusses the run-time performance of these methods on larger problems. Sections 4.1.1, 4.1.3 and 4.2.1 are based on Eckhause, *et al.* (2011). The remaining sections are unpublished work of Eckhause.

Chapter 5 applies the techniques outlined in Chapters 2 – 4 to actual carbon capture and storage projects eligible for public funding in the European Union. The first section provides an overview of the three major CCS technologies: pre-combustion, post-combustion and oxyfuel. The second section details some of the specific projects currently being undertaken in the EU. Section 3 outlines the solution approaches outlined in this dissertation to this real options problem. The fourth section describes the detailed subject matter expert interviews and survey results, necessary to obtain probabilities, costs, and knowledge-spillover data for the real options model. We then present the numerical results for these models, which include the cases with fixed budgets, flexible budgets and optimal budgets, along with knowledge-spillover cases. We then discuss the main conclusions about the advantages of managerial flexibility and cross-project learning, along with future applications of similar real options models applied to state-of-the-art energy technologies. Chapter 5 is based on the work of Eckhause and Herold (2011), with the exception of Section 5.6, which is unpublished work of Eckhause.

Chapter 6 provides a brief summary of the four critical chapters (Chapters 2-5). The research activities, particularly the CCS case study, have applications beyond the models presented here. As such, this last chapter concludes with suggested future research activities.

Chapter 2: Using Stochastic Dynamic Programming to Solve the Multi-Stage Real Options Model

This chapter first provides a description of a multi-stage real options model, followed by a description of the objective functions and data used in the models. We then provide a mathematical formulation of the Markov decision process for two budget allocation schemes (fixed and flexible) in Section 2.3. We then solve these real options problems for two numerical examples in Section 2.4. The final section provides a description of the model implementation. Chapter 2 is based on the work of Eckhause, *et al*. (2009).

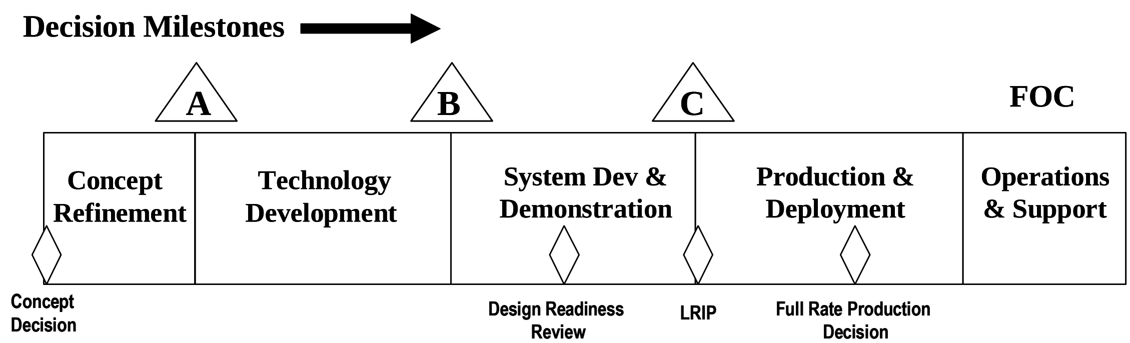
2.1 Problem Definition: The Multi-Stage Competition Model

Government acquisition managers often mitigate the technical risk associated with R&D acquisitions through a combination of formal milestone decision points and multi-source, parallel development acquisition strategies. However, a lack of formal models to address the optimal design of these competitions typically leads to *ad hoc*, qualitative solutions to these questions.

Real options valuation techniques provide an analytical framework to find optimal solutions to these problems. Real options strategies for managing R&D can also be viewed as deciding which projects should be funded and when this funding should occur. An important public sector example is in energy. The U.S. Department of Energy analyzes which alternative power generation technologies should be emphasized to meet the nation's environmental, energy-related, and security objectives. For example, in producing electricity, how much funding should be allocated to R&D in: carbon capture and storage (CCS), biofuels, solar energy, geothermal power, wind power, among others? This is essentially a multi-project competition over several stages (typically years). Often due to budgetary restrictions, it is not possible to fund each of these projects until completion. Instead, a decision must be made early on as to which projects should receive continued funding and at what levels in order to achieve some overall societal goals. This is a real options problem but one for which a public sector objective needs to be used. There are many choices for the overall objective such as the maximization of social welfare, minimization of total cost, or as discussed by Eckhause, *et al*. (2009), maximization of the probability that at least one of the projects succeeds.

As noted, government acquisition managers often mitigate the technical risk associated with R&D acquisitions through a combination of formal milestone decision points and parallel development strategies. For example, consider the Department of Defense's DOD 5000 acquisition process presented in Figure 2.1 (Department of Defense, 2001). After the Department of Defense has determined the new capability desired, multiple projects are initially awarded technology development and maturity contracts to perform the R&D required for successful development of the desired capability. At predetermined decision points, Milestones A and B, resulting technologies are evaluated to determine which, if any, projects are selected to continue R&D and capability development efforts. Milestone C decisions will typically evaluate finished prototypes and result in a final down-select to a single winning project to commence a low rate of initial production (LRIP) of the fielded capability. It is important to note that the winning project may be selected for criteria other than obtaining the highest or most robust technological maturity, such as possessing the technology with the lowest expected total cost or development schedule, or having the highest probability of successful implementation conditional upon their current level of technological maturity.

Figure 2.1: DOD 5000 Acquisition Process



While these multi-stage, multi-project competitions have proven useful for mitigating technical risk, acquisition managers must address a number of key questions to efficiently employ this strategy: How many projects should be initially funded? How many stages? How should funding be spread between stages? Which project should be funded after each decision point? The answers to these questions present difficult tradeoffs that must be faced. For example, are more projects, theoretically increasing the range of technical alternatives, or fewer, better-funded projects more likely to increase the probability of successfully acquiring the desired capability on time and within budget? Should more funds be spent in the R&D phase, ensuring a more robust technological solution, or in the product development phase, increasing the likelihood of a smoother implementation? Should the high-cost, mature technology project be selected over the low-cost, less mature technology project as the winner? Of course, the answers to these questions depend upon the precise nature of the given acquisition program. Real options techniques outlined in this chapter can provide an analytical framework to find optimal solutions to these problems.

2.1.1 Real Options Definitions

A basic call option represents a right, but not an obligation, to make a purchase at a future date (Dixit, 1994). There is the price paid to purchase this right, or option, along with the price paid to exercise a purchased option, which is the option's exercise (or strike) price. The exercise price is only paid if the option proves to be valuable at a later date, thus limiting the buyer's risk to the amount paid to initially purchase the option. Multi-stage, multi-project R&D competitions are similar in structure. The cost of issuing initial technology development contracts to a project represents the purchase price for that project option. A given project option is exercised upon the award of a subsequent contract to the project to continue development of the actual capability. The exercise price of this option is the amount of funding each winning project receives at each subsequent stage. In the simplest two-stage problem, the competition reduces to the selection of the optimal portfolio of simple call options to purchase and then exercise in the next stage.

Figure 2.2: Two-Stage Multi-Project Competition

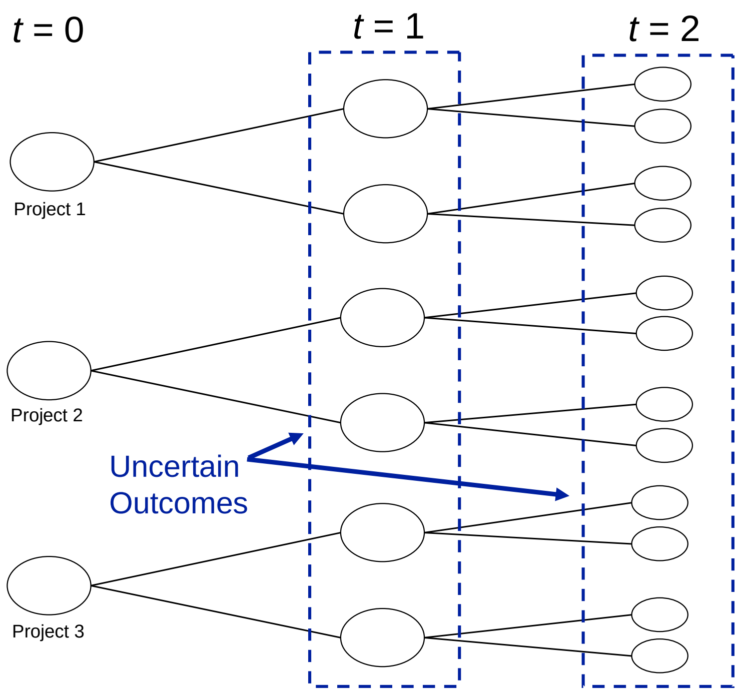


Figure 2.2 demonstrates such a two stage multi-project competition. If the objective of the acquisition program manager is to maximize the likelihood of successfully developing a desired capability in time period t=2, the manager must determine how many and which of the project options to purchase in period t=0 and then how many and which of the purchased project options to exercise in period t=1. If the competition is composed of several decision stages before the winning project(s) are selected, each project represents a complex[1] call option, as each subsequent exercising prior to the last stage, also represents an additional purchase of the option. While this may create potential computational problems as the state space

[1] A complex, or exotic, option is one that can be classified neither as a simple "European" (i.e., may only be exercised on the date of expiration) nor as a simple "American" option (i.e., can be exercised at any point up until expiration of the option) (Hull, 1997). One type of complex option, called a compound option, where the holder can purchase the right to a second option at a later date, is somewhat analogous to the real options problem described here.

expands, it does not change the formulation required to evaluate such problems. Fortunately, current computing power is sufficient to address the state spaces required for many realistic acquisition applications.

2.2 Project Progression Metric: Technology Readiness Level (TRL)

Before an optimal portfolio of project options to purchase and exercise can be identified, a metric must be employed to gauge the success of each project's R&D efforts. A common metric currently employed to assess the maturity of evolving technologies by many government agencies, especially the NASA and the Department of Defense, is the Technology Readiness Level (TRL). NASA (2011) uses nine TRLs to describe the maturity of an evolving technology. The Department of Defense, as by the Deputy Under Secretary of Defense for Science and Technology (2005), employs a slightly different definition, but the essence of the level progression is the same. The general concept behind a TRL progression is that at the beginning of technology development, general concepts are observed; then, the concepts are developed; the prototypes are designed and tested; and then the actual technology is tested and deployed. Table 2.1 provides a brief definition of each level, as defined by NASA.

Table 2.1: NASA Technology Readiness Levels (TRLs)

TRL	Definition
1	Basic principles observed and reported
2	Technology concept and/or application formulated
3	Analytical and experimental critical function and/or characteristic proof-of-concept
4	Component and/or breadboard validation in laboratory environment
5	Component and/or breadboard validation in relevant environment
6	System/subsystem model or prototype demonstration in a relevant environment
7	System prototype demonstration in a space environment
8	Actual system completed and “flight qualified” through test and demonstration
9	Actual system “flight proven” through successful mission operations

While the TRL metric has been used by NASA and the Department of Defense, there are of course other metrics that one could employ to gauge the completion level of a project. These measures might include measures related to earned value, number of successful prototypes developed or deployed or the like. Moreover, it is also true that the stochastic dynamic programming approach we propose can be used beyond these two application areas. Two other domains that lend themselves directly to such a methodology include IT management and R&D efforts in low-carbon technologies for the energy sector. In the first area, IT R&D managers may be concerned with fewer or different completion levels (e.g., software system concept, prototype development, alpha- and beta-level versions). In terms of funding R&D efforts for low-carbon technologies to produce power (e.g., tidal power, advanced solar or wind power), project managers also may have fewer or different levels of completion. For example, these levels might include: initial concept (taking into account how related to existing technology or novel), approval by a government regulatory agency, initial disbursement of funding to research laboratories and universities, prototype development, field deployment, market-ready product.

To mitigate the risks in developing new capabilities, many large-scale, expensive projects do not award a single contract that will progress from TRL 1 to TRL 9. Rather, the observations and concepts, along with the proof-of-concept and exploratory research, are usually done first, under smaller contract awards. If proven successful, or if sufficient progress is made, future contracts are awarded based on the preliminary success of the earlier TRL progression (this strategy is adopted by NASA and the Department of Defense, but is applicable to other public sector areas).

Although this multi-stage approach is sometimes used with a single project, it naturally leads to the multi-stage, multi-project contracts usually being employed. For example, during the beginning stages of a project's TRL progression, the cost of concept-development may be relatively small enough that the government agency can award several simultaneous contracts with a decision point for future contracts occurring when projects are expected to achieve TRL 6. Each project is assumed to choose whichever technology platform best suits its abilities to achieve its desired readiness level. Of course, it must be noted that TRL progression alone is not a substitute for quality of the work performed. Two competing developers or contractors may claim to have "successfully" reached a certain TRL, but one of the two may be vastly superior to the other. We assume this type of judgment is considered in the technology readiness assessment (Deputy Under Secretary of Defense for Science and Technology, 2005). Since TRLs are already commonly used for assessing technological maturity in multi-project competitions, we will also use this measure to assess competing projects within our formulation of multi-project, multi-stage acquisitions.

2.3 Mathematical Formulation

The general framework for our multi-project, multi-stage competition is as follows. We wish to potentially fund a number of projects, each with their own costs and probability of success over various stages of an R&D acquisition project so that the probability of achieving a specific predetermined level of success for the overall R&D project is maximized. The set of potential projects is represented by I. Using TRL as the measure of desired R&D success for each project in each stage, we assume that we wish to maximize the probability of achieving TRL 8 by the end of the acquisition process, as TRL 9 is usually reserved for proven, fielded systems, i.e., post initial acquisition. We should note that many other objectives are possible within this framework, such as minimizing expected cost or expected development schedule. Furthermore, let us assume there are certain funding decision periods that allow us to assess the level of maturity (success) of each funded project. There are T time periods in which decisions are made and an additional final time period $(T+1)$ in which outcomes are realized. At each of these time periods, t, we can decide whether or not to continue funding the projects currently funded (or even, by how much we should fund them) in the subsequent funding cycle.

We assume each project starts at a certain TRL, and can progress along the way according to a set of transitional probabilities relating to funding. Thus, the state of any project at the beginning of any time period is a value in the set $\mathbf{S} = \{0,1,2,3,4,5,6,7,8\}$, where $1,\ldots,8$ correspond to the current TRL achieved and 0 corresponds to no longer being funded (or possibly having been never funded). We allow for the possibility that the project may reach "success" (i.e., TRL 8) before the final stage ($T+1$). Whether or not this is possible for a particular instance of this problem can be specified by the probability mass functions (PMFs) for the transitions of each project. While we assume that we know the transitional probabilities from each state to every other state (i.e., the probability mass function of the TRL progression from one stage to the next) at every stage for every project, defining these PMFs can be challenging for many applications. However, many R&D intensive public sector acquisitions, such as aerospace and defense programs, already produce estimates of TRL success during source selection and R&D portfolio funding decisions. Typically, these are discrete PMFs, such as the probability that a program will achieve TRL 6 given a specific schedule or level of funding, that are obtained from subject matter experts and historical data. Weisbin, *et al*. (2004) describe such a process in the claim for the need for a systematic process for NASA technology portfolios. NASA's Strategic Assessment of Risk and Technology (START) approach for evaluating R&D investment decisions uses a peer review process to assign cumulative probability values to different performance range points as well as probabilities of project acceptance by the stakeholder once TRL 6 is achieved (Elfes, *et al*., 2006). Recognizing the need for better estimates for the likelihood that a technology development project successfully meets its milestones, NASA Ames Research Center is currently developing a Technology Development Risk Assessment (TDRA) tool to calculate TRL transitional probabilities as a function of time and budget (Mathias, *et al*., 2006). However, as current public sector R&D funding decisions use some form of qualitative or Delphi approach (Linstone and Turoff, 1975) to evaluate the probabilities of achieving a few specific program milestones (e.g., the probability of reaching TRL 6 by a specific point), we employ the simple, discrete PMFs that modelers will most likely obtain from subject matter experts and historical data.

This sequential decision model can be referred to as a Markov decision process (Puterman, 1994). The actions, rewards and transition probabilities depend only on the current state and actions, not on the past states occupied and past actions made. To the extent that those previous actions affect the transitional probabilities in our current state, we expand the state definition to include those effects. We will develop our formulations for determining the optimal portfolios of real options to purchase and exercise in multi-project, multi-stage competitions by initially examining a fairly restrictive version of the problem. We will then develop a formulation that relaxes many of the initial assumptions to better accommodate realistic acquisition programs. The PMF for each project is strictly determined by the funding decision for that project; we assume the funding decisions for the other projects do not impact that PMF.

While this assumption appears extreme for certain types of problems, modification of the state definitions can make this difficult restriction disappear. In our case study in Chapter 5, knowledge spillover, which is encouraged by the funding agency, implies that one project's technological progress can influence the PMFs of other projects. We incorporate that important feature by expanding the state definition to include not simply the project's current state, but the maximum state achieved by other relevant projects, at that time period. By expanding the state definition we are able to preserve the Markov process.

2.3.1 Model 1 Formulation: Fixed-Budget Problem

In this version of the multi-project, multi-stage competition, we assume that the total budget available to the acquisition manager at each stage is fixed and that the potential funding level for each project at every stage is also fixed at some predetermined level. The only decision available to the acquisition manger (or decision-maker) is whether or not to fund any specific project(s) at each stage. We define the following state variables and data for our formulation:

- Let $C_{it} \in S_i$ be the state of project i at time period t; we assume that $S_i = \{\, 0,1,2,3,4,5,6,7,8 \,\} \quad \forall\, i$
- Let $X_{it} \in \{0,1\}$ be the decision variable of whether to fund project i at time period t
- Let α_{it} represent the cost of funding project i at time period t
- Let B_t represent the R&D budget available for time period t

With these definitions, we make the following assumptions:

- *Assumption 1*: As previously stated, we assume we are provided the state transition probabilities. In other words, given for any state s_1 and any state s_2, we know the value of $\mathbf{P}\{C_{i,t+1} = s_2 \mid C_{it} = s_1,\, X_{it} = 1\}$. In other words, given that project i is in state s_1 at time period t, we know the probability of going to some other state s_2 if we fund the project in that time period.
- *Assumption 2*: $\forall\, s_1 \in S \setminus \{8\}$, $\mathbf{P}\{C_{i,t+1} = 0 \mid C_{it} = s_1,\, X_{it} = 0\} = 1$. In other words, a project not funded at time period t will necessarily be in state 0 in

the next time stage (and all subsequent stages), unless that project has already attained a TRL of 8.

- *Assumption 3*: We assume that if a project is in the unfunded state, then the option has "expired" and cannot be funded subsequently. Namely, for all time periods $t > 1$, $\mathbf{P}\{C_{i,t+1} = 0 \mid C_{it} = 0, X_{it}\} = 1$.
- *Assumption 4*: If a project reaches TRL 8 (or "success") before the final time period, then that project remains in the success state, regardless of additional funding, i.e., $\mathrm{P}\{C_{i,t+1} = 8 \mid C_{it} = 8, X_{it}\} = 1$.

Implicit in Assumption 1 is that the projects' state transition probabilities (and its associated costs) reflect the characteristics and variety of projects considered. Some project teams may have greater experience and more workers, and therefore likely more costly but more successful. Other projects may be "long shots" with limited resources, but potentially mitigated by lower costs.

At time period t, the state of the system can be described as all the combinations of states, with one for each project. That is,

$$C_t \in \left(\prod_{i \in I} S_i \right) = S \quad \forall t \tag{2.1}$$

For these combinations of states at time period t, we can choose a set of feasible funding decisions:

$$X(C_t) = \begin{cases} X_t \in \{0,1\}^I : \sum_{i \in I} \alpha_{it} X_{it} \le B_t \\ \qquad\qquad X_{it} = 0 \quad \text{if } C_{it} = 0 \qquad \forall\, i \in I \end{cases} \tag{2.2}$$

In other words, X_t is the set of all funding decisions made at time period t. The second constraint indicates that we do not fund a project that is already in state 0. Explicitly adding the constraint $X_{it} = 0$ if $C_{it} = 8 \quad \forall\, i \in I$ is unnecessary by an optimality argument, since it is implicitly considered in the objective function of maximizing overall project success. That is, letting $X_{it} = 1$ when $C_{it} = 8$ does not increase the objective function, but rather decreases the available budget. Nevertheless, in order to preserve fiduciary responsibility, we can include such a constraint. If the desire to conserve funding does not exist, then it still has no impact on an optimal solution.

If we wish to choose the optimal funding strategy to maximize the probability of at least one project reaching TRL 8 (i.e., success), then we can solve for the binary decision variables by formulating it as a stochastic dynamic program. We formulate the problem as

$$V_t(C_t) = \max_{X_t \in X(C_t)} \mathbf{E}\{V_{t+1}(C_{t+1}) \mid C_t, X_t\} \quad t = 1, \ldots, T \tag{2.3}$$

Calculating the value of this function is inherent to the stochastic dynamic program model itself. In other words, the model described here, while containing certain commonalities to all Markov decision processes, forms the basis of the solution algorithm techniques. In this SDP, the calculation of the expectation depends on the distribution of C_{t+1} conditioned on C_t and X_t, which we previously assumed as given. We note that maximizing the above objective function will always have a solution, since we are considering a set of feasible funding solutions over a discrete set of possibilities[2]. Therefore, complete enumeration—while not always desirable in practice—would guarantee an optimal solution.

Recall that the goal is to maximize the probability that at least one project achieves TRL 8. We assume if all projects fail to reach TRL 8, then we have failed to meet the goals of the R&D acquisition. Thus, we can state the boundary condition of the dynamic program as

$$V_{T+1}(C_{T+1}) = \begin{cases} 1 & \text{if} \quad C_{i,T+1} = 8 \quad \text{for some} \;\; i \in I \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

This condition assumes no "consolation" prize for a project reaching TRL 7, for instance. If the dynamic program is solved optimally, the probability that the goal is accomplished by the final time period is determined by the transitional probabilities and the R&D budgets for each time period (i.e., $B_1, B_2, ..., B_T$). Solved recursively, the value of V_1, the value of the initial state, therefore provides to solution the optimal success probability.

2.3.2 Model 2 Formulation: Multiple Funding Levels and Flexible Budgets

By relaxing two of our previous assumptions we are able to address a much wider class of problems that can accommodate the many variations that government decision-makers face. First, we permit some degree of budget flexibility. Though we continue to assume that the total budget for the entire planning horizon is fixed at a predetermined level, the budget can be spread as required between the two stages.

---

[2] The trivial solution of funding no projects in any time period is always feasible so long as the budget available in each time period is non-negative.

Next, we allow several distinct funding levels for each project at each stage, under the assumption that increasing a project's funding above some threshold is likely to positively increase its TRL transitional probabilities. One might argue that a decision-maker actually faces a continuum of funding levels for each project. However, there are at least two reasons why discrete funding levels are sufficient. From a theoretical standpoint, a continuum of funding levels can be sufficiently approximated discretely. In reality, transitional probabilities for TRL progression would exist for only a few funding levels, since they rely heavily on subject matter expertise and historical data. Thus, the number of funding levels for each project and stage is limited to the number of probability mass functions one is able to generate with reasonable accuracy. We define the following state variables and data for Model 2:

- Let $B = B_1$ denote the fixed budget available to the decision maker at the beginning of the R&D acquisition process.
- Let B_t be the budget *remaining* at time period t.
- Let α_{itl} denote the cost of funding project i at time period t at level l
- Let $X_{itl} \in \{0,1\}$ be the decision variable of whether to fund project i at time period t at level l

Extending the same four assumptions from Model 1 to multiple funding levels, we assume given for any state s_1 and any state s_2, we know the value of $\mathbf{P}\{C_{i,t+1} = s_2 \mid C_{it} = s_1, X_{itl} = 1\}$ for all funding levels l. If a project is not funded in a certain time period at any level l, i.e., $\sum_{l} X_{itl} = 0$, then Assumption 2 from Model 1 holds. The state of the system at time period t is

$$(C_t, B_t) \in \left(\prod_{i \in I} S_i \right) \times \mathbf{R}_+ \tag{2.5}$$

and the feasible decisions and budget transition at time period t can now be written as

$$X(C_t, B_t) = \begin{cases} (X_t, B_{t+1}) \in \{0,1\}^{I \times L} \times \mathbf{R}_+ : & \sum_{i \in I, l \in L} \alpha_{itl} X_{itl} \leq B_t \\ & \sum_{l \in L} X_{itl} \leq 1 \quad \forall\, i \in I \\ & X_{itl} = 0 \quad \text{if } C_{it} = 0 \quad \forall\, i \in I,\ l \in L \\ & B_{t+1} = B_t - \sum_{i \in I, l \in L} \alpha_{itl} X_{itl} \end{cases} \tag{2.6}$$

X_t represents the set of feasible funding decisions for all projects over all possible funding levels. Similar to Model 1, we can formulate this problem as a stochastic dynamic program, but with two sets of decision variables (X_t, B_{t+1}). The set $X(C_t, B_t)$ represents the feasible funding decisions given the state of all projects at time period t (i.e., C_t) and a budget remaining (i.e., B_t). Again, we are concerned with an optimal funding strategy to maximize the probability of at least one project reaching TRL 8. However, we now calculate that probability based on both the budgetary and funding decision flexibility. Thus, we have

$$V_t(C_t, B_t) = \max_{(X_t, B_{t+1}) \in X(C_t, B_t)} \mathbf{E}\{V_{t+1}(C_{t+1}, B_{t+1}) \mid C_t, X_t\} \quad t = 1, \ldots, T \tag{2.7}$$

In order to solve this dynamic program, we must discretize the budget component of the state variables. While this requirement could theoretically create significant state expansion problems rendering the SDP intractable, realistic applications can most likely be handled. For example, the decision-maker can discretize the budget components to reasonable sizes. One need not make that increment any smaller than the smallest combinations of the project costs over any time period. In the example below, \$0.1 million is a sufficiently small increment. Presumably, we may desire to limit the ability to spend large amounts of the budget in any one time period. Naturally, one can easily produce additional constraints to the feasible decisions to limit the amounts spent in each time period.

2.4 Numerical Examples

In this section we use numerical examples to illustrate the effectiveness of the formulations outlined in the previous section. These instances demonstrate the approach for both a simple, illustrative problem, as well as a larger, more computationally intense example. The purpose of the latter example is to indicate the efficiency and speed of the SDP model. Computational complexity is discussed in the next section. Algorithm implementation details for the actual case study are described in Chapter 5.

2.4.1 Model 1 (Fixed-Budget) Numerical Example

Suppose that the National Reconnaissance Office (NRO) decides to acquire a satellite with new sensing capabilities substantially out of the reach of current technology. Assuming the NRO is employing the DOD 5000 acquisition framework presented in Figure 2.1, they decide to pursue the following acquisition strategy. The NRO will request proposals from four project teams that detail their technical approach, proposed schedule, and cost bids for performing the R&D required to invent the new sensing capability. This type of acquisition strategy is typical for government agencies to employ. Each project will also submit a similar proposal and bid for actually developing the satellite. At the Milestone A decision, the NRO will determine which projects will actually receive a technology maturation contract to invent the new capability. The NRO will purchase a simple call option with each of these initial contracts it awards. At a predetermined Milestone B decision point, the NRO will evaluate each of the selected projects' prototypes and exercise one or more of their previously purchased options by awarding a follow-on contract to the winning project(s) selected to build the satellite. The NRO will decide whether to launch the satellite at the Milestone C decision point, at which time it is fielded. In essence, we are considering an acquisition with two stages and four projects. We will assume that the acquisition has already reached a certain technical maturity, so each project's project begins at TRL 4, with the goal of reaching TRL 8 by the end of the second stage. The budget available to the NRO for the first and second stages are fixed at $10 million and $20 million, respectively (i.e., $B_1 = \$10$ and $B_2 = \$20$), with decision makers facing the "use it or lose it" constraint not atypical of government budgets. Thus, with no budget flexibility or incentive to withhold funds, the NRO's acquisition managers will choose to exhaust their entire budget in each stage. Table 2.2 shows each project's stated costs for each stage. The conditional transitional probabilities for each project are presented in Tables A1 and A2 in Appendix A.

Table 2.2: Projects Costs for Each State (millions)

Project	Stage 1	Stage 2
Project 1	\$ 3.5	\$ 4.0
Project 2	\$ 3.7	\$ 6.9
Project 3	\$ 5.0	\$ 10.4
Project 4	\$ 2.3	\$ 6.3

Traditionally, the acquisition managers would construct a capability or requirements matrix and assign appropriate qualitative and quantitative values to each of the projects for comparison. Project selection in each stage would then typically be determined through either a weighted or un-weighted Delphi approach that estimates the state transition probabilities (Linstone and Turoff, 1975). While this approach allows acquisition managers the ability to carefully consider the qualitative merits of each project, it fails to ensure that the number and mix of projects selected actually maximizes the probability of a successful acquisition given the NRO's budget constraints. We determine the optimal portfolio of project options to purchase and exercise by solving $\max_{X_t \in X(C_t)} \mathbf{E}\{V_{t+1}(C_{t+1}) \mid C_t, X_t\}$. An optimal solution maximizes the expected value of the value function, which is the probability that at least one project achieves TRL 8. The results of the dynamic program for this two-stage, four-project problem are that the acquisition manager purchases options, by awarding contracts, on both Project 3 and Project 4 in the first stage. As it turns out, both options would be exercised in the second stage with the award of follow-on contracts regardless of their first-stage outcomes, since the total cost falls beneath the Stage 2 budget constraint. This acquisition strategy produces a 56% probability of success (i.e., $V_1 = 0.56$), with success defined as the likelihood that one of the projects will achieve TRL 8 at the end of the second stage. This 56% is computed as follows with P_3, P_4 denoted as the success probabilities for Projects 3 and 4, respectively. For example, Figure 2.3 and Figure 2.4 display the transition success probability for Projects 3 and 4. Using the transition probabilities in Figure 2.3, $P_3 = 0.1(0.1) + 0.1(0.1) + 0.5(0.3) + 0.2(0.7) + 0.1(1) = 0.41$. P_4 is calculated similarly using the values in Figure 2.4. Thus, we have that $1-(1-P_3)(1-P_4) = (1-(1-0.41)(1-0.255) = 0.56$.

Figure 2.3: Project 3's Transition Success Probabilities

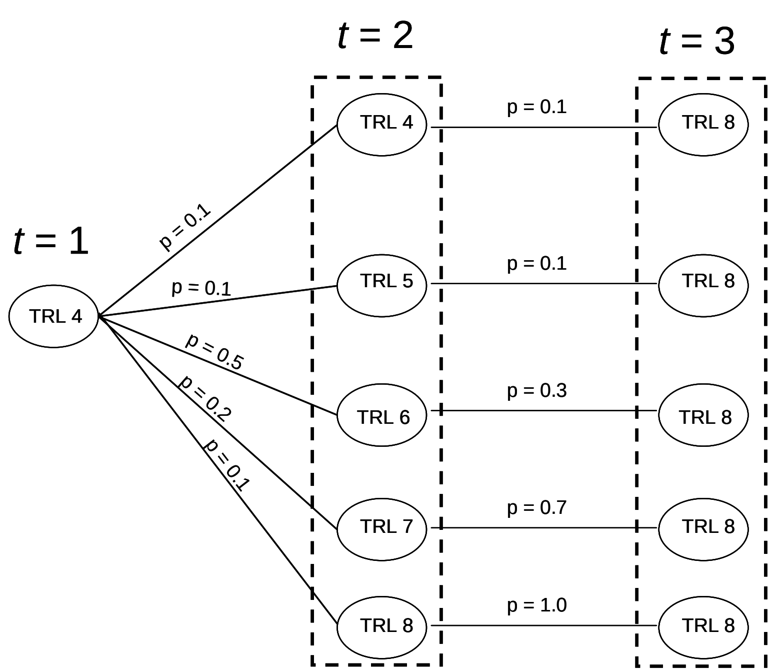
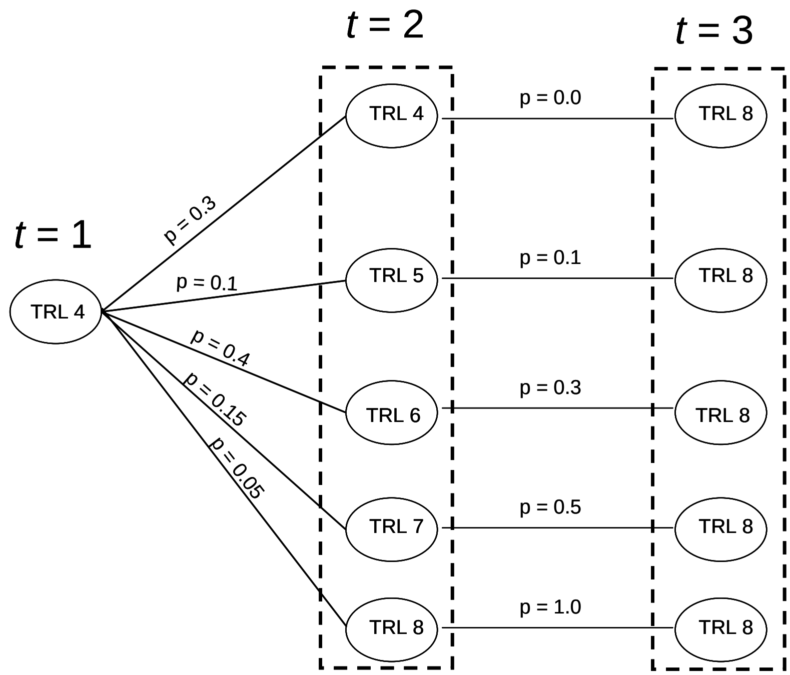


Figure 2.4: Project 4's Transition Success Probabilities



One interesting point to note is that in Stage 1, Project 3 has the highest cost and Project 4 has the lowest cost. Thus, funding them is not intuitively the obvious thing to do if one were to simply fund the cheapest projects first until the budget is exhausted (i.e., the "cherry-picking" approach). This result shows that it is the combination of costs as well as probabilities that need to be taken into consideration to arrive at an optimal decision.

For such a small problem, one can simply enumerate the state space, rather than solve the stochastic dynamic program. There are only $2^8 = 256$ unique funding possibilities, the vast majority of which are infeasible. One could simply select the feasible strategy with the largest value for the objective function. A subset of this enumeration is shown in Table 2.3. Obviously, larger problems can make better use of the reduction of states that need to be considered by solving a dynamic program.

Table 2.3: Enumeration of Model 1's Solutions

X_{11}	X_{21}	X_{31}	X_{41}	X_{12}	X_{22}	X_{32}	X_{42}	$V_2(C_2)$
0	0	0	0	0	0	0	0	0.00
1	0	0	0	1	0	0	0	0.14
1	1	0	1	1	1	0	1	0.47
1	0	1	0	1	0	1	0	0.49
0	1	1	0	0	1	1	0	0.52
0	***0***	***1***	***1***	***0***	***0***	***1***	***1***	***0.56***
0	1	1	1	0	0	1	1	infeasible
1	0	1	1	0	0	1	1	infeasible
1	1	1	1	1	1	1	1	infeasible

2.4.2 Model 2 (Flexible-Budget) Numerical Example

We now consider the more robust problem outlined in Model 2. Reconsidering our hypothetical NRO satellite procurement, we will assume that there are still two stages and four projects, but instead of one funding level, the NRO requests proposals from each project at different funding levels, to insulate the acquisition from pending budget cuts. Of course, the degree of technological maturity achieved will likely be reduced at lower levels of funding, but this will be reflected in the TRL transition probabilities associated with each funding level. For our example, we assume that the NRO receives as many as three funding options (four, if one counts deciding not to fund that project) for each project. We have retained the original four projects, but assume that each of the projects can also be funded at some specific higher or lower level of funding. Other than incorporating the additional funding levels, we will assume the NRO's acquisition strategy remains unchanged. Table 2.4 shows the costs for funding at the low, medium and high levels for each of the projects in both time periods. Again, each project begins at TRL 4. The conditional transitional probabilities for each project at each funding level are presented in Table A3 and

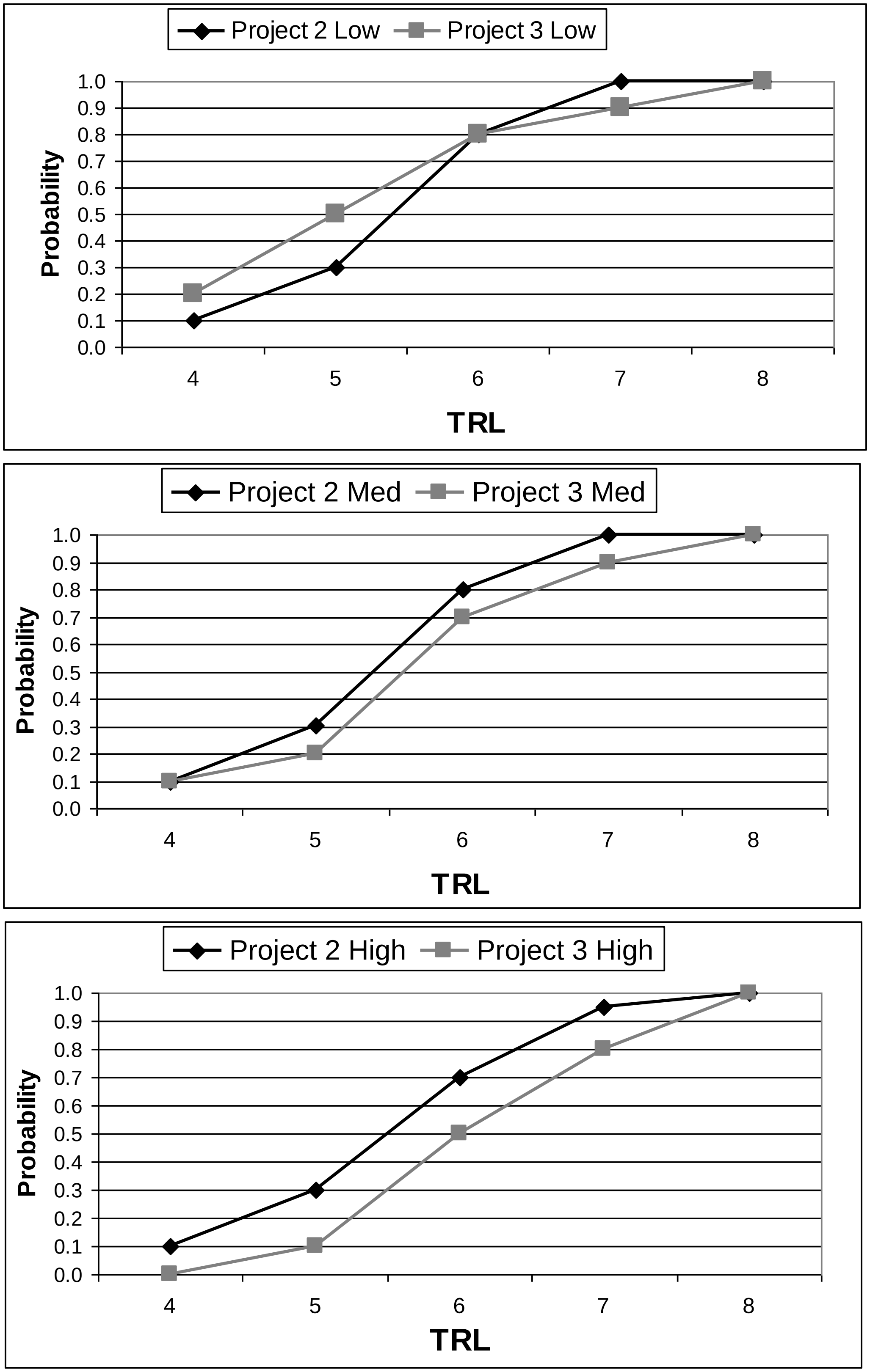
Table A4 in Appendix A. However, we also assume that the NRO's previous total acquisition budget of $30 million can be spent over the two stages without restriction. It is important to note that the costs in the medium funding levels in Table 2.4 correspond to the costs in Table 2.2. This overlap permits us to see explicitly the benefits of increased managerial flexibility. As our cost values have significance at the $0.1 million level, we can safely discretize the budget to $0.1 million without loss of scenario feasibility.

Table 2.4: Project Costs at Three Funding Levels in Each Stage (millions)

Project	Stage 1 Low	Stage 1 Med	Stage 1 High	Stage 2 Low	Stage 2 Med	Stage 2 High
Project 1	$ 2.5	$ 3.5	$ 5.0	$ 3.0	$ 4.0	$ 5.0
Project 2	$ 3.2	$ 3.7	$ 5.2	$ 6.9	$ 6.9	$ 7.9
Project 3	$ 3.0	$ 5.0	$ 9.2	$ 10.4	$ 10.4	$ 10.4
Project 4	$ 1.8	$ 2.3	$ 2.8	$ 5.3	$ 6.3	$ 7.3

The optimal first-stage solution (since we enumerated all funding possibilities, it is a unique optimum) in this example is to purchase options, by awarding contracts, to Project 1 and Project 3 at the highest possible funding level, and Project 4 at the medium funding level. An option is not purchased on Project 2, which at first glance may seem counter-intuitive given the relative cost vs. Project 3, which is funded. As shown in Figure 2.5, the rationale for this is Project 3's stochastic dominance over Project 2 for most of the TRL levels. Maximizing the value function (i.e., the probability that at least one project reaches TRL 8), we find $V_1 = 0.71$ or that there will be a 71% chance of at least one project at TRL 8 at the end of the acquisition program.

Figure 2.5: Cumulative Distribution Functions for Projects 2 and 3



Comparing these results to the Model 1 example clearly demonstrates the value of increasing managerial flexibility in these kinds of acquisitions. By allowing budget flexibility, the NRO's acquisition managers are able to fund an additional project (Project 1) in the first stage, even at their highest funding levels. The most surprising result, however, is that we maximize our probability of success by spending more on the first stage ($16.5 million) than the second stage ($B_2 = \$13.5\,\text{million}$). Since actual government acquisitions are typically structured with increasing budgets in each subsequent stage, even when program managers are able to retain unused funds, we produce an optimal strategy that would not likely have been discovered using the current Delphi-based decision process. Lastly, with a more flexible budget as well as the allowance for multiple funding levels, the success probability increases from 56% to 71%.

Another advantage of using this real options technique is that the optimal portfolio of options to exercise in the second stage can be easily solved after incorporating the realized TRLs after the first stage. This provides additional managerial flexibility since the acquisition manager can significantly alter his or her initial acquisition strategy as new information arrives. As our results show, the ability to include budget flexibility and multiple funding options in this example provides a significantly larger value for the objective function in Model 2.

2.4.3 Using Common Numerical Examples to Compare the SDP and IP Formulations

In the following chapter we formulate the real options problem with a fixed budget (Model 1) as an integer program. In order to compare the approaches computationally, we introduce a common problem in this section. In the above numerical examples, we demonstrated the approach on four-project, two-time period problems. The definition of the states mapped directly from the TRLs definitions. In other words, each project could theoretically be among the states $\mathbf{S} = \{0,1,2,3,4,5,6,7,8\}$, though in the numerical example there was only a non-zero probability of being in states $\{0,4,5,6,7,8\}$ given a funding decision.

For the integer programming problem, we consider a two-project, time-period problem and a three-project, three-time period, where the states space for each project is defined as $\mathbf{S} = \{0,1,2,3,4\}$, where states 1, 2, 3, and 4 implicitly map to TRLs 5, 6, 7 and 8, respectively.[3] The reason for this deviation is to be consistent with the notation in Eckhause, *et al.* (2011) where the TRL concept was not utilized. For computational complexity and problem size concerns, the key is that, for each project, there are five possible state outcomes instead of six. In Chapter 3, we introduce and motivate the integer programming version of this real option problem, and use the above state definitions for the numerical examples. We mention it here as a point of reference when we ultimately compare the run-times and computational complexity of the stochastic dynamic programming formation and the integer programming model.

[3] We define the final state, 4, in the possible project states, $\{0,1,2,3,4\}$, as the “success” state.

2.5 Algorithm Implementation and Computational Complexity

The dynamic program employs the backward induction method in the standard manner (see, for example, Puterman, 1994). It begins in the final time period ($t = 2$). For every C_2 (i.e., all possible states for the four projects at the beginning of time period 2) and a given remaining budget, B_2, we calculate the feasible set of actions, X_2, that maximizes the probability that at least one project reaches TRL 8 (i.e., we minimize the probability that all projects fail). In other words, for each C_2, we solve for

$$\max_{(X_2, B_3) \in X(C_2, B_2)} 1 - \prod_{i \in I} \left(1 - \mathbf{P}\{C_{i,3} = 8 \mid C_{it}, X_{i2l}\}\right) \quad \forall l \tag{2.8}$$

The optimal action's probability of success, given a set of outcomes and remaining budget, becomes the second-stage value function. That is, for each feasible (C_2, B_2), we calculate

$V_2(C_2, B_2) = \max_{(X_2, B_3) \in X(C_2, B_2)} 1 - \prod_{i \in I} \left(1 - \mathbf{P}\{C_{i,3} = 8 \mid C_{it}, X_{i2l}\}\right) \quad \forall l$. It is worth noting that this product can be written as the sum of logs, in which case the objective function becomes additive, which would help with computation.

In the first stage, for each set of funding actions, X_1, and its associated cost, we find the value that maximizes $V_1(C_1, B_1)$ by calculating

$$\max_{(X_1, B_2) \in X(C_1, B_1)} \mathbf{E}\{V_2(C_2, B_2) \mid C_1, X_1\} = \max_{(X_1, B_2) \in X(C_1, B_1)} \sum_{C_2 \in S} V_2(C_2, B_2)\, \mathbf{P}\{C_2 \mid C_1, X_1\} \tag{2.9}$$

In other words, we calculate the value of those feasible actions in state 1 by summing the probabilities of the outcomes given the funding action multiplied by the associated $V_2(C_2, B_2)$ calculated previously. The algorithm sums the probabilities since the state outcomes are mutually exclusive. This procedure would continue for all prior time periods if the acquisition problem had three or greater funding intervals.

While the numerical examples in the previous section describes a problem that can solved using the methods described in this chapter, these problems were still small enough that the run-time performance of the model for a larger number of projects and time periods needs to be explored. The case study in Chapter 5 will demonstrate the robustness of the model to adapt to large, real-world problems. In our numerical experiments, this model does well for problems with small numbers of projects, outcomes and actions. It seems likely that the number of possible projects and actions would be modest for large acquisitions. Also, since simple, discrete PMFs are likely the type of data available for such a decision process, the number of possible outcomes is probably limited to only a handful of identifiable outcomes.

In terms of the computational complexity involved, consider the following. Suppose that there are v projects, o possible outcomes (i.e., the possible TRLs achieved in the following state), a actions (i.e., the set of funding levels, including not funding) and b number of possible budget increments (simply the total budget divided by the budget increment—\$0.1 million in the Model 2 numerical example). For the Model 2 example, when $b = 300$, $a = 4$, $o = 6$, $v = 4$ and $t = 2$, there are potentially 200 million iterations[4], though many are eliminated due to budget or state infeasibilities. A C++ implementation on a 2.0 GHz dual-CPU with 2.0 GB of RAM runs in about two seconds. For $b = 6000$, there is a 20-fold increase in the number of

[4] Here, an "iteration" refers to the number of times in the inner-most for-loop the SDP enters.

iterations, and the computational time increases linearly to roughly 30 seconds, still quite tolerable for solving such problems. For a five-project, five-time period example of Model 1 (where the budget for each time period is fixed and the decision is simply “fund” or “no fund”), $a = 2$, $o = 6$, $v = 5$ and $t = 2$, there are roughly 6 billion potential iterations. The run time for a similar C++ implementation on the same 2.0 GHz dual-CPU with 2.0 GB of RAM is about 14 seconds. A more thorough explanation of the number of iterations, state size, run-times for the SDP, and its performance compared to an integer programming approach are presented at the end of the next chapter and in Chapter 4.

In Chapter 3, we present a mixed-integer programming formulation of the fixed-budget real options problem. The integer programming formulation provides an alternative method, which could be valuable for some instances, while allowing more easily for the addition of side constraints. We compare the run times of this stochastic dynamic programming implementation with integer programming formulations. In Chapter 4, we provide further detail on the comparison of the two approaches for certain classes of problems; we present detailed results on the state space, run-times and computational complexity issues.

Chapter 3: An Integer Programming Approach for Evaluating the Multi-Stage Real Options Model

This chapter provides an equivalent integer programming formulation for the real options problem presented in Chapter 2. We present a general formulation of the model in Section 3.2. A small instance of the problem is described in order to demonstrate the approach in Section 3.3, followed by a larger example in Section 3.4. After briefly discussing the advantages of the integer programming approach in Section 3.5, numerical examples are provided for two sample problems in Section 3.6, along with a comparison of run-time performance with the equivalent SDP formulation. Chapter 3 is primarily based on the work of Eckhause, *et al*. (2011).

3.1 Rationale for the Integer Programming Approach

In Chapter 1, we discussed several of the motivating factors for modeling and solving the multi-project, multi-time period real options problem as an integer program, not just simply presenting it as a Markov decision process and solving it using stochastic dynamic programming, as we did in Chapter 2. There are at least three motivating factors for this alternative formulation. The first is that the formulation of the problem, originally modeled as a Markov decision process, but reformulated as an integer program, presents a guide for the conversion of other problems traditionally solved using stochastic dynamic programming as integer programs. It is well established that stochastic programs whose outcomes do not depend on the decisions made can be written as linear programs (Birge and Louveaux, 1997). Certain knapsack problems can be solved either using an integer programming formulation or a pseudo-polynomial time dynamic program (Nemhauser and Wolsey, 1988). The formulation of the problem presented in Chapter 2, however, is solved using stochastic dynamic programming, which presents its conversion to an integer program as a novel undertaking. As we demonstrate in this chapter, the linearization constraints require the creation of path dependencies that eliminate the "memoryless" property of the Markov decision process. Nevertheless, we show that these linearization constraints only require the addition of continuous, linearly-constrained, non-integer, variables.

The second reason for the reformulation of this problem as an integer program is the relatively easy incorporation of side constraints, should they need to be added to a real options problem. For a clean energy example, we could consider certain funding restrictions, such as: the funding of two wind projects may be mutually exclusive; the funding of a certain wind project implies necessarily the funding of another; certain projects can commence funding in the initial time period only, while others are not so restricted, etc. While it is true these types of constraints can be incorporated into an SDP model, the addition of such constraints for sensitivity analysis can require some significant changes to the structure of the SDP. On the other hand, these constraints can more easily be incorporated into an IP with a simple algebraic expression of the relationship. Moreover, with the advances in integer programming methods over the last few decades, it is anticipated that an IP formulation could be a viable approach for even larger problems.

Finally, the third major reason for the IP formulation has to do with a two-level formulation, as we discussed in Chapter 1. Consider for example a government agency which first must decide its anticipated budget levels *a priori*. Having fixed these annual budgets, a real options problem can then be used to determine how to best allocate the various projects in support of an overall public sector goal or goals. We are unaware in the literature of a two-level problem in which the lower-level is an SDP. Significant recent work in multilevel programming has focused on this bottom level as an optimization (Brotcorne, *et al*., 2008) or an equilibrium problem. In the latter case, such problems are called mathematical programs with equilibrium constraints (MPEC) (Luo, *et al*., 1996) and include both optimization as well as game theoretic models. While most MPECs have included lower-level problems and have received considerable attention in the last few years due to their applicability in a variety of fields, often the lower-level problem is assumed to have Karush-Kuhn-Tucker (KKT) conditions (see, for instance, Bazaraa, *et al*., 1979) in order to be able to characterize the solution set of the lower-level problem. A recent example in energy in which the lower-level problem's KKT conditions get moved to the upper level and then converted to disjunctive constraints can be found in (Gabriel and Leuthold, 2010); a mixed-integer linear program resulted. In the current setting, no such KKT conditions are available to the SDP formulation of our problem, so necessarily a different approach is needed. In particular, the reformulation of the SDP into an integer program is what is accomplished. Thus, we develop the required IP formulation for solving a lower-level SDP in this context. We will then convert this two-level problem into a one-level IP problem. The formal proof of this property is in Chapter 4. This chapter follows the approach and notation from Eckhause, *et al*. (2011).

3.2 General Integer Program Formulation

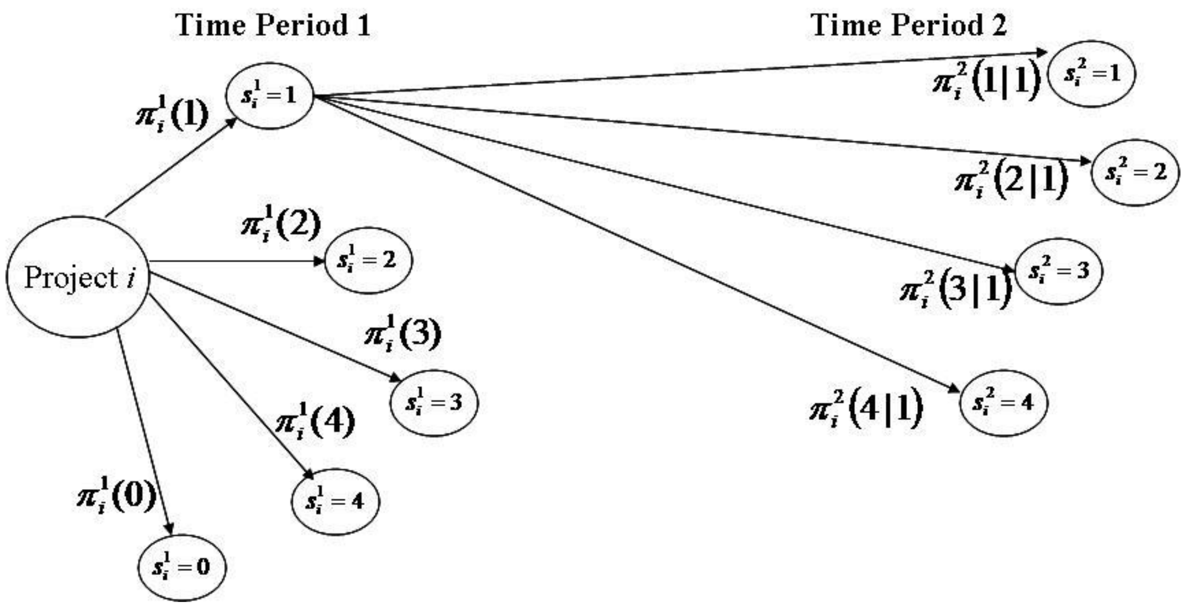
The following nonlinear, integer program is a formulation of the real options project selection problem presented in Chapter 2 and formulated as a stochastic dynamic program. Suppose we wish to decide whether to fund N projects over T time periods such that the probability of success (i.e., that at least one project reaches a specific success state by the end of the T^{th} time period) is maximized. There is one funding level for each project and a fixed budget for each time period.

The following notation is used for the input data:

- The set of possible project outcomes (states) is: $\mathbf{S} = \{0,1,2,...,S\}$, where 0 denotes the state of the project in all subsequent time periods if it was not funded in the first time period.
- $\pi_i^t\left(s_i^t \mid s_i^{t-1}\right)$ is the probability of project *i* (where $i = 1,..., N$) achieving state $s_i^t \in S$ at time period *t* given the project was in state s_i^{t-1} at time period $t-1$.

Figure 3.1 depicts these probabilities for a given project *i* where $\mathbf{S} = \{0,1,2,3,4\}$.

Figure 3.1: Project *i*'s Outcomes



- $v(s_1^T, s_2^T, ..., s_N^T)$ is the value of being in state $(s_1^T, s_2^T, ..., s_N^T)$ in the final time period. Since the objective is to have at least one project succeed, and there are no "consolation prizes:"

$$v(s_1^T, s_2^T, ..., s_N^T) = \begin{cases} 1 & \text{if } s_i^T = \text{S for any } i \\ 0 & \text{otherwise} \end{cases}$$

  $s_1^T = S$, $s_2^T = S$, or $s_N^T = S$ means that project 1 or project 2 or project *N* has achieved the success state.

- c_i^t is the cost of funding project *i* at time period *t*.

- B_t is the budget for time period *t*.

The decision variables that represent the values to be optimized in the objective function are:

- $x^t_{i,(s_1^{t-1},s_2^{t-1},...,s_N^{t-1})}$ is the binary decision variable to fund project *i* (where $i = 1,..., N$) at time period *t*, conditioned on the set of states[5] achieved by all projects at the end of $t-1$.

The resulting variable $p\left(s_1^T, s_2^T,..., s_N^T\right)$ is the probability of being in state $\left(s_1^T, s_2^T,..., s_N^T\right)$ in the final time period given the set of funding decisions $x^1_{i,(s_1^0,s_2^0,...,s_N^0)},..., x^t_{i,(s_1^{t-1},s_2^{t-1},...,s_N^{t-1})},...x^T_{i,(s_1^{T-1},s_2^{T-1},...,s_N^{T-1})}$ for all $i = 1,..., N$.

In the final time period for each project, there is only a "failure" (0) and a "success" (S) outcome. Thus, there are $2^N - 1$ possible meaningful outcomes, indexed by j , where at least one of the projects has achieved success. For example, if there are three projects, there would be $2^3 - 1 = 7$ outcomes: $(0,0,S),(0,S,0)...(S,S,S)$, where each project's final state less than S represents a "failure." Based on each of the $2^N - 1$ meaningful outcomes, $\tilde{P}_j$ is the probability of achieving the j^{th} outcome. For example, in a problem with two projects, the objective function would be:

$$\max \quad \sum_{s_1^T,s_2^T} v\left(s_1^T, s_2^T\right) p\left(s_1^T, s_2^T\right) = \sum_{j=1}^{3} \tilde{P}_j = \sum_{s_1^T}^{S-1} p\left(s_1^T, S\right) + \sum_{s_2^T}^{S-1} p\left(S, s_2^T\right) + p\left(S, S\right) \qquad (3.1)$$

Given the above definitions, we have the following general objective function:

$$\max \quad \sum_{s_1^T,s_2^T,...s_N^T} v\left(s_1^T, s_2^T,..., s_N^T\right) p\left(s_1^T, s_2^T,..., s_N^T\right) = \sum_{j=1}^{2^N-1} \tilde{P}_j \qquad (3.2)$$

[5] The set of initial states $\left(s_1^0, s_2^0,..., s_N^0\right)$ can be heterogeneous across projects *i*, though in our examples they are not.

In (3.2) the summation is only over the final time period, T, since success prior to T is maintained for all subsequent time periods.

The integer program with the objective function in (3.1) requires a number of nonlinear constraints. The first set of constraints defines the probabilities of terminating in the $2^N - 1$ possible success outcomes (i.e., at least on project in the state S) shown in (3.2).

In each time period, *t*, the set of projects N can be separated into two subsets, N_1^t and N_2^t, such that $N_1^t \cup N_2^t = N$ and $N_1^t \cap N_2^t = \varnothing$. N_1^t comprises the set of projects that were funded at time period *t*; N_2^t were the projects that were not funded at *t*. If a project is funded in a given time period *t* then the project was necessarily funded for all previous time periods, $1,...,t-1$. Therefore, $N_1^{t+1} \subseteq N_1^t$ for all *t*.

Let $D_1^t = \{1,2,...,|N_1^t|\}$ index the mapped elements of N_1^t; likewise, $D_2^t = \{1,2,...,|N_2^t|\}$ indexes the mapped elements of N_2^t. For example, for $N = 5$, if $N_1^t = \{1,3,5\}$ and $N_2^t = \{2,4\}$, then $D_1^t = \{1,2,3\}$ and $D_2^t = \{1,2\}$. Similarly, we define the states of the projects in D_1^t and D_2^t as $\tilde{D}_1^t$ and $\tilde{D}_2^t$, respectively. Thus, $\tilde{D}_1^t = \{s_1^{t-1}, s_2^{t-1},..., s_{D_1^t}^{t-1}\}$ and $\tilde{D}_2^t = 0$ for all elements of D_2^t. For each value of D_1^t, there are $\binom{N}{N_1^t}\left(S^{N_1^t}\right)$ possible $\tilde{D}_1^t$ sets.

We can now define a set of constraints that describe the success outcomes for every time period. Consider the probability, $p^t\left(\tilde{D}_1^t, \tilde{D}_2^t\right)$, of the event that at least one of the funded projects D_1^t achieved success for the first time in that time period, *t* (i.e., there were no successful projects at $t-1$ or earlier). Since the events are mutually exclusive, the constraints for all possible $\left(\tilde{D}_1^t, \tilde{D}_2^t\right)$ at time period, *t*, are:

$$p\left(\tilde{D}_1^T, \tilde{D}_2^T\right) = \sum_{t=1}^{T} p^t\left(\tilde{D}_1^t, \tilde{D}_2^t\right) \tag{3.3a}$$

where for any given time period, $\tilde{t}$, the probability of success is defined by (3.4) below.

To illustrate how constraint (3.4) works, we consider a simple example with three projects and three time periods, where each project has three states, $(0,1,2)$, where $S = 2$ and any unfunded project goes to state 0. As shown in Table 3.1, we consider the case where Projects 1 and 3 are funded in the first time period and Project 2 is not. Then, at $\tilde{t} = 2$, only Project 1 is funded. In other words,

$$\begin{aligned}
&\left(D_1^1, D_2^1\right) = \left((1,3),2\right) \text{(project indices)} \\
&\left(\tilde{D}_1^1, \tilde{D}_2^1\right) = \left((1,1),0\right) \text{(project states)} \\
&\left(D_1^2, D_2^2\right) = \left(1,(2,3)\right) \\
&\left(\tilde{D}_1^2, \tilde{D}_2^2\right) = \left(2,(0,0)\right)
\end{aligned} \tag{3.3b}$$

From Table 3.2, for each time period, Project 1 has probability of 0.6 to stay in state 1 and 0.4 to go to state 2 (success state), if funded. Project 3 has probability 0.5 for both staying in state 1 and going to state 2.

Table 3.1: A Funding Strategy for Obtaining $p^2\left(\tilde{D}_1^2, \tilde{D}_2^2\right) = p^2\left((2),(0,0)\right)$

Time Period	Project 1	Project 2	Project 3
1	Funded	Not Funded	Funded
2	Funded	Not Funded	Not Funded

Table 3.2: State Transition Probabilities for Three Projects, where $S = 2$

Project	P(s^{t+1} = 1 | s^t = 1, x^t = 1)	P(s^{t+1} = 2 [success] | s^t = 1, x^t = 1)
Project 1	0.6	0.4
Project 2	0.7	0.3
Project 3	0.5	0.5

According to (3.4), the probability of being in a success state for the first time at the end of $\tilde{t} = 2$, given the decisions in Table 3.1, is $p^2\left(\tilde{D}_1^2, \tilde{D}_2^2\right) = p^2\left((2),(0,0)\right) = (1 - (0.6))(0.5)(0.6) = 0.12$. This value, $p^2\left(\tilde{D}_1^2, \tilde{D}_2^2\right)$, is therefore equal to: i) 1 – probability Project 1 was unsuccessful in the second time period, multiplied by ii) the probability Projects 1 and 3 were unsuccessful in the first time period. However, this product must be multiplied by a product of binary decisions variables that is 1 if and only if $x_1^1 = x_3^1 = x_1^2 = 1$ and $x_2^1 = x_2^2 = x_3^2 = 0$. So,

$p^2\left(\tilde{D}_1^2, \tilde{D}_2^2\right) = p^2\left((2),(0,0)\right) = (1 - (0.6))(0.5)(0.6) = 0.12$ if and only if $x_1^1 = x_3^1 = x_1^2 = 1,\ x_2^2 = x_3^2 = 0$, and 0 otherwise.

In the more general case, we have the following:

$$
\begin{aligned}
&p^{\tilde{t}}\left(\tilde{D}_1^{\tilde{t}}, \tilde{D}_2^{\tilde{t}}\right) = A \times B \times C \times D, \text{ where:} \\
&A = \left(1 - \sum_{s_1^{\tilde{t}}, s_2^{\tilde{t}}, \ldots, s_{D_1^{\tilde{t}}}^{\tilde{t}}}^{S-1} \prod_{i=1}^{D_1^{\tilde{t}}} \pi_i^{\tilde{t}}\left(s_i^{\tilde{t}} \mid s_i^{\tilde{t}-1}\right)\right) \\
&B = \left(\prod_{t=1}^{\tilde{t}-1} \sum_{s_1^t, s_2^t, \ldots, s_{D_1^t}^t}^{S-1} \prod_{i=1}^{D_1^t} \pi_i^t\left(s_i^t \mid s_i^{t-1}\right)\right) \qquad (3.4) \\
&C = \left(\prod_{t=1}^{\tilde{t}} \prod_{i=1}^{D_1^{\tilde{t}}} x_{i,\left(\tilde{D}_1^t, \tilde{D}_2^t\right)}^t\right) \\
&D = \left(\prod_{i=1}^{D_2^{\tilde{t}}} 1 - x_{i,\left(\tilde{D}_1^{\tilde{t}-1}, \tilde{D}_2^{\tilde{t}-1}\right)}^{\tilde{t}}\right)
\end{aligned}
$$

In other words, $p^{\tilde{t}}\left(\tilde{D}_1^{\tilde{t}}, \tilde{D}_2^{\tilde{t}}\right)$ is the probability of achieving success for the first time at period $\tilde{t}$ for the set of funded and unfunded projects $\left(\tilde{D}_1^{\tilde{t}}, \tilde{D}_2^{\tilde{t}}\right)$. It is equal to the product of the following terms:

A. The probability of all successful outcomes for projects $\tilde{D}_1^{\tilde{t}}$ at time period $\tilde{t}$, which is simply 1 minus the probability of all unsuccessful outcomes (i.e., states 1 through $S-1$). In the example, it is: (1-(0.6)).

B. The probability of all outcomes where no project is in the success state, S, before time $\tilde{t}$ (i.e, $t = 1, \ldots, \tilde{t} - 1$), for all projects in $\tilde{D}_1^t$. In the example, it is $\left((0.6)(0.5)\right)$ for Projects 1 and 3 in time period 1.

C. A product of binary variables that equals 1 if the projects in the set $\tilde{D}_1^t$ have been funded up to and including time period $\tilde{t}$, and 0 otherwise. In the example, this is the product $x_1^1 x_3^1 x_1^2 = 1$.

D. The product of binary variables that equals 1 if the projects in the set $\tilde{D}_2^{\tilde{t}}$ are not funded at time period $\tilde{t}$, which is 1 only if *all* are not funded at time $\tilde{t}$, and 0 otherwise.

Thus, for a given $\left(\tilde{D}_1^{\tilde{t}}, \tilde{D}_2^{\tilde{t}}\right)$, the value $A \times B \times C \times D$ is nonzero if exactly projects $\tilde{D}_1^{\tilde{t}}$ are funded at time period, $\tilde{t}$.

The constraints (3.4) can be compactly expressed for each time period *t* by the recursive equation

$$P^{t+1} = M^t\left(X^t\right)P^t \tag{3.5}$$

where $M^t\left(X^t\right)$ is the matrix of all the decision variables (expressed in the vector X^t) and the probabilities $\pi_i^t\left(s_i^t \mid s_i^{t-1}\right)$, expressed as M^t , and P^t is the vector of probabilities.

This nonlinear, integer program also requires the following set of budget constraints:

$$\sum_{i=1}^{N} c_i^t x_{i,(s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1})}^t \le B_t \qquad \forall\, s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1} \in \{0, \ldots, S\} \tag{3.6}$$

in addition to the set of funding constraints:

$$x_{i,(s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1})}^t = 0 \quad \text{if any} \quad s_i^{t-1} = S \;\; \forall\, i \tag{3.7}$$

which ensures that no projects are funded once one or more projects achieve the success state.[6] One final set of constraints are the binary restrictions on the funding variables:

$$x^{t}_{i,(s_1^{t-1},s_2^{t-1},\ldots,s_N^{t-1})} \in \{0,1\} \;\; \forall i, \; \forall s_i^{t-1} \in \{0,1,\ldots,S\} \qquad (3.8)$$

This problem will always have an optimal solution, since i) there are a finite number of binary $x^{t}_{i,(s_1^{t-1},s_2^{t-1},\ldots,s_N^{t-1})}$, and ii) a feasible solution will always exist (e.g., fund no projects if budget is too small).

3.3 Two-Project, Two-Time Period Example

To illustrate the general formulation, consider a two-project, two-time period example ($N = 2$, $T = 2$) with possible project states $S = \{0,1,2,3,4\}$, where 4 denotes the success state and 0 denotes the state of the project in the second period if it was not funded in the first. Assuming the initial states of the projects and the transition probabilities are known, the following nonlinear, binary integer program results:

[6] These constraints are consistent with the assumption made in Chapter 2.

$$\max \quad \sum_{s_1^T, s_2^T} v(s_1^T, s_2^T)\, p(s_1^T, s_2^T) = \sum_{j=1}^{2^N-1} \widetilde{P}_j = \sum_{s_1^T=0}^{S-1} p(s_1^2, 4) + \sum_{s_2^T=0}^{S-1} p(4, s_2^2) + p(4,4) \tag{3.9a}$$

$$\text{s.t.} \quad p(s_1^2, 4) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(s_1^2 \mid s_1^1)\pi_2^2(4 \mid s_2^1) x_1^1 x_2^1 x_{1,(s_1^1,s_2^1)}^2 x_{2,(s_1^1,s_2^1)}^2 \quad \forall s_1^2 \in \{1,2,3\} \tag{3.9b}$$

$$p(4, s_2^2) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(4 \mid s_1^1)\pi_2^2(s_2^2 \mid s_2^1) x_1^1 x_2^1 x_{1,(s_1^1,s_2^1)}^2 x_{2,(s_1^1,s_2^1)}^2 \quad \forall s_2^2 \in \{1,2,3\} \tag{3.9c}$$

$$p(4,4) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(4 \mid s_1^1)\pi_2^2(4 \mid s_2^1) x_1^1 x_2^1 x_{1,(s_1^1,s_2^1)}^2 x_{2,(s_1^1,s_2^1)}^2 + \pi_1^1(4)\pi_2^1(4) x_1^1 x_2^1$$

$$p(4,0) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(4 \mid s_1^1) x_1^1 x_2^1 x_{1,(s_1^1,s_2^1)}^2 \left(1 - x_{2,(s_1^1,s_2^1)}^2\right) + \pi_1^1(4) x_1^1 + \tag{3.9d}$$

$$\sum_{s_1^1=1}^{3} \pi_1^1(s_1^1)\pi_1^2(4 \mid s_1^1) x_1^1 (1 - x_2^1) x_{1,(s_1^1,0)}^2$$

$$p(0,4) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_2^2(4 \mid s_2^1) x_1^1 x_2^1 \left(1 - x_{1,(s_1^1,s_2^1)}^2\right) x_{2,(s_1^1,s_2^1)}^2 + \pi_2^1(4) x_2^1 + \tag{3.9e}$$

$$\sum_{s_2^1=1}^{3} \pi_2^1(s_2^1)\pi_2^2(4 \mid s_2^1)(1 - x_1^1) x_2^1 x_{2,(0,s_2^1)}^2$$

$$x_i^1 \ge x_{i,(s_1^1,s_2^1)}^2 \quad i = 1,2 \quad \forall\, s_1^1, s_2^1 \in \{0,1,2,3,4\} \tag{3.9f}$$

$$B_1 \ge c_1^1 x_1^1 + c_2^1 x_2^1 \tag{3.9g}$$

$$B_2 \ge c_1^2 x_{1,(s_1^1,s_2^1)}^2 + c_2^2 x_{2,(s_1^1,s_2^1)}^2 \quad \forall s_1^1, s_2^1 \in \{0,1,2,3,4\} \tag{3.9h}$$

$$x_{i,(4,s_2^1)}^2 = 0 \quad i = 1,2 \quad \forall s_2^1 \in \{0,1,2,3,4\} \tag{3.9i}$$

$$x_{i,(s_1^1,4)}^2 = 0 \quad i = 1,2 \quad \forall s_1^1 \in \{0,1,2,3,4\} \tag{3.9j}$$

$$x_i^1, x_{i,(s_1^1,s_2^1)}^2 \in \{0,1\} \quad i = 1,2 \quad \forall s_1^1, s_2^1 \in \{0,1,2,3,4\} \tag{3.9k}$$

The first five constraints (3.9a)-(3.9e) define the probabilities of ending up in a success state (i.e., the state 4). Additionally, the following constraints are added:

- No funding a project in the second time period if it was not funded in the first (3.9f)
- Budget constraints (3.9g)-(3.9h)
- Funding ceases once a project has achieved a success state (3.9i)-(3.9j)

This optimization problem could be solved directly using an integer programming method, noting that the above integer program is nonlinear. However, all nonlinearities involve the product of a series of binary variables. In the following subsection, we demonstrate our technique for making all of these constraints linear by adding additional constraints and variables.

3.3.1 Removing Nonlinearities in the Constraints

The above nonlinear problem can be transformed to make it computationally easier to solve. Though it will still remain an integer program, we demonstrate how to remove all nonlinearities from the integer program in this section. This approach is a specific case of results demonstrated by Glover and Woolsey (1974). Moreover, the additional variables that are required to remove the nonlinearities grow exponentially with the number of time periods and projects; but they only need linear restrictions, not integer ones. We prove that property in the next subsection.

We first note that the above nonlinear, integer program has different types of decision variables multiplied by one another. There are cases where we multiply:

- both projects' first and second time period decision variables (four binary variables)
- both projects' first time period decision variables, but only one project's second time period (three binary variables)
- both projects' first time period decision variables only (two binary variables)

Linearizing Two Binary Variables

There are cases where the constraints contain the product of two binary variables, such as (3.9c). These represent the constraints on decisions of the first time period. These bilinear terms can be handled by introducing a new variable and constraints.

For example, in (3.9c), a new variable w_1 is defined so that $w_1 = 1$ if and only if $x_1 = x_2 = 1$. Specifically, we add the following constraints (see Williams, 1999 for this and other logic constraints and Yu *et al*., 2008 for a specific example in transportation modeling):

$$
\begin{aligned}
& w_1 \le x_1^1 \\
& w_1 \le x_2^1 \\
& w_1 \ge x_1^1 + x_2^1 - 1
\end{aligned}
\tag{3.10}
$$

The other bilinear terms in (3.2) can be handled in a similar way:

$$
\begin{aligned}
& w_2 \le x_1^1 \\
& w_2 \le \left(1 - x_2^1\right) \\
& w_2 \ge x_1^1 + \left(1 - x_2^1\right) - 1 \\
& \\
& w_3 \le \left(1 - x_1^1\right) \\
& w_3 \le x_2^1 \\
& w_3 \ge \left(1 - x_1^1\right) + x_2^1 - 1 \\
& w_1, w_2, w_3 \in \{0,1\}
\end{aligned}
\tag{3.11}
$$

Linearizing Three Binary Variables

The two-project, two-time period integer program has constraints where there is the product of three binary variables. These constraints result from the possibility that one of the two projects is not funded in the first time period and the other project is funded in both, for example (3.9d). Linearization variables and constraints are added in a similar fashion to the product of two binary variables. The set of variables y_{1,s_1^1} is defined as $y_{1,s_1^1} = 1 \Leftrightarrow x_1^1 = \left(1 - x_2^1\right) = x_{1,\left(s_1^1,0\right)}^2 = 1$ by adding the constraints:

$$
\begin{aligned}
&y_{1,s_1^1} \le x_1^1 \\
&y_{1,s_1^1} \le \left(1 - x_2^1\right) \\
&y_{1,s_1^1} \le x_{1,(s_1^1,0)}^2 \\
&y_{1,s_1^1} \ge x_1^1 + \left(1 - x_2^1\right) + x_{1,(s_1^1,0)}^2 - 2 \\
&y_{1,s_1^1} \in \{0,1\}
\end{aligned}
\tag{3.12}
$$

Similarly, the y_{2,s_2^1} variables are defined as $y_{2,s_2^1} = 1 \Leftrightarrow \left(1 - x_1^1\right) = x_2^1 = x_{2,\left(0,s_2^1\right)}^2 = 1$ with the constraints:

$$
\begin{aligned}
&y_{2,s_2^1} \le \left(1 - x_1^1\right) \\
&y_{2,s_2^1} \le x_2^1 \\
&y_{2,s_2^1} \le x_{2,(0,s_2^1)}^2 \\
&y_{2,s_2^1} \ge x_1^1 + \left(1 - x_2^1\right) + x_{2,(0,s_2^1)}^2 - 2 \\
&y_{2,s_2^1} \in \{0,1\}
\end{aligned}
\tag{3.13}
$$

Linearizing Four Binary Variables

For the constraints that have the product of all four decision variables, e.g., $x_1^1 x_2^1 x_{1,(s_1^1,s_2^1)}^2 x_{2,(s_1^1,s_2^1)}^2$ in (3.9a), a new of z variables is used. Specifically, $z_{1,(s_1^1,s_2^1)}$ is defined as $z_{1,(s_1^1,s_2^1)} = 1 \Leftrightarrow x_1^1 = x_2^1 = x_{1,(s_1^1,s_2^1)}^2 = x_{2,(s_1^1,s_2^1)}^2 = 1$. This definition for $z_{1,(s_1^1,s_2^1)}$ is done by adding the following constraints:

$$
\begin{aligned}
&z_{1,(s_1^1,s_2^1)} \le x_1^1 \\
&z_{1,(s_1^1,s_2^1)} \le x_2^1 \\
&z_{1,(s_1^1,s_2^1)} \le x_{1,(s_1^1,s_2^1)}^2 \\
&z_{1,(s_1^1,s_2^1)} \le x_{2,(s_1^1,s_2^1)}^2 \\
&z_{1,(s_1^1,s_2^1)} \ge x_1^1 + x_2^1 + x_{1,(s_1^1,s_2^1)}^2 + x_{2,(s_1^1,s_2^1)}^2 - 3 \\
&z_{1,(s_1^1,s_2^1)} \in \{0,1\}
\end{aligned}
\tag{3.14}
$$

The $z_{1,(s_1^1,s_2^1)}$ variables eliminate the need to have the product of the four binary variables ($x_1^1 x_2^1 x_{1,(s_1^1,s_2^1)}^2 x_{2,(s_1^1,s_2^1)}^2$) in the constraints. In the next section, we prove that $z_{1,(s_1^1,s_2^1)}$ and the other linearization variables (w , y) need only be linear variables and not binary. For now, however, we assume that $z_{1,(s_1^1,s_2^1)} \in \{0,1\}$.

Following the approach above, we set $z_{2,(s_1^1,s_2^1)} = 1$ if and only if $x_1^1 = x_2^1 = x_{1,(s_1^1,s_2^1)}^2 = \left(1 - x_{2,(s_1^1,s_2^1)}^2\right) = 1$ (in other words, $x_{2,(s_1^1,s_2^1)}^2 = 0$) by adding the constraints:

$$
\begin{aligned}
& z_{2,(s_1^1,s_2^1)} \le x_1^1 \\
& z_{2,(s_1^1,s_2^1)} \le x_2^1 \\
& z_{2,(s_1^1,s_2^1)} \le x_{1,(s_1^1,s_2^1)}^2 \\
& z_{2,(s_1^1,s_2^1)} \le \left(1 - x_{2,(s_1^1,s_2^1)}^2\right) \\
& z_{2,(s_1^1,s_2^1)} \ge x_1^1 + x_2^1 + x_{1,(s_1^1,s_{22}^1)}^2 + \left(1 - x_{2,(s_1^1,s_2^1)}^2\right) - 3 \\
& z_{2,(s_1^1,s_2^1)} \in \{0,1\}
\end{aligned}
\tag{3.15}
$$

Likewise, the case when $x_1^1 = x_2^1 = \left(1 - x_{1,(s_1^1,s_2^1)}^2\right) = x_{2,(s_1^1,s_2^1)}^2 = 1$ is handled by introducing the variables $z_{3,(s_1^1,s_2^1)}$ and the constraints:

$$
\begin{aligned}
& z_{3,(s_1^1,s_2^1)} \le x_1^1 \\
& z_{3,(s_1^1,s_2^1)} \le x_2^1 \\
& z_{3,(s_1^1,s_2^1)} \le \left(1 - x_{1,(s_1^1,s_{22}^1)}^2\right) \\
& z_{3,(s_1^1,s_2^1)} \le x_{2,(s_1^1,s_2^1)}^2 \\
& z_{3,(s_1^1,s_2^1)} \ge x_1^1 + x_2^1 + \left(1 - x_{1,(s_1^1,s_2^1)}^2\right) + x_{2,(s_1^1,s_2^1)}^2 - 3 \\
& z_{3,(s_1^1,s_2^1)} \in \{0,1\}
\end{aligned}
\tag{3.16}
$$

The end result is that the two-project, two-time period real options problem modeled in (3.9) can be formulated as the following *linear*, integer program:

$$
\begin{aligned}
\max \quad & \sum_{s_1^T, s_2^T} v(s_1^T, s_2^T)\, p(s_1^T, s_2^T) = \sum_{j=1}^{2^N-1} \widetilde{P}_j = \sum_{s_1^T=0}^{S-1} p(s_1^2, 4) + \sum_{s_2^T=0}^{S-1} p(4, s_2^2) + p(4,4) \\
\text{s.t.} \quad & p(s_1^2, 4) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(s_1^2 \mid s_1^1)\pi_2^2(4 \mid s_2^1) z_{1,(s_1^1,s_2^1)} \quad \forall s_1^2 \in \{1,2,3\} \\
& p(4, s_2^2) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(4 \mid s_1^1)\pi_2^2(s_2^2 \mid s_2^1) z_{1,(s_1^1,s_2^1)} \quad \forall s_2^2 \in \{1,2,3\} \\
& p(4,4) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(4 \mid s_1^1)\pi_2^2(4 \mid s_2^1) z_{1,(s_1^1,s_2^1)} + \pi_1^1(4)\pi_2^1(4) w_1 \\
& p(4,0) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(4 \mid s_1^1) z_{2,(s_1^1,s_2^1)} + \pi_1^1(4) x_1^1 + \\
& \qquad \sum_{s_1^1=1}^{3} \pi_1^1(s_1^1)\pi_1^2(4 \mid s_1^1) y_{1,s_1^1} \\
& p(0,4) = \sum_{s_1^1, s_2^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_2^2(4 \mid s_2^1) z_{3,(s_1^1,s_2^1)} + \pi_2^1(4) x_2^1 + \\
& \qquad \sum_{s_2^1=1}^{3} \pi_2^1(s_2^1)\pi_2^2(4 \mid s_2^1) y_{2,s_2^1} \\
& z_{1,(s_1^1,s_2^1)} \le x_i^1 \qquad z_{2,(s_1^1,s_2^1)} \le x_i^1 \qquad z_{3,(s_1^1,s_2^1)} \le x_i^1 \quad i = 1,2 \\
& z_{1,(s_1^1,s_2^1)} \le x_{1,(s_1^1,s_2^1)}^2 \qquad z_{2,(s_1^1,s_2^1)} \le x_{1,(s_1^1,s_2^1)}^2 \qquad z_{3,(s_1^1,s_2^1)} \le \left(1 - x_{1,(s_1^1,s_2^1)}^2\right) \\
& z_{1,(s_1^1,s_2^1)} \le x_{2,(s_1^1,s_2^1)}^2 \qquad z_{2,(s_1^1,s_2^1)} \le \left(1 - x_{2,(s_1^1,s_2^1)}^2\right) \quad z_{3,(s_1^1,s_2^1)} \le x_{2,(s_1^1,s_2^1)}^2 \\
& z_{1,(s_1^1,s_2^1)} \ge x_1^1 + x_2^1 + x_{1,(s_1^1,s_2^1)}^2 + x_{2,(s_1^1,s_2^1)}^2 - 3 \\
& z_{2,(s_1^1,s_2^1)} \ge x_1^1 + x_2^1 + x_{1,(s_1^1,s_2^1)}^2 + \left(1 - x_{2,(s_1^1,s_{22}^1)}^2\right) - 3 \\
& z_{3,(s_1^1,s_2^1)} \ge x_1^1 + x_2^1 + \left(1 - x_{1,(s_1^1,s_{22}^1)}^2\right) + x_{2,(s_1^1,s_{22}^1)}^2 - 3 \\
& y_{1,s_1^1} \le x_1^1 \qquad y_{1,s_2^1} \le \left(1 - x_1^1\right) \qquad y_{1,s_1^1} \le \left(1 - x_2^1\right) \qquad y_{1,s_2^1} \le x_2^1 \\
& y_{1,s_1^1} \le x_{1,(s_1^1,0)}^2 \qquad y_{1,s_2^1} \le x_{2,(0,s_2^1)}^2 \\
& y_{1,s_1^1} \ge x_1^1 + \left(1 - x_2^1\right) + x_{1,(s_1^1,0)}^2 - 2 \qquad y_{1,s_2^1} \ge x_1^1 + \left(1 - x_2^1\right) + x_{2,(0,s_2^1)}^2 - 2 \\
& x_i^1 \ge x_{i,(s_1^1,s_2^1)}^2 \quad i = 1,2 \quad \forall\, s_1^1, s_2^1 \in \{0,1,2,3,4\} \\
& B_1 \ge c_1^1 x_1^1 + c_2^1 x_2^1 \\
& B_2 \ge c_1^2 x_{1,(s_1^1,s_2^1)}^2 + c_2^2 x_{2,(s_1^1,s_2^1)}^2 \quad \forall s_1^1, s_2^1 \in \{0,1,2,3,4\} \\
& x_{i,(4,s_2^1)}^2 = 0 \quad i = 1,2 \quad \forall s_2^1 \in \{0,1,2,3,4\} \\
& x_{i,(s_1^1,4)}^2 = 0 \quad i = 1,2 \quad \forall s_1^1 \in \{0,1,2,3,4\} \\
& x_i^1, x_{i,(s_1^1,s_2^1)}^2 \in \{0,1\} \quad i = 1,2 \quad \forall s_1^1, s_2^1 \in \{0,1,2,3,4\} \\
& 0 \le y \le 1 \quad 0 \le z \le 1
\end{aligned}
\tag{3.17}
$$

It is important to note that the IP in (3.17) involves fixed, individual budgets for each time period; it is not effective for the flexible allocations. In the following section we show why the constraints that $0 \le w \le 1$, $0 \le y \le 1$, and $0 \le z \le 1$ are sufficient and the binary restrictions are not necessary. As a result, the number of binary (or integer) decision variables is limited to only the number of decisions at each time period given the current state.

3.3.2 Proof that the Linearization Variables Need No Binary Restrictions

The general formulation in Section 3.2, as explicitly shown in the two-project, two-time period example, contains the products of binary variables, resulting in nonlinear, non-convex constraints. As demonstrated in the previous section, this nonlinear problem can be transformed to potentially make it computationally much easier to solve. It is worth noting that the number of additional variables that are required to remove the nonlinearities grow exponentially with the number of time periods and projects; but these additional variables only need linear restrictions, not integer ones.

The construction of the *w*, *y*, and *z* variables are identical in structure. Namely, given a product of a series of *n* binary variables, $x_1 x_2 \cdots x_n$, the values of this product are such that either $x_1 x_2 \cdots x_n = 0$ or $x_1 x_2 \cdots x_n = 1$. Moreover, $x_1 x_2 \cdots x_n = 1$ if and only if $x_1 = x_2 = \ldots = x_n = 1$. Therefore, we can define a variable, *u*, which will be equivalent to $x_1 x_2 \cdots x_n$, by adding the following constraints:

$$\begin{aligned}
& u \le x_1 \\
& u \le x_2 \\
& \quad \vdots \\
& u \le x_n \\
& u \ge x_1 + \ldots + x_n - (n-1) \\
& u, x_1, \ldots, x_n \in \{0,1\}
\end{aligned} \tag{3.18}$$

The above constraints ensure that $u = 0$ if and only if $x_1 x_2 \cdots x_n = 0$ and $u = 1$ if and only $x_1 x_2 \cdots x_n = 1$. Extending Gabriel and Leuthold's (2010) method of linearizing binary variables to the linearization of products of binary variables, we claim that the solution for u in (3.18) is equivalent to its value in (3.19).

$$\begin{aligned}
& u \le x_1 \\
& u \le x_2 \\
& \quad \vdots \\
& u \le x_n \\
& u \ge x_1 + \ldots + x_n - (n-1) \\
& 0 \le u \le 1 \\
& x_1, \ldots, x_n \in \{0,1\}
\end{aligned} \tag{3.19}$$

Theorem 3.1: *The solution set for u in (3.18) is equivalent to its value in (3.19).*

Proof: Since the values $0 \text{ and } 1 \in [0,1]$, clearly $SOL(3.18) \subseteq SOL(3.19)$. To show $SOL(3.19) \subseteq SOL(3.18)$, we suppose not. Then, $u \notin \{0,1\}$ implies $0 < u < 1$. Since $u > 0$, then by the binary restrictions on the x variables, $x_1 = \ldots = x_n = 1$, since $x_i \ge u$ for all $i = 1, \ldots n$. However, the constraint that $u \ge x_1 + \ldots + x_n - (n-1)$ implies that $u \ge 1 + \ldots + 1 - (n-1) = n - (n-1) = 1$. Thus, for any $u > 0 \Rightarrow u \ge 1$, which implies that $u = 1$, and contradicts our claim that $0 < u < 1$. ■

The reason for applying these linearization techniques is to be able to reduce run-times in optimization packages that can solve (to global optimality) both linear and nonlinear integer programs. For the numerical examples outlined later in this chapter, the run-times for the globally optimal, nonlinear formulations were dramatically longer than their linearized equivalents. While these problems were not tested for every solver package, linearizing the integer formulation can provide an increase in solution speed—at least for some solvers. Of course, the linearization constraints must not exceed the virtual memory accessible to the solver.

3.4 Three-Project, Three-Time Period Problem

For the purposes of considering a larger, more realistic setting, this section describes a three-project, three-time period problem, where S is the set of states such that $S = \{0,1,2,3,4\}$, identical to the set S in the two-project, two-time period problem outlined in Section 3.3. Our objective function for this problem is also similar to the two-project, two-time period case, namely,

$$\begin{aligned}
&\max \sum_{s_1^T, s_2^T, s_3^T} v(s_1^T, s_2^T, s_3^T)\, p(s_1^T, s_2^T, s_3^T) = \sum_{j=1}^{2^N-1} \tilde{P}_j = \\
&\sum_{s_2^3, s_3^3=0}^{3} p^3(4, s_2^3, s_3^3) + \sum_{s_1^3, s_3^3=0}^{3} p^3(s_1^3, 4, s_3^3) + \sum_{s_1^3, s_2^3=0}^{3} p^3(s_1^3, s_2^3, 4) + \\
&\sum_{s_1^3=0}^{3} p^3(s_1^3, 4, 4) + \sum_{s_2^3=0}^{3} p^3(4, s_2^3, 4) + \sum_{s_3^3=0}^{3} p^3(4, 4, s_3^3) + p^3(4,4,4)
\end{aligned} \tag{3.20}$$

Using the general time period constraints we could set up the constraints provided in (3.5) for each time period, along with appropriate budget constraints, i.e.,

$$\begin{aligned}
&\sum_i c_i^1 x_i^1 \le B_1 \\
&\sum_i c_i^2 x_{i,(s_1^1, s_2^1, s_3^1)}^2 \le B_2 \\
&\sum_i c_i^3 x_{i,(s_1^2, s_2^2, s_3^2)}^3 \le B_3
\end{aligned} \tag{3.21}$$

and solve for the optimal funding decisions. However, this requires solving a nonlinear, non-convex optimization problem with nonlinear constraints. As shown previously, we can linearize these constraints. For a three-project, three-time period problem the constraints are more complicated than for two projects in two time periods, because the number of linear variables and linearization constraints grows exponentially with both projects *and* time periods. The additional variables and constraints occur due to the tracking of all decisions over three time periods for three projects. The specific constraints are provided in the next section.

3.4.1 Constraint Construction

Our objective function is to maximize the probability of being in a successful state at the end of the third time period. These can be obtained directly from the funding decision variables ($x_i^1, x^2_{i,(s_1^1,s_2^1,s_3^1)}$, and $x^3_{i,(s_1^2,s_2^2,s_3^2)}$, again where $x^t_{i,(s_1^{t-1},s_2^{t-1},s_3^{t-1})}$ represents the funding decision for project *i* at time *t* given that three projects achieved state $\left(s_1^{t-1},s_2^{t-1},s_3^{t-1}\right)$ at the end of the previous time period) and the probabilities of success. However, to avoid nonlinearities, variables that map to every funding decision over all three time periods are needed. For each component of the objective function (e.g., $p(0,4,2)$) it is necessary to define the funding decisions and transition probabilities under which that state could be reached. It is also necessary to construct linearization constraints $w, y,$ and z to represent reaching a success state in the first, second or third time period, respectively.

w Constraints

In the first time period, we must decide whether to fund the projects (i.e., choose the values for (x_1^1, x_2^1, x_3^1)). This set of funding decisions can be represented by a set of constraints w_j where $j = 1, \ldots, 8$. The w constraints correspond to the $2^3 = 8$ funding decisions possible in the first time period. For example, the constraints:

$$\begin{aligned} &w_1 \le x_1^1 \\ &w_1 \le x_2^1 \\ &w_1 \le x_3^1 \\ &w_1 \ge x_1^1 + x_2^1 + x_3^1 - 2 \\ &0 \le w_1 \le 1 \end{aligned} \tag{3.22}$$

ensure that $w_1 = 1 \Leftrightarrow x_1^1 = 1, x_2^1 = 1, x_3^1 = 1$, or $(1,1,1)$. We can assume these w (along with the subsequent y and z) variables are linear, rather than binary, due to the proof in Section 3.3. Similar constraints are constructed for the other w_j constraints such that:

$$\begin{aligned} &w_2 = 1 \Leftrightarrow (1,1,1) \\ &w_2 = 1 \Leftrightarrow (1,1,0) \\ &w_3 = 1 \Leftrightarrow (1,0,1) \\ &w_4 = 1 \Leftrightarrow (0,1,1) \\ &w_5 = 1 \Leftrightarrow (1,0,0) \\ &w_6 = 1 \Leftrightarrow (0,1,0) \\ &w_7 = 1 \Leftrightarrow (0,0,1) \\ &w_8 = 1 \Leftrightarrow (0,0,0) \end{aligned} \tag{3.23}$$

As we will observe in Section 3.5, branching on all combinations of first time period funding decisions greatly decreases the run-times of the IP optimization. Thus, it is an important property that the w_j variables are mutually exclusive. That is, there exists only one j such that $w_j = 1$, and therefore, $\sum_j w_j = 1$. Thus, the integer programming formulation need only iterate on, at most, eight first-stage funding possibilities.

y Constraints

Identical reasoning can be applied to the y and z constraints. There are seven sets of y variables corresponding to the $2^3 - 1$ non-trivial funding decisions made in the first time period (we do not denote the trivial case where no projects were funded, as there would be no second time period decisions to be made). For example, the $y_{1,(s_1^1,s_2^1,s_3^1),j}$ (where $j = 1,\ldots,2^3 = 8$) variables correspond the decision of which projects to fund in the second time period given that: (a) all three projects were funded in the first time period (i.e., $w_1 = 1$) and (b) the state at the beginning of time period 2 is denoted as $\left(s_1^1, s_2^1, s_3^1\right)$. The set of decision of which projects to fund in the second time period (i.e., $\left(x^2_{1,(s_1^1,s_2^1,s_3^1)}, x^2_{2,(s_1^1,s_2^1,s_3^1)}, x^2_{3,(s_1^1,s_2^1,s_3^1)}\right)$), can be represented by:

$$\begin{aligned}
&y_{1,(s_1^1,s_2^1,s_3^1),1} = 1 \Leftrightarrow (1,1,1) \\
&y_{1,(s_1^1,s_2^1,s_3^1),2} = 1 \Leftrightarrow (1,1,0) \\
&\vdots \\
&y_{1,(s_1^1,s_2^1,s_3^1),8} = 1 \Leftrightarrow (0,0,0)
\end{aligned} \tag{3.24}$$

Similar y variables and constraints are constructed for cases where other first time period funding situations arise, i.e., when $w_2 = 1$, $w_3 = 1$, …, $w_7 = 1$. Again, the trivial case ($w_8 = 1$) needs no second time period constraints, since no projects were funded in the first time period. The index on *j* varies on the number of possible projects available to be funded in that case. For the case where $w_2 = 1$, we have $y_{2,(s_1^1,s_2^1,0),j}$, where $j = 1,\ldots,2^2 = 4$, since $x^2_{3,(s_1^1,s_2^1,0)} \equiv 0$ (i.e., project 3 cannot be funded in the second time period if it was not funded in the first time period). For $y_{2,(s_1^1,s_2^1,0),j}$, the constraints are:

$$\begin{aligned} y_{2,(s_1^1,s_2^1,0),1} &= 1 \Leftrightarrow (1,1) \\ y_{2,(s_1^1,s_2^1,0),2} &= 1 \Leftrightarrow (1,0) \\ y_{2,(s_1^1,s_2^1,0),3} &= 1 \Leftrightarrow (0,1) \\ y_{2,(s_1^1,s_2^1,0),3} &= 1 \Leftrightarrow (0,0) \end{aligned} \tag{3.25}$$

Corresponding definitional constraints can be constructed for $y_{3,(s_1^1,0,s_3^1),j}$ and $y_{4,(0,s_2^1,s_3^1),j}$. In these cases, $y_{3,(s_1^1,0,s_3^1),j}$ corresponds to the case where projects 1 and 3 were funded in the first time period (and achieved state $\left(s_1^1, 0, s_3^1\right)$); $y_{4,(0,s_2^1,s_3^1),j}$ corresponds the case where projects 2 and 3 were funded in the first time period.

Similar cases where only one project was funded in the first time period can be constructed. For the case where the first project was the only one funded, there are variables $y_{5,(s_1^1,0,0),j}$, where $j = 1,2$. The constraints are:

$$\begin{aligned} y_{5,(s_1^1,0,0),1} &= 1 \Leftrightarrow x^2_{1,(s_1^1,0,0)} = 1 \\ y_{5,(s_1^1,0,0),2} &= 1 \Leftrightarrow x^2_{1,(s_1^1,0,0)} = 0 \end{aligned} \tag{3.26}$$

Finally, we create corresponding constraints for the variables $y_{6,(0,s_2^1,0),j}$ and $y_{7,(0,0,s_3^1),j}$, to represent the decision variables when only project 2 and only project 3 were funded in the first time period, respectively.

z Constraints

The z constraints map to decisions over all three time periods. Since the z variables consider the funding over the outcome of both the first and second stages, the variables must include the relevant state information for each of the funding decisions. This makes the set of z variables much larger than y, which in turn was larger than the number of w variables. Consider the case where both $w_1 = 1$ and $y_{1,(s_1^1,s_2^1,s_3^1),1} = 1$. Since all three projects were funded in the second time period, the state of the system at the beginning of the third time period (i.e., the end of the second time period) is denoted as (s_1^2, s_2^2, s_3^2). The set of variables $z_{1,(s_1^1,s_2^1,s_3^1,s_1^2,s_2^2,s_3^2),j}$, where $j = 1,\ldots,2^3 = 8$ corresponds to the third time period funded decisions $(x^3_{1,(s_1^2,s_2^2,s_3^2)}, x^3_{2,(s_1^2,s_2^2,s_3^2)}, x^3_{3,(s_1^2,s_2^2,s_3^2)})$ in the following manner:

$$\begin{aligned} &z_{1,(s_1^1,s_2^1,s_3^1,s_1^2,s_2^2,s_3^2),1} = 1 \Leftrightarrow (1,1,1) \\ &z_{1,(s_1^1,s_2^1,s_3^1,s_1^2,s_2^2,s_3^2),2} = 1 \Leftrightarrow (1,1,0) \\ &\qquad\vdots \\ &z_{1,(s_1^1,s_2^1,s_3^1,s_1^2,s_2^2,s_3^2),8} = 1 \Leftrightarrow (0,0,0) \end{aligned} \tag{3.27}$$

when $w_1 = 1$ and $y_{1,(s_1^1,s_2^1,s_3^1),1} = 1$. Similar constructions exist for, as an example, $z_{2,(s_1^1,s_2^1,s_3^1,s_1^2,s_2^2,0),j}$, $\mathrm{j} = 1,\ldots,2^2 = 4$ for $w_1 = 1$ and $y_{1,(s_1^1,s_2^1,s_3^1),2} = 1$. Since $y_{1,(s_1^1,s_2^1,s_3^1),2} = 1$, our only decision in the third time period is whether to fund the first two projects (i.e., $x^3_{1,(s_1^2,s_2^2,0)}, x^3_{2,(s_1^2,s_2^2,0)}$, since $x^3_{3,(s_1^2,s_2^2,0)} \equiv 0$).

It is worth noting that we would, in theory, need to construct sets of z constraints for each possible y variable with a j index value. Since $y_{1,(s_1^1,s_2^1,s_3^1),j}$ has $j=1,\ldots,8$, $y_{2,(s_1^1,s_2^1,0),j}, y_{3,(s_1^1,0,s_3^1),j}, y_{4,(0,s_2^1,s_3^1),j}$ has $j=1,\ldots,4$, and $y_{5,(s_1^1,0,0),j}, y_{6,(0,s_2^1,0),j}, y_{7,(0,0,s_3^1),j}$ has $j=1,2$, there could be $8+4+4+4+2+2+2=26$ set of z constraints. However, several of these funding decision paths do not need to be defined by z variables, as there is no nontrivial funding decision to make at that point. For example, if $y_{1,(s_1^1,s_2^1,s_3^1),8}=1$, which corresponds to $\left(x^2_{1,(s_1^1,s_2^1,s_3^1)}, x^2_{2,(s_1^1,s_2^1,s_3^1)}, x^2_{3,(s_1^1,s_2^1,s_3^1)}\right)=(0,0,0)$, then no project can be funded in the third time period since none were funded in the second. Thus, the corresponding z variables will not be used in any of the linear constraint construction, as it does represent a meaningful funding situation in the third time period. Nevertheless, for consistency of notation in mapping to previous time periods, the z variables map according to the 26 possible indices. For example, if $w_3=1$ (first and third projects were funded in the first time period), and $y_{3,(s_1^1,0,s_3^1),1}=1$ (both were funded in the second time period as well), the set of variables $z_{13,(s_1^1,0,s_3^1,s_1^2,0,s_3^2),j}$ is such that for the pair of binary variables $\left(x^3_{1,(s_1^2,0,s_3^2)}, x^3_{3,(s_1^2,0,s_3^2)}\right)$, the following mapping occurs:

$$
\begin{aligned}
z_{13,(s_1^1,0,s_3^1,s_1^2,0,s_3^2),1} &= 1 \Leftrightarrow (1,1) \\
z_{13,(s_1^1,0,s_3^1,s_1^2,0,s_3^2),2} &= 1 \Leftrightarrow (1,0) \\
z_{13,(s_1^1,0,s_3^1,s_1^2,0,s_3^2),3} &= 1 \Leftrightarrow (0,1) \\
z_{13,(s_1^1,0,s_3^1,s_1^2,0,s_3^2),4} &= 1 \Leftrightarrow (0,0)
\end{aligned}
\tag{3.28}
$$

In other words, the set of z_{13} variables represents the first funding decision in the second time period (i.e., $(1,1)$) and the third funding possibility in the first time period (i.e., $(x_1^1, x_2^1, x_3^1) = (1,0,1)$). Thus, it is $8+4+1=13$ in the ordering scheme.

As stated previously, the number of x binary decision variables grows exponentially with the number of projects and possible state outcomes. On the other hand, the linearization variables ($w, y,$ and z) grow exponentially with projects, state outcomes *and* time periods, since it is necessary to include a funding history from the first time period to the final time period those linearization constraints cover. As a result, the number of linear variables for even small problems can become quite large. For instance, while the number of binary variables for a three-project, three-time period problem with five possible outcomes for each project (including the "not funded" outcome) is $3+3(5^3)+3(5^3)=753$, the number[7] of continuous variables is $O(10^4)$.

3.4.2 Linking the Objective Function to the Funding Constraints

The successful state probabilities that define the objective function must be linked to the linearization constraints. For the sake of brevity, we show how to construct some sample constraints, rather than all of the constraints, though all follow a similar logic. The LINGO code for the full integer program is available in Appendix B.

For example, the probability of reaching the state $(S,S,S)=(4,4,4)$ (which represents all three projects being in the success state) is calculated by the variable

[7] This notation follows computational complexity descriptions found in Nemhauser and Wolsey (1988).

$p(4,4,4)$. Since the assumption is that no projects are funded once one or more has reached the success state, the only way $\left(s_1^3, s_2^3, s_3^3\right) = (4,4,4)$ is for all three projects to reach success is simultaneously. Thus, all three projects must achieve state 4 after the first stage, or the second stage or the third stage. In other words, $p(4,4,4) =$

- all projects reached state 4 after being funded in the first stage:

$$\pi_1^1(4)\pi_2^1(4)\pi_3^1(4)w_1 \tag{3.29a}$$

- or, no projects reached state 4 in the first stage and they were all funded in the second stage and then they all reached 4:

$$\sum_{s_1^1,s_2^1,s_3^1=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_3^1(s_3^1)\pi_1^2(s_1^1,4)\pi_2^2(s_2^1,4)\pi_3^2(s_3^1,4)y_{1,(s_1^1,s_2^1,s_3^1),1} \tag{3.29b}$$

- or, no projects reached it in the first or second stage, but they all reached state 4 (and were all funded) in the final (third) stage:

$$\sum_{s_1^1,s_2^1,s_3^1=1}^{3} \sum_{s_1^2,s_2^2,s_3^2=1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_3^1(s_3)\pi_1^2(s_1^1,s_1^2)\pi_2^2(s_2^1,s_2^2)\pi_3^2(s_3^1,s_3^2)\pi_1^3(s_1^2,4)\pi_2^3(s_2^2,4)\pi_3^3(s_3^2,4)z_{1,(s_1^1,s_2^1,s_3^1,s_1^2,s_2^2,s_3^2),1} \tag{3.29c}$$

The constraints are the sums of mutually exclusive scenarios, so the probabilities can be summed without over-counting.

For the $p(2,4,0)$ element, it is important to note that the system can only be in state $(2,4,0)$ at the end of the third time period if Project 2 achieved success in the third time period. Had Project 2 reached success prior to the final time period, funding for the other projects would have ceased for all other projects, and Project 1 would have reverted to state 0. On the other hand, Project 3 was not funded at least one time interval prior to Project 2 achieving state 4. The constraint needs to reflect those possibilities. In other words, $p(2,4,0) =$

- Project 3 was never funded:

$$\sum_{s_1^1, s_2^1 = 1}^{3} \sum_{s_1^2, s_2^2 = 1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_1^2(s_1^1, s_1^2)\pi_2^2(s_2^1, s_2^2)\pi_1^3(s_1^2, 2)\pi_2^3(s_2^2, 4) z_{9,(s_1^1, s_2^1, 0, s_1^2, s_2^2, 0),1} \tag{3.30a}$$

- Project 3 was funded in the first time period only:

$$\sum_{s_1^1, s_2^1, s_3^1 = 1}^{3} \sum_{s_1^2, s_2^2 = 1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_3^1(s_3^1)\pi_1^2(s_1^1, s_1^2)\pi_2^2(s_2^1, s_2^2)\pi_1^3(s_1^2, 2)\pi_2^3(s_2^2, 4) z_{2,(s_1^1, s_2^1, s_3^1, s_1^2, s_2^2, 0),1} \tag{3.30b}$$

- Project 3 was funded in the first two time periods only:

$$\sum_{s_1^1, s_2^1, s_3^1 = 1}^{3} \sum_{s_1^2, s_2^2, s_3^2 = 1}^{3} \pi_1^1(s_1^1)\pi_2^1(s_2^1)\pi_3^1(s_3^1)\pi_1^2(s_1^1, s_1^2)\pi_2^2(s_2^1, s_2^2)\pi_3^2(s_3^1, s_3^2)\pi_1^3(s_1^2, 2)\pi_2^3(s_2^2, 4) z_{1,(s_1^1, s_2^1, s_3^1, s_1^2, s_2^2, s_3^2),2} \tag{3.30c}$$

We use these linearization constraints to map from the probability of success variables that are contained in the objective function to the binary funding decision variables.

3.5 Advantages of the IP Formulation

As the details of the previous section indicates, there is a fair amount of effort required to formulate even a two-project, two-time period problem as an IP; and a three-project, three-time period problem is quite involved. In these formulations, we assumed that the budgets for each time period, B_1 and B_2 (and B_3 for the three-time period formulation), were fixed. For this case, the SDP is likely the best approach for problems of any considerable size. However, there are cases where a project manager may be looking for the optimal budget allocation within each time period given a total overall budget, along with an optimal set of funding strategies, representing a problem optimized at two levels. Suppose the project manager has the opportunity to optimize the proposed budgets for each time period. In order to optimize the probability of success over all possible budget allocations, the IP formulation can be modified rather easily. By setting B_1 and B_2 to continuous variables with their sum equal to a determined value (i.e., $B_1 + B_2 = B$), where the total budget, B, is given, we can modify the IP formulation to include only one more linear constraint and two more continuous variables, as we will show in Chapter 4. Since this is a two-level problem, with the lower-level being an SDP, perhaps the only way to solve this is to enumerate all possible budget increments and solve the resulting SDPs.

The formal proof showing the equivalence between a two-level budget problem and the one-level IP formulation, along with efficient solution search techniques when solving an embedded set of SDPs, is provided in Chapter 4. We nevertheless introduce these concepts here, since we compare the run-time performance on various sample problems using both the IP and SDP approaches in this chapter.

In Chapter 2, and in Eckhause, *et al*. (2009), we provide the SDP formulation and solution techniques for both the fixed-budget problem ("Model 1") and the flexible-budget problem ("Model 2"), but not a budget-optimal allocation problem (which we refer to as "Model 3"). In the flexible-budget (Model 2) case, the amount of budget necessary in each time period, B_t (where $\sum_{t=1}^{T} B_t = B$ for some fixed overall budget, B), is determined with certainty only when the state of the system is known (i.e., at that time period) and an optimal decision is identified. This flexibility offers the greatest set of feasible funding decisions, as each time period's budget need not be allocated in advance. Of course, solving for the optimal budget (Model 3) provides greater flexibility than the fixed-budget problem (Model 1), as the feasible region is larger, since the B_t variables are fixed in Model 1. As such, an optimal objective function value [8] (Z^*) to each of the three problem cases has the following property, assuming the same sets of transition probabilities, time periods, costs, and total budget:

$$Z^*_{\text{Model1}} \le Z^*_{\text{Model3}} \le Z^*_{\text{Model2}} \tag{3.31}$$

This relationship demonstrates the benefit of this type of increased managerial flexibility. While the fixed-budget ("Model 1") and the flexible-budget ("Model 2") problems are possibly more efficiently solved using an SDP approach, the numerical results indicate that the IP formulation may be the preferred approach for some problems when solving for the budget-optimal problem ("Model 3") along with the optimal funding strategy. We test both approaches on several problems in the next chapter. In the next section, we compare the approaches applied to the fixed-budget problem.

3.6 Numerical Examples

We solve the proposed IP formulation and compare its performance to that received by the SDP approach using the case study outlined in Chapter 2 and by Eckhause, *et al.* (2009). We have modified the parameters of the numerical problems to better facilitate the comparison and scalability of these kinds of problems. Specifically, two numerical problems are solved: a two-project, two-time period problem and a three-project, three-time period example. Both models were

---

[8] The optimal objective function value Z^* is distinguished from the continuous decision variables z mentioned previously.

constructed in LINGO and solved using both LINGO's solver and XPRESS-MP's solver by converting the LINGO model's code into .mps format for XPRESS-MP.

For the two-project, two-time period ($N = 2, T = 2$) and the three-project, three-time period ($N = 3, T = 3$) SDP formulations, the feasible decisions at each time period, given the state of all projects (denoted by S^t, where $S = \{0,1,2,3,4\}$) are:

$$X(S^t) = \begin{cases} X_i^t \in \{0,1\}: \ \sum_{i=1}^{N} c_i^t X_i^t \leq B_t \\ \qquad\qquad X_i^t = 0 \quad \text{if } S_i^t = 0 \qquad \forall\, i \in N \end{cases} \tag{3.32}$$

This problem is then solved for each decision X_i^t over all time periods, T, such that:

$$V^t(S^t) = \max_{X^t \in X(S^t)} \mathbf{E}\{V^{t+1}(S^{t+1}) \mid S^t, X^t\} \quad t = 1, \ldots, T \tag{3.33}$$

Since the objective function is to maximize the probability of reaching the success state (in this example, state 4) in the final time period, the boundary condition for the dynamic program is:

$$V^{T+1}(S^{T+1}) = \begin{cases} 1 & \text{if} \quad S_i^{T+1} = 4 \ \text{ for some } \ i \in N \\ 0 & \text{otherwise} \end{cases} \tag{3.34}$$

3.6.1 Problem 1: A Two-Project, Two-Time Period Example

In this section, we solve a two-project, two-period numerical example to illustrate the approach one would take for solving larger problems. The states are $\mathbf{S} = \{0,1,2,3,4\}$, where 4 defines the success state and the 0 state corresponds to not being funded. Table 3.3 shows the costs and budget for this problem in millions.

Table 3.4 shows the first stage transition probabilities. Table 3.5 shows the second stage conditional probabilities, where the row represents the state reached at the end of time period one, and the column is the state achieved at the end of the second stage, if the project is funded.

Table 3.3: Two-Project, Two-Time Period Costs and Budget

Project	Time Period 1	Time Period 2
Project 1	$ 5.8	$ 3.0
Project 2	$ 5.8	$ 2.0
Budget	$ 12.0	$ 3.0

Table 3.4: First Time Period State Transitional Probabilities

Project	State	Prob
Project 1	1	0.30
	2	0.40
	3	0.25
	4	0.05
Project 2	1	0.40
	2	0.40
	3	0.15
	4	0.05

Table 3.5: Second Time Period Conditional State Transition Probabilities

Project 1	**State 1**	**State 2**	**State 3**	**State 4**
State 1	0.30	0.20	0.50	0.00
State 2	-	0.20	0.70	0.10
State 3	-	-	0.35	0.65
State 4	-	-	-	1.00
Project 2	**State 1**	**State 2**	**State 3**	**State 4**
State 1	0.40	0.30	0.15	0.15
State 2	-	0.30	0.35	0.35
State 3	-	-	0.45	0.55
State 4	-	-	-	1.00

This integer program is solved in LINGO and XPRESS-MP very quickly, as the number of integer variables is $2+2\left(5^2\right)=52$, and some of these binary variables have trivial solutions (e.g., variables representing the decision variable of a project already in the success state). An optimal solution yields the following results:

$$\begin{aligned} Z^*_{\text{Model1}} &= 0.449625 \\ p(4,0) &= 0.201875 \\ p(0,4) &= 0.24525 \\ p(4,4) &= 0.0025 \end{aligned} \tag{3.35}$$

By inspection, it is clear that the only possible successful outcomes after two time periods are $(4,0)$, $(0,4)$ and $(4,4)$ because the budget only allows for funding one of the two projects in the second time period (total costs = \$5, total budget = \$3). Thus, the unfunded (i.e., less successful) project goes to state 0, unless both projects achieve state 4 after the first time period which happens with probability $(0.05)(0.05)=0.0025$ (see “State 4” probabilities in Table 3.4).

3.6.2 Problem 2: A Three-Project, Three-Time Period Example

Unlike the previous example, the three-project, three-time period ($N=3,\ T=3$) integer programming formulation is a large enough problem that solving by inspection for the optimal funding options is not likely possible. The example listed below provides a framework whereby larger problems can be modeled. Like the two-project, two-time period problem, the states are $\mathbf{S}=\{0,1,2,3,4\}$, where 4 defines the success state and the 0 state corresponds to not being funded. Table 3.6 shows the costs and budgets for the three projects over the three time periods. Table 3.7 shows the first stage transition probabilities. Table 3.8 shows the second stage conditional probabilities. Table 3.9 shows the third stage conditional probabilities.

Table 3.6: Three-Project, Three-Time Period Problem Costs and Budget

Projects	Time Period 1	Time Period 2	Time Period 3
Project 1	$ 5.0	$ 6.0	$ 8.0
Project 2	$ 4.0	$ 2.0	$ 4.0
Project 3	$ 3.0	$ 2.5	$ 4.0
Budget	$ 10.00	$ 10.00	$ 10.00

Table 3.7: First Time Period State Transitional Probabilities

Project	State	Probability
Project 1	1	0.40
	2	0.30
	3	0.30
	4	0.00
Project 2	1	0.50
	2	0.40
	3	0.10
	4	0.00
Project 3	1	0.45
	2	0.45
	3	0.10
	4	0.00

Table 3.8: Second Time Period Conditional State Transition Probabilities

Project 1	State 1	State 2	State 3	State 4
State 1	0.30	0.40	0.30	0.00
State 2	--	0.50	0.35	0.15
State 3	--	--	0.60	0.40
State 4	--	--	--	1.00

Project 2	State 1	State 2	State 3	State 4
State 1	0.40	0.35	0.25	0.00
State 2	--	0.60	0.30	0.10
State 3	--	--	0.80	0.20
State 4	--	--	--	1.00

Project 3	State 1	State 2	State 3	State 4
State 1	0.35	0.40	0.25	0.00
State 2	--	0.45	0.45	0.10
State 3	--	--	0.75	0.25
State 4	--	--	--	1.00

Table 3.9: Third Time Period Conditional State Transition Probabilities

Project 1	State 1	State 2	State 3	State 4
State 1	0.50	0.30	0.20	0.00
State 2	--	0.30	0.40	0.30
State 3	--	--	0.60	0.40
State 4	--	--	--	1.00

Project 2	State 1	State 2	State 3	State 4
State 1	0.40	0.40	0.20	0.00
State 2	--	0.50	0.40	0.10
State 3	--	--	0.75	0.25
State 4	--	--	--	1.00

Project 3	State 1	State 2	State 3	State 4
State 1	0.50	0.25	0.25	0.00
State 2	--	0.40	0.45	0.15
State 3	--	--	0.80	0.20
State 4	--	--	--	1.00

As Table 3.10 shows, using LINGO or XPRESS-MP to solve this example with the integer program formulation with no branching takes a significant amount of processor time. However, dividing the untreated problem into smaller subproblems makes finding an optimal solution very fast. The budget in the first time period is \$10.0 and the costs of the three projects are \$5.0, \$4.0 and \$3.0, respectively. Thus, we can consider solving for the optimal solution by considering the 2^3 first time period funding decisions. In fact, one can see that an optimal first time-period funding decision for this example must be among the following: $\left(x_1^1, x_2^1, x_3^1\right) = (1,1,0)$, $(1,0,1)$ or $(0,1,1)$, since $\left(x_1^1, x_2^1, x_3^1\right) = (1,1,1)$ is infeasible (first time period costs exceed \$10) and other combinations would fund fewer projects.

As noted previously, the number of binary variables in the three-project, three-time period example is $3 + 3\left(5^3\right) + 3\left(5^3\right) = 753$. While branching on the first time period variables only reduces the number of integers by 3 out of 753, the structure for calculating optimal funding actions based on the condition outcomes becomes much easier for both LINGO and XPRESS-MP to exploit. The value for the objective function, some of the critical decision variables and the relevant success states are:

$$
\begin{aligned}
&Z^*_{\text{Model 1}} = 0.476845 \\
&(x_1^1, x_2^1, x_3^1) = (1,0,1) \\
&p(4,0,0) = 0.3906 \\
&p(0,0,4) = 0.074695 \\
&p(4,0,4) = 0.01155
\end{aligned}
\tag{3.36}
$$

3.6.3 Solution Run-Times and Comparison with the SDP Approach

For the two-project, two-time period numerical example described, the solution in LINGO runs very quickly. The three-project, three-time period problem runs more slowly, unless we exploit the structure of the problem and solve for the branching on the first time period, which greatly speeds up the solution time. Table 3.10 provides the run-times for the above three-project, three-time period numerical example when solved in both LINGO and XPRESS-MP, with and without branching on the first time period variables. All runs were made on a 2.0 GHz dual-processor with 2.0 GB of RAM running Windows XP. While the number of linearization variables (w, y, z) is large, the run-times for the three-project, three-time period problems are quite low once we branch on the first time period variables prior to using LINGO or XPRESS-MP. This branching technique is inherent to the solution methods in solving an SDP, so results are repeated for the "branching" and "no branching" cases.

Table 3.10: Run-Times for Numerical Examples[9]

Numerical Example	IP (LINGO)		IP (XPRESS-MP)		SDP (Coded in C++)	
	CPU Sec.	Iterations	CPU Sec.	Iterations	CPU Sec.	State Var.
2 Project, 2 Time Period (no branching)	<1	476	<1	63	<1	50
3 Project, 3 Time Period (no branching)	5,116	726,970	1,249	1,171,868	2	375
3 Project, 3 Time Period (with branching)	39	97,795	2	1,624	2	375

For both examples, the stochastic dynamic program implements backward induction in the standard manner (Puterman, 1994). Given that there are N projects, S possible states (or outcomes) for each project and T time periods, there are $S^N T$ state variables and S^N possible states for each funding decision. The SDP does quite well for the problems with fixed budgets due to the Markov nature of the problem. However, as the number of state variables at each time period is S^N, computational

[9] Iterations refer to pivots on the constraint matrix (LINGO, 2011).

complexity grows polynomially with the increase in the number of states and exponentially with the number of projects.

When an optimal budget allocation must be solved, we are required to run a series of embedded SDPs, each of which is as computationally complex as the fixed-budget SDP. Since we must search over a range of possible budgets, the budget must be discretized into sufficiently small increments. In this case, the performance time of the SDP will be decreased. As we demonstrate in Chapter 4, when considering the budget-optimal real options problem, the IP formulation can be used effectively to solve this two-level problem, as we prove its equivalence to a one-level IP. Indeed, in many cases, it solves considerably faster than the SDP. However, as the complexity of the problem grows and the IP approach becomes intractable, efficient search techniques for solving the two-level problem, where the lower-level problem is an SDP, are needed. Those techniques are also developed in Chapter 4, and used on an actual set of carbon capture and storage technology projects in Chapter 5.

Chapter 4: Budget-Optimal Allocations for the Multi-Stage Real Options Model: A Two-Level Problem

As described in Chapter 1, this dissertation covers three types of budget allocation methods for the multi-stage real options problem: fixed ("Model 1"), flexible ("Model 2") and budget-optimal ("Model 3"). In Chapter 2, we provided the SDP formulation for the fixed-budget and flexible-budget real options problem. In Chapter 3, we provided the equivalent IP formulation for the fixed-budget real options problem. In this chapter, we present both IP and SDP formulations of the budget-optimal real options problem ("Model 3"). The first section describes how the budget-optimal problem can be modeled a one-level IP. Section 4.2 compares the run-times and complexity of the IP and SDP. Section 4.3 outlines methods for reducing the run-times of the two-level SDP formulation.

4.1 Motivation for the Two-Level Problem: Optimal Budgets

In Chapter 3, the integer programming solution for the multi-stage real options model assumes a fixed-budget allocation for each of the time periods under which the problem is being solved. This allocation is an important problem in itself, as not only the amount of the total budget, B is important, but also the specific values for each time period t, B_t, where $\sum_t B_t = B$. In Chapter 2, we assumed in the flexible-budget ("Model 2") numerical example that the budget available for both stages could be spread among the stages in whatever manner was optimal. This flexibility allowed us to demonstrate the approach on a problem with a larger state-space. While one can imagine such flexibility for a two-time period problem such as the example in Chapter 2, it is more difficult to imagine such budgetary freedom for multiple time periods, especially if each time period is related to months or years of development.

The characteristics of the budget allocation problem form a higher-level hierarchy to the optimal selection of projects solved in Chapter 3. Fortuny-Amat and McCarl (1981) present five criteria for situations which could be properly represented by a multi-level programming model. The five criteria are: i) two of more decision makers with not necessarily identical goals; ii) each decision maker only has control over certain variables; iii) the decision process is carried out in two stages: the higher level announces its actions and the lower level responds; iv) the higher level's objective is to select a plan that optimizes the lower level's rational response; and v) the higher level decision maker knows the objective function and constraints of the subproblem. All five criteria potentially exist in the multi-stage real options problem. As stated previously, this budget-optimal problem differs from the flexible-budget problem (Model 2) in Chapter 2, since we assign each time period's budget in advance. In this case, the discretion to decide upon the budget for each time period exists only at the beginning of the multi-stage competition, not at the beginning of each time period, which was the case of the flexible-budget problem (Model 2).

4.1.1 An Illustration of the Two-Level Problem

The two-project, two-time period and three-project, three-time period problems we formulated in Chapter 3 provides an optimal funding strategy and the probability of success given certain characteristics of the projects' costs and technological progression. Considering the problem with two time periods, suppose that we had a total budget, B (where $B = B_1 + B_2$) and we had the ability to spread the funding between the time periods optimally, but we must specify them in advance (hence, an *a priori* budget allocation). We can consider this as a two-level problem where the upper-level problem is:

$$\begin{aligned} \max \quad & a_1 B_1 + a_2 B_2 + a_3 \alpha(B_1) \\ \text{s.t.} \quad & B_1 + B_2 \le B \\ & x \in SOL(B_1) \end{aligned} \tag{4.1}$$

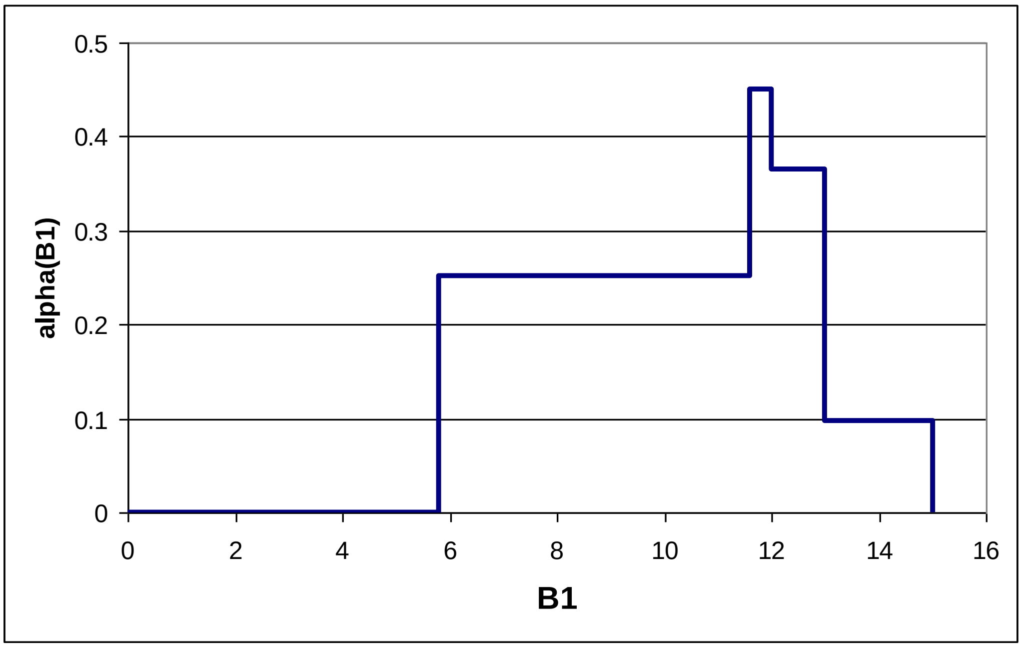
We denote the optimal value function to the lower-level integer programming problem as problem $\alpha(B_1)$, which is equivalent to solving the integer program (3.9). Since $B = B_1 + B_2$, given a value for B_1, we immediately know B_2. Thus, we can describe α as a function simply of B_1. In (4.1), a_1 and a_2 correspond to possible coefficients we might apply to the selection of B_1 and B_2 (e.g., a discount rate for delayed funding). In what follows, however, we set $a_1 = a_2 = 0$ and $a_3 = 1$ to simplify things, while noting that in the fixed-budget problem described in Chapter 3 that $B_1 = 12$ and $B_2 = 3$. If $B_1 + B_2 = 15$, and we specify a given B_1 (and thus, B_2), we can consider the optimization of the *a priori* budget allocation. Figure 4.1 shows the "cityscape" solution to the subproblem for varying values of B_1. As it turns out, its fixed-budget allocation from Chapter 3, $B_1 = 12$, is optimal since

$$\max_{B_1} \quad \alpha(B_1) = 0.449625 \tag{4.2}$$

which would be determined by solving the two-level problem in (4.1). Figure 4.1 provides the optimal objective function $\alpha(B_1)$ for all possible values of B_1 (and thus,

B_2). For T time periods, the figure would require optimizing over a $T-1$ dimensional "cityscape" function to the find the value over which $\alpha(B_1, B_2, \ldots, B_{T-1})$ is greatest.

Figure 4.1: Solution to the Subproblem $\alpha(B_1)$



However, as we prove in this chapter, this problem can be easily converted into a one-level integer programming problem by bringing the budgets for each time period into the lower level problem and making them continuous variables. The result of this equivalence is that the two-level problem is now only slightly more computationally difficult than the fixed-budget problem, when solved as an IP. However, there is no known way of converting this to a one-level problem when the lower-level problem is solved using SDP methods. As a result, the run-times for the IP for certain size problems can be significantly faster than the SDP approach. We compare the run-times for several problems in Section 4.2, after we formally prove the equivalence of the two-level problem to a one-level IP.

4.1.2 Converting the Budget-Optimal Allocation Problem to a One-Level Integer Program

In order to establish the potential advantage of the IP formulation for the budget-optimal problem, we first formally prove the equivalence of the budget-optimal two-level problem in (4.1) to a one-level IP. This property allows us to solve a one-level integer program to compare it with the two-level problem solved as a series of SDPs.

Suppose we have a budget at time period t denoted by B_t. We denote the vector of budgets over all time periods as $\vec{B} = (B_1, B_2, ..., B_T)$, for a given total budget available, B. For a fixed-budget allocation, $\vec{B}$, we can solve the real options integer programming problem formulated in Section 3.2 to find the optimal decisions of which projects to fund. Using the notation from Chapter 3, we denote the objective function for this problem, represented in (3.2), as $\alpha(\vec{B})$, since the solution depends on the budgets for each time period. The integer program can then be written generically as:

$$\begin{aligned} \alpha(\vec{B}) = \max\ & d^T x \\ \text{s.t.}\ & x \in X(\vec{B}) \end{aligned} \tag{4.3}$$

where d is the appropriate vector corresponding to the values in (3.2), x is the feasible funding decisions and $X(\vec{B})$ is the feasible space (of the real options integer program shown given in (3.2) – (3.8) in Section 3.2), given the set of budget allocations, $\vec{B}$. As shown in Chapter 3, using linearization variables, the general integer program in Section 3.2 can be expressed with linear constraints and objective function (e.g., the two-project, two-time period problem given in (3.17)). Thus, (4.3) is expressed with a linear objective and $X(\vec{B})$ has linear constraints.

The two-level problem can be written as:

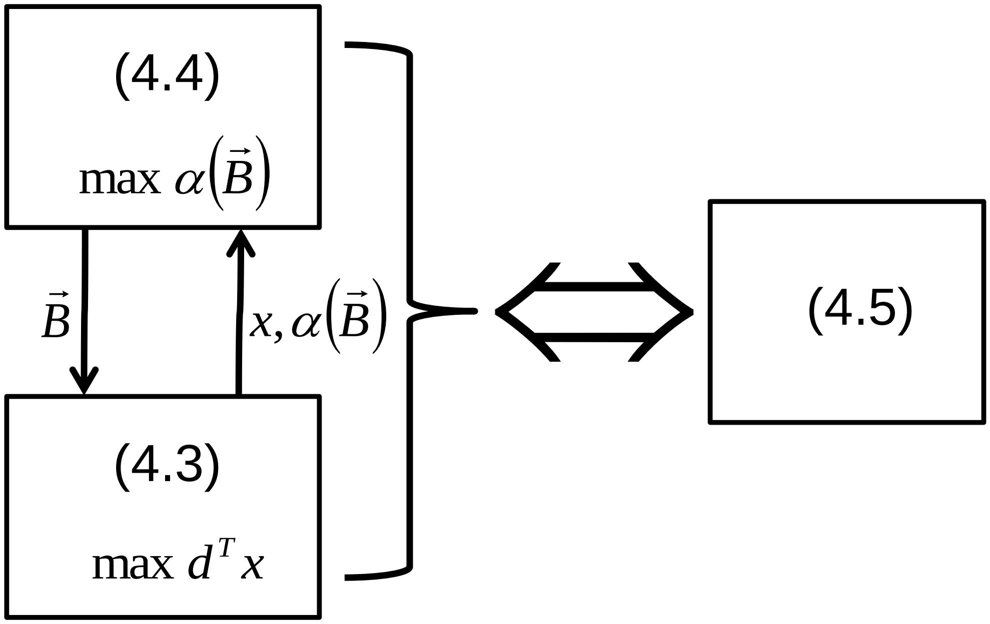
$$\begin{aligned} Z_{\text{two-level}} = \max\ & \alpha(\vec{B}) \\ \text{s.t.}\ & B_1 + \ldots + B_T \le B \\ & \vec{B} = (B_1, B_2, \ldots, B_T) \ge 0 \end{aligned} \tag{4.4}$$

In other words, the two-level problem finds the maximum value for $\alpha(\vec{B})$—that is, the maximum probability that at least one project succeeds—over all possible budget allocations, $\vec{B}$, that are feasible (i.e., less than or equal to B). Combining the components of (4.3) and (4.4), we can now write a one-level problem as:

$$\begin{aligned} Z_{\text{one-level}} = \max\ & d^{\mathrm{T}} x \\ \text{s.t.}\ & B_1 + \ldots + B_T \le B \\ & \vec{B} \ge 0 \\ & x \in X(\vec{B}) \end{aligned} \tag{4.5}$$

Figure 4.2 shows the conceptual relationship between the upper-level and lower-level problems of the two-level problem in (4.4) and its equivalence to (4.5).

Figure 4.2: Two-Level Problem Structure



Remark 4.1: There is always a solution to problems (4.3) – (4.5) by the following argument. The feasible region is always nonempty, since $x = 0$, $\vec{B} = 0$ (fund nothing with no budget) is always feasible. There are a finite number of values for the binary funding decisions, x, (all x variables are binary by Theorem 3.1). Each $B_t \in [0, B]$, which is a compact set. Thus, by the Weierstrass theorem, a solution exists for each fixed x. Given that there is a finite number of values for x, the result is shown. We show in the next theorem that the optimal objective functions to the two-level problem (4.4) and the one-level problem (4.5) are identical.

Theorem 4.1: *The optimal objective function value, $Z_{\text{two-level}}$, to the two-level problem (4.4) is equal to the optimal objective function value, $Z_{\text{one-level}}$, to the one-level problem (4.5). Moreover, the solution set to (4.4) is equal to the solution set to (4.5). That is, $SOL(4.4) = SOL(4.5)$.*

Proof: First, we prove that $Z_{\text{two-level}} \le Z_{\text{one-level}}$. Let $\vec{B}^*$ be an optimal solution to the two-level problem in (4.4). We thus have that $Z_{\text{two-level}} = \alpha\left(\vec{B}^*\right)$ and that $B_1^* + \ldots + B_T^* \le B$. Solving (4.3) using the budget allocation $\vec{B}^*$, we denote x^* as an optimal solution to this problem. Thus, we have that $Z_{\text{two-level}} = \alpha\left(\vec{B}^*\right) = d^T x^*$ and $x^* \in X\left(\vec{B}^*\right)$. So, $\left(\vec{B}^*, x^*\right)$ is a feasible solution to problem (4.5) with the objective function value of $d^T x^* = \alpha\left(\vec{B}^*\right)$, as all constraints are satisfied. Since this solution is feasible solution to (4.5), though not necessarily optimal, we get that $Z_{\text{one-level}} \ge d^T x^* = \alpha\left(\vec{B}^*\right) = Z_{\text{two-level}}$, because the objective function in (4.5) can do at least well as $d^T x^*$.

Now we show that $Z_{\text{one-level}} \le Z_{\text{two-level}}$. Let $\left(\vec{B}^*, x^*\right)$ be an optimal solution to (4.5), i.e., the one-level problem. Solving (4.3) after fixing the budgets to $\vec{B}^*$, we claim that x^* must be an optimal solution to (4.3), which gives $\alpha\left(\vec{B}^*\right) = d^T x^*$.

To prove that x^* is a solution to (4.3), we assume not. Then, there exists a $\hat{x} \in X\left(\vec{B}^*\right)$ such that $d^T \hat{x} > d^T x^*$. So, $\left(\vec{B}^*, \hat{x}\right)$ is a feasible solution to (4.5) (since $\hat{x} \in X\left(\vec{B}^*\right)$), satisfying $d^T \hat{x} > d^T x^*$. However, this contradicts the fact that $\left(\vec{B}^*, x^*\right)$ is an optimal solution to the one-level problem (4.5). Therefore, we must indeed have that $\alpha\left(\vec{B}^*\right) = d^T x^*$.

Since $\left(\vec{B}^*, x^*\right)$ is an optimal solution to (4.5), $\vec{B}^*$ is a feasible solution to (4.4), since it must satisfy $B_1^* + \ldots + B_T^* \le B$. Since $\vec{B}^*$ is a feasible—but not necessarily optimal—solution to (4.4), we have that $Z_{\text{two-level}} \geq \alpha\left(\vec{B}^*\right) = d^T x^* = Z_{\text{one-level}}$.

Therefore, $Z_{\text{two-level}} = \alpha\left(\vec{B}^*\right) = d^T x^* = Z_{\text{one-level}}$.

To show that $SOL(4.4) \subseteq SOL(4.5)$, we know that $(\vec{B}^*, x^*)$ must be a feasible solution to (4.5). Assume it is feasible, but not optimal. Then, we have that $Z_{\text{two-level}} = \alpha\left(\vec{B}^*\right) = d^T x^* < Z_{\text{one-level}}$, which is a contradiction. Thus, any optimal solution to (4.4) must be an optimal solution to (4.5). Similarly, to show that $SOL(4.5) \subseteq SOL(4.4)$, we know that $\vec{B}^*$ must be a feasible solution to (4.4). If $\vec{B}^*$ is not an optimal solution to (4.4), then we have that $Z_{\text{one-level}} = d^T x^* = \alpha\left(\vec{B}^*\right) < Z_{\text{two-level}}$, which is also a contradiction. Therefore, any optimal solution to (4.5) must be an optimal solution to (4.4). Therefore, $SOL(4.4) = SOL(4.5)$. ■

While $Z_{\text{two-level}} = Z_{\text{one-level}}$, a solution $\vec{B}^*$ in (4.4) may not be the same as a solution in (4.5), since an optimal solution is not necessarily unique. As is shown in Figure 4.1, an optimal solution for the two-level problem is $\vec{B}^* = (11.9, 3.1)$. However, solving the one-level problem can produce the solution $\vec{B}^* = (12.0, 3.0)$. While the solutions to (4.4) and (4.5) are not unique, the solution sets are identical.

The interpretation of this equivalence is important, but relatively straightforward. For the integer programming problem, we can simply add the constraint that $B_1 + \ldots + B_T \leq B$ to the original problem formulated in Chapter 3 and solve what is in effect a two-level problem. From a modeling perspective, this additional constraint is trivial to add. Nevertheless, despite the addition of only a few continuous variables and one constraint, it can somewhat increase the run-time for the IP, as the feasible region increases. The reason the budget-optimal problem can be written as a one-level program is that it assumes that a single decision maker (in this case, the real options problem optimizer) is deciding upon both the optimal funding strategy and the optimal budget. Since there is no longer explicitly a higher level decision maker with "control" over certain variables (i.e., the budgets), the budget-optimal problem does not fit the classical requirements for a two-level problem (Fortuny-Amat and McCarl, 1981).

While we do not prove it formally, objective functions other than the one in (4.3) and (4.4) can be used while preserving the conversion of the two-level problem to a one-level problem. For example, we could have the objective function: $Z_{\text{one-level}} = Z_{\text{two-level}} = \max \sum_t a_t B_t + \alpha\left(\vec{B}\right)$, where a_t is perhaps some scalar representing the time-value of money. Additionally, budget constraints can be incorporated into the one-level problem while removing the equivalence to the two-level problem. For example, we could have a constraint that $B_1 \le B_2 \le ... \le B_T$, ensuring that the budgets increase over time. The flexibility to add these constraints easily in the IP formulation highlights an advantage of over the SDP approach.

Unlike for the IP, for the SDP, the additional, budget-optimal constraint is hugely significant. Since the budgets for each time period are intended to be optimal, but dynamically determined, the budget-optimal problem in (4.5) is not identical to the flexible-budget problem (i.e., Model 2) shown in Chapter 2, where the budget allocation for that time period does not need to be decided (and is not decided) until the previous time period. In a general setting, to solve this budget-optimal problem using stochastic dynamic programming requires solving for every possible budget allocation to find the allocation that optimizes $\alpha(\vec{B})$, or the probability of success. In the next section, we compare IP and SDP run-times for the three-project, three-time period problem introduced in Chapter 3 and then consider the advantages and limits of the two approaches.

4.2 Comparing the Performance of the IP and SDP Formulations of the Budget-Optimal Allocation Problem

In this section, we first compare the run-times for the numerical example introduced in Section 3.6.2. The IP formulation appears to do quite well in comparison with the SDP for problems of a certain size. The main difficulty with using the IP for all large problems is the exponentially increasing number of linearization constraints. Methods for solving larger problems are discussed at the end of this section.

4.2.1 Three-Project, Three-Time Period Examples

We now consider the identical three-project, three-time period problem introduced in Section 3.6.2, but with the possibility for budget-optimal solutions. Recalling the costs from Table 3.6, we now add the flexibility for the budgets:

Table 4.1: Costs Identical, but now with Flexible Budgets

Projects	Time Period 1	Time Period 2	Time Period 3
Project 1	$ 5.0	$ 6.0	$ 8.0
Project 2	$ 4.0	$ 2.0	$ 4.0
Project 3	$ 3.0	$ 2.5	$ 4.0
Total Budget = $30.0	B_1	B_2	B_3

where the total budget is fixed at \$30. Note that instead of $B_1 = B_2 = B_3 = \$10$, as we had in Section 3.6.2, we now simply have that $B_1 + B_2 + B_3 \le B = \$30$.

In order to handle this additional constraint, the integer programming problem in Appendix B simply includes the budget-optimal constraint and the budgets for each time period become continuous variables. To get run-time data for each problem solved, we employ both LINGO and XPRESS-MP solvers. Our results indicate that XPRESS-MP tends to be faster for these budget-optimal IP problems. However, LINGO has a global, nonlinear, mixed-integer nonlinear programming solver, which we used in the sensitivity analyses; so both solvers' results are included for all runs each solver can handle. As in the fixed-budget problem solved in Section 3.6.2, using either solver for this example with no specified branching on the first-stage variables can increase both XPRESS-MP's and LINGO's solvers run-time by a factor of 10 or more. However, as with the fixed-budget problem in Chapter 3, dividing the untreated problem into smaller subproblems makes finding an optimal solution straightforward.

The budget in the first time period can be as high as \$30.0 and the costs of the three projects are \$5.0, \$4.0 and \$3.0, respectively. Thus, we can consider solving for the optimal solution by branching on the $2^3 = 8$ first time period funding decisions (and can exclude the trivial solution $(x_1^1, x_2^1, x_3^1) = (0,0,0)$, since that will have objective value equal to 0). The most time-consuming branch (between 50%-90% of the total run-time for both LINGO and XPRESS-MP) is branching on the $(x_1^1, x_2^1, x_3^1) = (1,1,1)$ case, as that represents the greatest number of choices for the second and third time-period variables.

As noted in Chapter 3, this three-project, three-time period problem has

$$3\,(\text{first time period}) + 3\left(5^3\right)(\text{second time period}) + 3\left(5^3\right)(\text{third time period}) = 753$$

binary variables. While branching on the first time period only reduces the number of variables by 3 out of the 753 integer variables, the structure for calculating optimal funding actions based on the condition outcomes becomes much easier for both LINGO and XPRESS-MP to exploit. The value for the objective function, some of the critical decision variables and the relevant success states are:

$$\begin{aligned} Z^* &= 0.5391175 \\ (x_1^1, x_2^1, x_3^1) &= (1,0,1) \\ B_1 &= 8.0 \\ B_2 &= 8.5 \\ B_3 &= 13.5 \end{aligned} \tag{4.6}$$

It is interesting to note that while the first time funding decision results are identical to those in (3.36)—namely, fund Project 1 and Project 3—the objective function is higher: approximately 0.54 instead of approximately 0.48. This improvement, of course, stems from the ability to specify the budget in an optimal way—in this case, having more funds available in the final time period. While the number of linearization variables is large, the run-times shown in Table 4.2 for the three-project, three-time period problems are quite reasonable once we branch on the first time period variables prior to using LINGO or XPRESS-MP, even when solving for the budget-optimal allocation.

We also compare the results of the IP formulation to that of the SDP in Table 4.2. To summarize the SDP problem size, there are N projects, S possible states (or outcomes) for each project and T time periods. Therefore, there are $S^N T$ state variables and S^N possible states for each funding decision (X^t) The SDP does quite well for the problems with fixed budgets, due to the Markovian nature of the problem. When the *a priori* budget allocation must be optimized, we are required to run a series of embedded SDPs, each of which is as computationally complex as the fixed-budget SDP. Since the algorithm must search over a range of possible budgets, the budget must be discretized into sufficiently small increments. For the three-project, three-time period example, we used a budget increment of $0.5 million, as that value represents the greatest common factor (GCF) for the costs of the projects over all time periods. Thus, in this example, this increment is sufficiently small that the embedded SDP will provide an optimal solution equal to the IP formulation's optimal solution. If smaller increments are required, more efficient search techniques need to be employed in order for the SDP run-times to be manageable. Reducing the number of iterations and improving the two-level SDP run-times are addressed in detail later in this chapter. The run-times for the two-project, two-time period numerical example are also shown, though their run-times were sufficiently small that no significant information about which method is faster can likely be inferred. The values for an optimal B_1 and B_2 were 11.6 and 3.4, respectively, with $Z^* = 0.449625$. However, the fixed allocation in Chapter 3 with $B_1 = 12.0$ and $B_2 = 3.0$ is also optimal, as Figure 4.1 demonstrates.

Table 4.2 provides the run times for the two-project, two-time period example and the three-project, three-time period numerical example when solved as an IP, in both LINGO and XPRESS-MP and compares them to that of the SDP for both the fixed-budget results from Chapter 3 and with the budget-optimal results here. All runs in

Table 4.2 were made on a 2.0 GHz dual processor with 2.0 GB of RAM using Windows XP.

Table 4.2: Run-Times for IP and SDP Models with Fixed-Budget and Budget-Optimal Allocations

Numerical Example	IP (LINGO)		IP (XPRESS-MP)		SDP (Coded in C++)	
	CPU Sec.	Iterations	CPU Sec.	Iterations	CPU Sec.	State Var.
2 Project, 2 Time Period (fixed budgets)	<1	476	<1	63	<1	50
2 Project, 2 Time Period (budget-optimal)	<1	1088	<1	124	<1	52
3 Project, 3 Time Period (fixed budgets)	39	97,795	2	1,624	2	375
3 Project, 3 Time Period (budget-optimal)	91	230,185	14	18,926	634	378

For the IP, the run-time for the budget-optimal problem is considerably greater than the fixed-budget problem, which is to some degree surprising, since the number of continuous variables is increased only by three (B_1, B_2, and B_3), the number of constraints is increased by one ($B_1 + B_2 + B_3 \le B$) and no new binary variables are introduced. Indeed, it is likely that the larger feasible region due to the introduction of the budget variables makes for the longer run-times. The run-time for the SDP, on the other hand, grows even more considerably. Since the budgets for each time period are continuous variables, there are potentially an uncountably infinite number of values each budget could assume (Royden, 1988). In order to solve the lower-level SDP, one must discretize the budget. However, even with a budget increment of \$0.5 million, the budget-optimal problem required solving approximately 500 subproblems of the fixed-budget real options problem for the three-project, three-time period problem. Hence, there was an increase from roughly two seconds to over 10 minutes in run-times.

For problems of this size, the IP approach appears to be the faster method, at least in terms of run-times. While explicitly formulating the linearization constraints requires a fair amount of effort, the coding can be generalized. Also, the addition of side constraints can more easily be handed than in the SDP formulation. While run-times for IP may be better, there is a difficulty in the exponential growth of the linearization constraints. The limits of these ranges are discussed in the rest of this section.

4.2.2 Comparing Run-Times for Larger Problems

The solution time for the three-project, three-time period problem with optimal budgets was lower for the IP than it was for the SDP. As this section shows, based on the numerical examples tried, this property appears to hold for problems that were ultimately solved using the IP approach. For example, the five-project, five-time period problem in Table 4.5 was not solved due insufficient memory (4 GB). However, the number of linearization constraints is $O\left(10^5\right)$ and the number of linear variables is $O\left(10^4\right)$, even though the number of binary variables was fewer than $O\left(10^3\right)$. Since the number of variables for the IP and the number of states for the SDP grow exponentially with the number of projects, we would expect the run-times for both the IP and the SDP to grow exponentially as a function of the number of projects. As we show in Figure 4.3, this property seems to hold numerically for the sample problems. With the fixed-budget real options problem, SDP problem size only grows *linearly* with the number of time periods. However, when one solves the budget-optimal allocation problem using an SDP formulation, the increase in time periods produces an *exponential* increase in run-time, since the number of SDP subproblems increases exponentially, as we show in this chapter. Nevertheless, for very small problems (e.g., the two-project, two-time period example), either the SDP or the IP method is sufficiently fast, even when solving for the budget-optimal allocation. Since it would appear the IP performs better for somewhat larger problems, it is important to evaluate yet larger problems to determine i) when the IP approach might no longer be as efficient as the SDP and ii) when the problem size becomes too large for a typical computer to solve such problems as IPs.

We first increased the size to a *four*-project, *three*-time period problem, similar to the three-project problem illustrated in the previous section. The first three projects' cost and transitional probability data are identical to those in the three-project, three-time period problem solved previously and are provided in Tables 3.4-3.7. The fourth project's probability transition data are provided in Table 4.3. The cost for funding Project 4 is, along with total budget for all three time periods ($40 million), is given in Table 4.4.

Table 4.3: Transition Probability Data for Project 4

Time Period 1	
State	**Probability**
1	0.30
2	0.40
3	0.30
4	0.00

Time Period 2	State 1	State 2	State 3	State 4
State 1	0.40	0.35	0.20	0.05
State 2	--	0.50	0.40	0.10
State 3	--	--	0.70	0.30
State 4	--	--	--	1.00

Time Period 3	State 1	State 2	State 3	State 4
State 1	0.40	0.35	0.20	0.05
State 2	--	0.40	0.40	0.20
State 3	--	--	0.70	0.30
State 4	--	--	--	1.00

Table 4.4: Cost Project Data for All Four Projects

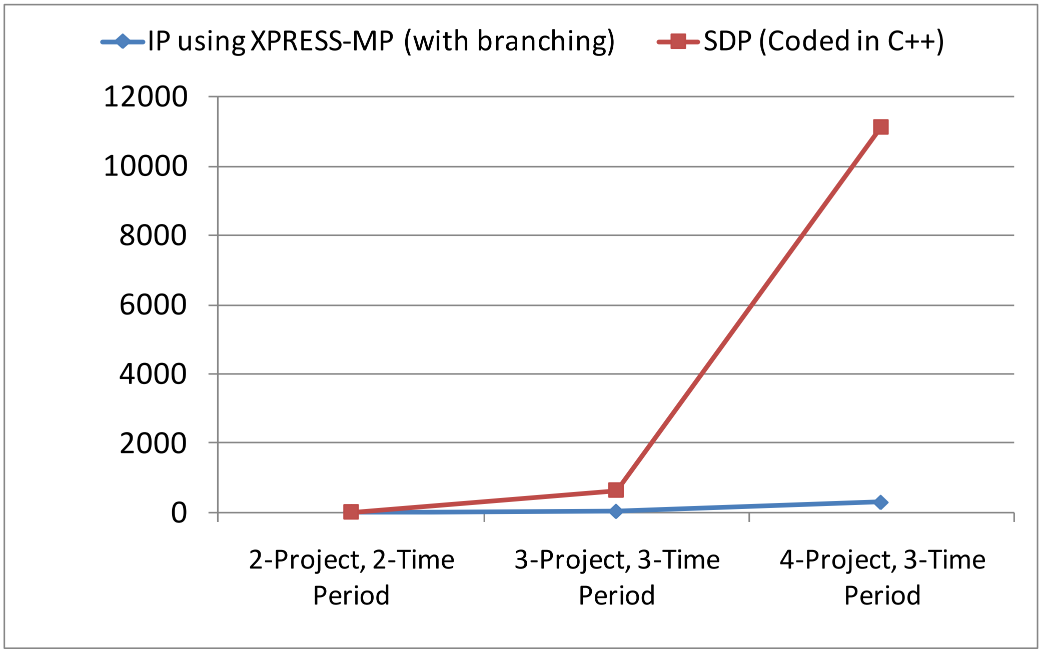
Projects	Time Period 1	Time Period 2	Time Period 3
Project 1	$ 5.0	$ 6.0	$ 8.0
Project 2	$ 4.0	$ 2.0	$ 4.0
Project 3	$ 3.0	$ 2.5	$ 4.0
Project 4	$ 2.8	$ 3.4	$ 5.4
Total Budget = $40.0	B_1	B_2	B_3

For the budget-optimal *four*-project, *three*-time period problem, the IP performs more efficiently than the SDP. With a total budget of $40 million and a budget increment of $0.1 million (the greatest common divisor of the project costs), the SDP solved to optimality in 11,143 seconds (approximately 3 hours). For the same problem, the IP formulation took 284 seconds (between 4-5 minutes) using XPRESS-MP when branching on all of the possible first time period solutions. These results, along with the results from Table 4.2, are shown in Figure 4.3. This reduction is both significant in absolute terms and the fact that the IP version in XPRESS-MP ran in 2.5% of the time the SDP did. Nevertheless, despite running reasonably quickly, the generation of over one million constraints (mostly linearization constraints) required roughly 800 megabytes of memory. The objective function, first time period variables and budget allocations are:

$$\begin{aligned} Z^* &= 0.677959 \\ (x_1^1, x_2^1, x_3^1, x_4^1) &= (1,1,1,0) \\ B_1 &= 12.0 \\ B_2 &= 16.0 \\ B_3 &= 12.0 \end{aligned} \tag{4.7}$$

Figure 4.3 shows the comparative run-times for three cases between the IP and the SDP for the budget-optimal problem. While the run-times for both methods increase nonlinearly with the number of projects and time periods, the SDP approach becomes much slower.

Figure 4.3: Run-Times (CPU Seconds) for the IP and SDP Approaches for the Optimized Budget Allocation Problem



We attempted to solve a five-project, five-time period problem as an IP; however, the linearization constraints grew far too quickly and exhausted the computer's 2 gigabytes of memory. The number of linearization constraints and memory required for the budget-optimal IP sample problems are provided in Table 4.5. While attempting to solve this problem in its nonlinear version would likely require far fewer linearization constraints and variables, as we show in the next section, the run-times for smaller problems were sufficiently long that this approach would not be viable, at least for the LINGO nonlinear, nonconvex, global solver we utilized. It is certainly possible that other solvers, such as the Branch and Reduce Optimization Navigator (BARON, 2011), could do better than LINGO's non-convex global optimizer for this problem, however, we were limited to the solvers for which we have an unrestricted license.

Table 4.5: Constraints and Memory Requirements for the Sample Budget-Optimal IP Formulations[10]

Numerical Example	Linearization Constraints (Approximate)	Memory Needed (MB)
2 Project, 2 Time Period	600	0.1
3 Project, 3 Time Period	100,000	15
4 Project, 3 Time Period	1,000,000	800
5 Project, 5 Time Period	1,000,000,000	> 4000

Based on the results of these modestly sized sample problems, the IP in (4.5) solves significantly faster than the equivalent SDP. Additionally, with limited sensitivity analysis on the costs or transitional probabilities this pattern did not change. While there is a clear limit on the size of solvable problems using the IP approach, the SDP formulation encounters difficulties for even smaller problems. While these results do not represent a comprehensive set of results over all mixed-integer linear and nonlinear solvers, it demonstrates that the linearized IP approach has a range over which it appears to be the preferred method. It also highlights the need for efficient solution techniques when solving this two-level problem as an SDP, which we address later in this chapter.

4.2.3 Evaluating Approaches for Improving the IP Run-Times on Larger Problems

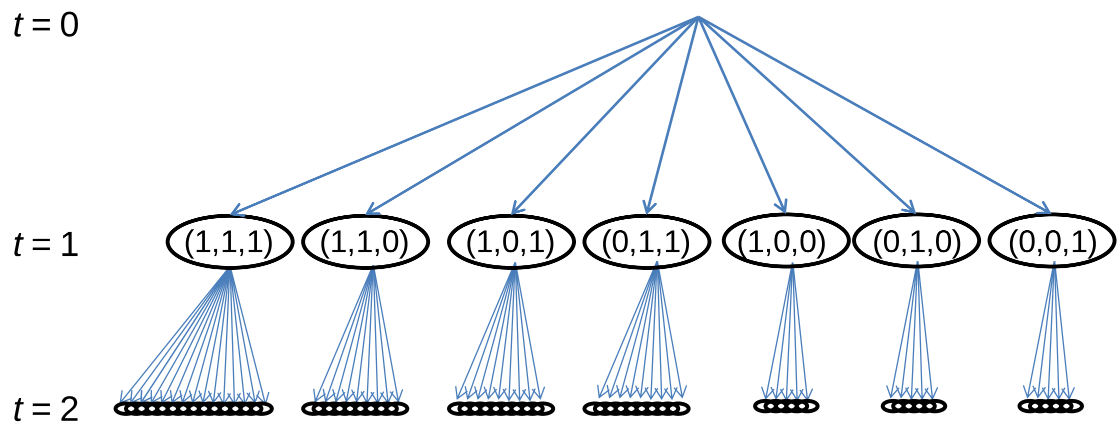
Several techniques can be considered to improve the run-times of the integer programming version of the multi-stage real options problem described in Section 3.2

[10] Since there are a small number (relative to the linearization constraints) of other constraints, the numbers of constraints shown are rounded estimates. Again, the five-project, five-time period problem was not successfully solved.

with objectives and constraints in (3.1) – (3.8). However, excessively long run-times is only one difficulty faced when solving these problems as IPs. The other primarily difficulty is the enumeration of all of the linearization constraints, which requires to varying degrees some amount of automation of the constraints generation for larger problems. Moreover, enumeration of the linearization constraints requires an increasing number of constraints and continuous variables that grows exponentially as the number of projects, states or time periods increases.

Perhaps the most important improvement in run-time comes from branching on the first time period binary variables (i.e., all possible solutions for $\left(x_1^1, x_2^1, x_3^1\right)$, except for the trivial case of $\left(0,0,0\right)$). While exploiting this structure would seem to be an obvious technique in an IP solver using branch-and-bound algorithms, based on the greatly improved run-times in both LINGO and XPRESS-MP, it appears that the solvers did not use this approach. While it is unclear whether other solvers would need this exogenous branching, it would appear that at least some solvers are unable to identify this structure in the problem quickly. As Figure 4.4 shows, the number of feasible solutions is greatly reduced when only considering one of the branches of the first-stage decision variables. This massive reduction in the feasible region occurs because the second-stage (as well as all subsequent stages) continuous variables and linearization constraints are path-dependent (though the binary variables are not). As Table 3.10 showed, run-times for both LINGO and XPRESS-MP were roughly two orders of magnitude lower after branching on $\left(x_1^1, x_2^1, x_3^1\right)$.

Figure 4.4: Branching on the First Time Period Constraints Reduces Run-Time



Another approach that could potentially improve the run-times and certainly reduce the number of constraints and continuous variables is to simply solve the problem as the original mixed-integer (in this case, binary) nonlinear program (MINLP). Even if the run-times solving it as an MINLP were no better than the linearized version of the problem, such as the IP shown in (3.17), the lack of linearization constraints could make it a viable modeling option for problems such as the five-project, five-time period problem where the number of linearization constraints greatly exceeded the computer's memory.

The MINLP approach was applied to the two-project, two-time period example introduced in Section 3.6.1 using LINGO (XPRESS-MP does not have a global MINLP solver). Additionally, the MINLP approach was applied to a nonlinear version of a *three*-project, *two*-time period problem with fixed budgets[11]. In both

[11] The MINLP version of the three-project, two-time period fixed budget problem is located in Appendix C. The linearized (IP) version of this problem is located in Appendix D.

cases, we branched on the possible solutions for the first time period. Table 4.6 compares the run-time for both cases.

Table 4.6: Fixed-Budget Problem Run-Times (CPU Seconds) for the Linearized IP and MINLP Formulations

Numerical Example	Linearized IP	MINLP
2 Project, 2 Time Period	<1	4
3 Project, 2 Time Period	11	1,825

Based on these limited results, we conclude that the linearized approach is necessary to get reasonable run-times for the three-project, three-time period problem, at least when using the LINGO solver. Moreover, based on these results, while a nonlinear version of the five-project, five-time period would have many fewer constraints than the IP, the run-times for smaller MINLP problems were sufficiently large that it appears there is no reasonable chance that an MINLP version of the problem can successfully produce a solution in an acceptable amount of time. Again, other solvers more specialized in nonlinear optimization, such as BARON, may perform better than LINGO when solving this problem as a MINLP.

Another option we performed was solving in LINGO the MINLP version of the fixed-budget (Model 1) and budget-optimal (Model 3) IP formulations. We evaluated LINGO's local solutions options in terms of solution quality and run-time performance. As Table 4.7 shows, the run-times for the local solution approach were short, but not always optimal. The general conclusion is the local MINLP approach provides fast, but not necessarily global solutions. Therefore, it is not generally viable.

For the three-project, two-time period problem with fixed budgets shown in Table 4.6, we ran a more extensive set of cases by modifying the costs, transition probability and budget data. We obtained local solutions using the MINLP and compared them with the global solution from the IP[12]. Table 4.7 contains the results for five three-project, two-time period example problems with fixed and optimized budgets. For about half of the examples, the MINLP's local solution's objective equaled that of the IP's global solution. The costs, budgets and transition probabilities for the five sample problems are given in Appendix E.

Table 4.7: Comparison of the MINLP's Locally Optimal Solutions with the IP's Globally Optimal Solutions for the Problems in Appendix E

Budget Type	Problem #	MINLP Solution's Objective (Local)	IP Solution's Objective (Global)	MINLP Run-Time (Sec)	IP Run-Time (Sec)
Fixed	1	0.4496	0.5296	<1	<1
	2	0.4321	0.5296	<1	<1
	3	0.4298	0.5090	<1	<1
	4	0.5540	0.5650	<1	<1
	5	0.5785	0.5785	<1	<1
Budget-Optimal	1	0.5799	0.5799	<1	2
	2	0.5296	0.5296	<1	<1
	3	0.4755	0.5730	<1	<1
	4	0.5491	0.6246	<1	2
	5	0.6710	0.6710	<1	2

For the budget-optimal problem, improving the SDP performance beyond the times in this section is particularly important for very large-sized problems where the IP approach is likely not a viable option. The IP could be difficult, if not impossible, to use for such problems, since the linearization constraints in the IP would be too numerous. Similarly, the MINLP's run-times might be too long for globally

[12] LINGO has a local MINLP solver setting. We compared the MINLP local solutions with the linearized IP's global solution (with no optimality gap) to verify optimality.

optimally solutions, or its bounds on optimality might not be sufficiently tight. Therefore, we explore techniques to improve the run-times of the budget-optimal, two-level SDP in the next section.

4.3 Improving the Budget-Optimal Problem Run-Time when Solved as an SDP

Section 4.2 outlined some of the advantages for solving the budget-optimal problem as a one-level IP formulation in (4.5) instead of a two-level SDP version in (4.4). As mentioned previously, the optimal budget allocation is determined in the upper level of the two-level problem in (4.4). While establishing the equivalence of the SDP version of the problem to an IP version illuminates a set of methods for translating other SDPs into IPs (especially in cases where the IP may solve more quickly), there are several cases where modeling this multi-stage real options problem as an SDP is particularly useful.

First, from the numerical results, there is strong evidence that the fixed-budget (i.e., the original “Model 1” from Chapter 2) real options problem solves faster—or at least, no slower—as an SDP than as an IP. Second, as far as we know, the flexible-budget problem (“Model 2” from Chapter 2), where the budget available for that time period is only decided upon at the time period, is only solvable as an SDP without a prohibitively large number of binary variables. Moreover, the flexible-budget problems are often solved quickly as SDPs. Thirdly, it seems quite reasonable that if the budget-optimal problem (“Model 3”) is part of a sensitivity analysis on the value of budget flexibility, presented along with the fixed and fully flexible budget options, then the SDP is the more likely and useful approach. In summary, if we wish to solve all three budget problems (as we do in Chapter 5's CCS case study), then the SDP approach is the only one that can handle all three problem-types. If the fixed-budget ("Model 1") and the budget-optimal ("Model 3") problems are the only ones being solved, then using the IP may be preferred since run-times are lower for some problems. Table 4.8 summarizes the run-time results for the three-project, three-time period problem for all three models.

Table 4.8: Run-Times (CPU Seconds) for the Three-Project, Three-Time Period Problem for All Three Budget Allocation Problems

Budget Problem Type	IP (XPRESS-MP)	SDP (Coded in C++)
Fixed (Model 1)	2	2
Flexible (Model 2)	N/A	4
Budget-Optimal (Model 3)	14	634

While the SDP approach in preferred in certain circumstance, for the budget-optimal problem, the IP approach we are faced with a two-level problem, which can be converted into a one-level problem with continuous budget levels when solved as an IP—and with only moderate increases in run-times. With the SDP, we must first discretize the budget, and then—absent any additional information—solve for all possible combinations.

In the next subsection, we establish a simple bound on the number of times the lower-level SDP problem (i.e., the fixed-budget problem shown in (2.1) – (2.4)) must be solved. Improvements to the bound are then described and proven in the subsequent section, along with the introduction of improved search rules which can greatly reduce the solution time. Performance results from tests on a series of sample problems are then given. Those results are then applied to improve the solution times for the CCS case study in the following chapter.

4.3.1 A Simple Bound on the Iterations of the Lower-Level Problem: The Budget-Increment Method

As introduced earlier in this chapter, the two-level problem for solving for the budget-optimal problem can be written as:

$$
\begin{aligned}
Z_{\text{two-level}} = \max\ & \alpha\left(\vec{B}\right) \\
\text{s.t. }\ & B_1 + \ldots + B_T \le B \\
& \vec{B} \ge 0
\end{aligned}
$$

where $\alpha\left(\vec{B}\right)$ represents the optimal objective value to the real options problem given a set of budgets, $\vec{B}$. If solving this problem as an IP, this is equivalent to a one-level problem, as we proved in Theorem 4.1, since the constraint that $B_1 + \ldots + B_T \le B$ can be incorporated directly in the constraints of the lower-level problem. For the SDP, there is no way to explicitly consider this constraint, so enumerating "all possible" budgets levels is required.

Since the lower-level SDP (i.e., the fixed-budget problem in (2.1) – (2.4)) is solved for a given set of budgets over all time periods, represented by $\vec{B}$, it is important to describe a bound on the number of potential budget combinations the SDP must solve. Since we must discretize the potential budget allocation, an important factor is the budget increment. Ideally, one presumably wants a large budget increment in order to keep the number of lower-level SDPs that must be solved to a manageable amount. On the other hand, too large an increment—without analyzing the solutions in that interval—could result in "skipping over" a potentially budget-optimal allocation.

Costs for projects, along with the budgets for each time period, can be considered discrete, since eventually these values must be specified in some monetary value (e.g., \$100.54 million is \$100,540,000; \$456.56 is 45,656 cents). In the illustrative problems presented earlier in the chapter, we set the budget increment as the greatest common factor (GCF) of all of the costs of the projects. Due to the discrete values these costs can assume, a GCF among the costs (perhaps as small as 1 U.S. cent, for example) is guaranteed to exist. We again denote the c_i^t as the cost of funding project i at time period t. Further, we denote the term $gcf\,(a_1, a_2)$ as the greatest common factor of a_1 and a_2. Given a budget increment (or grid width), b, such that $b = gcf\,(c_1^1,...,c_N^1, c_1^2,...,c_N^2,...,c_1^T,...,c_N^T)$, we denote M as a positive integer such that $M \le \frac{B}{b}$ and that $M + 1 > \frac{B}{b}$. For example, if $N = T = 2$, $(c_1^1, c_2^1, c_1^2, c_2^2) = (4.2, 6.4, 2.2, 3.8)$ and $B = 10$, then $b = gcf\,(4.2, 6.4, 2.2, 3.8) = 0.2$ and $M = \frac{10}{0.2} = 500$. We can then write the following theorem.

Theorem 4.2: *There exists a set of nonnegative integers* $(m_1, m_2,..., m_T)$, *with* $m_t \le M \ \forall t$ *and* $m_1 b + m_2 b + ... + m_T b \le B$, *such that* $(m_1 b, m_2 b,..., m_T b)$ *is an optimal budget allocation to the two-level problem (4.4).*

Proof: As shown in Remark 4.1, there always exists an optimal solution for (4.4). Thus, there is an optimal $\vec{B}^* = (B_1^*, \ldots, B_T^*)$ such that $B_1^* + \ldots + B_T^* \le B$. It suffices to show that B_t^* can be replaced with $m_t b$ for some $m_t \in \mathbf{Z}_+$ and $m_t \le M$. Since a solution to (4.4) must also satisfy (3.6), there is an optimal $x^* = \left(x_1^t, ..., x_N^t\right)^T$ with

$c_1^t x_1^t + c_2^t x_2^t + ... + c_N^t x_N^t \le B_t^* \Rightarrow \frac{c_1^t}{b} x_1^t + \frac{c_2^t}{b} x_2^t + ... + \frac{c_N^t}{b} x_N^t \le \frac{B_t^*}{b}$. Since $\frac{c_1^t}{b}, \frac{c_2^t}{b}, ..., \frac{c_N^t}{b}$

are all integer-valued, the sum $\frac{c_1^t}{b} x_1^t + \frac{c_2^t}{b} x_2^t + ... + \frac{c_N^t}{b} x_N^t$ must also be an integer.

Therefore, $\frac{c_1^t}{b} x_1^t + \frac{c_2^t}{b} x_2^t + ... + \frac{c_N^t}{b} x_N^t \le \frac{B_t^*}{b}$ implies

$\frac{c_1^t}{b} x_1^t + \frac{c_2^t}{b} x_2^t + ... + \frac{c_N^t}{b} x_N^t \le \left\lfloor \frac{B_t^*}{b} \right\rfloor \equiv m_t$, where $m_t \in \mathbf{Z}_+$ and $m_t = \left\lfloor \frac{B_t^*}{b} \right\rfloor \le \frac{B_t^*}{b} \le \frac{B}{b} = M$.

Therefore, the feasible region for (4.3) will be the same if $m_t b$ replaces B_t^*. Thus,

$\alpha(m_1 b, \ldots, m_T b) = \alpha(B_1^*, \ldots, B_T^*) \equiv d^T x^*$. ■

In other words, we have constraint $\left(m_1 + m_2 + ... + m_T\right)b \le Mb \le B < (M+1)b$. Because of the integrality of m_t and M, we can simply consider the number of combinations of the form $m_1 + m_2 + ... + m_T = M$, where each m_t is a nonnegative integer. This value can be considered as the number of ways to put M "items" into T "buckets." This combinatorial analysis problem is equivalent to the "stars and bars" problem, where the number of distinct T-tuples of nonnegative integers whose sum is M is given by the binomial coefficient (Feller, 1968):

$$\binom{M+T-1}{M} = \frac{\left(M+T-1\right)!}{M!\left(T-1\right)!} \tag{4.8}$$

With Theorem 4.2, the equation in (4.8) guarantees one upper bound on the number of lower-level SDPs that must be solved for a given total budget B and an appropriate budget increment b. As we will show in Section 4.3.2, using another approach may produce a better upper bound under some circumstances. For the three-project, three-time period example in Section 4.2.1, where $B = \$30.0$ (million) and $b = \$0.5$ (million), we have that $M = \frac{30}{0.5} = 60$ and $T = 3$. Therefore, the number of lower-level SDPs solved was $\frac{(M+T-1)!}{M!(T-1)!} = \frac{62!}{(60)!2!} = 1{,}891$, since we used the budget-increment method. For the more complicated four-project, three-time period example in Section 4.2.2, $M = 400$, and therefore, the upper-bound on the number of lower-level SDPs solved was $\frac{402!}{(400)!2!} = 80{,}601$ for the budget-increment approach. Even though the lower-level SDP for a fixed budget is solvable in a fraction of a second, the very high number of possible funding combinations is the reason the solution time approached three hours.

It is worth noting how this differs in computational complexity from the flexible-budget SDP presented in (2.5) – (2.8) in Section 2.3.2. Due to the Markov nature of the flexible-budget SDP, the computational complexity should only increase linearly with the addition of time periods, which is significantly better than the increase in the *number* (shown in (4.8)) of lower-level SDPs for the budget-optimal problem. For example, if the four-project, three-time period problem added one additional time period and were a four-time period problem, the number of lower-level SDPs to solve would increase from 80,601 to $\frac{403!}{(400)!3!} = 10{,}827{,}401$ for the same number of budget increments ($M = 400$), more than a hundred-fold increase.

Fortunately, the equation in (4.8) only represents an upper bound for one approach for finding the number of lower-level SDPs necessary to solve. In the next section, we provide methods to reduce the number of lower-level SDP problems that must be solved. From Theorem 4.3 shown below, it can be shown that the necessary number of lower-level problems can be much lower. As the previous example illustrated, such techniques are useful for solving problems with even a modest number of time periods, and are certainly necessary for solving the budget-optimal real options problem for CCS projects in Chapter 5.

4.3.2 An Approach for Reducing the Number of Iterations on the Lower-Level Problem: The Cost-Coefficient Method

In this section, we apply an alternative approach for determining the number of lower-level (fixed-budget) SDPs that must be solved when optimizing the two-level problem in (4.4). This approach creates another upper bound, which can be compared to the budget-increment method, thereby allowing the two-level problem to be solved by whichever method produces the fewest lower-level SDPs. We call this approach the cost-coefficient method.

As Figure 4.1 illustrated, the solution to the subproblem $\alpha\left(\vec{B}\right)$ (which is the real options project selection for a fixed set of budgets, $\vec{B}$) of course depends on the budget allocation. Additionally, the budget allocation only affects one set of constraints, shown in (3.6) as $\sum_{i=1}^{N} c_i^t x_{i,(s_1^{t-1},s_2^{t-1},\ldots,s_N^{t-1})}^t \le B_t \quad \forall\, s_1^{t-1}, s_2^{t-1},\ldots, s_N^{t-1} \in \{1,\ldots,S\}$.

Therefore, the value for $\alpha\left(\vec{B}\right)$, which represents the objective function, is flat over certain intervals of its domain ($\vec{B}$) and only changes when there is a change in one of the binary decision variables $x_{i,(s_1^{t-1},s_2^{t-1},\ldots,s_N^{t-1})}^t$ in (3.6).

While there are a large number of binary variables for each time period specifically, there are only N cost variables, c_i^t, relating to each of the N projects. The next theorem indicates that we need only consider sensitivities on c_i^t. So, any change in the objective function must be a result of some $\sum_{i=1}^{N} c_i^t x_{i,(s_1^{t-1},s_2^{t-1},\ldots,s_N^{t-1})}^t$ becoming feasible (or infeasible). Therefore, we can reduce the number of values for B_t so that we only consider all binary combinations of c_i^t. Since there are N binary variables, there are 2^N combinations at each time period, t. As a result there are, at most, $2^N 2^N \cdots 2^N = 2^{N+N+\ldots+N} = 2^{NT}$ possible combinations for an N project, T project problem. Moreover, for the final time period, it is not necessary (assuming no incentive to conserve the budget) to iterate through all of the final time period's possible funding strategies. Allocating the entire "remaining" budget to the final time period is sufficient, since a lower budget for the final time period will not increase the objective function. That is, once we have chosen values for $B_1, B_2, \ldots, B_{T-1}$, we can simply let $B_T = \max\left(B - (B_1 + B_2 + \ldots + B_{T-1}), 0\right)$. This decrease has the effect of reducing the number of lower-level problems to—at most—$2^{N(T-1)}$. Theorem 4.3 below proves a general property illustrated by the following small example. Given a set of constraints:

$$\begin{aligned} &c_1 x_1 + c_2 x_2 \le n \\ &c_1 x_3 + c_2 x_4 \le n \\ &x_i \in \{0,1\} \end{aligned} \tag{4.9}$$

the set of values (x_1, x_2) is feasible if and only if $(x_3, x_4) = (x_1, x_2)$ is feasible.

Theorem 4.3: *For a given time period,* t*, and a set of states,* $\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1} \in \{0, \ldots, S\}$*, the binary variables* $x^t_{i,(\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1})}$ *satisfy* $\sum_{i=1}^{N} c_i^t x^t_{i,(\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1})} \le B_t$ *if and only if* $\sum_{i=1}^{N} c_i^t x^t_{i,(s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1})} \le B_t$ *is feasible for all* $s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1} \in \{0, \ldots, S\}$.

Proof: Obviously, if $\sum_{i=1}^{N} c_i^t x^t_{i,(s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1})} \le B_t$ is feasible for all $s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1} \in \{0, \ldots, S\}$, then it is true for any specific set, $\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1} \in \{0, \ldots, S\}$. Now suppose there exist N binary variables, $x^t_{i,(\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1})}$ where $\sum_{i=1}^{N} c_i^t x^t_{i,(\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1})} \le B_t$ for a given set of $\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1} \in \{0, \ldots, S\}$. Then, we define the vector $\hat{c}_i^t$ of size N where:

$$\hat{c}_i^t = \begin{cases} c_i^t & \text{if } x^t_{i,(\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1})} = 1 \\ 0 & \text{if } x^t_{i,(\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1})} = 0 \end{cases} \quad \forall\, i \tag{4.10}$$

such that $\sum_{i=1}^{N} \hat{c}_i^t \le B_t$. Since $\sum_{i=1}^{N} \hat{c}_i^t \le B_t$ does not depend on the specific values of the index $\hat{s}_1^{t-1}, \hat{s}_2^{t-1}, \ldots, \hat{s}_N^{t-1} \in \{0, \ldots, S\}$, it holds for all sets of $s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1} \in \{0, \ldots, S\}$. Therefore, we have that $\sum_{i=1}^{N} c_i^t x^t_{i,(s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1})} \le B_t$ for all $s_1^{t-1}, s_2^{t-1}, \ldots, s_N^{t-1} \in \{0, \ldots, S\}$. ■

Solving for all lower-level SDPs using the budget-increment method provides a feasible budget allocation for all time periods. However, for the cost-coefficient method, the bound of $2^{N(T-1)}$ could be lower because many of the $2^{N(T-1)}$

combinations may be infeasible because it is possible that $\sum_{t=1}^{T-1}\sum_{i=1}^{N} c_i^t x_{i,(s_1^{t-1},s_2^{t-1},\ldots,s_N^{t-1})}^t \geq B$

for some values of $x_{i,(s_1^{t-1},s_2^{t-1},\ldots,s_N^{t-1})}^t$. Additionally, if we assume funding for a project cannot occur in time period $t+1$ if it did not occur at t, then further combinations could be eliminated.

On the other hand, as we discuss in Chapter 5, $2^{N(T-1)}$ assumes: i) a single funding level and ii) that there is only one value, c_i^t, for a given project i and time t. However, if cost is also determined by the current state of the project, then the cost-coefficient vectors must be written as $c_{i,s_i^{t-1}}^t$. These two conditions do not hold for the CCS real options model presented in Chapter 5. Consequently, we develop additional methods for managing run-times for the budget-optimal CCS real options problem in Section 5.6.

Since either the cost-coefficient method in this section or the budget-increment method from Section 4.3.1 will produce a budget-optimal solution to the SDP version of the two-level problem in (4.4), we can compare the methods to determine which method produces the fewer lower-level SDPs that must be solved. The next section shows whether the SDP run-times from Section 4.2 to determine if the run-times can be improved using the cost-coefficient method.

4.3.3 Testing the Improved Performance of the Two-Level SDP

As stated previously, for the three-project, three-time period problem in Section 4.2.1, the run-time is based on using a budget-increment method. As outlined in Section 4.3.1, this approach resulted in 1,891 lower-level, fixed-budget SDPs to solve. Using the cost-coefficient method described in the previous section, we can solve to optimality in no more than $2^{N(T-1)} = 2^{(3)(2)} = 2^6 = 64$ SDPs. Since $\sum_i c_i^1 + \sum_i c_i^2 < B$, we are not able to eliminate any of the 64 lower-level SDPs, because no cost coefficient combination exceeds the total budget in the first two time periods. The run-time results for the three-project, three-time period and the four-project, three-time period budget-optimal problems are shown in Table 4.9 for the different solution approaches.

Table 4.9: Comparison of the Run-Times for the Budget-Optimal Problem under Solution Approaches

Numerical Example	IP (XPRESS-MP)	SDP (Budget Increment Approach)		SDP (Cost Coefficent Approach)	
	Total CPU Sec.	Total CPU Sec.	Lower-Level SDPs	Total CPU Sec.	Lower-Level SDPs
3 Project, 3 Time Period	14	634	1,891	29	64
4 Project, 3 Time Period	284	11,143	80,601	216	256

The problems in Table 4.9 were solved sufficiently quickly using the cost-coefficient approach that additional methods to reduce run-times are not necessary. However, the cost-coefficient approach still requires an exponentially increasing number of lower-level SDPs as the number of time periods grows. Thus, larger problems would need to reduce the number of lower-level SDPs in order to be practical to solve. For example, a five-project, five-time period problem could have as many as $2^{N(T-1)} = 2^{20} = 1{,}048{,}576$ lower-level SDPs. Of course, many of these lower-level problems will be infeasible because they violate the total budget or other logical constraints.

While the run-times for the cost-coefficient method are comparable to that of the IP, further reductions in the IP could be gained. Moreover, it is relatively straightforward to add side constraints and easily run with many with different solvers, which is not the case for SDP. Nevertheless, intelligent search techniques must be implemented to solver larger problems for the budget-optimal allocation, which we implement for the CCS case study in the next chapter.

Solution run-time for fixed-budget, flexible-budget, and budget-optimal real options problems for a case study are provided in the following chapter. Specifically, in the next chapter, the application of the multi-stage real options model to a series of CCS projects in the European Union is presented after the types of technologies are described, specific projects are identified, and the costs, transition probabilities and knowledge spillover are described. Solving all three types of budget allocations demonstrates the merit of the real options approach, especially the improved value of budgetary flexibility.

Chapter 5: Optimal Funding Strategies for Carbon Capture and Storage (CCS) Projects in the European Union: A Real Options Case Study

Among the various technologies for CO_2 abatement available, carbon capture and storage (CCS) technologies are expected to hold significant abatement potential if they reach market maturity within the next several years. One barrier to the large scale implementation of the technology is the lack of demonstration projects that validate the technologies. Several projects in the European Union (EU) are currently under development to implement the CCS technology on a large scale and may be subject to public funding under EU support initiatives. These CCS projects may try to develop any combination of three types of operating levels: pilot, demonstration and full-scale, representing progressing levels of electric power generation capability. Several projects have commenced at the pilot project level, with full-scale commercial operation levels planned for approximately 2020. While CCS projects outside the EU exist, such as FutureGen 2.0 in the United States (DoE, 2011), those projects would not be subject to the same EU public funding agency, so they are excluded from consideration.

In this chapter, we apply the analytical funding decision methods outlined in the previous chapters to a series of CCS projects in the European Union. Moreover, since these projects are working on both competing and complementary technologies with progressive levels of improvement, such a framework is ideal for the multi-stage real options competition. Prior to introducing the specific real options model, it is important to describe the current CCS, as well as carbon capture, transport and storage (CCTS) technologies in sufficient detail.

The first section of this chapter provides an introduction to CCS technologies and describes three specific ones: post-combustion, pre-combustion, and oxyfuel. Section 5.2 describes the actual CCS projects that are being undertaken and the data available and assumptions based on expert elicitation. Section 5.3 outlines the three specific multi-stage real options model, based on the framework presented in this dissertation. Section 5.4 presents the data for the model obtained through subject-matter expert interviews. Results are then presented in Section 5.5 with sensitivities for many alternatives: technology type, funding level, and budget allocation and flexibility. In Section 5.6, we then provide statistics on run-times and model complexity, along with a heuristic for improving the run-times for the budget-optimal problem, followed by a summary section on the policy implications of the results. This chapter is based on the work of Eckhause and Herold (2011), with the exception of Section 5.6, which is entirely unpublished work of Eckhause.

There are a total of eight CCS projects modeled in this chapter, but they are analyzed in two separate problems: three projects for one technology and five for two related technologies. While the project types, models, data, complexity and run-times are discussed in detail, Table 5.1 provides a summary of the breadth of the problems that were solved. The large ranges on run-times and problem sizes illustrate the curse of dimensionality common to many Markov decision processes (Puterman, 1994).

Table 5.1: Bounds on Problem Sizes and Run-Times Modeled in Chapter 5

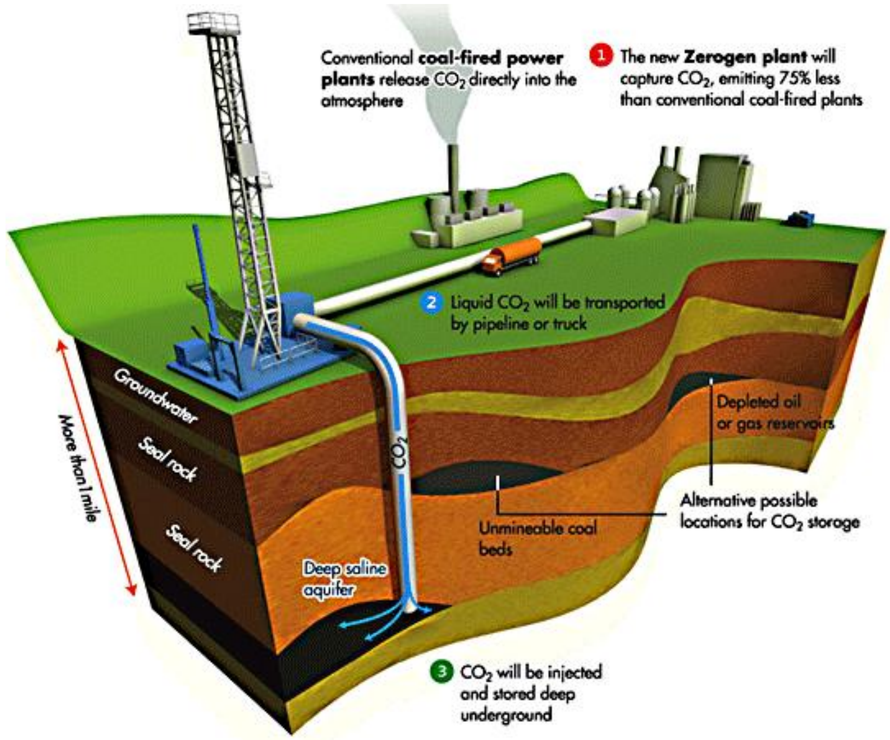
	3 Pre-Combustion Projects	5 Post-Combustion & Oxyfuel Projects
Smallest Problem: State Variables	90	196
Largest Problem: State Variables	2,122,200	23,708,160
Shortest Run-Time (CPU Seconds)	<1	<1
Longest Run-Time (CPU Seconds)	1,873	1,828

5.1 Overview of Three CCS Storage Technologies

The global efforts to combat climate change have led to a high degree of innovation and investment in the energy sector (IEA, 2011). In recent years, the development of renewable energy sources, among other measures, have already led to a significant reduction in CO_2 emissions from electricity production within the European Union (PWC, 2010). Another strategy to further reduce the emissions from burning fossil fuels in power plants is through the technology of CO_2 capture, transport and storage. These technologies can be applied to power plants and CO_2-intensive industries to capture a 90% share of the CO_2 emissions in the flue gas[13]. Storing this CO_2 in underground reservoirs mitigates its impact as a greenhouse gas effect in the earth's atmosphere. Figure 5.1 provides a diagram of how CCS and CCTS technologies are typically designed. Note that transportation of the CO_2 can occur via truck or pipeline. Depleted oil or gas beds, or deep saline aquifers are ideal locations for storing spent CO_2.

[13] Flue gas refers to combustion exhaust gas produced by power plants that is generally released to the atmosphere.

Figure 5.1: Overview of CCS Plant (Zerogen, 2011)



The high potential for CCS in the global fight against climate change was outlined by the Intergovernmental Panel on Climate Change (IPCC, 2005 and IPCC 2007) and the International Energy Agency (IEA, 2008). Both institutions reached the conclusion that CCS could provide a high share of CO_2 abatement in the 21st century and that the technologies would significantly lower the global costs of climate change mitigation. In this dissertation, we focus on the first generation capture technologies: post-combustion, pre-combustion and oxyfuel.

5.1.1 Post-Combustion CCS Technology

Post-combustion technologies separate the CO_2 out of the flue gas after combustion. This process is comparable to flue gas desulphurization, which has long has been mandatory for power plants to filter SO_2 emissions. It was first applied in the 1980s for the capture of CO_2 from ammonia production plants for commercial uses of CO_2. Once removed from ammonia, the captured CO_2 is used in food production (e.g., to carbonate soft drinks and soda water). Post-combustion chemical absorption technologies represent the most commercially available CO_2 capture technologies. However, the technology so far has only be used for the treatment of very clean gas mixtures containing no or few impurities such as dust, SO_2, NO_2 (Kanniche, *et al.*, 2010). Currently, plants operating with this technology are capable of capturing between 1000 to 4000 tons of CO_2 per day. However, to comply with the emissions of a 1 GW lignite-powered plant, upscaling to 13,000 tons of CO_2 per day would be required (Vallentin, 2007). An advantage of the post-combustion implementation is that the technology can be retrofitted to existing power plants. It can also be retrofitted to other plants that produce large amounts of CO_2, such as iron or cement manufacturers.

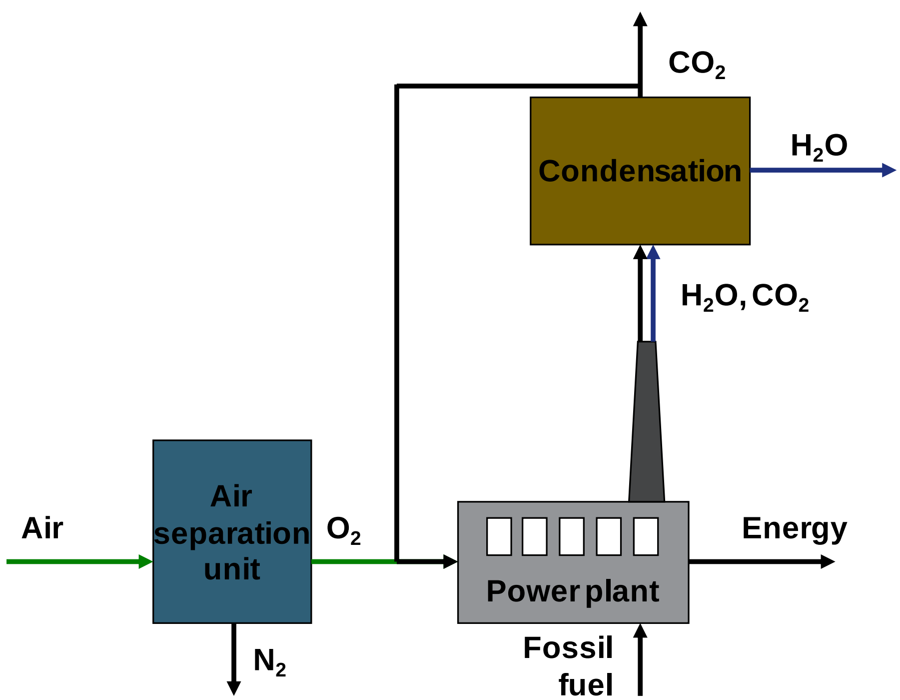
5.1.2 Pre-Combustion CCS Technology

Pre-combustion capture refers to the treatment of CO_2 and H_2 after the gasification process of coal, biomass or the steam reformation of natural gas. Due the limited number of power plants operating with this technology, the coal-based internal gasification combined cycle (IGCC) technology itself is still in the demonstration phase. Due to the increasing process complexity, proven refinery-based plants are not based on coal, but on natural gas or liquid hydro-carbons; and the hydrogen is used in the chemical industry instead of power generation. The high investment costs would need to be reduced to a level in which they allow for competition with other capture technologies (e.g., by developing economic and efficient hydrogen selective membranes). Currently, post-combustion offers significantly less expensive investment costs, as is shown later in this chapter.

5.1.3 Oxyfuel CCS Technology

The oxyfuel process aims at the separation of gases before the combustion. By combusting fuels in a pure O_2 and CO_2 atmosphere (with up to 60% as O_2), one achieves a sequestration-ready gas stream, containing simply CO_2 and H_2O. The water vapor can then be easily removed by simply cooling the gas. In this sense, the oxyfuel process has elements of both pre-combustion and post-combustion technologies.

Figure 5.2: The Oxyfuel Process



Nevertheless, there remain several open questions. First, the presence of incondensable gases (oxygen, nitrogen, argon) in the CO_2 flow transported in the supercritical state can cause vibrations and shock loads in the pipeline (Kanniche, *et al.,* 2010). Secondly, because of the separation required, there is reduced efficiency, which may further decrease if additional SO_2 removal is required. Thirdly, so far, the technology has not been demonstrated on a larger scale than the demonstration level, so there may be unforeseen technical problems. For oxyfuel, the actual CCS component is still in the pre-demonstration phase; only a limited number of pilot projects have been realized for power plants (Herold, *et al.*, 2010a). Therefore, its impact on the marginal CO_2 abatement curve remains uncertain (Baker, 2009).

5.2 CCS Initiatives in the European Union

The International Energy Agency (IEA, 2008) has reached the conclusion that reducing global CO_2 emissions by 50 percent by 2050 compared to 1990 levels would be greatly aided by the commercial availability of CCS technologies. Otherwise, society could face additional mitigation costs of up to $1.28 trillion over the next 40 years. To reach that target, the IEA (2009) Blue Map scenario outlines ambitious development plans in CCS demonstration over the next ten years. Specifically, a total investment in 100 capture plants, with a minimum of 10,000 km of pipelines and the storage of 1.2 billion metric tons of CO_2, are required to make CCS a serious abatement technology by 2050. As the industry has failed to realize demonstration projects on the scale and scope required to meet the Blue Map target, the EU, among other governments, has committed billions of Euros to co-finance CCS demonstration projects (Herold, *et al.*, 2010a). Particularly noteworthy are the following:

- The European Energy Program for Recovery (EEPR), part of the European Economic Recovery Plan presented by the European Commission on November 26, 2008, allocates €1.05 billion to six CCS demonstration projects. Five of the six schemes have been awarded an initial subsidy of 180 million Euros each, with additional funding coming from national governments. The Italian project, Enel, will receive 100 million Euros (Reuters, 2009).
- On February 3, 2010, EU member states agreed on the use of the revenues from the sale of 300 million CO_2 certificates from the New Entrants Reserve of the EU Emission Trading Scheme (ETS) (NER300, 2010). Earnings will finance CCS demonstration projects (200 million certificates) and innovative renewable energy technologies (100 million certificates). Depending on the allowance price, several billion Euros could become available for CCS. The agreement proposes to fund eight CCS projects, with at least one, but not more than three, of each capture technology.
- There is a portfolio of national support schemes, which provide funding on the basic research level, as well as for implementation of the technology in pilot and demonstration plants (ZEP, 2010).

These programs are supposed to accelerate the CCS demonstration and commercialization not only by providing financial support, but through technology transfer. Projects that receive public funding must disclose the acquired knowledge the developed technologies, while making it available to competitors. This requirement means that subsequent projects will likely benefit from more advanced and cheaper technology. This knowledge transfer is supposed to be greatest for the first CCS projects, as uncertainty on costs, performance and feasibility is highest. This stipulation implies that projects that receive public funding should neither be too risky, nor should they be too insignificant. These projects should also not apply components which have already reached a high level of maturity. Projects that are too risky have a high chance of running out of money without achieving the target, while projects that are too small or of marginal technical innovation will likely not provide much additional insight and, thus, will not lead to a return on the public investment.

Allocating the available billions of Euros among a portfolio of CCS projects therefore raises a series of questions: the number and scale of projects to be funded by the EU, the optimal funding level of the projects, the level of flexibility of spending this money over time, and the optimal timing of abandoning unsuccessful or delayed projects. Projects differ in many respects other than the capture technology. For instance, the plant size within these projects varies from small pilot plants to mid-size demonstration projects to projects on a commercial scale (ZEP, 2008). Thus, possible CCS projects differ not only in costs, but the extent of supplemental public funding required. Finally, they also have different probabilities of successful realization, which depend on both funding level and knowledge gain from the other projects.

Taking the perspective of a funding agency, we employ a real options framework for determining an optimal funding strategy for project selection for the development of full-scale carbon capture plants. Specifically, we formulate and solve a stochastic dynamic program for obtaining optimal funding solutions in order to achieve at least one successfully operating full-scale CCS plant by a target year (in this case, 2022). Using a subject-matter-expert interview approach, we obtain the needed data on projects costs and technology success probabilities. The SDP model is solved for multiple budgets and budget allocation schemes. Sensitivities on knowledge spillover, where projects' costs and transition probabilities may be improved based on the progress of a competing project, are presented. In the next section, we describe the real options model, which is used to determine the optimal funding for the CCS portfolio.

5.3 Applying the Real Options Model to the CCS Projects

While virtually all public-sector initiatives involve some risk, ones that involve uncertain technological capabilities are particularly perilous. Controlling for the risk is critical to the overall success of the technology, especially for a long-term, one-of-a-kind R&D activity (Ceylan and Ford, 2002). Funding a series of projects is not an uncommon way for government managers to mitigate the technical risk associated with R&D by establishing decision points and multi-project, parallel development strategies (Department of the Navy, 1998).

While there exists a robust literature on using real options to mitigate risk (Dixit and Pindyck, 1994; Huchzermeier and Loch, 2001), and even using real options approaches to mitigate emissions for power generation (Burchett and Biswas, 2002), there are many unique aspects to public-sector R&D which differs from the private-sector undertakings, including valuation of non-market traded goods, which we outlined in Chapter 2. For the CCS projects, we employ the SDP framework introduced in Chapter 2 (Eckhause, *et al.*, 2009) to solve a multi-project, multi-time period competition in which each projects has its own costs, probabilities of success and states. The objective function we employ for the CCS projects is similar to the one employed in the Chapter 2 examples in that we are maximizing the probability that at least one project is "successful." Our work expands upon that objective function to consider competing technologies, knowledge spillover, state-dependent probabilities, and solving a budget-optimal allocation problem.

Given the emphasis in Chapter 3 and some of Chapter 4 on the IP approach, it is important to the state the reasons why this approach was not implemented for the CCS projects. First and foremost, there are six time periods in which funding decisions are made. Since using the linearization constraints illustrated in Chapter 3 requires creating path-dependent variables, the number of variables becomes unwieldy. For example, the five-project, five-time period problem in Chapter 4 far exceeded the memory available (2 GB). While some of the CCS project cases modeled have fewer than five projects, the number of states for the projects ranges from 6 to 14, significantly more than the five-state projects used in the problems in Chapter 4. As we noted in Table 5.1, the number of state (and decision) variables can be as large as 20 million. Secondly, in addition to solving the fixed-budget and budget-optimal problems, we wish to solve the flexible-budget real options problem, which is, as far as we know, essentially unsolvable using an IP formulation.

Considering this CCS project funding problem as a multi-project, multi-stage competition, each stage represents a decision time period for the funding of a project. Each project funding decision represents an option to the EU. The cost of purchasing each option is the amount of initial funding required for each project's development. An option is exercised through the award of a continuation of funding. We seek an optimal portfolio of options (CCS projects) to fund at each stage in order to maximize the overall success of the R&D capability.

5.3.1 Model 1: The Fixed-Budget Real Options Formulation

Suppose the funding available at each time period (e.g., every two years) is fixed. That is, the amount of subsides available must be allocated during that time period and cannot be saved for (or borrowed from) future time periods. Using the model presented in Eckhause and Herold (2011), whose model is an extension of Eckhause, *et al*. (2009), we define the following terms:

- Let there be $t = 1,...,T$ time periods over which funding decisions for the $i = 1,...,I$ CCS projects are made. Funding for project i may occur at different levels, $l \in L$.
- Let $s_{it} \in S$ be the state of project i at time period t from a set of possible project states, S. We denote the state of all projects at time period t as S_t.
- Let $s_{\text{full}} \in S$ denote the state where a project has reached full-scale capacity.
- Let $X_{itl} \in \{0,1\}$ be the binary decision variable whether fund project i at level l at time period t.
- Let c_{itlS_t} be the cost of funding project i at level l time period t, given that the state of system at time t is S_t. For cases of knowledge spillover, this cost vector is a function of multiple projects, not simply project i.

- Let B_t be the budget available for time period t.

Based on the expert elicitations and data available in the literature outlined in Section 5.4, we are able to obtain the estimates for costs c_{itlS_t} for each of the projects. Additionally, these elicitations provide us the state transition probabilities for each of the projects, i.e., $\mathbf{P}[s_{it+1} = s \mid S_t, X_{itl} = 1]$. In other words, given the current state of all projects (or perhaps some subset of those projects) and a funding decision, we know the probability of project i winding up in state s for all $s \in S$. The values for B_t are based on assumptions about funding levels, though we perform several sensitivity analyses on those values. Therefore, we have the set of feasible funding decisions at time period, t:

$$X(S_t) = \begin{cases} (X_t) \in \{0,1\}^{I \times L} : & \sum_{i,l} c_{itlS_t} X_{itl} \le B_t \\ & \sum_{l \in L} X_{itl} \le 1 \quad \forall\, i \in I \end{cases} \tag{5.1}$$

We choose our objective function to maximize the probability of having at least one successful CCS project with fully functional capabilities at the end of the final time period. We can then write our value function for the stochastic dynamic program with a standard recursion equation (Puterman, 1994):

$$V_t(S_t) = \max_{X_t \in X(S_t)} \mathbf{E}\{V_{t+1}(S_{t+1}) \mid S_t, X_t\} \quad t = 1, \ldots, T \tag{5.2}$$

Since our objective is to have at least one project reach state s_{full}, we have failed to meet the objective if this property does not hold. Thus, we write the boundary condition of the stochastic dynamic program as[14]:

$$V_{T+1}(S_{T+1}) = \begin{cases} 1 & \text{if} \quad s_{i,T+1} = s_{\text{full}} \quad \text{for some project } i \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

This condition provides no "consolation" prize for a project reaching any level below the successful full-scale implementation, nor does it provide "extra credit" for having multiple projects reach that objective. While other objective functions are possible, this formulation ensures an efficient use of funding towards one successful implementation.

5.3.2 Model 2: The Flexible-Budget Real Options Formulation

In the model presented in the previous subsection, we assumed that the budget for each time period was fixed. Following the flexible-budget formulation provided in Eckhause, *et al.* (2009), we now suppose that the total public funding budget available (denoted as B) is fixed, but that the budgets for each time period, B_t, do not need to be determined in advance. We simply add the constraint: $B = B_1 + \ldots + B_T$. For this problem, our feasible region (and thus, our state definition) must include a budget variable indicating the budget remaining to the decision maker at each time period. Therefore, at time period, t, the state of the system is $(S_t, B_t) \in \left(\prod_{i \in I} S \right) \times \mathbf{R}_+$; and the set of feasible funding decisions for time period, t, is:

[14] The boundary condition is for the final observed time period, $T+1$. While decisions are only made for $t = 1, \ldots, T$, the outcome of the decisions made at time T is realized at time $T+1$.

$$X(S_t, B_t) = \begin{cases} (X_t, B_{t+1}) \in \{0,1\}^{I \times L} \times \mathbf{R}_+ : & \sum_{i,l} c_{itlS_t} X_{itl} \le B_t \\ & \sum_{l \in L} X_{itl} \le 1 \quad \forall\, i \in I \\ & B_{t+1} = B_t - \sum_{i,l} c_{itlS_t} X_{itl} \end{cases} \tag{5.4}$$

The boundary condition for the stochastic dynamic program follows as:

$$V_t(S_t, B_t) = \max_{(X_t, B_{t+1}) \in X(S_t, B_t)} \mathbf{E}\{V_{t+1}(S_{t+1}, B_{t+1}) \mid S_t, X_t\} \quad t = 1, \ldots, T \tag{5.5}$$

In order to solve this dynamic program, a discrete budget increment must be a component of the state variables. While this addition creates a significant state expansion, reasonable increments could be handled in our cases. Moreover, the IP formulation is essentially untenable for the flexible-budget problem, as it would require the creation of a *binary* variable for every possible budget increment, which could be as many as 20 million binary variables for one of the aforementioned CCS problems (Table 5.1).

This budget flexibility can provide a great benefit, since it can increase the value of the objective function as the feasible region is much larger. Indeed, the ability to combine budgets to consider more scenarios can greatly increase the overall success probability. Of course, complete flexibility for the budgets in each time period may not be realistic. However, even some degree of budget autonomy could greatly increase the probability of successful project implementation, as we demonstrate in the numerical results in Section 5.5.

5.3.3 Model 3: The Budget-Optimal Real Options Formulation

Eckhause, *et al.* (2009) provided formulations for the fixed and flexible real options models we apply to the CCS projects in this chapter. There is the potential for another budget allocation scheme: one must specify the budgets *a priori*, but there is the ability to optimize them so that the overall probability of a fully functional CCS project is maximized. In this sense, we are solving a series of fixed-budget problems outlined in Section 5.3.1 and choosing the allocation that provides the greatest objective function. However, this allocation must be specified *a priori*; so it will not provide as great a benefit as the flexible-budget model.

On the other hand, it is likely that the fiscal and political freedom to have budgets remain unspecified until the year of the funding decisions is not realistic. This budget-optimal allocation problem can be considered as a two-level problem (Fortuny-Amat and McCarl, 1981), where the upper-level problem is the budget allocation and the lower-level is the fixed-budget real options model. Letting $\mathbf{B}$ represent the vector of budgets available at each time period (i.e., $\mathbf{B} = (B_1,...,B_T)$) and $\alpha(\mathbf{B})$ be a solution to the fixed-budget real options model presented in Section 5.3.1, the budget-optimal real options model (with objective function $Z_{\text{two-level}}$) can be written as a two-level problem:

$$
\begin{aligned}
Z_{\text{two-level}} = \max\ & \alpha(\mathbf{B}) \\
\text{s.t.}\ \ & B_1 + \ldots + B_T \leq B \\
& B_1, \ldots B_T \geq 0
\end{aligned}
\tag{5.6}
$$

Since $\alpha(\mathbf{B})$ is solved as a stochastic dynamic program, we know of no way to explicitly incorporating the constraint $B_1 + \ldots + B_T \leq B$ (as one could with an integer program, as we showed in Chapter 4). Thus, we solve for an optimal budget by solving a series of fixed-budget stochastic dynamic programs. Techniques for reducing the number of fixed-budget subproblems, along with solution run-times, are discussed in the Sections 5.5 and 5.6; these results follow from the methods described in Chapter 4. In the next section, we describe the methods we used to obtain transition probabilities, costs and other necessary inputs to the three models outlined in this section.

5.4 Expert Elicitations and Survey Results

A survey of experts on the likelihood of a technical breakthrough and the associated costs or the performance of a new technology is part of the standard assessment tools in science, business and politics (Duong *et al.*, 2007). Chan, *et al.* (2011) define the expected cost and efficiency of different capture technologies in 2010 and 2030 as benchmarks and ask subject matter experts how those will change under public R&D funding scenarios. They conclude that the most important area of demonstration is the IGCC, but that most funds should be allocated to the most mature and market-ready post-combustion technology. While public funding might help to demonstrate the technologies on the medium- and large-scales, the experts expect it to contribute little towards the reduction of the CCS investment costs. Hansson and Bryngelsson (2009) collect expert opinions on economic, technical and institutional aspects along the CCS value chain. They find excessive optimism regarding the large-scale, commercial availability of the CCS technology and underestimation of uncertainties among the 24 experts. This conclusion is of particular importance, as those experts also often advise policy makers. Baker, *et al.*

(2009) apply expert elicitation to analyze the impact of public R&D for CCS on the future costs of CO_2 abatement. Their expert panel strongly disagrees on the expected CCS investment costs, but according to the authors, cost uncertainty does not pose a barrier to a successful CCS application.

To address the question of an optimal funding strategy for a predefined portfolio of CCS demonstration projects, we are interested not only in the expected cost of the technologies, but also in the probability that such projects will be realized. To our knowledge, no previous study provides such data; we therefore use individual judgments from six subject matter experts from research and academia to a predetermined questionnaire on the required information[15]. This approach is in contrast to the closely related Delphi method, where a consensus among experts is reached during a number of judgment rounds (Dalkey, 1969). While individual elicitations rule out the problem of group dynamics, it leaves the interpretation of the results to the authors of the questionnaire. This approach increases the risk of misinterpretation of results due to a biased or less informed analyst (Chan, *et al.*, 2011). In the following subsections, we present the outcomes of the expert elicitation we conducted. The results are manifested as success probabilities of implementing each of the CCS technologies successfully for the first time in coal-fueled power plants; these probabilities change based on the available budget.

[15] We would like to thank: Dr. Joachim Geske, Dr. Peter Markewitz, and Dr. Stefan Vögele from Forschungszentrum Jülich (Jülich Research Center in Jülich, Germany); Mr. Michael Trompelt from the Technische Universität Bergakademie Freiberg (Freiberg University of Mining and Technology); and two anonymous experts for their estimates of the CCS costs and success probabilities for the projects considered in this chapter.

5.4.1 Cost Data for the CCS Technologies

Costs and success probabilities for the first CCS demonstration projects are highly speculative and depend on many factors, e.g., the scale and scope of the project, the political and public option towards the technology, the storage situation, available funding, the expected availability of technology substitutes like renewable energy technologies, and the anticipated carbon prices. In this dissertation, we abstract from any existing cost, technological, and institutional related hurdles existing along the transport and storage part of the CCS value chain and focus solely on the construction of the power plants using the CO_2 capture technologies. This focus is based on the type of uncertainty around each of the steps along the value chain. The transport and storage of CO_2 is a mature technology, but faces high regulatory and legal hurdles. Also, across Europe, citizens affected by potential, nearby storage sites typically strongly oppose storage. This resistance has led to a number of delayed or cancelled projects (Herold, *et al*., 2010a).

In this chapter, we focus on the technical uncertainty only surrounding carbon capture. Nevertheless, a diverse set of announced carbon capture projects remains, which will test different capture technologies on various scales and levels of maturity. Therefore, we develop individual project data from a generalized questionnaire on CCS costs and the probability of implementing the technology for the first time. As a starting point, we used the cost estimates for a 400 MW coal-fired CCS plants presented in Tzimas (2009), shown in Table 5.2. We then asked the experts how upscaling or downscaling the size of a CCS plant will affect those costs.

Table 5.2: Investment Cost of Different Plants with and without CO_2 Capture (Tzimas, 2009)[16]

	Investment Costs for Demonstration Project [€/kW]	Efficiency [%]
Standard coal plant	1,478	46
Post-combustion plant	2,500	35
Pre-combustion plant	2,700	35
Oxyfuel plant	2,900	35

The simple, unweighted, mathematical mean of the experts' cost estimates are shown in Table 5.3. It shows that all three technologies benefit from economies of scale. For the most mature technology—post-combustion capture—the penalty for downscaling is lowest. The pre-combustion capture is best implemented on medium and large-scale power generation, which is based on the complexity of applying the IGCC technology to solid fuels, which has been implemented in only a few projects with unsatisfying results (Herold, *et al.*, 2010b).

Table 5.3: Estimated Costs for the First CCS Demonstration Project by Technology in €/kW[17]

	125 MW	250 MW	500 MW	750 MW	1000 MW
Post-combustion	3000	2875	2500	2125	1875
Pre-combustion	3780	3240	2700	2295	1890
Oxyfuel	3770	3480	2900	2465	2175

The investment cost advantage of larger projects is nevertheless outweighed by the sharp decreased in the project success probability for building the first carbon capture plant. Technology success, as presented in Table 5.4, is defined as reaching a

[16] "€/kW" means Euros (in 2008) per kilowatt generated.
[17] The starting values for the 500 MW plants are taken from Tzimas (2009).

capture rate of 90 percent and staying within the predefined cost threshold in € per kW installed capacity outlined in Table 5.3. All the panel experts estimate a significantly higher chance of technology success for smaller power plants and the corresponding capture unit.

Table 5.4: CCS Technology Success Probabilities

	125 MW	250 MW	500 MW	750 MW	1000 MW
Post-combustion	100%	86%	75%	68%	60%
Pre-combustion	81%	69%	58%	46%	41%
Oxyfuel	98%	81%	68%	54%	48%

In a further step the experts were asked to estimate how a change in the budget (for instance, due to a higher public funding) would affect the technology success probabilities. These estimates are based on the supposition that the scale and scope of the project are unaffected by a change in the budget. If this assumption holds, a higher budget is assumed to increase the project success probability and vice versa; Table 5.5 provides a summary of those surveys. However, a higher budget can also allow for more components to be tested; if this is the case, no answer on the change in the overall success probability can be calculated. The lowest probability of success is defined for the "no funding" case. The decision to undertake such a project is external to the funding agency and therefore not considered in the model. We define an increase in the investment budget of 20 percent per kW as public "Funding Level 1" and an increase in the budget of 40 percent as public "Funding Level 2."

Table 5.5: Technology Success Probabilities of the First Demonstration Project Subject to Changes in the Budgets Estimated in Table 5.2

	- 20% (No Funding)	- 10%	€/kW for 500 MW (Funding Level 1)	+ 10%	+ 20% (Funding Level 2)
Post-combustion	65%	68%	75%	80%	85%
Pre-combustion	46%	50%	58%	62%	66%
Oxyfuel	55%	59%	68%	73%	77%

In Table 5.4, the project success probabilities are estimated to be at their maximum for the smallest project stage available. Successful completion of this stage therefore allows companies to gain a first experience with the innovation itself on a small scale and at lower total costs. The experience the company gains may allow for developing the technology further by lowering its investment costs. Table 5.6 reflects the expert panel's opinion on how the successful completion of a previous project stage influences the technology success probabilities and the costs of the subsequent technology stages. For example, for post-combustion, a company successfully completing the pilot stage can expect a 20 percent increase in the success probability of the following, demonstration stage, while the costs of building this demonstration plant will also drop by 5 percent. This learning effect is of paramount importance for the real options model and the optimal funding strategy, as it determines whether it is beneficial to test the technology on various stages or not.

Table 5.6: Change in Technology Success Probabilities and Costs for the Subsequent Stage Given Successful Completion of Current Stage

	Change in Technology Success Probability	Change in Technology Costs [€/kW]
Post-combustion	+ 20%	- 5%
Pre-combustion	+ 17%	- 5%
Oxyfuel	+ 20%	- 10%

Finally, we asked which of the European electricity companies planning to build a CCS demonstration project is a leader in knowledge and experience about each capture technology. This leadership in early adoption translates, according to the experts, into a 5 percent increased technology success probability for that company for all three stages. The companies listed in Table 5.7 have a history of early innovation. RWE, for instance, gained experience with IGCC technology during the 1980s and 1990s, while Vattenfall has already successfully implemented the oxyfuel technology in a 30 MW pilot plant in Germany.

Table 5.7: Electricity Supplier Estimated to be Most Advanced for Each Technology Line

	Innovation Leader	Increased Probability of Success
Post-combustion	E.ON	5 %
Pre-combustion	RWE	5 %
Oxyfuel	Vattenfall	5 %

5.4.2 Effects of Knowledge Spillover

One of the main reasons for public funding of R&D initiatives is so that firms can recover the full benefit of their research investment. The innovating firm creates knowledge externalities, or so-called spillovers (Jaffe, *et al.*, 2006). Then, the improvement results in benefits beyond those enjoyed by the original firm (Stern, 2007). In the absence of recovering these benefits, there will be insufficient funding of R&D by private firms. Public funding is therefore designed to compensate for the under investment. In the presence of pollution control, society might benefit not only from R&D spillovers, but also from lower emissions. Nevertheless, Fischer (2008) concludes that R&D policy support would only be justified if: i) some spillovers are realizable; and ii) at least a moderate pollution internalization policy is in place. Otherwise, public support in favor of R&D is not justified as any progress in pollution control lacks incentives for its application. Thus, an intense R&D policy cannot necessarily compensate for a deficient internalization policy.

Reis and Traca (2008) assert that it is in the inability to appropriate the returns (i.e., reaping profits and protecting from imitation) received from R&D that is a key deterrent to innovative undertakings and, by extension, to economic growth. Policy can respond to that innovation market failure by enforcing intellectual property rights or by funding R&D. Because the companies under consideration for the CCS technology projects receive EU funding, they are thus obliged to make patent-protected technologies developed in these projects available to competitors, in the form of compulsory licensing (IZ Klima, 2010).

In this study, we distinguish between learning among projects applying the same capture technology and cross-technology learning, meaning that a successfully operating post-combustion plant will, for example, have a positive impact on costs or success probability of the oxyfuel projects. For cross-technology effects, we there are divergent opinions. First, some of our experts expect no cross-technology learning. This view is based on the assumption that the successful implementation of one technology significantly lowers the need for a second technology. Indeed, there is only one dominating flue gas desulphurization technology applied today, whereas in the beginning of the diffusion process of that technology different options competed (Rubin, *et al.*, 2005). With respect to CCS, there may be exceptional cases. For instance, the pre-combustion technology is not applicable to an existing power plant other than IGCC, and therefore may require a complementing technology such as post-combustion or oxyfuel capture. On the other hand, some industrial processes, like hydrogen and biofuel production, or advanced technologies for iron and steel production use, would require pre-combustion capture.

We therefore group the technologies into substitutable technologies (post-combustion and oxyfuel capture) and the complementing technology, pre-combustion capture. The results of our expert interviews on cross-technology spillover effects for the technology success probabilities are shown in Table 5.8. For the cross-technology cases, all experts expect little-to-no impact.

Table 5.8: Knowledge Spillover: Impact of a First 500 MW Plant Successfully Operating on the Success Probabilities for Third Party Projects

	Post-combustion	Pre-combustion	Oxyfuel
Post-combustion	+ 20%	0%	0%
Pre-combustion	0%	+ 17%	0%
Oxyfuel	0%	0%	+ 20%

As Table 5.9 shows, the effect of the first successfully operating, large-scale demonstration project on the cost of subsequent plants is positive or zero. Our experts do not expect a benefit from pre-combustion plants on the competing post-combustion and oxyfuel technology, nor vice versa. Within the post-combustion and oxyfuel process, however, we find technologies that are more related to each other, as both processes rely on conventional thermal power plant technology. Thus, there is an expected decrease in costs across these two technologies.

Table 5.9: Knowledge Spillover: Impact of a First 500 MW Plant Successfully Operating on the Technology Costs for Third Party Projects

	Post-combustion	Pre-combustion	Oxyfuel
Post-combustion	- 8%	0%	- 3%
Pre-combustion	0%	- 8%	0%
Oxyfuel	- 3%	0%	- 8%

Overall, the answers of the experts to our questionnaire present a consistent picture, especially for the estimated changes in capital costs and the success probabilities. Whether this consensus occurred randomly or was based on similar underlying assumptions is not entirely clear, due to modest sample size. Nevertheless, the positive effect of the first successfully operating project on similar projects' costs and success probabilities is notable; and it could justify public funding if this knowledge would be shared among other projects and companies.

5.4.3 Technology State Definitions

We translate the data presented in Sections 5.4.1 and 5.4.2 into the state definitions introduced in Section 5.3.1, distinguishing the projects according to their project size and the level of maturity of the applied CCS technology. We classify the selected projects into three groups, according to the following criteria. The descriptions of the three categories are:

1. The pilot project stage covers projects on a small stage, below 125 MW and a high level of innovative technologies to be tested, as is the case for the 30 MW Schwarze Pumpe oxyfuel project in Germany.

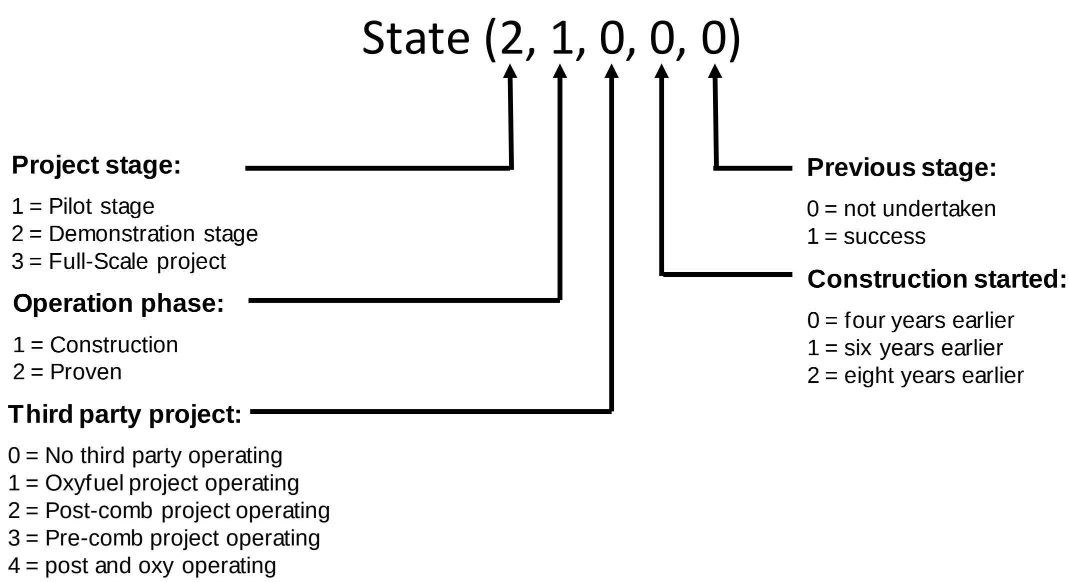
2. The demonstration project stage covers projects which aim at the demonstration of the CCS technology at a larger scale. Examples include the planned 250 MW post-combustion capture projects in Belchatow, Poland and the 250 MW Oxyfuel project in Jänschwalde, Germany.
3. The full-scale project stage aims at the commercial roll-out of the CCS technology. This stage covers projects rated at 500 MW and the retrofitting of pre-combustion technology to capture-ready IGCC plants constructed on in the second stage.

In the model, we assume construction of projects can take four, six or eight years, depending on the project stage. For pilot plants, only four years of construction is allowed. Afterwards, the projects is considered as having reached its technology success level (denoted as "proven"), or it is still in the construction phase. From the perspective of the funding agency, the latter case is considered as failure and no additional funding would be provided. This outcome does not mean that the project is going to be abandoned entirely. Rather, whether the company continues the project or not is a decision internal to the company (and external to the funding agency), and therefore not considered in our model. The same mechanism applies to projects in the demonstration and full-scale stages, with a maximum of six and eight years of construction, respectively.

However, at the demonstration and full-scale stages, the set of funding decisions is extended by defining additional milestones after four years for demonstration and after four and six years for full-scale projects respectively. Thus, we have the following states:

1. The project reached the "proven" state. If so, no need for an additional intervention from the funding agency arises.
2. The project is not finished yet, but a third party project has reached completion of the stage. In this case there remains no need to further fund the unfinished project.
3. No project has successfully completed the current stage. In this case, the funding agency can decide to continue funding.

Figure 5.3: Technology State Definition Example



The stage reached by other (i.e., third party) projects can affect the costs and success probabilities of the projects that would be funded in subsequent periods, based on the assumptions of technology knowledge spillover outlined in Tables 5.8 and 5.9. Unlike the other components of state definition shown in Figure 5.3, the third party state is determined exogenously; that is, it is determined by examining the states of all the other projects. This structure allows for smaller state size, which favorably affects computational complexity and run-time. We present all costs and transition probabilities using the definitions in Figure 5.3—with and without knowledge spillover—for the "Oxy 1" project in Appendix F. Those data are applied to the projects shown in Table 5.10, which are based on announced projects (ZEP, 2008).

Table 5.10: Modeled CCS Projects[18]

Project	Unit Size [MW]			Expected Start of Operation		
	Stage 1	Stage 2	Stage 3	Stage 1	Stage 2	Stage 3
Oxy 1	30	300	1000	2008	2014	2018
Oxy 2	30	320	-	2012	2016	-
Post 1	-	250	-	-	2014	-
Post 2	-	450	900	-	2014	2018
Post 3	-	250	-	-	2016	-
Pre 1	-	450	-	-	2014	-
Pre 2	-	900 IGCC	300 PCC	-	2014	2016
Pre 3	-	1200 IGCC	900 PCC	-	2012	2014

5.5 Funding Scenarios and Numerical Results

In this section, we present the results from the models outlined in Section 5.3 using the data described in Section 5.4 on the eight actual CCS projects shown in Table 5.10.[19] Since the CCS technology for pre-combustion differs significantly from the post-combustion and oxyfuel technologies, we assume there exists a separate funding source for pre-combustion in order to maximize the probability that two CCS

[18] "Oxy" refers to the oxyfuel projects. "Post" refers to the post-combustion projects. "Pre" refers to the pre-combustion projects. "PCC" stands for pre-combustion capture.

[19] While these represent actual projects, the cost, probability and CCS efficiency data are based on the expert interviews. Thus, the project's names are generic so as not to identify any company's name with any project.

technologies will be realized. As a result, there are two sets of optimizations performed: one for the pre-combustion projects and for the post-combustion and oxyfuel projects. Additionally, since the budgets available in future years are not entirely known, we perform sensitivities on both the available budget and how that budget can be allocated. Specifically, we solve for optimal funding strategies for all three models formulated in Section 5.3: fixed-budget, flexible-budget, and budget-optimal allocations.

For both the pre-combustion and the combined post-combustion/oxyfuel cases, we employ the multi-stage competition using an identical approach. The costs for each of the projects depend not only on knowledge spillover, but on the levels at which the project are funded. Due to limited levels of funding which are likely to be manifested, the number of funding levels we model is typically restricted to two (high funding and low funding), in addition to no funding of the project, for each time period. As the following results show for both the pre-combustion and post-combustion/oxyfuel results, the probability of successful completion of a full-scale CCS plant greatly depends on available budget and, moreover, how that budget is allocated during the time periods.

5.5.1 Pre-Combustion Capture Projects without Knowledge Spillover

There are three pre-combustion CCS projects that we considered in our real options model (denoted as “Pre 1,” “Pre 2” and “Pre 3” in Table 5.10). None of the projects is assumed to be able to progress to full-scale without public funding (i.e., the probability of success with no funding is zero). Since the Pre 1 project does not plan to achieve full-scale operation, the objective function's value for funding this project extends only to Pre 1's ability to create knowledge spillover. As shown in Tables 5.8 and 5.9, the spillover can increase the probability of success and decrease the cost on the full-scale stage for other projects. Therefore, it is suboptimal to fund "Pre 1" in cases where the spillover is assumed to be nonexistent.

Table 5.11 shows the potential funding levels for the fixed-budget and flexible-budget cases. The total budget in the bottom row represents the sum of each time period's budget and the total amount available over all time periods for the flexible-budget case (e.g., 3910€ million). For the fixed-budget cases, the budget is usually allocated in equal amounts or in amounts with larger funding for the initial "ramping-up" time periods, as that is when the costs may be greatest (Rubin, *et al.*, 2006). In the pre-combustion cases, the internal gasification plant itself is considered as an innovative technology and represents the major share of the investment. Therefore, higher funding levels are needed in the beginning.

Table 5.11: Budget Cases Modeled for Pre-Combustion Projects[20]

Period	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]
1 (2010)	100	200	300	400	910	910	910
2 (2012)	100	200	300	400	400	910	910
3 (2014)	100	200	300	400	400	400	910
4 (2016)	100	200	300	400	400	400	400
5 (2018)	100	200	300	400	400	400	400
6 (2020)	100	200	300	400	400	400	400
Total Budget	**600**	**1200**	**1800**	**2400**	**2910**	**3420**	**3930**

Due to the flexibility of the real options approach we employ and the relatively high level of success probabilities (especially given the option of delaying the funding

[20] The final decision time period ($T = 6$) maps to the year 2020, but the results are not realized until 2022 ($T + 1$).

of projects), the overall value of the objective function (i.e., the probability that at least one pre-combustion CCS project is performing at full-scale by the end of the sixth time period—that is, 2022) is quite high for the larger budget cases. However, it is important to notice a vastly improved performance of the flexible-budget approach in terms of the objective function. This difference is particularly noticeable for more limited budgets, since the amount available in a given time period could be restricted to only a few possibilities. Of course, as the budgets increase, the managerial benefit of flexible budgets decreases, as there is sufficient funding in most time periods to fund multiple projects. Indeed, as the budget increases, the performance gap between the fixed and flexible budgets narrows, as shown in Table 5.12. With infinite budgets for each time period, the flexible and fixed budgets would have identical objective function values. Table 5.12 provides the results with the total budget provided; again, the breakouts for the budgets for each time period are provided in Table 5.11. Computational complexity and solution run-times for these cases, along with the budget-optimal allocation, are presented in Section 5.6.

Table 5.12: Optimal Objective Function Values (Success Probabilities) for Pre-Combustion Projects with Fixed and Flexible Budgets

Budget [m €]	Fixed-Budget Case	Flexible-Budget Case
0	0	0
600	0	0.405
1200	0	0.781
1800	0	0.903
2400	0.844	0.937
2910	0.902	0.962
3420	0.943	0.968
3930	0.962	0.974

Naturally, the allocation of the fixed budget is very rigid. In some cases, this somewhat arbitrary distribution of the budget among the time periods does not allow for the opportunity for even one project to reach full-scale operations. A more thoughtful allocation—that takes into consideration optimal budgeting for each time period—would be necessary to improve the fixed-budget case. One option is a completely flexible-budget case, where the budget for each time period is only decided at that time. Another possible model is the one presented in Section 5.3.3: it is an optimal, *a priori* budget, which is essentially the optimal, fixed-budget case. It does not permit for the ability to have complete budget flexibility, as one must specify the budgets at the beginning of the multi-stage competition. However, it permits for an optimal allocation of budgets, along with determining how much the real options model's objective function is improved by budget flexibility versus simply optimal upfront budgeting. This difference determines to a large degree the value of managerial flexibility, which is vital to overall R&D success (Tseng, *et al.*, 2005).

When we solve for the budget-optimal model presented in Section 5.3.3, we know the solution cannot be better than the flexible-budget problem. The reason the flexible-budget problem acts as an upper-bound on the budget-optimal problem is that in the flexible-budget problem, the budget allocation for time period t is determined at time period t, when the state of the system, $(s_{1t},...,s_{It})$, is known. The budget-optimal problem must determine an optimal budget before each $(s_{1t},...,s_{It})$ is realized. As we know of no way to incorporate the budget constraint into the lower-level SDP explicitly, we applied heuristics to reduce the number of lower-level problems to be solved in order to avoid complete enumeration when the problem size was too large, while ensuring the budget-optimal solution was within 2% of optimality. Indeed, as shown in Table 5.22, using the cost-coefficient method to determine all possible budget allocations, there would be as many as $O(10^7)$ lower-level SDPs that would need to be solved. However, since the optimal flexible-budget objective function value acts as an upper-bound[21] for the budget-optimal problem, the objective values in Table 5.12 provide sufficient conditions to determine if our heuristic is close to an optimal solution. As Table 5.13 demonstrates, our solutions to the budget-optimal problem produce optimal objective values that are very close to the flexible-budget solutions' optimal objectives shown in Table 5.12. In addition, the first time period funding decision variables are the same for the flexible-budget and budget-optimal cases for the results shown in Tables 5.12 and 5.13. These results point to the importance of having the ability to optimize the available budget in advance (in the absence of completely flexible budgets). Table 5.13 also provides the actual optimal budgets for each case considered in Table 5.12.

[21] The optimal objective function value to any fixed-budget problem serves as a lower-bound, since it represents a feasible budget allocation.

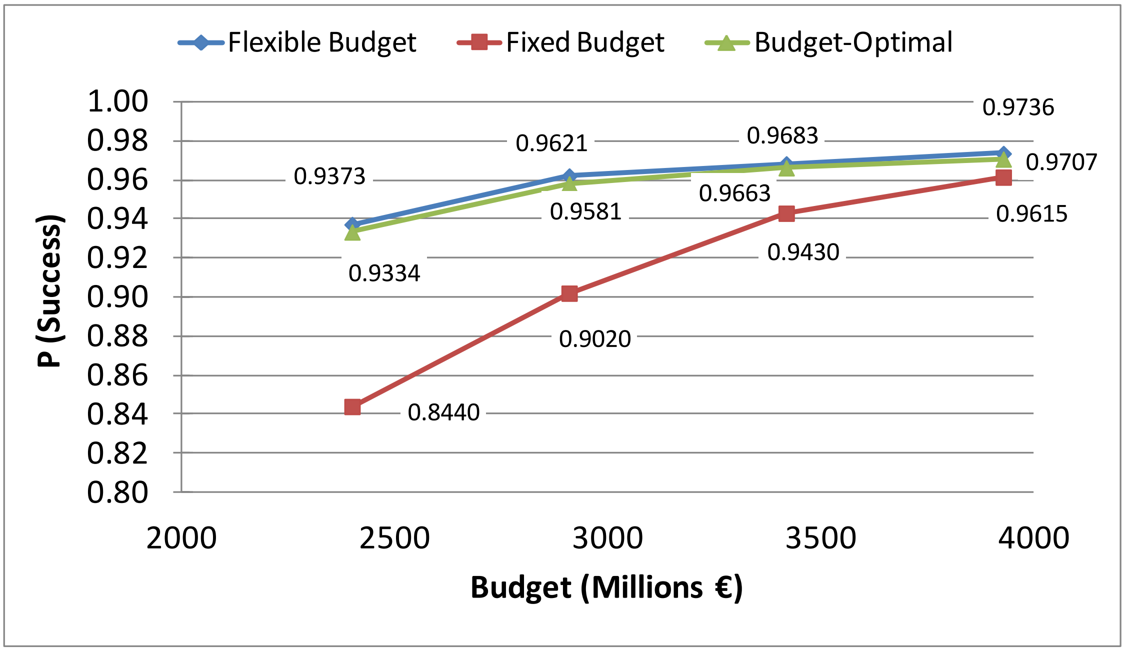
Table 5.13: Optimal Objectives and Budgets for Pre-Combustion Projects under the Budget-Optimal Problem[22]

Period	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]
1	324	324	732	732	732	1056	1056
2	0	0	0	423	423	423	423
3	84	345	324	423	747	747	1140
4	84	345	324	324	423	648	747
5	84	84	324	324	423	423	423
6	24	102	96	174	162	123	141
Total Budget	**600**	**1200**	**1800**	**2400**	**2910**	**3420**	**3930**
Budget-Optimal Objective	**0.405***	**0.779***	**0.889**	**0.933**	**0.958**	**0.966**	**0.971**
Flexible-Budget Objective	**0.405**	**0.781**	**0.903**	**0.937**	**0.962**	**0.968**	**0.974**

The number of lower-level SDPs (i.e., the fixed-budget real options model) solved varied from 710 (for a budget of 600€ million) to approximately 74,000 (for the 2910€ million budget)[23]. Further detail on the run-times and solution heuristic are provided in Section 5.6. Figure 5.4 shows the objective functions for all three budget allocation cases when the budget was 2400€ million or greater. In order to show the very small gap between the budget-optimal and flexible-budget results, the solutions for the smaller budget cases are not shown. Again, the flexible-budget objective serves as an upper bound for the budget-optimal solution. Any fixed-budget solution provides a lower bound to the budget-optimal objective, since any fixed-budget solution is a feasible, but not necessarily optimal, solution to the budget-optimal solution.

[22] *Budgets of 600€ million and 1200€ million were solved to optimality, since all feasible lower-level SDPs were solved. For all other budget cases, the heuristics outlined in Section 5.6 were employed.
[23] Since the heuristic trimmed cost combinations until the number of lower-level SDPs was fewer than about 100,000, there were fewer lower-level problems solved for the 3420€ million and 3930€ million budget cases. Details about the heuristic are provided in Section 5.6.

Figure 5.4: Comparison of Objective Functions for Pre-Combustion Projects under Different Budget Allocation Rules



5.5.2 Post-Combustion and Oxyfuel Capture Projects without Knowledge Spillover

As Table 5.9 notes, there are the three post-combustion and two oxyfuel projects that we model using the real options framework outlined in Section 5.3. Like the pre-combustion projects, not all post-combustion and oxyfuel projects are planning to attempt full-scale completion. As a result, the funding strategies of Oxy 1 and Post 2 are the only projects under consideration for the non-spillover case. Table 5.14 shows the possible budget allocations considered for the funding of these projects.

Table 5.14: Budget Cases Modeled for Post-Combustion and Oxyfuel Projects

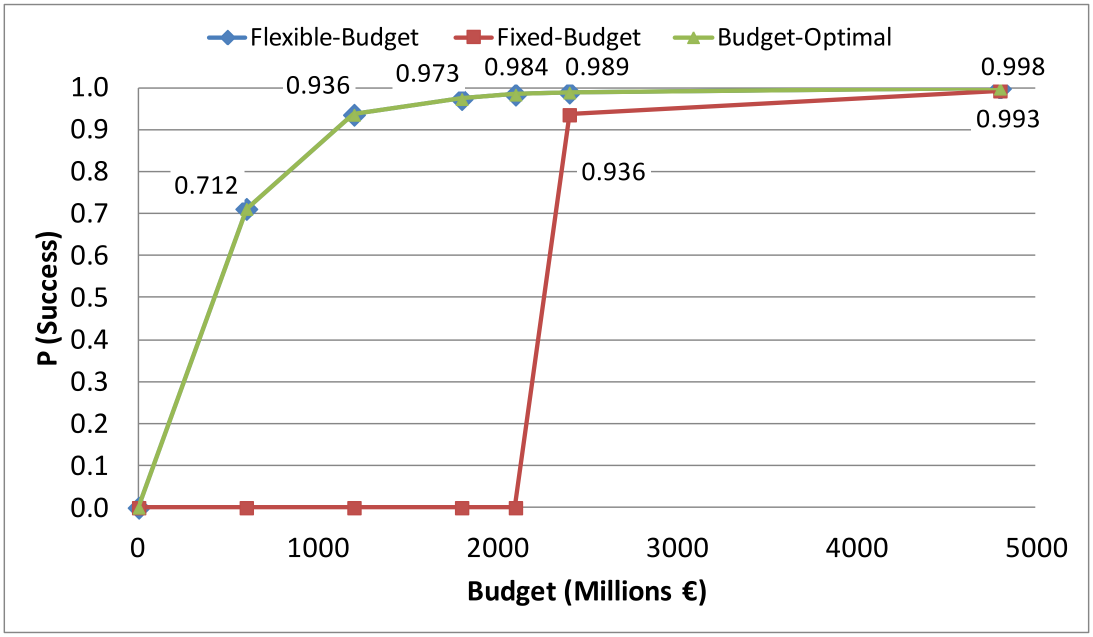
Period	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]
1	100	200	300	350	400	1000
2	100	200	300	350	400	1000
3	100	200	300	350	400	1000
4	100	200	300	350	400	600
5	100	200	300	350	400	600
6	100	200	300	350	400	600
Total Budget	**600**	**1200**	**1800**	**2100**	**2400**	**4800**

Since the post-combustion technology is more mature than pre-combustion, the probability of success for the post-combustion projects tends to be higher—at least for those scenarios with large budgets (Table 5.12 vs. Table 5.15). The costs for the upscaling of the post-combustion and oxyfuel projects, however, are higher than for pre-combustion, even though they are technologically less uncertain. Therefore, as Table 5.15 and Figure 5.5 indicate, the optimal probability of success approaches 1.0, once the budget becomes large enough to have multiple funding options. It is again important to note the significant improvement in the objective function when we have a flexible-budget allocation.

Table 5.15: Optimal Objective Function Values for Post-Combustion and Oxyfuel Projects with Fixed and Flexible Budgets

Budget [m €]	Fixed-Budget Case	Flexible-Budget Case
0	0	0
600	0	0.712
1200	0	0.936
1800	0	0.973
2100	0	0.984
2400	0.936	0.989
4800	0.993	0.998

Figure 5.5: Comparison of Objective Functions for Post-Combustion and Oxyfuel Projects under Fixed and Flexible Budgets



As the approach did for pre-combustion, solving for the budget-optimal allocation for the post-combustion and oxyfuel projects produces an improvement in the objective function over the fixed-budget allocation. In fact, as Table 5.16 indicates, for all of the budget cases we modeled—except the 4800€ million case—the optimal budget produced an objective function equal to the flexible-budget case. For the 4800€ million case, the solutions was within 2×10^{-4}, or 0.01%.

Table 5.16: Optimal Objectives and Budgets for Post-Combustion and Oxyfuel Projects for the Budget-Optimal Problem

Period	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]	Budget [m €]
1	0	0	0	435	0	190
2	150	435	585	0	735	380
3	0	0	0	450	0	890
4	0	380	450	450	450	0
5	435	380	380	380	380	1651
6	15	5	385	385	835	1689
Total Budget	**600**	**1200**	**1800**	**2100**	**2400**	**4800**
Budget-Optimal Objective	**0.712**	**0.936**	**0.973**	**0.984**	**0.989**	**0.998**
Flexible-Budget Objective	**0.712**	**0.936**	**0.973**	**0.984**	**0.989**	**0.998**

Since the budget-optimal problem requires solving a large number of SDPs, we applied heuristic approaches to reduce the number of lower-level problems; these techniques were similar to the ones we employed on the pre-combustion projects. Namely, for smaller total budgets (i.e., 600 and 1200€ million cases) we considered all cost coefficient combinations of funding strategies for each time period (i.e., all combinations of c_{itlS_t} for each t), provided that the sum of those strategies was feasible (i.e., within the budget). For small budgets, this strategy works very well. For larger budgets, we needed to reduce the number of combinations solved with a lower-level SDP, as run-times grow exponentially with the increase in budget. As the flexible-budget case represents an upper bound on the value of the budget-optimal solution, the heuristic obtained an optimal budget allocation in all but the 4800€ million case (where it was within 0.01% of optimality), as its objective function was equal to the flexible-budget result. Again, that property represents a sufficient, but not necessary, condition for the heuristic to have an optimal budget allocation. Since the flexible-budget objective value represents an upper-bound, the budget-optimal objective must be optimal if it is equal to the flexible-budget objective value, and thus represents a sufficient condition. This condition is not necessary, however, since there are cases when an optimal budget-optimal objective (according to Theorem 4.3) is less than the flexible-budget problem's value (e.g., the total budget case of 1200€ million in Table 5.12). The potential gap in the objective values of these two problems results from the lack of temporal flexibility in the budget-optimal case.

As we will show in Section 5.6, run-times were similar to the pre-combustion cases: the lower-level SDP solved more quickly for the post-combustion and oxyfuel projects, but there were more SDPs for each case that need to be solved. Run-times ranged from two seconds (for the 600€ million budget case) to approximately 30 minutes (for the 4800€ million budget case). In short, the heuristics managed to solve all cases to optimality (or near optimality) within a reasonable amount of time.

5.5.3 Effects of Knowledge Spillover

Based on the experts' opinions on knowledge spillover represented in Tables 5.8 and 5.9, it would seem reasonable to expect an increased objective function value (i.e., probability of success) since the costs decrease and transition probabilities increase under such a technical transfer assumption. The results of the model, however, show very little effect in most cases. The knowledge spillover cases in these tables refer to the increased success probabilities and decreased costs assumed in Tables 5.8 and 5.9. Those data show that the probability of a project reaching successful completion increases between 17-20% if another project of the same technology type has already done so. In addition, there is a cost reduction between 3-

8% for those projects. Table 5.17 shows the results for the pre-combustion projects with fixed and flexible budgets; Table 5.18 shows the corresponding results for the post-combustion and oxyfuel projects. Because the results for the budget-optimal allocation were nearly identical to the flexible-budget problem, we tested the effects of knowledge spillover for the fixed-budget and flexible-budget problems only (i.e., we omitted the budget-optimal model runs for knowledge spillover).

Table 5.17: Increase in Objective Function's Value with Knowledge Spillover for Pre-Combustion Projects

Budget [m €]	Fixed-Budget Problems			Flexible-Budget Problems		
	No Spillover (A)	Knowledge Spillover (B)	Increase (B-A)	No Spillover (A)	Knowledge Spillover (B)	Increase (B-A)
0	0	0	0	0	0	0
600	0	0	0	0.405	0.405	0
1200	0	0	0	0.781	0.781	<0.001
1800	0	0	0	0.903	0.903	0.001
2400	0.844	0.844	0	0.937	0.938	0.001
2910	0.902	0.903	<0.001	0.962	0.963	0.001
3420	0.943	0.944	0.001	0.968	0.969	0.001
3930	0.962	0.963	0.001	0.974	0.975	0.001

Table 5.18: Increase in Objective Function's Value with Knowledge Spillover for Post-Combustion and Oxyfuel Projects

Budget [m €]	Fixed-Budget Problems			Flexible-Budget Problems		
	No Spillover (A)	Knowledge Spillover (B)	Increase (B-A)	No Spillover (A)	Knowledge Spillover (B)	Increase (B-A)
0	0	0	0	0	0	0
600	0	0	0	0.712	0.712	0
1200	0	0	0	0.936	0.936	0
1800	0	0	0	0.973	0.974	0.001
2100	0	0.720	0.720	0.984	0.984	0.000
2400	0.936	0.936	<0.001	0.989	0.990	0.001
4800	0.993	0.993	<0.001	0.998	0.999	<0.001

It appears counterintuitive that the addition of knowledge spillover did not materially affect the results in most cases. The one case where the effect was very significant (the 2100€ million fixed budget for post-combustion and oxyfuel) occurs because the reduction in costs from roughly 360€ million to 340€million allows a budget of 350€ in each time period (i.e., 2100€ million total budget) to be sufficient to fund a project to full-scale completion, provided another project was making sufficient progress. However, in most cases, the extra savings is not enough to fund an entirely new project; so the previous funding strategy usually remained unchanged after the spillover assumptions were added, even in the cases with flexible budgets. Secondly, while the probability of fully completing the stage increases once another project has done so (Table 5.8), the improved odds of success can only occur with sufficient means to fund multiple projects concurrently. Indeed, the ability to fund projects agilely under tight budgets is one of the benefits of the real options approach. Finally, the number of projects considered in these two cases was limited, and there is no knowledge transfer between pre-combustion and post-combustion or oxyfuel.

Certainly the results in this chapter do not imply knowledge spillover is insignificant in all multi-stage R&D competitions, or even for those competitions involving the development of CCS technologies. Knowledge spillover would likely be more significant in a case where the number of projects was greater; and the probability of two or more projects making significant progress would be higher. If the knowledge spillover more significantly affected the probabilities—especially in the cases where the probability of adequate technological progression was not particularly high—then the probability of success could be considerably larger for the spillover case versus the no-spillover case.

5.6 Model Complexity, State Size and Solution Run-Times

As we outlined in Chapter 4, there is a limit on the size of the models that were tested using the linearized integer programming approach, even though it has the obvious natural benefit of solving the budget-optimal allocation problem more directly. The models we solved for both the pre-combustion and the post-combustion and oxyfuel projects exceeded this limit. In Table 4.5, we note that a five-project, five-time period problem with each project having five state variables would require linearization constraints (approximately 1 billion) and variables that exceeded computer memory (4 GB). For both sets of CCS projects modeled there were six time periods. Moreover, while some cases we solved had fewer than five projects, all of the projects had more than five states. The number of projects and the state variables for each project are provided in Table 5.19 for both the pre-combustion and the post-combustion and oxyfuel cases. The state variables map to the components shown in Figure 5.3. The number of states differs among projects since not all projects complete all stages. The third party state is determined exogenously (i.e., from the states of the other projects) and, therefore, does not increase the state space. The non-spillover case had fewer projects, as noted in the table.

Table 5.19: Number of States for Pre-Combustion and Post-Combustion/Oxyfuel Projects

Project	States	Spillover Case Only
Oxy 1	14	
Oxy 2	7	X
Post 1	6	X
Post 2	14	
Post 3	6	X

Project	States	Spillover Case Only
Pre 1	9	
Pre 2	10	
Pre 3	6	X

Thus, in the case of the post-combustion and oxyfuel projects, the number of projects, state variables and time periods equaled or exceeded the number in the five-project, five-time period, five-state variable problem that was computationally too complex for the IP formulation. While the non-spillover cases, especially the pre-combustion one, were perhaps solvable on a 4 GB machine using an IP approach ((4.3) for the fixed-budget problem or (4.5) for the budget-optimal problem), they would not be scalable if more projects were later added. Of course, the run-times for the SDP are quite manageable, provided we did not solve for the budget-optimal allocation problem—something that the IP may handle more readily for smaller problems. However, since the IP approach was not feasible in this case, we needed to apply the approach from Chapter 4 to these cases.

5.6.1 Complexity of Fixed-Budget, Flexible-Budget, and Budget-Optimal Problems

For the spillover cases with a fixed budget, the total number of states at each time period was $14 \cdot 7 \cdot 14 \cdot 6 \cdot 6 \approx 50{,}000$ for post-combustion and oxyfuel and $9 \cdot 10 \cdot 6 = 540$ for pre-combustion. However, for the case of flexible budgets, the number of states increases by the size of the budget increment. Thus, for a budget as high as 4000€ million, the number of state variables per time period was as high as $O(10^7)$ for the post-combustion and oxyfuel cases with knowledge spillover. For the non-spillover cases with fixed budgets, the state size of these problems is very modest: $O(10^2)$. The addition of the flexible budget, even with a very small budget increment (1€ million), increases the state size to $O(10^5)$, which is well within the means of a 4 GB laptop with four 1.73 GHz processors. Table 5.20 shows the number of states (which equals the number of decision variables in the SDP) and run-times for the fixed-budget problems. Table 5.21 provides the number of states and run-times for the flexible-budget problems solved in this chapter. For the flexible-budget case, the run-times and state variables are shown for the largest budgets considered (i.e., 3930€ million for the pre-combustion projects and 4800€ million for the post-combustion and oxyfuel projects), as those cases have the longest run-times and greatest number of states. Because of the Markovian property of the SDP, only the current and the next time periods' state variables need to be stored in a memory at any one time. Nevertheless, due to the large number of state variables for even one time period, the budget increment for the post-combustion and oxyfuel projects with knowledge spillover must be 10€ million; with a smaller budget increment of 1€ million, 4 GB of RAM was insufficient to solve this problem.

Table 5.20: Number of State Variables and Run-Times for the Fixed-Budget CCS Problems

CCS Projects	With Spillover?	State Variables (Each Period)	State Variables (Total)	Run-Time (CPU Sec.)
Pre-Combustion	**No**	90	540	<1
Post-Combustion & Oxyfuel	**No**	196	1,176	<1
Pre-Combustion	**Yes**	540	3,240	<1
Post-Combustion & Oxyfuel	**Yes**	49,392	296,352	12

Table 5.21: Number of State Variables and Run-Times for the Flexible-Budget CCS Problems

CCS Projects	With Spillover?	Budget Increment (€ Million)	State Variables (Each Period)	State Variables (Total)	Run-Time (CPU Sec.)
Pre-Combustion	No	1	353,700	2,122,200	8
Post-Combustion & Oxyfuel	No	1	940,800	5,644,800	46
Pre-Combustion	Yes	1	2,122,200	12,733,200	63
Post-Combustion & Oxyfuel	Yes	10	23,708,160	142,248,960	974

Since the budget-optimal problem is obtained by solving a set of lower-level fixed-budget problems, the number of decision variables for the budget-optimal problem is identical to the fixed-budget problem. The increase in run-time and complexity for the budget-optimal problem is strictly due to the large *number* of lower-level, fixed-budget SDPs that must be solved. Those results are presented in the Section 5.6.2.

It is worth noting that combining all eight projects into one real options model with flexible budgets and a budget increment of 10€ million would have $14 \cdot 7 \cdot 14 \cdot 6 \cdot 6 \cdot 9 \cdot 10 \cdot 6 \cdot 900 \approx O\left(10^{10}\right)$ state variables per time period, far exceeding a computer's 4GB of memory.[24] Since pre-combustion technology differs significantly, separating the funding budgets is suitable. However, for a larger set of projects, the solution time will eventually become unmanageable as the states suffer from the curse of dimensionality (Puterman, 1994).

[24] At four bytes for each single-point floating variable, the memory requirements to store one time period's state variables for this problem would be at least 100 GB.

Nevertheless, the size of the budget-optimal allocation problem requires a very large number of lower-level SDPs to be solved. In Chapter 4, we outlined two approaches: solving for all budget combinations using a discrete budget increment, or solving for all combinations of the cost coefficients. The former approach was not viable. Since costs were in terms of whole millions of Euros, and the greatest common factor was typically 1€ million, for even the smallest budget of 600€ million, according to (4.8) there would be, $\frac{(M+T-1)!}{M!(T-1)!} = \frac{605!}{(600)!5!} \approx 600{,}000{,}000{,}000$ lower-level problems.

As we show, obtaining budgets from all cost coefficient combinations, while fewer than the number of discrete budget combinations, does not necessarily produce a sufficiently small number of lower-level SDPs that need to be solved. Since each project typically had multiple funding levels at each time period, there were a large number of cost combinations, each requiring the solving of the lower-level SDP. For post-combustion, Table 5.22 shows the possible number of costs each project could have in that time period. While some of these costs only can occur when the project is in a certain state, what state a project occupies at a given time period cannot be determined *a priori*. Additionally, each project may not necessarily be funded in that time period, which represents a cost combination (0€, for that project).

Table 5.22: Number of Budget Possibilities Using the Cost-Coefficient Method for the Full-Scale CCS Projects Based on Each Project's Funding Levels

Project	Time Period 1	2	3	4	5		Total
Pre 1	3	3	5	5	5	=	4,134,375
Pre 2	3	7	7	5	5		
Oxy 1	5	5	9	9	7	=	8,859,375
Post 2	1	5	5	5	5		

The total number of budget possibilities is simply the product of all possible cost combinations for each of the projects over each of the time periods. As we noted in Chapter 4, it is not necessary to consider the final time period's cost combinations, since the final time period's budget, B_6, is simply $B_6 = \max\left(B - (B_1 + B_2 + B_3 + B_4 + B_5), 0\right)$. Even though the lower-level SDP solves very quickly (less than 10^{-1} seconds), solving such a large number of cases requires heuristics to obtain a manageable number of lower-level problems, since a full consideration of all cases would take roughly 100,000 seconds, or around one full day. The next section describes the methods used and the run-times for the budget-optimal problems for both sets of CCS technologies.

5.6.2 Approaches to Improving Solution Speeds for the Budget-Optimal Problem

As mentioned in Chapter 4, an obvious way to reduce the number of lower-level SDPs necessary is to remove from consideration all budget combinations that are infeasible because the allocation exceeds to the total budget. This method is particularly useful for cases where the total budget is a small amount compared to the sum of the average cost of each project over each time period. Indeed, that was the only technique we applied for the cases where the total budgets were either 600€ million or 1200€ million. The resulting number of SDPs for these cases was relatively small; and the run-times were short.

For cases where the total budget was larger, the number of feasible combinations grows quickly. Based on the desire to have relatively manageable run-times, it was necessary to develop a set of rules to obtain verifiably close-to-optimal solutions. The budget-increment approach (where optimality is guaranteed by Theorem 4.2) or the cost-coefficient approach (which is optimal by Theorem 4.3) requires solving for all lower-level SDPs specified by the theorems. However, we can solve for a subset of the lower-level problems using the cost-coefficient approach, while checking the optimality gap by calculating the upper bound provided by the flexible-budget solution's objective. We therefore outline the following heuristic approach for solving budget-optimal problems as an SDP, which was applied to several of the budget-optimal problems in this chapter. The number of lower-level SDPs and their associated run-times are presented in Tables 5.23 and 5.24.

Heuristic 5.1:

- ***Step 0***: Solve the flexible-budget SDP for the identical problem (i.e., for the same total budget). This solution acts as an upper bound for a budget-optimal objective. Solve the fixed-budget problem to obtain a lower bound and approximate run-time for this lower-level SDP.[25]

- ***Step 1***: A) Compute the number of cost coefficient combinations that produce feasible budget allocations (i.e., the values of x_{it} such that $\sum_{t=1}^{T-1}\sum_{i=1}^{N} c_{it}x_{it} \leq B$).

  B) From the average run-time of the lower-level SDP (calculated in Step 0), determine if expected run-time is sufficiently fast. If so, solve for an optimal budget allocation and stop; solution will be optimal via from Theorem 4.3. If not, continue to Step 2.

- ***Step 2***: Reduce the number of cost combinations in the following manner: for each time period considered in isolation, calculate all possible budget allocations. Eliminate all cost coefficients, c_{it}, that are a combination of any other coefficients in that time period. For example, if there are five cost coefficients at time period t with $(c_{1t}, c_{2t}, c_{3t}, c_{4t}, c_{5t}) = (40, 120, 220, 340, 500)$, then eliminate $c_{4t} = 340$, since $c_{2t} + c_{3t} = c_{4t} = 340$. Calculate the reduced number of lower-level SDPs and expected run-time. If sufficiently small, solve for all budget-optimal allocations and go to Step 4; otherwise, go to Step 3.

- ***Step 3***: Eliminate the cost coefficient that is closest (but smaller) to another coefficient or combination of coefficients. For example, if the four cost coefficients at time t are $(c_{1t}, c_{2t}, c_{3t}, c_{4t}) = (75, 130, 200, 280)$, then eliminate

[25] The average run-time for a single fixed-budget SDP can be negligible (much less than one second). Thus, for the CCS projects modeled in this chapter, our estimate was calculated in terms of fixed-budget SDPs solved per CPU second.

$c_{3t} = 200$, since $c_{1t} + c_{2t} = 75 + 130 = 205$. Calculate the reduced number of lower-level SDPs and the corresponding expected run-time. If sufficiently small, go to Step 4; otherwise, repeat Step 3.[26]

- ***Step 4***: Compare the result to the flexible-budget result. If budget-optimal solution is not close to the flexible-budget result, return to Step 1 and increase the run-time threshold. For an infinite run-time threshold, a solution will be optimal, since the heuristic will only perform Step 1, which is optimal by Theorem 4.3. ■

Fortunately, as Tables 5.13 and 5.16 show, the budget-optimal objectives were sufficiently close to the flexible-budget values that the optimality gaps, if any, were small. For the pre-combustion cases, the largest optimality gap for any budget case (the 1800€ million budget case) was 1.5%; for post-combustion and oxyfuel it was a mere 0.01% (for the 4800€ million case). Since there is no known way of incorporating the budget allocation constraint into the SDP, the solution to the flexible-budget problem is likely the only certain optimality upper bound. Lower bounds are obtained by solving for the fixed-budget problem with largest objective function value at that point. Table 5.23 provides the run-times number of lower-level SDPs solved for every budget for pre-combustion projects; Table 5.24 provides those results for the post-combustion and oxyfuel projects. Appendix G lists the cost coefficients that exist and the ones that were eliminated for the larger budget cases for the post-combustion and oxyfuel projects.

[26] Note that in Step 2 and Step 3, the elimination of a cost coefficient at any time period reduces the number of lower-level SDPs by one-half, since the total number of combinations for N cost coefficients at each time period is $2^N = \left(2^{N-1}\right)2$.

Table 5.23: Run-Times and Number of Lower-Level SDPs Solved for the Pre-Combustion Projects Budget-Optimal Problems

Total Budget	600 [m €]	1200 [m €]	1800 [m €]	2400 [m €]	2910 [m €]	3420 [m €]	3930 [m €]
Run-Time (sec)	22	514	201	851	1,873	386	558
Lower-Level SDPs	710	24,376	8,708	38,608	73,922	16,815	20,518

Table 5.24: Run-Times and Number of Lower-Level SDPs Solved for the Post-Combustion and Oxyfuel Projects Budget-Optimal Problems

Total Budget	600 [m €]	1200 [m €]	1800 [m €]	2100 [m €]	2400 [m €]	4800 [m €]
Run-Time (sec)	2	19	205	119	157	1,828
Lower-Level SDPs	156	5,947	75,813	40,332	55,471	607,474

5.7 Policy Implications of Results

In this chapter, we combine subject matter expert interviews with a real options approach to find risk-minimizing public funding strategies for CCS. The results of the expert elicitations show that testing the innovative CCS technologies for the first time should best be carried out in small pilot plants. The knowledge gained in such projects at lower total costs (yet higher costs per kW installed) benefits projects at the demonstration and full-scale stage. The post-combustion capture technology, which is generally understood as being closest to the market, already shows the highest success probabilities at the lowest cost. This result is consistent with the literature, as well as the actual EU CCS funding strategy (Tzimas, 2009; RECCS, 2010). Under the EEPR, it was decided to co-finance four post-combustion projects, one pre-combustion and one oxyfuel project (European Commission, 2009).

The experts' answers regarding the cross-technology spillover effect show interesting, albeit divergent, opinions. While some experts expect some positive effects on the costs of subsequent projects between post-combustion and oxyfuel capture, other experts expect no effect at all. Those experts even outline a sharp decrease in the success probability if one technology is successfully introduced on the market. This possible outcome is explained by the necessity for only one proven capture technology in the power sector and the resulting stop in public funding for the substitutable technology.

In addition to the value of the real options approach for considering a set of CCS projects, the solutions we present point to the importance of allocating budgets in a way that best utilizes the projects' ability to make use of them. The results for the pre-combustion projects (Table 5.16) seem to indicate that allocating more funds in the early time periods produces an optimal budget allocation, while the post-combustion and oxyfuel projects (Table 5.13) have an optimal budget allocation that somewhat favors more funding in later time periods. In general, it is difficult to predict the appropriate allocation without explicitly solving the budget-optimal problem. While it is likely that the costs for the projects could accommodate several different budget allocations, it is important to identify how success probabilities change with funding levels, as the outcomes are greatly dependent upon it. Any study whose results affect actual funding decisions should consider a more detailed technology success probability distribution for each project, in the hopes of allocating a budget, and then funding projects, in a way that is fully optimal.

Nevertheless, there arises a natural conflict between the risk-minimizing CCS portfolio strategy from the perspective of the funding agency modeled in this chapter and the necessity for a credible funding strategy from the perspective of the firms: the

CCS technology is a complex compound of technologies which faces significant barriers towards its large-scale implementation. Those barriers consist not only in the form of technological hurdles, but also in the uncertain legal and institutional framework. Real options modeling may show that it is optimal to cancel ongoing or planned projects in case of underperformance or in the event of third-party projects reaching success. This strategy necessarily adds another dimension of uncertainty to the project planning process of the firms.

Another conflict arises from the regional distribution of potential projects. According to the European Commission, a maximum of one project may receive public funding in each member state. While this allotment appears to be a fair procedure at a policy level, it may result in an inefficient project portfolio. This inefficiency in portfolio selection may be further enforced due to individual delays in member state projects—for example, due to delays in the planning and approval process. Under the EEPR funding scheme, the European Commission limited this risk by only granting funding to projects which can prove by the end of 2011 that the technical, legal and regulatory framework allows the demonstration of the full carbon capture, transport and storage chain (European Commission, 2009). Finally, our results show that a budget-optimal allocation leads to objective function values very close to the fully flexibly budget allocation. The advantage of the budget-optimal allocation over the flexible allocation is an increase in the credibility of the funding scheme from the perspective of the firms undertaking such high risk projects.

Chapter 6: Conclusions and Extensions

The main chapters of this dissertation provide four fundamental categories of results. First, the value of the real options models demonstrates the benefits over qualitative, *ad hoc* methods commonly used, especially in government-sponsored, R&D intensive procurement (Ceylan and Ford, 2002). The methods applied in these chapters not only highlight the value of the delayed decisions common to real options valuation (for example, Ward, *et al*., 1995), but also establishes the importance of budget flexibility and an optimal budget allocation. The second major contribution is the development of equivalent methods for modeling such problems: stochastic dynamic programming and integer programming, which provides a roadmap for expressing similar Markov decision processes as integer programs. The third contribution is solving the otherwise, two-level, budget-optimal SDP as a one-level IP. Each approach has advantages with respect to the other; we provide specific guidelines where each method is likely the preferred technique. In general, the IP is more efficient when solving a large number of projects (or cost coefficients) while the SDP can be efficiently utilized using heuristics and solving for the flexible-budget problem as an upper bound. The fourth contribution is the application of the model to an actual set of technologically uncertain, R&D intensive projects being undertaken that are subject to public funding: pre-combustion, post-combustion and oxyfuel carbon capture and storage projects in the European Union. In addition to providing value for those concerned with the actual funding strategies of these CCS applications, the research we conducted provides illustrative processes for data collections, expert elicitation, and state definition techniques for risky R&D initiatives unrelated to clean energy.

6.1 Utility and Limitations of the Real Options Methods

Though government acquisition managers recognize they are using real options approaches, through multi-project, multi-stage competitions, to mitigate the technical risks associated with R&D intensive acquisition programs, there have been few analytical frameworks available for their use. This dissertation develops a general formulation of such competitions that may be readily solved through stochastic dynamic programming or integer programming to determine the optimal portfolio of project options to purchase and exercise.

The real options model approaches utilized in this dissertation provide a set of quantitative measures for appropriate levels of funding. The models presented, such as the flexible budget example in Section 2.4.2, illustrate the value in the wait-and-see approach fundamental to real options theory. On the other hand, these methods can often provide counter-intuitive results, such as funding a smaller set of projects initially, as opposed to spreading the funding over more projects where the diluted funds produce several unsuccessful projects more likely than fewer well-funded projects. The numerical example in Section 2.4.1 illustrates how this strategy could be optimal. Such conclusions are important to identify and analyze.

An additional benefit of the real options approaches presented in this work is that it can explicitly recognize the effect budgeting has on the solutions obtained from the models. Based on the numerical examples presented in Chapters 2 and 3, along with the actual CCS projects in Chapter 5, we find that allowing budget flexibility and multiple finding levels may increase the probability of program success. While the specific project selection and the budget are typically considered as separate problems, the ability to optimize the budget greatly increases our likelihood of success, which thereby decreases the chances of cost overruns (Tseng, *et al*., 2005). When the budgets must be determined in advance, the extent to which the lower-level problem (i.e., the real options funding strategy) can inform the upper-level problem (i.e., the budget allocation) is vitally important to a successful acquisition outcome.

Nevertheless, significant hurdles hinder the wide-spread adoption of such methods to R&D intensive acquisitions. Perhaps the most obvious difficulty is the ability for the acquisition managers to obtain well-defined probability distributions for the technical progress of the projects they evaluate. There are two mitigating strategies that can be employed when using the models presented here.

First, due to the temporal nature of these real options problems, any subsequent information that provides more reliable probability, cost or budget estimates can be incorporated into the model and solved to optimality from that point forward. While any potential modification of the data will not guarantee optimality for the previous time periods, the ability to be optimal from that intermediate time period forward provides a risk-mitigation strategy for employing these real options models. This flexibility extends not only to funding decisions, but potentially future budget allocations as well.

Secondly, robust optimization techniques could be utilized to obtain solutions that are within some range of optimality and feasibility. Since the budgets available and (to a lesser extent) project costs are often more certain than the estimates of the probability distributions, it is vital to find solutions that are "solution robust," which remain near-optimal for all scenarios of the input data (Mulvey, *et al*., 1995). Specifically, if the greatest uncertainly is in the probabilities of success (i.e., the objective function coefficients), a robust optimization could potentially make use of recent methods, such as Bertsimas and Sim (2003), who provide a method for obtaining for robust solutions on IPs containing n binary variables that solves in at most $n+1$ instances of the original IP.

When using the SDP approach, one can make assumptions on the uncertainties of the transition probabilities (Bertsimas, *et al*., 2011). For example, Nilim and El Ghaoui (2005) formulate "robust" SDPs with explicit bounds on optimality that can be achieved with practically no extra computing cost beyond the original SDP. Moreover, they derive a similar bound for an uncertainty set with a finite number of possible values for the transition matrices (an assumption that several problems in this dissertation could likely assume). Such methods could be employed to help provide robust solutions for the multi-stage real options problems discussed in this dissertation.

A second major obstacle towards implementation is that our models assume to some extent the flexibility to down-select projects based on the performance of the portfolio or a specific project. In reality, such large R&D projects with public funding are not so agile, especially when the down-select is due to a competing project doing well, rather than specific poor performance of the project being reduced. Nevertheless, it can be handled in many cases, though the risk of reduced or discontinued funding due to these strategies would likely increase the upfront contractual costs (Ceylan and Ford, 2002). For the purposes of our examples, we assume full transparency of the strategies employed by the real options model user (e.g., the public funding agency). For the CCS projects, we assumed internal resources were included in the developments of the plants, but were not sufficient to continue the R&D efforts without continued public support.

A third obstacle is that the future budgets (whether fixed or flexible) may be uncertain, especially in the later years. Fortunately, the SDP approach can handle this uncertainty by calculating a set of solutions that optimizes the objective function. If the budget for each time period can be described by a probability distribution, then the Markov process can be incorporated as part of the solution with these budget uncertainties. However, the computational complexity of the model will increase, as the solution must optimize over all possible future budgets. Incorporating this realistic scenario into the multi-stage competition in this dissertation could prove to be an insightful extension to public-sector managers facing fiscal uncertainty.

6.2 Formulation and Modeling Approaches

Practitioners can employ the stochastic dynamic programming or integer programming formulations in this dissertation to determine an optimal combination of projects to fund and how much funding each project should receive within an R&D portfolio to achieve a given objective. In general terms, the SDP approach's advantages over the IP formulation are: it can handle larger sized problems; it has shorter run-times for the fixed-budget problem; and it can readily incorporate a flexible-budget state variable. Additionally, the memoryless property that the SDP utilizes ensures that the growth in the computational complexity of the model will grow only linearly with the number of time periods. This path-independent property does not hold if the IP uses linearization constraints. While solving for the budget-optimal problem requires solving a large number of lower-level SDPs, each lower-level problem can be solved independently, which can take advantage of parallel processing in a more obvious manner than the IP.

An advantage of the IP approach over the more traditional SDP approach is the ability to identify the optimal *a priori* budget for each time period given a fixed budget for the overall project portfolio. Like the SDP approach, the IP model can be used iteratively after each decision point has been reached to fine-tune the optimal strategy conditional upon each funded project's progress up to that point. Also, modeling this type of real options problem as an IP more readily facilitates "what-if" analysis by easily incorporating additional side constraints to the problem (e.g., the budget for a given time period must be within a certain range; funding of one project is conditional on the funding decision of another at a certain time period). If a practitioner anticipates building models with many of these side constraints, the IP approach could be preferred, since the set-up time for a completely new SDP may be greater than modifying an existing IP. The set-up time for an IP, however, can be quite large when the number of time periods is significant, as even semi-automated construction of the linearization variables can be cumbersome.

While it is certainly true that the LINGO and XPRESS-MP solvers typically can only solve modestly-sized problems (due to the large number of linearization constraints and continuous variables), continual improvements in commercial IP solvers beyond merely faster processor time does suggest larger problems may be yet be tractable in the future. Moreover, techniques for obtaining more concise 0-1 linearizations are a field of active research. For example, Adams and Forrester (2005) propose a method for linearizing a mixed 0-1 cubic program with n variables in $\frac{n(n+1)}{2}$ additional continuous variables and $\frac{n(n+1)}{2}$ inequality constraints. While the product of binary variables is often greater than three in the IP version of the real options model, (3.4) can be written so that only the objective function is nonlinear. Such an implementation could potentially reduce the number of linearization constraints and increase the size of problems an IP model could solve.

Additionally, the problems solved in this dissertation were done so on a typical laptop computer, not on a cluster of super computers. While faster computers would no doubt solve the SDP formulations more quickly, they would also provide a larger space for which these real options problems can be solved using IP approaches, especially if those computers could utilize a diverse set of commercial IP solvers.

Other solution techniques can be employed to reduce the excessive IP run-times or solve larger-sized problems. In Chapter 4, we solved the nonlinear version of the integer program with lackluster results; however, other, more specialized, nonlinear solvers may perform better. Local solutions were obtained quickly, but those solutions were often not globally optimal. Relaxing the binary constraints produced unusable solutions, though it could be utilized as part of a branching and bounding approach. Taking logarithms of the binary variables eliminates the multiplication of variables, but results in nonlinear constraints. With further research, some combination of these approaches could expand the region over which the IP formulation would be the preferred solution approach for the optimal *a priori* budget allocation real options problem.

In Chapter 5, we solved the formulated real options problem for a detailed set of pre-combustion, post-combustion and oxyfuel CCS projects. While the run-times for the budget-optimal problem could be lengthy, the solution heuristics that were implemented produced optimal (or near-optimal) solutions with reasonable run-times. However, the successful implementation of Heuristic 5.1 for the budget-optimal problem does not imply such models can be applied to larger sets of projects with little concern for the computational complexity and state size of the problem. Due to the technologically unique aspects of pre-combustion, we evaluated those funding strategies separately and from a different source of funding. This separation approach allows the models' complexity to be greatly reduced, since SDPs state size suffers from the "curse of dimensionality" when two sets of projects are combined. A reason for the increased number of states is the partial path-dependency for some CCS projects. These dependencies occur because some projects' transitional probabilities are affected by: i) how long has a project been in the "construction" phase and ii) whether the project completed the previous phase. However, a *complete* path-dependency for all projects (e.g., knowing the mix of projects for each time period) would destroy the Markov property and would require total enumeration.

Even without path-dependencies, one can imagine that the addition of a few more projects, states or funding levels would make the problem intractable to solve using the SDP or IP approaches presented. At that point, some form of approximate dynamic programming or other heuristic approach could be employed to get meaningful solutions (Powell, 2009). A specific example of a solution technique is to assign a value to only a subset of the possible states (e.g., the value of being in the state where any project has achieved state S are all identical, regardless of which states the other projects occupy). In effect, we would have a composite state that represents a set of states with assumed homogenous properties. Fortunately, for the projects evaluated in Chapter 5, such approximation or simplification techniques were not necessary.

6.3 Extending the Real Options Models to Other Problem Classes

An obvious extension of the real options model would be to other technologically uncertain, research-intensive, and publically funded (either partially or wholly) projects whose outcome is difficult or impossible to monetize using traditional real options approaches. As we noted in Chapter 5, the economic value of the CCS projects extends beyond those specific plants' ability to capture and store CO_2, as the technology acquired will transfer to a larger set of operators. Moreover, while even this transfer knowledge can be valued, the funding for these initiatives is often divorced from strictly monetary considerations (Post, *et al*., 2004). Thus, objective functions such as the maximization of project success employed in our models are appropriate for these types of initiatives, as traditional methods (e.g., NPV) often do not capture the long-range value of such R&D programs (Vonortas and Hertzfeld, 1998).

While the real options framework presented in this dissertation is suitable for many practical applications beyond CCS or other energy applications, it also serves as a foundation for further research, in the hopes of addressing larger classes of problems. One example would be solving for optimal funding decisions while considering entirely different types of systems (e.g., a centralized or de-centralized air traffic control system). The techniques can also be extended to consider the optimal number of projects to fund from a pool of identical projects, where the transitional probability matrices are more complex than the ones discussed here, but are difficult to distinguish among the separate projects.

The problems addressed in this dissertation typically have one decision-maker (e.g., the program manager of a federal agency). For a broader class of problems, there could be a set of decision-makers with multiple objectives (e.g., a National Science Foundation panel). In cases where those objectives conflict, relevant game theory must be utilized to appropriately address the outcomes (Cottle, *et al.*, 1992).

Of course, multiple objectives could arise from a single decision-maker with competing goals (Cohon, 2003). For example, the Centers for Medicare and Medicaid Services manage a portfolio of research projects with the joint goals of increasing the quality of care while reducing the overall cost of service delivery—goals which are often in opposition in the healthcare arena. Developing models that explicitly accommodate such complex objectives would be valuable for applying the techniques developed in the dissertation in these contexts. Possible solutions may involve applying weights to each objective, which would effectively reduce the objective function to a single quantitative metric that can be solved through the methods presented. More realistic applications would likely include non-constant weights for each of the objectives for different combinations of outcomes to adequately reflect the convex relationship that may characterize the tradeoff among different objectives. This approach may require multi-objective optimization techniques that explicitly incorporate the convex relationship among different objectives.

Appendix A: Transition Probabilities for Numerical Examples in Chapter 2

Table A1: Model 1's First Stage Transition Probabilities

First Stage Outcomes	TRL	Prob	First Stage Outcomes	TRL	Prob
Project 1	4	0.20	**Project 3**	4	0.10
	5	0.30		5	0.10
	6	0.40		6	0.50
	7	0.10		7	0.20
	8	0.00		8	0.10
Project 2	4	0.10	**Project 4**	4	0.30
	5	0.20		5	0.10
	6	0.50		6	0.40
	7	0.20		7	0.15
	8	0.00		8	0.05

Table A2: Model 1's Second Stage Transition Probabilities

Second Stage Outcomes	TRL	Previous TRL	Prob	Second Stage Outcomes	TRL	Previous TRL	Prob
Project 1	4	4	0.30	**Project 3**	4	4	0.20
	5	4	0.40		5	4	0.40
	6	4	0.20		6	4	0.20
	7	4	0.10		7	4	0.10
	5	5	0.40		8	4	0.10
	6	5	0.35		5	5	0.40
	7	5	0.25		6	5	0.35
	6	6	0.30		7	5	0.15
	7	6	0.50		8	5	0.10
	8	6	0.20		6	6	0.30
	7	7	0.40		7	6	0.40
	8	7	0.60		8	6	0.30
	8	8	1.00		7	7	0.30
					8	7	0.70
Project 2	4	4	0.10		8	8	1.00
	5	4	0.30				
	6	4	0.40	**Project 4**	4	4	0.40
	7	4	0.20		5	4	0.30
	5	5	0.30		6	4	0.20
	6	5	0.20		7	4	0.10
	7	5	0.50		5	5	0.50
	6	6	0.20		6	5	0.30
	7	6	0.70		7	5	0.10
	8	6	0.10		8	5	0.10

7	7	0.35	6	6	0.40
8	7	0.65	7	6	0.30
8	8	1.00	8	6	0.30
			7	7	0.50
			8	7	0.50
			8	8	1.00

Table A3: Model 2's First Stage Transition Probabilities

LOW				MIDDLE				HIGH		
Project	**TRL**	**Prob**		**Project**	**TRL**	**Prob**		**Project**	**TRL**	**Prob**
Project 1	4	0.30		Project 1	4	0.20		Project 1	4	0.20
	5	0.20			5	0.30			5	0.20
	6	0.45			6	0.40			6	0.30
	7	0.05			7	0.10			7	0.20
	8	0.00			8	0.00			8	0.10
Project 2	4	0.10		Project 2	4	0.10		Project 2	4	0.10
	5	0.20			5	0.20			5	0.20
	6	0.50			6	0.50			6	0.40
	7	0.20			7	0.20			7	0.25
	8	0.00			8	0.00			8	0.05
Project 3	4	0.20		Project 3	4	0.10		Project 3	4	0.00
	5	0.30			5	0.10			5	0.10
	6	0.30			6	0.50			6	0.40
	7	0.10			7	0.20			7	0.30
	8	0.10			8	0.10			8	0.20
Project 4	4	0.30		Project 4	4	0.30		Project 4	4	0.20
	5	0.20			5	0.10			5	0.20
	6	0.30			6	0.40			6	0.40
	7	0.20			7	0.15			7	0.15
	8	0.00			8	0.05			8	0.05

Table A4: Model 2's Second Stage Transition Probabilities

LOW				MIDDLE				HIGH			
Project	**Stage 2 TRL**	**Stage 1 TRL**	**Prob**	**Project**	**Stage 2 TRL**	**Stage 1 TRL**	**Prob**	**Project**	**Stage 2 TRL**	**Stage 1 TRL**	**Prob**
Project 1	4	4	0.40	Project 1	4	4	0.30	Project 1	4	4	0.20
	5	4	0.30		5	4	0.40		5	4	0.30
	6	4	0.20		6	4	0.20		6	4	0.30
	7	4	0.10		7	4	0.10		7	4	0.20
	8	4	0.00		8	4	0.00		8	4	0.00
	5	5	0.50		5	5	0.40		5	5	0.40
	6	5	0.40		6	5	0.35		6	5	0.30
	7	5	0.10		7	5	0.25		7	5	0.20

	8	5	0.00		8	5	0.00		8	5	0.10
	6	6	0.40		6	6	0.30		6	6	0.25
	7	6	0.50		7	6	0.50		7	6	0.50
	8	6	0.10		8	6	0.20		8	6	0.25
	7	7	0.50		7	7	0.40		7	7	0.50
	8	7	0.50		8	7	0.60		8	7	0.50
	8	8	1.00		8	8	1.00		8	8	1.00
Project 2	4	4	0.10	Project 2	4	4	0.10	Project 2	4	4	0.10
	5	4	0.30		5	4	0.30		5	4	0.20
	6	4	0.40		6	4	0.40		6	4	0.50
	7	4	0.20		7	4	0.20		7	4	0.20
	8	4	0.00		8	4	0.00		8	4	0.00
	5	5	0.30		5	5	0.30		5	5	0.20
	6	5	0.20		6	5	0.20		6	5	0.30
	7	5	0.50		7	5	0.50		7	5	0.40
	8	5	0.00		8	5	0.00		8	5	0.10
	6	6	0.20		6	6	0.20		6	6	0.20
	7	6	0.70		7	6	0.70		7	6	0.65
	8	6	0.10		8	6	0.10		8	6	0.15
	7	7	0.35		7	7	0.35		7	7	0.30
	8	7	0.65		8	7	0.65		8	7	0.70
	8	8	1.00		8	8	1.00		8	8	1.00
Project 3	4	4	0.20	Project 3	4	4	0.20	Project 3	4	4	0.20
	5	4	0.40		5	4	0.40		5	4	0.40
	6	4	0.20		6	4	0.20		6	4	0.20
	7	4	0.10		7	4	0.10		7	4	0.10
	8	4	0.10		8	4	0.10		8	4	0.10
	5	5	0.40		5	5	0.40		5	5	0.40
	6	5	0.35		6	5	0.35		6	5	0.35
	7	5	0.15		7	5	0.15		7	5	0.15
	8	5	0.10		8	5	0.10		8	5	0.10
	6	6	0.30		6	6	0.30		6	6	0.30
	7	6	0.40		7	6	0.40		7	6	0.40
	8	6	0.30		8	6	0.30		8	6	0.30
	7	7	0.30		7	7	0.30		7	7	0.30
	8	7	0.70		8	7	0.70		8	7	0.70
	8	8	1.00		8	8	1.00		8	8	1.00
Project 4	4	4	0.40	Project 4	4	4	0.40	Project 4	4	4	0.30
	5	4	0.40		5	4	0.30		5	4	0.30
	6	4	0.10		6	4	0.20		6	4	0.20
	7	4	0.10		7	4	0.10		7	4	0.15
	8	4	0.00		8	4	0.00		8	4	0.05
	5	5	0.50		5	5	0.50		5	5	0.40
	6	5	0.30		6	5	0.30		6	5	0.30
	7	5	0.15		7	5	0.10		7	5	0.15
	8	5	0.05		8	5	0.10		8	5	0.15
	6	6	0.40		6	6	0.40		6	6	0.30
	7	6	0.35		7	6	0.30		7	6	0.35

	8	6	0.25		8	6	0.30		8	6	0.35
	7	7	0.55		7	7	0.50		7	7	0.45
	8	7	0.45		8	7	0.50		8	7	0.55
	8	8	1.00		8	8	1.00		8	8	1.00

Appendix B: LINGO Code for Three-Project, Three-Time Period Integer Program

```
model:

DATA:
M = 10000.0;
nTRL = 4; nTRLP1 = 5;
nTRLM1 = 3;
nV = 3;
nTP = 3;
nF = 8;
nFFFF = 4;
nFF = 2;
ENDDATA

SETS:
COST/1..nV/: C1, C2, C3;     !Cost matrix for each project at each period;
TRL/1..nTRL/:  FP11, FP21, FP31;
TRLP1/1..nTRLP1/;
TRLM1/1..nTRLM1/;
FUND/1..nF/: W;   ! Logic variables
F4/1..nFFFF/;
F2/1..nFF/;
TRLP1M(TRLP1,TRLP1,TRLP1): OC3, X12, X22, X32, X13, X23, X33;
TRL2(TRL,TRL);
TRL3(TRL,TRL,TRL);
TRL3P1(TRLP1,TRLP1,TRLP1);
TRLM1P2(TRLM1,TRLM1);
TRLM1P3(TRLM1,TRLM1,TRLM1);
TRLM1P4(TRLM1,TRLM1,TRLM1,TRLM1);
TRLM1P5(TRLM1,TRLM1,TRLM1,TRLM1,TRLM1);
TRLM1P6(TRLM1,TRLM1,TRLM1,TRLM1,TRLM1,TRLM1);
TRLMATRIX2(TRL,TRL): P12, P22, P32, P13, P23, P33;   !Second and third stage probability
matrices

!Logic variables that end funding in the second stage;
TY1(TRLM1,TRLM1,TRLM1,FUND): Y1;
TY2(TRLM1,TRLM1,F4): Y2, Y3, Y4;
TY3(TRLM1,F2): Y5, Y6, Y7;

!Logic variables that have at least one funding in the third time period;
TZ1(TRLM1,TRLM1,TRLM1,TRLM1,TRLM1,TRLM1,FUND): Z1;
TZ2(TRLM1,TRLM1,TRLM1,TRLM1,TRLM1,F4): Z2, Z3, Z4;
TZ3(TRLM1,TRLM1,TRLM1,TRLM1,F2): Z5, Z6, Z7;
!TZ4(TRLM1,TRLM1,TRLM1): Z8;
TZ5(TRLM1,TRLM1,TRLM1,TRLM1,F4): Z9, Z13, Z17;
TZ6(TRLM1,TRLM1,TRLM1,F2): Z10, Z11, Z14, Z15, Z18, Z19;
!TZ7(TRLM1,TRLM1): Z12, Z16, Z20;
TZ8(TRLM1,TRLM1,F2): Z21, Z23, Z25;
!TZ9(TRLM1): Z22, Z24, Z26;
ENDSETS

DATA:
FP11, FP21, FP31 = @OLE('C:\ProbsTRL5-8_2.xls', 'fpone', 'fptwo', 'fpthree');
P12, P22, P32, P13, P23, P33 = @OLE('C:\ProbsTRL5-8_2.xls', 'pr12', 'pr22', 'pr32', 'pr13',
'pr23', 'pr33');
C1, C2, C3 = @OLE('C:\CostsTRL5-8_2.xls', 'cost1', 'cost2', 'cost3');
B1, B2, B3 = @OLE('C:\CostsTRL5-8_2.xls', 'bud1', 'bud2', 'bud3');

ENDDATA

!objective function;
MAX = (OC3(4,4,4) + @SUM(TRLP1(H)| H #NE# 4: @SUM(TRLP1(I)| I #NE# 4: OC3(H,I,4) ))  +
       @SUM(TRLP1(I)| I #NE# 4: @SUM(TRLP1(J)| J #NE# 4: OC3(4,I,J) )) +
       @SUM(TRLP1(H)| H #NE# 4: @SUM(TRLP1(J)| J #NE# 4: OC3(H,4,J) )) +
       @SUM(TRLP1(H)| H #NE# 4: OC3(H,4,4)) +
       @SUM(TRLP1(I)| I #NE# 4: OC3(4,I,4)) +
```

```
       @SUM(TRLP1(J)| J #NE# 4: OC3(4,4,J))  );

!subject to;

   !1) OC3(4,I,J) constraint;
   @FOR(TRLM1P2(I,J): OC3(4,I,J) = @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,I)*FP31(J1)*P32(J1,J2)*P33(J2,J)*Z1(H1
,I1,J1,H2,I2,J2,1)) );
   !2) OC3(H,4,J);
@FOR(TRLM1P2(H,J): OC3(H,4,J) = @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,H)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*P33(J2,J)*Z1(H1
,I1,J1,H2,I2,J2,1)) );
   !3) OC3(H,I,4);
   @FOR(TRLM1P2(H,I): OC3(H,I,4) = @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,H)*FP21(I1)*P22(I1,I2)*P23(I2,I)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1
,I1,J1,H2,I2,J2,1)) );
   !4) OC3(4,4,J);
   @FOR(TRLM1(J): OC3(4,4,J) = @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*P33(J2,J)*Z1(H1
,I1,J1,H2,I2,J2,1)) );
   !5) OC3(4,I,4);
   @FOR(TRLM1(I): OC3(4,I,4) = @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,I)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1
,I1,J1,H2,I2,J2,1)) );
   !6) OC3(H,4,4);
   @FOR(TRLM1(H): OC3(H,4,4) = @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,H)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1
,I1,J1,H2,I2,J2,1)) );
   !7) OC3(4,4,4);
   OC3(4,4,4) = FP11(4)*FP21(4)*FP31(4)*W(1) + @SUM(TRLM1P3(H1,I1,J1):
FP11(H1)*P12(H1,4)*FP21(I1)*P22(I1,4)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,1))+@SUM(TRLM1P6(H1,I1,J1,
H2,I2,J2):
P11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1,
I1,J1,H2,I2,J2,1));
   !8) OC3(4,5,J);
   @FOR(TRLM1(J): OC3(4,5,J) = @SUM(TRLM1P4(H1,J1,H2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP31(J1)*P32(J1,J2)*P33(J2,J)*Z13(H1,J1,H2,J2,1)) +
@SUM(TRLM1P5(H1,I1,J1,H2,J2):
P11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*FP31(J1)*P32(J1,J2)*P33(J2,J)*Z3(H1,I1,J1,H2,J2,1)) +
@SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,J2)*P33(J2,J)*Z1(H1,I1,J1,H2,
I2,J2,3)) );
   !9) OC3(4,I,5);
   @FOR(TRLM1(I): OC3(4,I,5) = @SUM(TRLM1P4(H1,I1,H2,I2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,I)*Z9(H1,I1,H2,I2,1)) +
@SUM(TRLM1P5(H1,I1,J1,H2,I2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,I)*FP31(J1)*Z2(H1,I1,J1,H2,I2,1)) +
@SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,I)*FP31(J1)*P32(J1,J2)*Z1(H1,I1,J1,H2,
I2,J2,2)) );
   !10) OC3(4,5,5);
    OC3(4,5,5) = @SUM(TRLM1P2(H1,H2): FP11(H1)*P12(H1,H2)*P13(H2,4)*Z21(H1,H2,1)) +
@SUM(TRLM1P3(H1,J1,H2): FP11(H1)*P12(H1,H2)*P13(H2,4)*FP31(J1)*Z14(H1,J1,H2,1)) +
@SUM(TRLM1P4(H1,J1,H2,J2): P11(H1)*P12(H1,H2)*P13(H2,4)*FP31(J1)*P32(J1,J2)*Z13(H1,J1,H2,J2,2))
+ @SUM(TRLM1P3(H1,I1,H2): FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*Z10(H1,I1,H2,1)) +
@SUM(TRLM1P4(H1,I1,H2,I2): FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*Z9(H1,I1,H2,I2,2))
+ @SUM(TRLM1P4(H1,I1,J1,H2): FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*FP31(J1)*Z5(H1,I1,J1,H2,1))
+ @SUM(TRLM1P5(H1,I1,J1,H2,I2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*FP31(J1)*Z2(H1,I1,J1,H2,I2,2)) +
@SUM(TRLM1P5(H1,I1,J1,H2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*FP31(J1)*P32(J1,J2)*Z3(H1,I1,J1,H2,J2,2)) +
@SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,J2)*Z1(H1,I1,J1,H2,I2,J2,5))
+ @SUM(TRLM1P5(H1,I1,J1,I2,J2):
FP11(H1)*P12(H1,4)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,J2)*Y1(H1,I1,J1,1)) +
@SUM(TRLM1P4(H1,I1,J1,I2): FP11(H1)*P12(H1,4)*FP21(I1)*P22(I1,I2)*FP31(J1)*Y1(H1,I1,J1,2)) +
@SUM(TRLM1P4(H1,I1,J1,J2): FP11(H1)*P12(H1,4)*FP21(I1)*FP31(J1)*P32(J1,J2)*Y1(H1,I1,J1,3)) +
@SUM(TRLM1P3(H1,I1,J1): FP11(H1)*P12(H1,4)*FP21(I1)*FP31(J1)*Y1(H1,I1,J1,5)) +
@SUM(TRLM1P3(H1,I1,I2): FP11(H1)*P12(H1,4)*FP21(I1)*P22(I1,I2)*Y2(H1,I1,1)) +
@SUM(TRLM1P2(H1,I1): FP11(H1)*P12(H1,4)*FP21(I1)*Y2(H1,I1,2)) +
@SUM(TRLM1P3(H1,J1,J2): FP11(H1)*P12(H1,4)*FP31(J1)*P32(J1,J2)*Y3(H1,J1,1)) +
```

```
@SUM(TRLM1P2(H1,J1): FP11(H1)*P12(H1,4)*FP31(J1)*Y3(H1,J1,2)) + @SUM(TRLM1(H1):
FP11(H1)*P12(H1,4)*Y5(H1,1)) + @SUM(TRLM1P2(I1,J1): FP11(4)*FP21(I1)*FP31(J1)*W(1)) +
@SUM(TRLM1(I1): FP11(4)*FP21(I1)*W(2)) + @SUM(TRLM1(J1): FP11(4)*FP31(J1)*W(3)) + FP11(4)*W(5);
    !11) OC3(5,4,J);
    @FOR(TRLM1(J): OC3(5,4,J) = @SUM(TRLM1P4(I1,J1,I2,J2):
FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*P33(J2,J)*Z17(I1,J1,I2,J2,1)) +
@SUM(TRLM1P5(H1,I1,J1,I2,J2): FP11(H1)*FP21(I1)*P22(I1,I2)*P23(I2,4)*
FP31(J1)*P32(J1,J2)*P33(J2,J)*Z4(H1,I1,J1,I2,J2,1)) + @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*P33(J2,J)*Z1(H1,I1,J1,H2,
I2,J2,4)) );
    !12) OC3(H,4,5);
    @FOR(TRLM1(H): OC3(H,4,5) = @SUM(TRLM1P4(H1,I1,H2,I2):
P11(H1)*P12(H1,H2)*P13(H2,H)*FP21(I1)*P22(I1,I2)*P23(I2,4)*Z9(H1,I1,H2,I2,1)) +
@SUM(TRLM1P5(H1,I1,J1,H2,I2): FP11(H1)*P12(H1,H2)*P13(H2,H)*FP21(I1)*P22(I1,I2)*P23(I2,4)*
            FP31(J1)*Z2(H1,I1,J1,H2,I2,1)) + @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,H)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*Z1(H1,I1,J1,H2,
I2,J2,2)) );
    !13) OC3(5,4,5);
    OC3(5,4,5) =  @SUM(TRLM1P2(I1,I2): FP21(I1)*P22(I1,I2)*P23(I2,4)*Z23(I1,I2,1)) +
@SUM(TRLM1P3(I1,J1,I2): FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*Z18(I1,J1,I2,1)) +
@SUM(TRLM1P4(I1,J1,I2,J2):
FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*Z17(I1,J1,I2,J2,2)) +
@SUM(TRLM1P3(H1,I1,I2): FP11(H1)*FP21(I1)*P22(I1,I2)*P23(I2,4)*Z11(H1,I1,I2,1)) +
@SUM(TRLM1P4(H1,I1,H2,I2): FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,I2)*P23(I2,4)*Z9(H1,I1,H2,I2,3))
+ @SUM(TRLM1P4(H1,I1,J1,I2): FP11(H1)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*Z6(H1,I1,J1,I2,1))
+ @SUM(TRLM1P5(H1,I1,J1,H2,I2):
P11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*Z2(H1,I1,J1,H2,I2,3)) +
@SUM(TRLM1P5(H1,I1,J1,I2,J2):
P11(H1)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*Z4(H1,I1,J1,I2,J2,2)) +
@SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
P11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*Z1(H1,I1,J1,H2,I2,J2,6)) +
@SUM(TRLM1P5(H1,I1,J1,H2,J2):
FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,4)*FP31(J1)*P32(J1,J2)*Y1(H1,I1,J1,1)) +
@SUM(TRLM1P4(H1,I1,J1,H2): FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,4)*FP31(J1)*Y1(H1,I1,J1,2)) +
@SUM(TRLM1P4(H1,I1,J1,J2): FP11(H1)*FP21(I1)*P22(I1,4)*FP31(J1)*P32(J1,J2)*Y1(H1,I1,J1,4)) +
@SUM(TRLM1P3(H1,I1,J1): FP11(H1)*FP21(I1)*P22(I1,4)*FP31(J1)*Y1(H1,I1,J1,6)) +
@SUM(TRLM1P3(H1,I1,H2): FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,4)*Y2(H1,I1,1)) +
@SUM(TRLM1P2(H1,I1): FP11(H1)*FP21(I1)*P22(I1,4)*Y2(H1,I1,3)) +
@SUM(TRLM1P3(I1,J1,J2): FP21(I1)*P22(I1,4)*FP31(J1)*P32(J1,J2)*Y4(I1,J1,1)) +
@SUM(TRLM1P2(I1,J1): FP21(I1)*P22(I1,4)*FP31(J1)*Y4(I1,J1,2)) + @SUM(TRLM1(I1):
FP21(I1)*P22(I1,4)*Y6(I1,1)) + @SUM(TRLM1P2(H1,J1): FP11(H1)*FP21(4)*FP31(J1)*W(1)) +
@SUM(TRLM1(H1): FP11(H1)*FP21(4)*W(2)) + @SUM(TRLM1(J1): FP21(4)*FP31(J1)*W(4)) + FP21(4)*W(6);
   !14) OC3(H,5,4);
   @FOR(TRLM1(H): OC3(H,5,4) = @SUM(TRLM1P4(H1,J1,H2,J2):
P11(H1)*P12(H1,H2)*P13(H2,H)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z13(H1,J1,H2,J2,1)) +
          @SUM(TRLM1P5(H1,I1,J1,H2,J2): FP11(H1)*P12(H1,H2)*P13(H2,H)*FP21(I1)*
FP31(J1)*P32(J1,J2)*P33(J2,4)*Z3(H1,I1,J1,H2,J2,1)) + @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,H)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1,I1,J1,H2,
I2,J2,3)) );
   !15) OC3(5,I,4);
   @FOR(TRLM1(I): OC3(5,I,4) = @SUM(TRLM1P4(I1,J1,I2,J2):
FP21(I1)*P22(I1,I2)*P23(I2,I)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z17(I1,J1,I2,J2,1)) +
@SUM(TRLM1P5(H1,I1,J1,I2,J2): FP11(H1)*FP21(I1)*P22(I1,I2)*P23(I2,I)*
    FP31(J1)*P32(J1,J2)*P33(J2,4)*Z4(H1,I1,J1,I2,J2,1)) + @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,I2)*P23(I2,I)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1,I1,J1,H2,
I2,J2,4)) );
   !16) OC3(5,5,4);
   OC3(5,5,4) = @SUM(TRLM1P2(J1,J2): FP31(J1)*P32(J1,J2)*P33(J2,4)*Z25(J1,J2,1)) +
@SUM(TRLM1P3(I1,J1,J2): FP21(I1)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z19(I1,J1,J2,1)) +
@SUM(TRLM1P4(I1,J1,I2,J2):
FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z17(I1,J1,I2,J2,3)) +
@SUM(TRLM1P3(H1,J1,J2): FP11(H1)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z15(H1,J1,J2,1)) +
@SUM(TRLM1P4(H1,J1,H2,J2):
FP11(H1)*P12(H1,H2)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z13(H1,J1,H2,J2,3)) +
@SUM(TRLM1P4(H1,I1,J1,J2): FP11(H1)*FP21(I1)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z7(H1,I1,J1,J2,1)) +
            @SUM(TRLM1P5(H1,I1,J1,H2,J2):
P11(H1)*P12(H1,H2)*FP21(I1)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z3(H1,I1,J1,H2,J2,3)) +
@SUM(TRLM1P5(H1,I1,J1,I2,J2):
P11(H1)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z4(H1,I1,J1,I2,J2,3)) +
@SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
P11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1,I1,J1,H2,I2,J2,7)) +
```

```
            @SUM(TRLM1P5(H1,I1,J1,H2,I2):
FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,1)) +
@SUM(TRLM1P4(H1,I1,J1,H2): FP11(H1)*P12(H1,H2)*FP21(I1)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,3)) +
@SUM(TRLM1P4(H1,I1,J1,I2): FP11(H1)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,4)) +
             @SUM(TRLM1P3(H1,I1,J1): FP11(H1)*FP21(I1)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,7)) +
@SUM(TRLM1P3(H1,J1,H2): FP11(H1)*P12(H1,H2)*FP31(J1)*P32(J1,4)*Y3(H1,J1,1)) +
@SUM(TRLM1P2(H1,J1): FP11(H1)*FP31(J1)*P32(J1,4)*Y3(H1,J1,3)) + @SUM(TRLM1P3(I1,J1,I2):
FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,4)*Y4(I1,J1,1)) + @SUM(TRLM1P2(I1,J1):
FP21(I1)*FP31(J1)*P32(J1,4)*Y4(I1,J1,3)) + @SUM(TRLM1(J1): FP31(J1)*P32(J1,4)*Y7(J1,1)) +
@SUM(TRLM1P2(H1,I1): FP11(H1)*FP21(I1)*FP31(4)*W(1)) + @SUM(TRLM1(H1): FP11(H1)*FP31(4)*W(3)) +
@SUM(TRLM1(I1): FP21(I1)*FP31(4)*W(4)) + P31(4)*W(7);
   !17) OC3(4,4,5);
   OC3(4,4,5) = @SUM(TRLM1P4(H1,I1,H2,I2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,4)*  Z9(H1,I1,H2,I2,1)) +
@SUM(TRLM1P5(H1,I1,J1,H2,I2):
P11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*Z2(H1,I1,J1,H2,I2,1)) +
             @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*P23(I2,4)*
FP31(J1)*P32(J1,J2)*Z1(H1,I1,J1,H2,I2,J2,2)) + @SUM(TRLM1P4(H1,I1,J1,J2):
FP11(H1)*P12(H1,4)*FP21(I1)*P22(I1,4)*FP31(J1)*P32(J1,J2)*Y1(H1,I1,J1,1))
+@SUM(TRLM1P3(H1,I1,J1): FP11(H1)*P12(H1,4)*FP21(I1)*P22(I1,4)*FP31(J1)*Y1(H1,I1,J1,2)) +
@SUM(TRLM1P2(H1,I1): FP11(H1)*P12(H1,4)*FP21(I1)*P22(I1,4)*Y2(H1,I1,1)) + @SUM(TRLM1(J1):
FP11(4)*FP21(4)*FP31(J1)*W(1)) + FP11(4)*FP21(4)*W(2);
   !18) OC3(4,5,4);
   OC3(4,5,4) = @SUM(TRLM1P4(H1,J1,H2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z13(H1,J1,H2,J2,1)) +
@SUM(TRLM1P5(H1,I1,J1,H2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*FP31(J1)*P32(J1,J2)*P33(J2,4)*
Z3(H1,I1,J1,H2,J2,1)) + @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*P13(H2,4)*FP21(I1)*P22(I1,I2)*
FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1,I1,J1,H2,I2,J2,3)) + @SUM(TRLM1P4(H1,I1,J1,I2):
FP11(H1)*P12(H1,4)*FP21(I1)*P22(I1,I2)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,1)) +
@SUM(TRLM1P3(H1,I1,J1): FP11(H1)*P12(H1,4)*FP21(I1)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,3)) +
@SUM(TRLM1P2(H1,J1): FP11(H1)*P12(H1,4)*FP31(J1)*P32(J1,4)*Y3(H1,J1,1)) + @SUM(TRLM1(I1):
FP11(4)*FP21(I1)*FP31(4)*W(1)) + FP11(4)*FP31(4)*W(3);
   !19) OC3(5,4,4);
   OC3(5,4,4) = @SUM(TRLM1P4(I1,J1,I2,J2):
FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*P33(J2,4)*Z17(I1,J1,I2,J2,1)) +
@SUM(TRLM1P5(H1,I1,J1,I2,J2):
FP11(H1)*FP21(I1)*P22(I1,I2)*P23(I2,4)*FP31(J1)*P32(J1,J2)*P33(J2,4)*
Z4(H1,I1,J1,I2,J2,1)) +  @SUM(TRLM1P6(H1,I1,J1,H2,I2,J2):
FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,I2)*P23(I2,4)*
FP31(J1)*P32(J1,J2)*P33(J2,4)*Z1(H1,I1,J1,H2,I2,J2,4)) + @SUM(TRLM1P4(H1,I1,J1,H2):
FP11(H1)*P12(H1,H2)*FP21(I1)*P22(I1,4)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,1)) +
@SUM(TRLM1P3(H1,I1,J1): FP11(H1)*FP21(I1)*P22(I1,4)*FP31(J1)*P32(J1,4)*Y1(H1,I1,J1,4)) +
@SUM(TRLM1P2(I1,J1): FP21(I1)*P22(I1,4)*FP31(J1)*P32(J1,4)*Y4(I1,J1,1)) + @SUM(TRLM1(H1):
FP11(H1)*FP21(4)*FP31(4)*W(1)) + FP21(4)*FP31(4)*W(4);

   !Constraints linking all 8 W's to decision variables in first stage (X11,X21,X31), e.g, W(2)
= (1,1,0);
   !The W's correspond to those funding decisions where the "success" stage is reached after
the first time period;

   W(1) <= X11;
   W(1) <= X21;
   W(1) <= X31;
   X11 + X21 + X31 - W(1) <= 2;
   W(2) <= X11;
   W(2) <= X21;
   W(2) <= (1-X31);
   X11 + X21 + (1-X31) - W(2) <= 2;
   W(3) <= X11;
   W(3) <= (1-X21);
   W(3) <= X31;
   X11 + (1-X21) + X31 - W(3) <= 2;
   W(4) <= (1-X11);
   W(4) <= X21;
   W(4) <= X31;
   (1-X11) + X21 + X31 - W(4) <= 2;
   W(5) <= X11;
   W(5) <= (1-X21);
```

```
   W(5) <= (1-X31);
   X11 + (1-X21) + (1-X31) - W(5) <= 2;
   W(6) <= (1-X11);
   W(6) <= X21;
   W(6) <= (1-X31);
   (1-X11) + X21 + (1-X31) - W(6) <= 2;
   W(7) <= (1-X11);
   W(7) <= (1-X21);
   W(7) <= X31;
   (1-X11) + (1-X21) + X31 - W(7) <= 2;
   W(8) <= (1-X11);
   W(8) <= (1-X21);
   W(8) <= (1-X31);
   (1-X11) + (1-X21) + (1-X31) - W(8) <= 2;

   !Constraints linking all 8 (really, 7) Y(I)'s to decision variables in second stage
(X12,X22,X32);
   !These variables correspond to funding decisions where the "success" stage is reached in the
2nd stage;

   !Y1 Constraints;
   @FOR(TRLM1P3(H,I,J): @FOR(FUND(F): Y1(H,I,J,F) <= X11) );
   @FOR(TRLM1P3(H,I,J): @FOR(FUND(F): Y1(H,I,J,F) <= X21) );
   @FOR(TRLM1P3(H,I,J): @FOR(FUND(F): Y1(H,I,J,F) <= X31) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,1) <= X12(H,I,J) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,1) <= X22(H,I,J) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,1) <= X32(H,I,J) );
   @FOR(TRLM1P3(H,I,J): X11 + X21 + X31 + X12(H,I,J) + X22(H,I,J) + X32(H,I,J) - Y1(H,I,J,1) <=
5);
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,2) <= X12(H,I,J) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,2) <= X22(H,I,J) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,2) <= (1-X32(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): X11 + X21 + X31 + X12(H,I,J) + X22(H,I,J) + (1-X32(H,I,J)) -
Y1(H,I,J,2) <= 5);
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,3) <= X12(H,I,J) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,3) <= (1-X22(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,3) <= X32(H,I,J) );
   @FOR(TRLM1P3(H,I,J): X11 + X21 + X31 + X12(H,I,J) + (1-X22(H,I,J)) + X32(H,I,J) -
Y1(H,I,J,3) <= 5);
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,4) <= (1-X12(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,4) <= X22(H,I,J) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,4) <= X32(H,I,J) );
   @FOR(TRLM1P3(H,I,J): X11 + X21 + X31 + (1-X12(H,I,J)) + X22(H,I,J) + X32(H,I,J) -
Y1(H,I,J,4) <= 5);
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,5) <= X12(H,I,J));
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,5) <= (1-X22(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,5) <= (1-X32(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): X11 + X21 + X31 + X12(H,I,J) + (1-X22(H,I,J)) + (1+X32(H,I,J)) -
Y1(H,I,J,5) <= 5);
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,6) <= (1-X12(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,6) <= X22(H,I,J) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,6) <= (1-X32(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): X11 + X21 + X31 + (1-X12(H,I,J)) + X22(H,I,J) + (1-X32(H,I,J)) -
Y1(H,I,J,6) <= 5);
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,7) <= (1-X12(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,7) <= (1-X22(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,7) <= X32(H,I,J) );
   @FOR(TRLM1P3(H,I,J): X11 + X21 + X31 + (1-X12(H,I,J)) + (1-X22(H,I,J)) + X32(H,I,J) -
Y1(H,I,J,7) <= 5);
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,8) <= (1-X12(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,8) <= (1-X22(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): Y1(H,I,J,8) <= (1-X32(H,I,J)) );
   @FOR(TRLM1P3(H,I,J): X11 + X21 + X31 + (1-X12(H,I,J)) +(1-X22(H,I,J))+ (1-X32(H,I,J)) -
Y1(H,I,J,8)<= 5);

   !Y2 Constraints;
   @FOR(TRLM1P2(H,I): @FOR(F4(F): Y2(H,I,F) <= X11) );
   @FOR(TRLM1P2(H,I): @FOR(F4(F): Y2(H,I,F) <= X21) );
   @FOR(TRLM1P2(H,I): @FOR(F4(F): Y2(H,I,F) <= (1-X31)) );
   @FOR(TRLM1P2(H,I): Y2(H,I,1) <= X12(H,I,5) );
   @FOR(TRLM1P2(H,I): Y2(H,I,1) <= X22(H,I,5) );
```

```
@FOR(TRLM1P2(H,I): X11 + X21 + (1-X31) + X12(H,I,5) + X22(H,I,5) - Y2(H,I,1) <= 4);
@FOR(TRLM1P2(H,I): Y2(H,I,2) <= X12(H,I,5) );
@FOR(TRLM1P2(H,I): Y2(H,I,2) <= (1-X22(H,I,5)) );
@FOR(TRLM1P2(H,I): X11 + X21 + (1-X31) + X12(H,I,5) + (1-X22(H,I,5)) - Y2(H,I,2) <= 4);
@FOR(TRLM1P2(H,I): Y2(H,I,3) <= (1-X12(H,I,5)) );
@FOR(TRLM1P2(H,I): Y2(H,I,3) <= X22(H,I,5) );
@FOR(TRLM1P2(H,I): X11 + X21 + (1-X31) + (1-X12(H,I,5)) + X22(H,I,5) - Y2(H,I,3) <= 4);
@FOR(TRLM1P2(H,I): Y2(H,I,4) <= (1-X12(H,I,5)) );
@FOR(TRLM1P2(H,I): Y2(H,I,4) <= (1-X22(H,I,5)) );
@FOR(TRLM1P2(H,I): X11 + X21 + (1-X31) + (1-X12(H,I,5)) + (1-X22(H,I,5)) - Y2(H,I,4) <= 4);

!Y3 Constraints;
@FOR(TRLM1P2(H,J): @FOR(F4(F): Y3(H,J,F) <= X11) );
@FOR(TRLM1P2(H,J): @FOR(F4(F): Y3(H,J,F) <= (1-X21)) );
@FOR(TRLM1P2(H,J): @FOR(F4(F): Y3(H,J,F) <= X31) );
@FOR(TRLM1P2(H,J): Y3(H,J,1) <= X12(H,5,J) );
@FOR(TRLM1P2(H,J): Y3(H,J,1) <= X32(H,5,J) );
@FOR(TRLM1P2(H,J): X11 + (1-X21) + X31 + X12(H,5,J) + X32(H,5,J) - Y3(H,J,1) <= 4);
@FOR(TRLM1P2(H,J): Y3(H,J,2) <= X12(H,5,J) );
@FOR(TRLM1P2(H,J): Y3(H,J,2) <= (1-X32(H,5,J)) );
@FOR(TRLM1P2(H,J): X11 + (1-X21) + X31 + X12(H,5,J) + (1-X32(H,5,J)) - Y3(H,J,2) <= 4);
@FOR(TRLM1P2(H,J): Y3(H,J,3) <= (1-X12(H,5,J)) );
@FOR(TRLM1P2(H,J): Y3(H,J,3) <= X32(H,5,J) );
@FOR(TRLM1P2(H,J): X11 + (1-X21) + X31 + (1-X12(H,5,J)) + X32(H,5,J) - Y3(H,J,3) <= 4);
@FOR(TRLM1P2(H,J): Y3(H,J,4) <= (1-X12(H,5,J)) );
@FOR(TRLM1P2(H,J): Y3(H,J,4) <= (1-X32(H,5,J)) );
@FOR(TRLM1P2(H,J): X11 + (1-X21) + X31 + (1-X12(H,5,J)) + (1-X32(H,5,J)) - Y3(H,J,4) <= 4);

!Y4 Constraints;
@FOR(TRLM1P2(I,J): @FOR(F4(F): Y4(I,J,F) <= (1-X11)) );
@FOR(TRLM1P2(I,J): @FOR(F4(F): Y4(I,J,F) <= X21) );
@FOR(TRLM1P2(I,J): @FOR(F4(F): Y4(I,J,F) <= X31) );
@FOR(TRLM1P2(I,J): Y4(I,J,1) <= X22(5,I,J) );
@FOR(TRLM1P2(I,J): Y4(I,J,1) <= X32(5,I,J) );
@FOR(TRLM1P2(I,J): (1-X11) + X21 + X31 + X22(5,I,J) + X32(5,I,J) - Y4(I,J,1) < = 4);
@FOR(TRLM1P2(I,J): Y4(I,J,2) <= X22(5,I,J) );
@FOR(TRLM1P2(I,J): Y4(I,J,2) <= (1-X32(5,I,J)) );
@FOR(TRLM1P2(I,J): (1-X11) + X21 + X31 + X22(5,I,J) + (1-X32(5,I,J)) - Y4(I,J,2) < = 4);
@FOR(TRLM1P2(I,J): Y4(I,J,3) <= (1-X22(5,I,J)) );
@FOR(TRLM1P2(I,J): Y4(I,J,3) <= X32(5,I,J) );
@FOR(TRLM1P2(I,J): (1-X11) + X21 + X31 + (1-X22(5,I,J)) + X32(5,I,J) - Y4(I,J,3) < = 4);
@FOR(TRLM1P2(I,J): Y4(I,J,4) <= (1-X22(5,I,J)) );
@FOR(TRLM1P2(I,J): Y4(I,J,4) <= (1-X32(5,I,J)) );
@FOR(TRLM1P2(I,J): (1-X11) + X21 + X31 + (1-X22(5,I,J)) + (1-X32(5,I,J)) - Y4(I,J,4) < = 4);

!Y5 Constraints;
@FOR(TRLM1(H): @FOR(F2(F): Y5(H,F) <= X11) );
@FOR(TRLM1(H): @FOR(F2(F): Y5(H,F) <= (1-X21)) );
@FOR(TRLM1(H): @FOR(F2(F): Y5(H,F) <= (1-X31)) );
@FOR(TRLM1(H): Y5(H,1) <= X12(H,5,5));
@FOR(TRLM1(H): X11 + (1-X21) + (1-X31) + X12(H,5,5) - Y5(H,1) <= 3);
@FOR(TRLM1(H): Y5(H,2) <= (1-X12(H,5,5)) );
@FOR(TRLM1(H): X11 + (1-X21) + (1-X31) + (1-X12(H,5,5)) - Y5(H,1) <= 3);

!Y6 Constraints;
@FOR(TRLM1(I): @FOR(F2(F): Y6(I,F) <= (1-X11)) );
@FOR(TRLM1(I): @FOR(F2(F): Y6(I,F) <= X21) );
@FOR(TRLM1(I): @FOR(F2(F): Y6(I,F) <= (1-X31)) );
@FOR(TRLM1(I): Y6(I,1) <= X22(5,I,5));
@FOR(TRLM1(I): (1-X11) + X21 + (1-X31) + X22(5,I,5) - Y6(I,1) <= 3);
@FOR(TRLM1(I): Y6(I,2) <= (1-X22(5,I,5)) );
@FOR(TRLM1(I): (1-X11) + X21 + (1-X31) + (1-X22(5,I,5)) - Y6(I,2) <= 3);

!Y7 Constraints;
@FOR(TRLM1(J): @FOR(F2(F): Y7(J,F) <= (1-X11)) );
@FOR(TRLM1(J): @FOR(F2(F): Y7(J,F) <= (1-X21)) );
@FOR(TRLM1(J): @FOR(F2(F): Y7(J,F) <= X31) );
@FOR(TRLM1(J): Y7(J,1) <= X32(5,5,J));
@FOR(TRLM1(J): (1-X11) + (1-X21) + X31 + X32(5,5,J) - Y7(J,1) <= 3);
@FOR(TRLM1(J): Y7(J,2) <= (1-X32(5,5,J)) );
@FOR(TRLM1(J): (1-X11) + (1-X21) + X31 + (1-X32(5,5,J)) - Y7(J,2) <= 3);
```

```
!Z1 Constraints -- Z1 is largest variable set, it corresponds to fund-all, fund-all in the
first two stages;
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): @FOR(FUND(F): Z1(H1,I1,J1,H2,I2,J2,F) <= X11) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): @FOR(FUND(F): Z1(H1,I1,J1,H2,I2,J2,F) <= X21) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): @FOR(FUND(F): Z1(H1,I1,J1,H2,I2,J2,F) <= X31) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): @FOR(FUND(F): Z1(H1,I1,J1,H2,I2,J2,F) <= X12(H1,I1,J1)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): @FOR(FUND(F): Z1(H1,I1,J1,H2,I2,J2,F) <= X22(H1,I1,J1)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): @FOR(FUND(F): Z1(H1,I1,J1,H2,I2,J2,F) <= X32(H1,I1,J1)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,1) <= X13(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,1) <= X23(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,1) <= X33(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) +
X32(H1,I1,J1) +
  X13(H2,I2,J2) + X23(H2,I2,J2) + X33(H2,I2,J2) - Z1(H1,I1,J1,H2,I2,J2,1) <= 8);
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,2) <= X13(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,2) <= X23(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,2) <= (1-X33(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) +
X32(H1,I1,J1) +
  X13(H2,I2,J2) + X23(H2,I2,J2) + (1-X33(H2,I2,J2)) - Z1(H1,I1,J1,H2,I2,J2,2) <= 8);
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,3) <= X13(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,3) <= (1-X23(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,3) <= X33(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) +
X32(H1,I1,J1) +
  X13(H2,I2,J2) + (1-X23(H2,I2,J2)) + X33(H2,I2,J2) - Z1(H1,I1,J1,H2,I2,J2,3) <= 8);
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,4) <= (1-X13(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,4) <= X23(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,4) <= X33(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) +
X32(H1,I1,J1) +
  (1-X13(H2,I2,J2)) + X23(H2,I2,J2) + X33(H2,I2,J2) - Z1(H1,I1,J1,H2,I2,J2,4) <= 8);
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,5) <= X13(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,5) <= (1-X23(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,5) <= (1-X33(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) +
X32(H1,I1,J1) +
  X13(H2,I2,J2) + (1-X23(H2,I2,J2)) + (1-X33(H2,I2,J2)) - Z1(H1,I1,J1,H2,I2,J2,5) <= 8);
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,6) <= (1-X13(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,6) <= X23(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,6) <= (1-X33(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) +
X32(H1,I1,J1) +
  (1-X13(H2,I2,J2)) + X23(H2,I2,J2) + (1-X33(H2,I2,J2)) - Z1(H1,I1,J1,H2,I2,J2,6) <= 8);
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,7) <= (1-X13(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,7) <= (1-X23(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,7) <= X33(H2,I2,J2) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) +
X32(H1,I1,J1) + (1-X13(H2,I2,J2)) + (1-X23(H2,I2,J2)) + X33(H2,I2,J2) - Z1(H1,I1,J1,H2,I2,J2,7)
<= 8);
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,8) <= (1-X13(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,8) <= (1-X23(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): Z1(H1,I1,J1,H2,I2,J2,8) <= (1-X33(H2,I2,J2)) );
   @FOR(TRLM1P6(H1,I1,J1,H2,I2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) +
X32(H1,I1,J1) + (1-X13(H2,I2,J2)) + (1-X23(H2,I2,J2)) + (1-X33(H2,I2,J2)) -
Z1(H1,I1,J1,H2,I2,J2,8) <= 8);

   !Z2 Constraints;
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): @FOR(F4(F): Z2(H1,I1,J1,H2,I2,F) <= X11) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): @FOR(F4(F): Z2(H1,I1,J1,H2,I2,F) <= X21) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): @FOR(F4(F): Z2(H1,I1,J1,H2,I2,F) <= X31) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): @FOR(F4(F): Z2(H1,I1,J1,H2,I2,F) <= X12(H1,I1,J1)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): @FOR(F4(F): Z2(H1,I1,J1,H2,I2,F) <= X22(H1,I1,J1)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): @FOR(F4(F): Z2(H1,I1,J1,H2,I2,F) <= (1-X32(H1,I1,J1)) ) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): Z2(H1,I1,J1,H2,I2,1) <= X13(H2,I2,5) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): Z2(H1,I1,J1,H2,I2,1) <= X23(H2,I2,5) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) + (1-
X32(H1,I1,J1)) + X13(H2,I2,5) + X23(H2,I2,5) - Z2(H1,I1,J1,H2,I2,1) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): Z2(H1,I1,J1,H2,I2,2) <= X13(H2,I2,5) );
```

```
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): Z2(H1,I1,J1,H2,I2,2) <= (1-X23(H2,I2,5)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) + (1-
X32(H1,I1,J1)) +  X13(H2,I2,5) + (1-X23(H2,I2,5)) - Z2(H1,I1,J1,H2,I2,2) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): Z2(H1,I1,J1,H2,I2,3) <= (1-X13(H2,I2,5)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): Z2(H1,I1,J1,H2,I2,3) <= X23(H2,I2,5) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) + (1-
X32(H1,I1,J1)) + (1-X13(H2,I2,5)) + X23(H2,I2,5) - Z2(H1,I1,J1,H2,I2,3) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): Z2(H1,I1,J1,H2,I2,4) <= (1-X13(H2,I2,5)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): Z2(H1,I1,J1,H2,I2,4) <= (1-X23(H2,I2,5)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,I2): X11 + X21 + X31 + X12(H1,I1,J1) + X22(H1,I1,J1) + (1-
X32(H1,I1,J1)) + (1-X13(H2,I2,5)) + (1-X23(H2,I2,5)) - Z2(H1,I1,J1,H2,I2,4) <= 7);

   !Z3 Constraints;
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): @FOR(F4(F): Z3(H1,I1,J1,H2,J2,F) <= X11) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): @FOR(F4(F): Z3(H1,I1,J1,H2,J2,F) <= X21) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): @FOR(F4(F): Z3(H1,I1,J1,H2,J2,F) <= X31) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): @FOR(F4(F): Z3(H1,I1,J1,H2,J2,F) <= X12(H1,I1,J1)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): @FOR(F4(F): Z3(H1,I1,J1,H2,J2,F) <= (1-X22(H1,I1,J1)) ) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): @FOR(F4(F): Z3(H1,I1,J1,H2,J2,F) <= X32(H1,I1,J1)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): Z3(H1,I1,J1,H2,J2,1) <= X13(H2,5,J2) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): Z3(H1,I1,J1,H2,J2,1) <= X33(H2,5,J2) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + (1-X22(H1,I1,J1)) +
X32(H1,I1,J1) + X13(H2,5,J2) + X33(H2,5,J2) - Z3(H1,I1,J1,H2,J2,1) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): Z3(H1,I1,J1,H2,J2,2) <= X13(H2,5,J2) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): Z3(H1,I1,J1,H2,J2,2) <= (1-X33(H2,5,J2)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + (1-X22(H1,I1,J1)) +
X32(H1,I1,J1) + X13(H2,5,J2) + (1-X33(H2,5,J2)) - Z3(H1,I1,J1,H2,J2,2) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): Z3(H1,I1,J1,H2,J2,3) <= (1-X13(H2,5,J2)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): Z3(H1,I1,J1,H2,J2,3) <= X33(H2,5,J2) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + (1-X22(H1,I1,J1)) +
X32(H1,I1,J1) + (1-X13(H2,5,J2)) + X33(H2,5,J2) - Z3(H1,I1,J1,H2,J2,3) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): Z3(H1,I1,J1,H2,J2,4) <= (1-X13(H2,5,J2)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): Z3(H1,I1,J1,H2,J2,4) <= (1-X33(H2,5,J2)) );
   @FOR(TRLM1P5(H1,I1,J1,H2,J2): X11 + X21 + X31 + X12(H1,I1,J1) + (1-X22(H1,I1,J1)) +
X32(H1,I1,J1) + (1-X13(H2,5,J2)) + (1-X33(H2,5,J2)) - Z3(H1,I1,J1,H2,J2,4) <= 7);


   !Z4 Constraints;
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): @FOR(F4(F): Z4(H1,I1,J1,I2,J2,F) <= X11) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): @FOR(F4(F): Z4(H1,I1,J1,I2,J2,F) <= X21) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): @FOR(F4(F): Z4(H1,I1,J1,I2,J2,F) <= X31) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): @FOR(F4(F): Z4(H1,I1,J1,I2,J2,F) <= (1-X12(H1,I1,J1)) ) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): @FOR(F4(F): Z4(H1,I1,J1,I2,J2,F) <= X22(H1,I1,J1)) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): @FOR(F4(F): Z4(H1,I1,J1,I2,J2,F) <= X32(H1,I1,J1)) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): Z4(H1,I1,J1,I2,J2,1) <= X23(5,I2,J2) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): Z4(H1,I1,J1,I2,J2,1) <= X33(5,I2,J2) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): X11 + X21 + X31 + (1-X12(H1,I1,J1)) + X22(H1,I1,J1) +
X32(H1,I1,J1) + X23(5,I2,J2) + X33(5,I2,J2) - Z4(H1,I1,J1,I2,J2,1) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): Z4(H1,I1,J1,I2,J2,2) <= X23(5,I2,J2) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): Z4(H1,I1,J1,I2,J2,2) <= (1-X33(5,I2,J2)) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): X11 + X21 + X31 + (1-X12(H1,I1,J1)) + X22(H1,I1,J1) +
X32(H1,I1,J1) + X23(5,I2,J2) + (1-X33(5,I2,J2)) - Z4(H1,I1,J1,I2,J2,2) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): Z4(H1,I1,J1,I2,J2,3) <= (1-X23(5,I2,J2)) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): Z4(H1,I1,J1,I2,J2,3) <= X33(5,I2,J2) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): X11 + X21 + X31 + (1-X12(H1,I1,J1)) + X22(H1,I1,J1) +
X32(H1,I1,J1) + (1-X23(5,I2,J2)) + X33(5,I2,J2) - Z4(H1,I1,J1,I2,J2,3) <= 7);
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): Z4(H1,I1,J1,I2,J2,4) <= (1-X23(5,I2,J2)) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): Z4(H1,I1,J1,I2,J2,4) <= (1-X33(5,I2,J2)) );
   @FOR(TRLM1P5(H1,I1,J1,I2,J2): X11 + X21 + X31 + (1-X12(H1,I1,J1)) + X22(H1,I1,J1) +
X32(H1,I1,J1) + (1-X23(5,I2,J2)) + (1-X33(5,I2,J2)) - Z4(H1,I1,J1,I2,J2,4) <= 7);

   !Z5 Constraints;
   @FOR(TRLM1P4(H1,I1,J1,H2): @FOR(F2(F): Z5(H1,I1,J1,H2,F) <= X11) );
   @FOR(TRLM1P4(H1,I1,J1,H2): @FOR(F2(F): Z5(H1,I1,J1,H2,F) <= X21) );
   @FOR(TRLM1P4(H1,I1,J1,H2): @FOR(F2(F): Z5(H1,I1,J1,H2,F) <= X31) );
   @FOR(TRLM1P4(H1,I1,J1,H2): @FOR(F2(F): Z5(H1,I1,J1,H2,F) <= X12(H1,I1,J1)) );
   @FOR(TRLM1P4(H1,I1,J1,H2): @FOR(F2(F): Z5(H1,I1,J1,H2,F) <= (1-X22(H1,I1,J1)) ) );
   @FOR(TRLM1P4(H1,I1,J1,H2): @FOR(F2(F): Z5(H1,I1,J1,H2,F) <= (1-X32(H1,I1,J1)) ) );
   @FOR(TRLM1P4(H1,I1,J1,H2): Z5(H1,I1,J1,H2,1) <= X13(H2,5,5) );
   @FOR(TRLM1P4(H1,I1,J1,H2): X11 + X21 + X31 + X12(H1,I1,J1) + (1-X22(H1,I1,J1)) + (1-
X32(H1,I1,J1)) + X13(H2,5,5) - Z5(H1,I1,J1,H2,1) <= 6);
```

```
   @FOR(TRLM1P4(H1,I1,J1,H2): Z5(H1,I1,J1,H2,2) <= (1-X13(H2,5,5)) );
   @FOR(TRLM1P4(H1,I1,J1,H2): X11 + X21 + X31 + X12(H1,I1,J1) + (1-X22(H1,I1,J1)) + (1-
X32(H1,I1,J1)) + (1-X13(H2,5,5)) - Z5(H1,I1,J1,H2,2) <= 6);

   !Z6 Constraints;
   @FOR(TRLM1P4(H1,I1,J1,I2): @FOR(F2(F): Z6(H1,I1,J1,I2,F) <= X11) );
   @FOR(TRLM1P4(H1,I1,J1,I2): @FOR(F2(F): Z6(H1,I1,J1,I2,F) <= X21) );
   @FOR(TRLM1P4(H1,I1,J1,I2): @FOR(F2(F): Z6(H1,I1,J1,I2,F) <= X31) );
   @FOR(TRLM1P4(H1,I1,J1,I2): @FOR(F2(F): Z6(H1,I1,J1,I2,F) <= (1-X12(H1,I1,J1)) ) );
   @FOR(TRLM1P4(H1,I1,J1,I2): @FOR(F2(F): Z6(H1,I1,J1,I2,F) <= X22(H1,I1,J1)) );
   @FOR(TRLM1P4(H1,I1,J1,I2): @FOR(F2(F): Z6(H1,I1,J1,I2,F) <= (1-X32(H1,I1,J1)) ) );
   @FOR(TRLM1P4(H1,I1,J1,I2): Z6(H1,I1,J1,I2,1) <= X23(5,I2,5) );
   @FOR(TRLM1P4(H1,I1,J1,I2): X11 + X21 + X31 + (1-X12(H1,I1,J1)) + X22(H1,I1,J1) + (1-
X32(H1,I1,J1)) + X23(5,I2,5) - Z6(H1,I1,J1,I2,1) <= 6);
   @FOR(TRLM1P4(H1,I1,J1,I2): Z6(H1,I1,J1,I2,2) <= (1-X23(5,I2,5)) );
   @FOR(TRLM1P4(H1,I1,J1,I2): X11 + X21 + X31 + (1-X12(H1,I1,J1)) + X22(H1,I1,J1) + (1-
X32(H1,I1,J1)) + (1-X23(5,I2,5)) - Z6(H1,I1,J1,I2,2) <= 6);

   !Z7 Constraints;
   @FOR(TRLM1P4(H1,I1,J1,J2): @FOR(F2(F): Z7(H1,I1,J1,J2,F) <= X11) );
   @FOR(TRLM1P4(H1,I1,J1,J2): @FOR(F2(F): Z7(H1,I1,J1,J2,F) <= X21) );
   @FOR(TRLM1P4(H1,I1,J1,J2): @FOR(F2(F): Z7(H1,I1,J1,J2,F) <= X31) );
   @FOR(TRLM1P4(H1,I1,J1,J2): @FOR(F2(F): Z7(H1,I1,J1,J2,F) <= (1-X12(H1,I1,J1)) ) );
   @FOR(TRLM1P4(H1,I1,J1,J2): @FOR(F2(F): Z7(H1,I1,J1,J2,F) <= (1-X22(H1,I1,J1)) ) );
   @FOR(TRLM1P4(H1,I1,J1,J2): @FOR(F2(F): Z7(H1,I1,J1,J2,F) <= X32(H1,I1,J1)) );
   @FOR(TRLM1P4(H1,I1,J1,J2): Z7(H1,I1,J1,J2,1) <= X33(5,5,J2) );
   @FOR(TRLM1P4(H1,I1,J1,J2): X11 + X21 + X31 + (1-X12(H1,I1,J1)) + (1-X22(H1,I1,J1)) +
X32(H1,I1,J1) + X33(5,5,J2) - Z7(H1,I1,J1,J2,1) <= 6);
   @FOR(TRLM1P4(H1,I1,J1,J2): Z7(H1,I1,J1,J2,2) <= (1-X33(5,5,J2)) );
   @FOR(TRLM1P4(H1,I1,J1,J2): X11 + X21 + X31 + (1-X12(H1,I1,J1)) + (1-X22(H1,I1,J1)) +
X32(H1,I1,J1) + (1-X33(5,5,J2)) - Z7(H1,I1,J1,J2,2) <= 6);

   !Z9 Constraints;
   @FOR(TRLM1P4(H1,I1,H2,I2): @FOR(F4(F): Z9(H1,I1,H2,I2,F) <= X11) );
   @FOR(TRLM1P4(H1,I1,H2,I2): @FOR(F4(F): Z9(H1,I1,H2,I2,F) <= X21) );
   @FOR(TRLM1P4(H1,I1,H2,I2): @FOR(F4(F): Z9(H1,I1,H2,I2,F) <= (1-X31)) );
   @FOR(TRLM1P4(H1,I1,H2,I2): @FOR(F4(F): Z9(H1,I1,H2,I2,F) <= X12(H1,I1,5)) );
   @FOR(TRLM1P4(H1,I1,H2,I2): @FOR(F4(F): Z9(H1,I1,H2,I2,F) <= X22(H1,I1,5)) );
   @FOR(TRLM1P4(H1,I1,H2,I2): Z9(H1,I1,H2,I2,1) <= X13(H2,I2,5) );
   @FOR(TRLM1P4(H1,I1,H2,I2): Z9(H1,I1,H2,I2,1) <= X23(H2,I2,5) );
   @FOR(TRLM1P4(H1,I1,H2,I2): X11 + X21 + (1-X31) + X12(H1,I1,5) + X22(H1,I1,5) + X13(H2,I2,5)
+ X23(H2,I2,5) - Z9(H1,I1,H2,I2,1) <= 6);
   @FOR(TRLM1P4(H1,I1,H2,I2): Z9(H1,I1,H2,I2,2) <= X13(H2,I2,5) );
   @FOR(TRLM1P4(H1,I1,H2,I2): Z9(H1,I1,H2,I2,2) <= (1-X23(H2,I2,5)) );
   @FOR(TRLM1P4(H1,I1,H2,I2): X11 + X21 + (1-X31) + X12(H1,I1,5) + X22(H1,I1,5) + X13(H2,I2,5)
+ (1-X23(H2,I2,5)) - Z9(H1,I1,H2,I2,2) <= 6);
   @FOR(TRLM1P4(H1,I1,H2,I2): Z9(H1,I1,H2,I2,3) <= (1-X13(H2,I2,5)) );
   @FOR(TRLM1P4(H1,I1,H2,I2): Z9(H1,I1,H2,I2,3) <= X23(H2,I2,5) );
   @FOR(TRLM1P4(H1,I1,H2,I2): X11 + X21 + (1-X31) + X12(H1,I1,5) + X22(H1,I1,5) + (1-
X13(H2,I2,5)) + X23(H2,I2,5) - Z9(H1,I1,H2,I2,3) <= 6);
   @FOR(TRLM1P4(H1,I1,H2,I2): Z9(H1,I1,H2,I2,4) <= (1-X13(H2,I2,5)) );
   @FOR(TRLM1P4(H1,I1,H2,I2): Z9(H1,I1,H2,I2,4) <= (1-X23(H2,I2,5)) );
   @FOR(TRLM1P4(H1,I1,H2,I2): X11 + X21 + (1-X31) + X12(H1,I1,5) + X22(H1,I1,5) + (1-
X13(H2,I2,5)) + (1-X23(H2,I2,5)) - Z9(H1,I1,H2,I2,4) <= 6);

   !Z10 Constraints;
   @FOR(TRLM1P3(H1,I1,H2): @FOR(F2(F): Z10(H1,I1,H2,F) <= X11) );
   @FOR(TRLM1P3(H1,I1,H2): @FOR(F2(F): Z10(H1,I1,H2,F) <= X21) );
   @FOR(TRLM1P3(H1,I1,H2): @FOR(F2(F): Z10(H1,I1,H2,F) <= (1-X31)) );
   @FOR(TRLM1P3(H1,I1,H2): @FOR(F2(F): Z10(H1,I1,H2,F) <= X12(H1,I1,5)) );
   @FOR(TRLM1P3(H1,I1,H2): @FOR(F2(F): Z10(H1,I1,H2,F) <= (1-X22(H1,I1,5)) ) );
   @FOR(TRLM1P3(H1,I1,H2): Z10(H1,I1,H2,1) <= X13(H2,5,5) );
   @FOR(TRLM1P3(H1,I1,H2): X11 + X21 + (1-X31) + X12(H1,I1,5) + (1-X22(H1,I1,5)) + X13(H2,5,5)
- Z10(H1,I1,H2,1) <= 5);
   @FOR(TRLM1P3(H1,I1,H2): Z10(H1,I1,H2,2) <= (1-X13(H2,5,5)) );
   @FOR(TRLM1P3(H1,I1,H2): X11 + X21 + (1-X31) + X12(H1,I1,5) + (1-X22(H1,I1,5)) + (1-
X13(H2,5,5)) - Z10(H1,I1,H2,2) <= 5);
   !Z11 Constraints;
   @FOR(TRLM1P3(H1,I1,I2): @FOR(F2(F): Z11(H1,I1,I2,F) <= X11) );
   @FOR(TRLM1P3(H1,I1,I2): @FOR(F2(F): Z11(H1,I1,I2,F) <= X21) );
   @FOR(TRLM1P3(H1,I1,I2): @FOR(F2(F): Z11(H1,I1,I2,F) <= (1-X31)) );
```

```
   @FOR(TRLM1P3(H1,I1,I2): @FOR(F2(F): Z11(H1,I1,I2,F) <= (1-X12(H1,I1,5)) ) );
   @FOR(TRLM1P3(H1,I1,I2): @FOR(F2(F): Z11(H1,I1,I2,F) <= X22(H1,I1,5)) );
   @FOR(TRLM1P3(H1,I1,I2): Z11(H1,I1,I2,1) <= X23(5,I2,5));
   @FOR(TRLM1P3(H1,I1,I2): X11 + X21 + (1-X31) + (1-X12(H1,I1,5)) + X22(H1,I1,5) + X23(5,I2,5)
- Z11(H1,I1,I2,1) <= 5);
   @FOR(TRLM1P3(H1,I1,I2): Z11(H1,I1,I2,2) <= (1-X23(5,I2,5)) );
   @FOR(TRLM1P3(H1,I1,I2): X11 + X21 + (1-X31) + (1-X12(H1,I1,5)) + X22(H1,I1,5) + (1-
X23(5,I2,5)) - Z11(H1,I1,I2,2) <= 5);

   !Z13 Constraints;
   @FOR(TRLM1P4(H1,J1,H2,J2): @FOR(F4(F): Z13(H1,J1,H2,J2,F) <= X11) );
   @FOR(TRLM1P4(H1,J1,H2,J2): @FOR(F4(F): Z13(H1,J1,H2,J2,F) <= (1-X21)) );
   @FOR(TRLM1P4(H1,J1,H2,J2): @FOR(F4(F): Z13(H1,J1,H2,J2,F) <= X31) );
   @FOR(TRLM1P4(H1,J1,H2,J2): @FOR(F4(F): Z13(H1,J1,H2,J2,F) <= X12(H1,5,J1)) );
   @FOR(TRLM1P4(H1,J1,H2,J2): @FOR(F4(F): Z13(H1,J1,H2,J2,F) <= X32(H1,5,J1)) );
   @FOR(TRLM1P4(H1,J1,H2,J2): Z13(H1,J1,H2,J2,1) <= X13(H2,5,J2) );
   @FOR(TRLM1P4(H1,J1,H2,J2): Z13(H1,J1,H2,J2,1) <= X33(H2,5,J2) );
   @FOR(TRLM1P4(H1,J1,H2,J2): X11 + (1-X21) + X31 + X12(H1,5,J1) + X32(H1,5,J1) + X13(H2,5,J2)
+ X33(H2,5,J2) - Z13(H1,J1,H2,J2,1) <= 6);
   @FOR(TRLM1P4(H1,J1,H2,J2): Z13(H1,J1,H2,J2,2) <= X13(H2,5,J2) );
   @FOR(TRLM1P4(H1,J1,H2,J2): Z13(H1,J1,H2,J2,2) <= (1-X33(H2,5,J2)) );
   @FOR(TRLM1P4(H1,J1,H2,J2): X11 + (1-X21) + X31 + X12(H1,5,J1) + X32(H1,5,J1) + X13(H2,5,J2)
+ (1-X33(H2,5,J2)) - Z13(H1,J1,H2,J2,2) <= 6);
   @FOR(TRLM1P4(H1,J1,H2,J2): Z13(H1,J1,H2,J2,3) <= (1-X13(H2,5,J2)) );
   @FOR(TRLM1P4(H1,J1,H2,J2): Z13(H1,J1,H2,J2,3) <= 1-X33(H2,5,J2) );
   @FOR(TRLM1P4(H1,J1,H2,J2): X11 + (1-X21) + X31 + X12(H1,5,J1) + X32(H1,5,J1) + (1-
X13(H2,5,J2)) + X33(H2,5,J2) - Z13(H1,J1,H2,J2,3) <= 6);
   @FOR(TRLM1P4(H1,J1,H2,J2): Z13(H1,J1,H2,J2,4) <= (1-X13(H2,5,J2)) );
   @FOR(TRLM1P4(H1,J1,H2,J2): Z13(H1,J1,H2,J2,4) <= 1-X33(H2,5,J2) );
   @FOR(TRLM1P4(H1,J1,H2,J2): X11 + (1-X21) + X31 + X12(H1,5,J1) + X32(H1,5,J1) + (1-
X13(H2,5,J2)) + (1-X33(H2,5,J2)) - Z13(H1,J1,H2,J2,4) <= 6);

   !Z14 Constraints;
   @FOR(TRLM1P3(H1,J1,H2): @FOR(F2(F): Z14(H1,J1,H2,F) <= X11) );
   @FOR(TRLM1P3(H1,J1,H2): @FOR(F2(F): Z14(H1,J1,H2,F) <= (1-X21)) );
   @FOR(TRLM1P3(H1,J1,H2): @FOR(F2(F): Z14(H1,J1,H2,F) <= X31) );
   @FOR(TRLM1P3(H1,J1,H2): @FOR(F2(F): Z14(H1,J1,H2,F) <= X12(H1,5,J1)) );
   @FOR(TRLM1P3(H1,J1,H2): @FOR(F2(F): Z14(H1,J1,H2,F) <= (1-X32(H1,5,J1)) ) );
   @FOR(TRLM1P3(H1,J1,H2): Z14(H1,J1,H2,1) <= X13(H2,5,5) );
   @FOR(TRLM1P3(H1,J1,H2): X11 + (1-X21) + X31 + X12(H1,5,J1) + (1-X32(H1,5,J1)) + X13(H2,5,5)
- Z14(H1,J1,H2,1) <= 5);
   @FOR(TRLM1P3(H1,J1,H2): Z14(H1,J1,H2,2) <= (1-X13(H2,5,5)) );
   @FOR(TRLM1P3(H1,J1,H2): X11 + (1-X21) + X31 + X12(H1,5,J1) + (1-X32(H1,5,J1)) + (1-
X13(H2,5,5)) - Z14(H1,J1,H2,2) <= 5);

   !Z15 Constraints;
   @FOR(TRLM1P3(H1,J1,J2): @FOR(F2(F): Z15(H1,J1,J2,F) <= X11) );
   @FOR(TRLM1P3(H1,J1,J2): @FOR(F2(F): Z15(H1,J1,J2,F) <= (1-X21)) );
   @FOR(TRLM1P3(H1,J1,J2): @FOR(F2(F): Z15(H1,J1,J2,F) <= X31) );
   @FOR(TRLM1P3(H1,J1,J2): @FOR(F2(F): Z15(H1,J1,J2,F) <= (1-X12(H1,5,J1)) ) );
   @FOR(TRLM1P3(H1,J1,J2): @FOR(F2(F): Z15(H1,J1,J2,F) <= X32(H1,5,J1)) );
   @FOR(TRLM1P3(H1,J1,J2): Z15(H1,J1,J2,1) <= X33(5,5,J2) );
   @FOR(TRLM1P3(H1,J1,J2): X11 + (1-X21) + X31 + (1-X12(H1,5,J1)) + X32(H1,5,J1) + X33(5,5,J2)
- Z15(H1,J1,J2,1) <= 5);
   @FOR(TRLM1P3(H1,J1,J2): Z15(H1,J1,J2,2) <= (1-X33(5,5,J2)) );
   @FOR(TRLM1P3(H1,J1,J2): X11 + (1-X21) + X31 + (1-X12(H1,5,J1)) + X32(H1,5,J1) + (1-
X33(5,5,J2)) - Z15(H1,J1,J2,2) <= 5);

   !Z17 Constraints;
   @FOR(TRLM1P4(I1,J1,I2,J2): @FOR(F4(F): Z17(I1,J1,I2,J2,F) <= (1-X11)) );
   @FOR(TRLM1P4(I1,J1,I2,J2): @FOR(F4(F): Z17(I1,J1,I2,J2,F) <= X21) );
   @FOR(TRLM1P4(I1,J1,I2,J2): @FOR(F4(F): Z17(I1,J1,I2,J2,F) <= X31) );
   @FOR(TRLM1P4(I1,J1,I2,J2): @FOR(F4(F): Z17(I1,J1,I2,J2,F) <= X22(5,I1,J1)) );
   @FOR(TRLM1P4(I1,J1,I2,J2): @FOR(F4(F): Z17(I1,J1,I2,J2,F) <= X32(5,I1,J1)) );
   @FOR(TRLM1P4(I1,J1,I2,J2): Z17(I1,J1,I2,J2,1) <= X23(5,I2,J2) );
   @FOR(TRLM1P4(I1,J1,I2,J2): Z17(I1,J1,I2,J2,1) <= X33(5,I2,J2) );
   @FOR(TRLM1P4(I1,J1,I2,J2): (1-X11) + X21 + X31 + X22(5,I1,J1) + X32(5,I1,J1) + X23(5,I2,J2)
+ X33(5,I2,J2) - Z17(I1,J1,I2,J2,1) <= 6);
   @FOR(TRLM1P4(I1,J1,I2,J2): Z17(I1,J1,I2,J2,2) <= X23(5,I2,J2) );
   @FOR(TRLM1P4(I1,J1,I2,J2): Z17(I1,J1,I2,J2,2) <= (1-X33(5,I2,J2)) );
```

```
   @FOR(TRLM1P4(I1,J1,I2,J2): (1-X11) + X21 + X31 + X22(5,I1,J1) + X32(5,I1,J1) + X23(5,I2,J2)
+ (1-X33(5,I2,J2)) - Z17(I1,J1,I2,J2,2) <= 6);
   @FOR(TRLM1P4(I1,J1,I2,J2): Z17(I1,J1,I2,J2,3) <= (1-X23(5,I2,J2)) );
   @FOR(TRLM1P4(I1,J1,I2,J2): Z17(I1,J1,I2,J2,3) <= X33(5,I2,J2) );
   @FOR(TRLM1P4(I1,J1,I2,J2): (1-X11) + X21 + X31 + X22(5,I1,J1) + X32(5,I1,J1) + (1-
X23(5,I2,J2)) + X33(5,I2,J2) - Z17(I1,J1,I2,J2,3) <= 6);
   @FOR(TRLM1P4(I1,J1,I2,J2): Z17(I1,J1,I2,J2,4) <= (1-X23(5,I2,J2)) );
   @FOR(TRLM1P4(I1,J1,I2,J2): Z17(I1,J1,I2,J2,4) <= (1-X33(5,I2,J2)) );
   @FOR(TRLM1P4(I1,J1,I2,J2): (1-X11) + X21 + X31 + X22(5,I1,J1) + X32(5,I1,J1) + (1-
X23(5,I2,J2)) + (1-X33(5,I2,J2)) - Z17(I1,J1,I2,J2,3) <= 6);

   !Z18 Constraints;
   @FOR(TRLM1P3(I1,J1,I2): @FOR(F2(F): Z18(I1,J1,I2,F) <= (1-X11)) );
   @FOR(TRLM1P3(I1,J1,I2): @FOR(F2(F): Z18(I1,J1,I2,F) <= X21) );
   @FOR(TRLM1P3(I1,J1,I2): @FOR(F2(F): Z18(I1,J1,I2,F) <= X31) );
   @FOR(TRLM1P3(I1,J1,I2): @FOR(F2(F): Z18(I1,J1,I2,F) <= X22(5,I1,J1)) );
   @FOR(TRLM1P3(I1,J1,I2): @FOR(F2(F): Z18(I1,J1,I2,F) <= (1-X32(5,I1,J1)) ) );
   @FOR(TRLM1P3(I1,J1,I2): Z18(I1,J1,I2,1) <= X23(5,I2,5) );
   @FOR(TRLM1P3(I1,J1,I2): (1-X11) + X21 + X31 + X22(5,I1,J1) + (1-X32(5,I1,J1)) + X23(5,I2,5)
- Z18(I1,J1,I2,1) <= 5);
   @FOR(TRLM1P3(I1,J1,I2): Z18(I1,J1,I2,2) <= (1-X23(5,I2,5)) );
   @FOR(TRLM1P3(I1,J1,I2): (1-X11) + X21 + X31 + X22(5,I1,J1) + (1-X32(5,I1,J1)) + (1-
X23(5,I2,5)) - Z18(I1,J1,I2,2) <= 5);

   !Z19 Constraints;
   @FOR(TRLM1P3(I1,J1,J2): @FOR(F2(F): Z19(I1,J1,J2,F) <= (1-X11)) );
   @FOR(TRLM1P3(I1,J1,J2): @FOR(F2(F): Z19(I1,J1,J2,F) <= X21) );
   @FOR(TRLM1P3(I1,J1,J2): @FOR(F2(F): Z19(I1,J1,J2,F) <= X31) );
   @FOR(TRLM1P3(I1,J1,J2): @FOR(F2(F): Z19(I1,J1,J2,F) <= (1-X22(5,I1,J1)) ) );
   @FOR(TRLM1P3(I1,J1,J2): @FOR(F2(F): Z19(I1,J1,J2,F) <= X32(5,I1,J1)) );
   @FOR(TRLM1P3(I1,J1,J2): Z19(I1,J1,J2,1) <= X33(5,5,J2) );
   @FOR(TRLM1P3(I1,J1,J2): (1-X11) + X21 + X31 + (1-X22(5,I1,J1)) + X32(5,I1,J1) + X33(5,5,J2)
- Z19(I1,J1,J2,1) <= 5);
   @FOR(TRLM1P3(I1,J1,J2): Z19(I1,J1,J2,2) <= (1-X33(5,5,J2)) );
   @FOR(TRLM1P3(I1,J1,J2): (1-X11) + X21 + X31 + (1-X22(5,I1,J1)) + X32(5,I1,J1) + (1-
X33(5,5,J2)) - Z19(I1,J1,J2,2) <= 5);

   !Z21 Constraints;
   @FOR(TRLM1P2(H1,H2): @FOR(F2(F): Z21(H1,H2,F) <= X11) );
   @FOR(TRLM1P2(H1,H2): @FOR(F2(F): Z21(H1,H2,F) <= (1-X21)) );
   @FOR(TRLM1P2(H1,H2): @FOR(F2(F): Z21(H1,H2,F) <= (1-X31)) );
   @FOR(TRLM1P2(H1,H2): @FOR(F2(F): Z21(H1,H2,F) <= X12(H1,5,5)) );
   @FOR(TRLM1P2(H1,H2): Z21(H1,H2,1) <= X13(H2,5,5) );
   @FOR(TRLM1P2(H1,H2): X11 + (1-X21) + (1-X31) + X12(H1,5,5) + X13(H2,5,5) - Z21(H1,H2,1) <=
4);
   @FOR(TRLM1P2(H1,H2): Z21(H1,H2,2) <= (1-X13(H2,5,5)) );
   @FOR(TRLM1P2(H1,H2): X11 + (1-X21) + (1-X31) + X12(H1,5,5) + (1-X13(H2,5,5)) - Z21(H1,H2,2)
<= 4);

   !Z23 Constraints;
   @FOR(TRLM1P2(I1,I2): @FOR(F2(F): Z23(I1,I2,F) <= (1-X11)) );
   @FOR(TRLM1P2(I1,I2): @FOR(F2(F): Z23(I1,I2,F) <= X21) );
   @FOR(TRLM1P2(I1,I2): @FOR(F2(F): Z23(I1,I2,F) <= (1-X31)) );
   @FOR(TRLM1P2(I1,I2): @FOR(F2(F): Z23(I1,I2,F) <= X22(5,I1,5)) );
   @FOR(TRLM1P2(I1,I2): Z23(I1,I2,1) <= X23(5,I2,5) );
   @FOR(TRLM1P2(I1,I2): (1-X11) + X21 + (1-X31) + X22(5,I1,5) + X23(5,I2,5) - Z23(I1,I2,1) <=
4);
   @FOR(TRLM1P2(I1,I2): Z23(I1,I2,2) <= (1-X23(5,I2,5)) );
   @FOR(TRLM1P2(I1,I2): (1-X11) + X21 + (1-X31) + X22(5,I1,5) + (1-X23(5,I2,5)) - Z23(I1,I2,2)
<= 4);

   !Z25 Constraints;
   @FOR(TRLM1P2(J1,J2): @FOR(F2(F): Z25(J1,J2,F) <= (1-X11)) );
   @FOR(TRLM1P2(J1,J2): @FOR(F2(F): Z25(J1,J2,F) <= (1-X21)) );
   @FOR(TRLM1P2(J1,J2): @FOR(F2(F): Z25(J1,J2,F) <= X31) );
   @FOR(TRLM1P2(J1,J2): @FOR(F2(F): Z25(J1,J2,F) <= X32(5,5,J1)) );
   @FOR(TRLM1P2(J1,J2): Z25(J1,J2,1) <= X33(5,5,J2) );
   @FOR(TRLM1P2(J1,J2): (1-X11) + (1-X21) + X31 + X32(5,5,J1) + X33(5,5,J2) - Z25(J1,J2,1) <=
4);
   @FOR(TRLM1P2(J1,J2): Z25(J1,J2,2) <= (1-X33(5,5,J2)) );
```

```
   @FOR(TRLM1P2(J1,J2): (1-X11) + (1-X21) + X31 + X32(5,5,J1) + (1-X33(5,5,J2)) - Z25(J1,J2,2)
<= 4);

   @FOR(TRLM1P2(I,J): X12(5,I,J) = 0);
   @FOR(TRLM1P2(H,J): X22(H,5,J) = 0);
   @FOR(TRLM1P2(H,I): X32(H,I,5) = 0);
   @FOR(TRLM1P2(I,J): X13(5,I,J) = 0);
   @FOR(TRLM1P2(H,J): X23(H,5,J) = 0);
   @FOR(TRLM1P2(H,I): X33(H,I,5) = 0);

   @FOR(TRLM1(I): X12(5,I,5) = 0);
   @FOR(TRLM1(J): X12(5,5,J) = 0);
   @FOR(TRLM1(H): X22(H,5,5) = 0);
   @FOR(TRLM1(J): X22(5,5,J) = 0);
   @FOR(TRLM1(H): X32(H,5,5) = 0);
   @FOR(TRLM1(I): X32(5,I,5) = 0);
   @FOR(TRLM1(I): X13(5,I,5) = 0);
   @FOR(TRLM1(J): X13(5,5,J) = 0);
   @FOR(TRLM1(H): X23(H,5,5) = 0);
   @FOR(TRLM1(J): X23(5,5,J) = 0);
   @FOR(TRLM1(H): X33(H,5,5) = 0);
   @FOR(TRLM1(I): X33(5,I,5) = 0);

   @FOR(TRLM1P2(I,J): X12(4,I,J) = 0);
   @FOR(TRLM1P2(H,J): X22(H,4,J) = 0);
   @FOR(TRLM1P2(H,I): X32(H,I,4) = 0);
   @FOR(TRLM1P2(I,J): X13(4,I,J) = 0);
   @FOR(TRLM1P2(H,J): X23(H,4,J) = 0);
   @FOR(TRLM1P2(H,I): X33(H,I,4) = 0);

   @FOR(TRLM1(I): X12(4,I,4) = 0);
   @FOR(TRLM1(J): X12(4,4,J) = 0);
   @FOR(TRLM1(H): X22(H,4,4) = 0);
   @FOR(TRLM1(J): X22(4,4,J) = 0);
   @FOR(TRLM1(H): X32(H,4,4) = 0);
   @FOR(TRLM1(I): X32(4,I,4) = 0);
   @FOR(TRLM1(I): X13(4,I,4) = 0);
   @FOR(TRLM1(J): X13(4,4,J) = 0);
   @FOR(TRLM1(H): X23(H,4,4) = 0);
   @FOR(TRLM1(J): X23(4,4,J) = 0);
   @FOR(TRLM1(H): X33(H,4,4) = 0);
   @FOR(TRLM1(I): X33(4,I,4) = 0);

   @FOR(TRL3P1(H,I,J): X11 >= X12(H,I,J));
   @FOR(TRL3P1(H,I,J): X21 >= X22(H,I,J));
   @FOR(TRL3P1(H,I,J): X31 >= X32(H,I,J));
   @FOR(TRL3P1(H,I,J): X11 >= X13(H,I,J));
   @FOR(TRL3P1(H,I,J): X21 >= X23(H,I,J));
   @FOR(TRL3P1(H,I,J): X31 >= X33(H,I,J));

   !Budget constraints: First time period;
   C1(1)*X11 + C1(2)*X21 + C1(3)*X31 <= B1;

   !Second time period;
   @FOR(TRL3(H,I,J): C2(1)*X12(H,I,J) + C2(2)*X22(H,I,J) + C2(3)*X32(H,I,J) <= B2 );
   @FOR(TRL2(H,I): C2(1)*X12(H,I,5) + C2(2)*X22(H,I,5) <= B2 );
   @FOR(TRL2(I,J): C2(2)*X22(5,I,J) + C2(3)*X32(5,I,J) <= B2 );
   @FOR(TRL2(H,J): C2(1)*X12(H,5,J) + C2(3)*X32(H,5,J) <= B2 );
   @FOR(TRL(H): C2(1)*X12(H,5,5) <= B2);
   @FOR(TRL(I): C2(2)*X22(5,I,5) <= B2);
   @FOR(TRL(J): C2(3)*X32(5,5,J) <= B2);

   !Third time period;
   @FOR(TRL3(H,I,J): C3(1)*X13(H,I,J) + C3(2)*X23(H,I,J) + C2(3)*X33(H,I,J) <= B3 );
   @FOR(TRL2(H,I): C3(1)*X13(H,I,5) + C3(2)*X23(H,I,5) <= B3 );
   @FOR(TRL2(I,J): C3(2)*X23(5,I,J) + C3(3)*X33(5,I,J) <= B3 );
   @FOR(TRL2(H,J): C3(1)*X13(H,5,J) + C3(3)*X33(H,5,J) <= B3 );
   @FOR(TRL(H): C3(1)*X13(H,5,5) <= B3);
   @FOR(TRL(I): C3(2)*X23(5,I,5) <= B3);
   @FOR(TRL(J): C3(3)*X33(5,5,J) <= B3);
```

```
   !W,Y,Z variables <= 1;
   @FOR(FUND: W <= 1);

   @FOR(TY1: Y1 <= 1);
   @FOR(TY2: Y2 <= 1);
   @FOR(TY2: Y3 <= 1);
   @FOR(TY2: Y4 <= 1);
   @FOR(TY3: Y5 <= 1);
   @FOR(TY3: Y6 <= 1);
   @FOR(TY3: Y7 <= 1);

   @FOR(TZ1: Z1 <= 1);
   @FOR(TZ2: Z2 <= 1);
   @FOR(TZ2: Z3 <= 1);
   @FOR(TZ2: Z4 <= 1);
   @FOR(TZ3: Z5 <= 1);
   @FOR(TZ3: Z6 <= 1);
   @FOR(TZ3: Z7 <= 1);
   @FOR(TZ5: Z9 <= 1);
   @FOR(TZ6: Z10 <= 1);
   @FOR(TZ6: Z11 <= 1);
   @FOR(TZ5: Z13 <= 1);
   @FOR(TZ6: Z14 <= 1);
   @FOR(TZ6: Z15 <= 1);
   @FOR(TZ5: Z17 <= 1);
   @FOR(TZ6: Z18 <= 1);
   @FOR(TZ6: Z19 <= 1);
   @FOR(TZ8: Z21 <= 1);
   @FOR(TZ8: Z23 <= 1);
   @FOR(TZ8: Z25 <= 1);

   !Xs are binary variables;
   @BIN(X11);
   @BIN(X21);
   @BIN(X31);
   @FOR(TRLP1M: @BIN(X12));
   @FOR(TRLP1M: @BIN(X22));
   @FOR(TRLP1M: @BIN(X32));
   @FOR(TRLP1M: @BIN(X13));
   @FOR(TRLP1M: @BIN(X23));
   @FOR(TRLP1M: @BIN(X33));

END
```

Appendix C: LINGO Code for Three-Project, Two-Time Period Mixed-Integer Nonlinear Program

```
model:

DATA:
BIGM = 10000.0;
nTRL = 4; ! 4 TRLs (5 - 8), start at 5, 8 is success;
nTRLP1 = 5;
nV = 3;   ! three projects to consider;
nTP = 2;  ! two time periods;
ENDDATA

SETS:
COST/1..nV/: C1, C2;     !Cost matrix for each project/funding level at each time period;
TRL/1..nTRL/:  FP11, FP21, FP31;  !FP'S are first stage probability matrices for project;
TRLP1/1..nTRLP1/;
TRLP1M(TRLP1,TRLP1,TRLP1): OC1, OC2, X12, X22, X32;
TRLMATRIX2(TRL,TRL): P12, P22, P32; !Second stage probability matrices for project;
TRLMATRIX3(TRL,TRL,TRL): T3;
ENDSETS


DATA:
FP11, FP21, FP31 =
   @OLE('C:\Maryland\OptionsResearch\IPformulation\Spring09\2Vendors2TPs_2.xls', 'fponeone',
'fptwoone', 'fpthreeone');
P12, P22, P32 =
   @OLE('C:\Maryland\OptionsResearch\IPformulation\Spring09\2Vendors2TPs_2.xls', 'ponetwo',
'ptwotwo', 'pthreetwo');
B1, B2, C1, C2 =
    @OLE('C:\Maryland\OptionsResearch\IPformulation\Spring09\2Vendors2TPs_2.xls', 'bud1',
'bud2', 'cost1', 'cost2');


ENDDATA



!objective function;
MAX = OC2(1,1,4) + OC2(1,2,4) + OC2(1,3,4) + OC2(1,4,4) + OC2(1,5,4) + OC2(2,1,4) + OC2(2,2,4)
+ OC2(2,3,4) + OC2(2,4,4) + OC2(2,5,4) + OC2(3,1,4) + OC2(3,2,4) + OC2(3,3,4) + OC2(3,4,4) +
OC2(3,5,4) + OC2(4,1,4) + OC2(4,2,4) + OC2(4,3,4) + OC2(4,4,4) + OC2(4,5,4) + OC2(5,1,4) +
OC2(5,2,4) + OC2(5,3,4) + OC2(5,4,4) + OC2(5,5,4) + OC2(1,4,5) + OC2(2,4,5) + OC2(3,4,5) +
OC2(5,4,5) + OC2(1,4,3) + OC2(2,4,3) + OC2(3,4,3) + OC2(5,4,3) + OC2(1,4,2) + OC2(2,4,2) +
OC2(3,4,2) + OC2(5,4,2) + OC2(1,4,1) + OC2(2,4,1) + OC2(3,4,1) + OC2(5,4,1) + OC2(4,1,1) +
OC2(4,1,2) + OC2(4,1,3) + OC2(4,1,5) + OC2(4,2,1) + OC2(4,2,2) + OC2(4,2,3) + OC2(4,2,5) +
OC2(4,3,1) + OC2(4,3,2) + OC2(4,3,3) + OC2(4,3,5) + OC2(4,5,1) + OC2(4,5,2) + OC2(4,5,3) +
OC2(4,5,5) + OC2(4,4,1) + OC2(4,4,2) + OC2(4,4,3) + OC2(4,4,5);

!subject to;

!Second funding stage decisions/outcomes;
   @FOR(TRLMATRIX3(K,L,M): OC2(K,L,M) = @SUM(TRLMATRIX3(H,I,J):
FP11(H)*FP21(I)*FP31(J)*P12(H,K)*P22(I,L)*P32(J,M)*X11*X21*X31*X12(H,I,J)*X22(H,I,J)*X32(H,I,J)
)+ @SUM(TRLMATRIX2(H,I):
FP11(H)*FP21(I)*FP31(M)*P12(H,K)*P22(I,L)*X11*X21*X31*X12(H,I,M)*X22(H,I,M)*(1-X32(H,I,M)))+
@SUM(TRLMATRIX2(H,J):  FP11(H)*FP21(L)*FP31(J)*P12(H,K)*P32(J,M)*X11*X21*X31*X12(H,L,J)*(1-
X22(H,L,J))*X32(H,L,J))+ @SUM(TRLMATRIX2(I,J):
FP11(K)*FP21(I)*FP31(J)*P22(I,L)*P32(J,M)*X11*X21*X31*(1-X12(K,I,J))*X22(K,I,J)*X32(K,I,J))+
@SUM(TRL(H): FP11(H)*FP21(L)*FP31(M)*P12(H,K)*X11*X21*X31*X12(H,L,M)*(1-X22(H,L,M))*(1-
X32(H,L,M)))+
@SUM(TRL(I):  FP11(K)*FP21(I)*FP31(M)*P22(I,L)*X11*X21*X31*(1-X12(K,I,M))*X22(K,I,M)*(1-
X32(K,I,M)))+ @SUM(TRL(J):  FP11(K)*FP21(L)*FP31(J)*P32(J,M)*X11*X21*X31*(1-X12(K,L,J))*(1-
X22(K,L,J))*X32(K,L,J))+ FP11(K)*FP21(L)*FP31(M)*X11*X21*X31*(1-X12(K,L,M))*(1-X22(K,L,M))*(1-
X32(K,L,M)) );
```

```
@FOR(TRL(J): OC2(J,5,5) = @SUM(TRL(I): FP11(I)*P12(I,J)*X11*(1-X21)*(1-X31)*X12(I,5,5)) +
FP11(J)*X11*(1-X21)*(1-X31)*(1-X12(J,5,5)) );

@FOR(TRL(J): OC2(5,J,5) = @SUM(TRL(I): FP21(I)*P22(I,J)*(1-X11)*X21*(1-X31)*X22(5,I,5)) +
FP21(J)*(1-X11)*X21*(1-X31)*(1-X22(5,J,5)) );

@FOR(TRL(J): OC2(5,5,J) = @SUM(TRL(I): FP31(I)*P32(I,J)*(1-X11)*(1-X21)*X31*X32(5,5,I)) +
FP31(J)*(1-X11)*(1-X21)*X31*(1-X32(5,5,J)) );

@FOR(TRLMATRIX2(J,K): OC2(J,K,5) = @SUM(TRLMATRIX2(H,I):
FP11(H)*FP21(I)*P12(H,J)*P22(I,K)*X11*X21*(1-X31)*X12(H,I,5)*X22(H,I,5)) +
@SUM(TRL(H): FP11(H)*FP21(K)*P12(H,J)*X11*X21*(1-X31)*X12(H,K,5)*(1-X22(H,K,5))) +
@SUM(TRL(I): FP11(J)*FP21(I)*P22(I,K)*X11*X21*(1-X31)*(1-X12(J,I,5))*X22(J,I,5)) +
FP11(J)*FP21(K)*X11*X21*(1-X31)*(1-X12(J,K,5))*(1-X22(J,K,5)) );

@FOR(TRLMATRIX2(J,K): OC2(J,5,K) = @SUM(TRLMATRIX2(H,I):
FP11(H)*FP31(I)*P12(H,J)*P32(I,K)*X11*(1-X21)*X31*X12(H,5,I)*X32(H,5,I)) +
@SUM(TRL(H): FP11(H)*FP31(K)*P12(H,J)*X11*(1-X21)*X31*X12(H,5,K)*(1-X32(H,5,K))) +
@SUM(TRL(I): FP11(J)*FP31(I)*P32(I,K)*X11*(1-X21)*X31*(1-X12(J,5,I))*X32(J,5,I)) +
FP11(J)*FP31(K)*X11*(1-X21)*X31*(1-X12(J,5,K))*(1-X32(J,5,K)) );

@FOR(TRLMATRIX2(J,K): OC2(5,J,K) = @SUM(TRLMATRIX2(H,I): FP21(H)*FP31(I)*P22(H,J)*P32(I,K)*(1-
X11)*X21*X31*X22(5,H,I)*X32(5,H,I)) + @SUM(TRL(H): FP21(H)*FP31(K)*P22(H,J)*(1-
X11)*X21*X31*X22(5,H,K)*(1-X32(5,H,K))) + @SUM(TRL(I): FP21(J)*FP31(I)*P32(I,K)*(1-
X11)*X21*X31*(1-X22(5,J,I))*X32(5,J,I)) + FP21(J)*FP31(K)*(1-X11)*X21*X31*(1-X22(5,J,K))*(1-
X32(5,J,K)) );

!A project can only be funded in the second period if it is funded in the first;
@FOR(TRLP1M(I,J,K): X11 >= X12(I,J,K));
@FOR(TRLP1M(I,J,K): X21 >= X22(I,J,K));
@FOR(TRLP1M(I,J,K): X31 >= X32(I,J,K));

!Budget constraints;
C1(1)*X11 + C1(2)*X21 + C1(3)*X31 <= B1;
@FOR(TRLMATRIX3(I,J,K): C2(1)*X12(I,J,K) + C2(2)*X22(I,J,K) +  C2(3)*X32(I,J,K) <= B2 );
@FOR(TRLMATRIX2(I,J): C2(2)*X22(5,I,J) + C2(3)*X32(5,I,J) <= B2 );
@FOR(TRLMATRIX2(I,J): C2(1)*X12(I,5,J) + C2(3)*X32(I,5,J) <= B2 );
@FOR(TRLMATRIX2(I,J): C2(1)*X12(I,J,5) + C2(2)*X22(I,J,5) <= B2 );

!Branching on first time period binary variables;
X11=1;
X21=1;
X31=0;

!Xs are binary variables;
@BIN(X11);
@BIN(X21);
@BIN(X31);
@FOR(TRLP1M: @BIN(X12));
@FOR(TRLP1M: @BIN(X22));
@FOR(TRLP1M: @BIN(X32));
;
```

Appendix D: LINGO Code for Three-Project, Two-Time Period Integer Program

```
model:

DATA:
BIGM = 10000.0;
nTRL = 4; ! 4 TRLs (5 - 8), start at 5, 8 is success;
nTRLP1 = 5;
nV = 3;   ! three projects to consider;
nTP = 2;  ! two time periods;
nF = 8; ! eight funding decision combinations;
ENDDATA

SETS:
COST/1..nV/: C1, C2;     !Cost matrix for each project/funding level at each time period;
TRL/1..nTRL/:  FP11, FP21, FP31;  !FP'S are first stage probability matrices for project;
TRLP1/1..nTRLP1/;
TP/1..nTP/;
FUND/1..nF/: Y;
TRLP1M(TRLP1,TRLP1,TRLP1): OC1, OC2, X12, X22, X32;
TRLMATRIX2(TRL,TRL): P12, P22, P32; !Second stage probability matrices for project;
TRLMATRIX3(TRL,TRL,TRL): ZZZZ;
TRLMATRIX2F(TRL,TRL,TRL): Z2, Z3, Z4;
TRLMATRIX1F(TRL,TP): Z5, Z6, Z7;
TRLMATRIX4(TRL,TRL,TRL,FUND): Z1;

ENDSETS


DATA:
FP11, FP21, FP31 =
   @OLE('C:\Maryland\OptionsResearch\IPformulation\Spring09\3Vendors2TPs_2.xls', 'fponeone',
'fptwoone', 'fpthreeone');
P12, P22, P32 =
   @OLE('C:\Maryland\OptionsResearch\IPformulation\Spring09\3Vendors2TPs_2.xls', 'ponetwo',
'ptwotwo', 'pthreetwo');
B1, B2, C1, C2 =
    @OLE('C:\Maryland\OptionsResearch\IPformulation\Spring09\3Vendors2TPs_2.xls', 'bud1',
'bud2', 'cost1', 'cost2');

ENDDATA

!objective function;
MAX = OC2(1,1,4) + OC2(1,2,4) + OC2(1,3,4) + OC2(1,4,4) + OC2(1,5,4) + OC2(2,1,4) + OC2(2,2,4)
+ OC2(2,3,4) + OC2(2,4,4) + OC2(2,5,4) +
      OC2(3,1,4) + OC2(3,2,4) + OC2(3,3,4) + OC2(3,4,4) + OC2(3,5,4) + OC2(4,1,4) + OC2(4,2,4)
+ OC2(4,3,4) + OC2(4,4,4) + OC2(4,5,4) +
      OC2(5,1,4) + OC2(5,2,4) + OC2(5,3,4) + OC2(5,4,4) + OC2(5,5,4) + OC2(1,4,5) + OC2(2,4,5)
+ OC2(3,4,5) + OC2(5,4,5) + OC2(1,4,3) +
  OC2(2,4,3) + OC2(3,4,3) + OC2(5,4,3) + OC2(1,4,2) + OC2(2,4,2) + OC2(3,4,2) + OC2(5,4,2) +
OC2(1,4,1) + OC2(2,4,1) + OC2(3,4,1) +
  OC2(5,4,1) + OC2(4,1,1) + OC2(4,1,2) + OC2(4,1,3) + OC2(4,1,5) + OC2(4,2,1) + OC2(4,2,2) +
OC2(4,2,3) + OC2(4,2,5) + OC2(4,3,1) +
  OC2(4,3,2) + OC2(4,3,3) + OC2(4,3,5) + OC2(4,5,1) + OC2(4,5,2) + OC2(4,5,3) + OC2(4,5,5) +
OC2(4,4,1) + OC2(4,4,2) + OC2(4,4,3) +
  OC2(4,4,5);

!subject to;

   !Branch on the first time period binary variables;
   X11=0;
   X21=1;
   X31=1;

   !Second funding stage decisions/outcomes;
   @FOR(TRLMATRIX3(K,L,M): OC2(K,L,M) =
```

```
          @SUM(TRLMATRIX3(H,I,J):
FP11(H)*FP21(I)*FP31(J)*P12(H,K)*P22(I,L)*P32(J,M)*Z1(H,I,J,1))+
          @SUM(TRLMATRIX2(H,I):  FP11(H)*FP21(I)*FP31(M)*P12(H,K)*P22(I,L)*Z1(H,I,M,2))+

          @SUM(TRLMATRIX2(H,J):  FP11(H)*FP21(L)*FP31(J)*P12(H,K)*P32(J,M)*Z1(H,L,J,3))+
          @SUM(TRLMATRIX2(I,J):  FP11(K)*FP21(I)*FP31(J)*P22(I,L)*P32(J,M)*Z1(K,I,J,4))+
          @SUM(TRL(H):  FP11(H)*FP21(L)*FP31(M)*P12(H,K)*Z1(H,L,M,5))+
          @SUM(TRL(I):  FP11(K)*FP21(I)*FP31(M)*P22(I,L)*Z1(K,I,M,6))+
          @SUM(TRL(J):  FP11(K)*FP21(L)*FP31(J)*P32(J,M)*Z1(K,L,J,7))+
          FP11(K)*FP21(L)*FP31(M)*Z1(K,L,M,8) );

   @FOR(TRL(J): OC2(J,5,5) = @SUM(TRL(I): FP11(I)*P12(I,J)*Z5(I,1)) + FP11(J)*Z5(J,2) );
   @FOR(TRL(J): OC2(5,J,5) = @SUM(TRL(I): FP21(I)*P22(I,J)*Z6(I,1)) + FP21(J)*Z6(J,2) );
   @FOR(TRL(J): OC2(5,5,J) = @SUM(TRL(I): FP31(I)*P32(I,J)*Z7(I,1)) + FP31(J)*Z7(J,2) );

   @FOR(TRLMATRIX2(J,K): OC2(J,K,5) = @SUM(TRLMATRIX2(H,I):
FP11(H)*FP21(I)*P12(H,J)*P22(I,K)*Z2(H,I,1)) +
                                          @SUM(TRL(H): FP11(H)*FP21(K)*P12(H,J)*Z2(H,K,2)) +
                                          @SUM(TRL(I): FP11(J)*FP21(I)*P22(I,K)*Z2(J,I,3)) +
                                          FP11(J)*FP21(K)*Z2(J,K,4) );

   @FOR(TRLMATRIX2(J,K): OC2(J,5,K) = @SUM(TRLMATRIX2(H,I):
FP11(H)*FP31(I)*P12(H,J)*P32(I,K)*Z3(H,I,1)) +
                                          @SUM(TRL(H): FP11(H)*FP31(K)*P12(H,J)*Z3(H,K,2)) +
                                          @SUM(TRL(I): FP11(J)*FP31(I)*P32(I,K)*Z3(J,I,3)) +
                                          FP11(J)*FP31(K)*Z3(J,K,4) );

   @FOR(TRLMATRIX2(J,K): OC2(5,J,K) = @SUM(TRLMATRIX2(H,I):
FP21(H)*FP31(I)*P22(H,J)*P32(I,K)*Z4(H,I,1)) +
                                          @SUM(TRL(H): FP21(H)*FP31(K)*P22(H,J)*Z4(H,K,2)) +
                                          @SUM(TRL(I): FP21(J)*FP31(I)*P32(I,K)*Z4(J,I,3)) +
                                          FP21(J)*FP31(K)*Z3(J,K,4) );

   @FOR(FUND(I): Y(I) <= 1);

   @FOR(TRLMATRIX3(H,I,J):
     Z1(H,I,J,1) <= X11;
     Z1(H,I,J,1) <= X21;
     Z1(H,I,J,1) <= X31;
     Z1(H,I,J,1) <= X12(H,I,J);
     Z1(H,I,J,1) <= X22(H,I,J);
     Z1(H,I,J,1) <= X32(H,I,J);
     X11 + X21 + X31 + X12(H,I,J) + X22(H,I,J) + X32(H,I,J) - Z1(H,I,J,1) <= 5 );

   @FOR(TRLMATRIX3(H,I,J):
     Z1(H,I,J,2) <= X11;
     Z1(H,I,J,2) <= X21;
     Z1(H,I,J,2) <= X31;
     Z1(H,I,J,2) <= X12(H,I,J);
     Z1(H,I,J,2) <= X22(H,I,J);
     Z1(H,I,J,2) <= (1-X32(H,I,J));
     X11 + X21 + X31 + X12(H,I,J) + X22(H,I,J) + (1-X32(H,I,J)) - Z1(H,I,J,2) <= 5 );

   @FOR(TRLMATRIX3(H,I,J):
     Z1(H,I,J,3) <= X11;
     Z1(H,I,J,3) <= X21;
     Z1(H,I,J,3) <= X31;
     Z1(H,I,J,3) <= X12(H,I,J);
     Z1(H,I,J,3) <= (1-X22(H,I,J));
     Z1(H,I,J,3) <= X32(H,I,J);
     X11 + X21 + X31 + X12(H,I,J) + (1-X22(H,I,J)) + X32(H,I,J) - Z1(H,I,J,3) <= 5 );

   @FOR(TRLMATRIX3(H,I,J):
     Z1(H,I,J,4) <= X11;
     Z1(H,I,J,4) <= X21;
     Z1(H,I,J,4) <= X31;
     Z1(H,I,J,4) <= (1-X12(H,I,J));
     Z1(H,I,J,4) <= X22(H,I,J);
     Z1(H,I,J,4) <= X32(H,I,J);
     X11 + X21 + X31 +  (1-X12(H,I,J)) + X22(H,I,J) + X32(H,I,J) - Z1(H,I,J,4) <= 5 );
```

```
@FOR(TRLMATRIX3(H,I,J):
  Z1(H,I,J,5) <= X11;
  Z1(H,I,J,5) <= X21;
  Z1(H,I,J,5) <= X31;
  Z1(H,I,J,5) <= X12(H,I,J);
  Z1(H,I,J,5) <= (1-X22(H,I,J));
  Z1(H,I,J,5) <= (1-X32(H,I,J));
  X11 + X21 + X31 +  X12(H,I,J) + (1-X22(H,I,J)) + (1-X32(H,I,J)) - Z1(H,I,J,5) <= 5 );

@FOR(TRLMATRIX3(H,I,J):
  Z1(H,I,J,6) <= X11;
  Z1(H,I,J,6) <= X21;
  Z1(H,I,J,6) <= X31;
  Z1(H,I,J,6) <= (1-X12(H,I,J));
  Z1(H,I,J,6) <= X22(H,I,J);
  Z1(H,I,J,6) <= (1-X32(H,I,J));
  X11 + X21 + X31 +  (1-X12(H,I,J)) + X22(H,I,J) + (1-X32(H,I,J)) - Z1(H,I,J,6) <= 5 );

@FOR(TRLMATRIX3(H,I,J):
  Z1(H,I,J,7) <= X11;
  Z1(H,I,J,7) <= X21;
  Z1(H,I,J,7) <= X31;
  Z1(H,I,J,7) <= (1-X12(H,I,J));
  Z1(H,I,J,7) <= (1-X22(H,I,J));
  Z1(H,I,J,7) <= X32(H,I,J);
  X11 + X21 + X31 +  (1-X12(H,I,J)) + (1-X22(H,I,J)) + X32(H,I,J) - Z1(H,I,J,7) <= 5 );

@FOR(TRLMATRIX3(H,I,J):
  Z1(H,I,J,8) <= X11;
  Z1(H,I,J,8) <= X21;
  Z1(H,I,J,8) <= X31;
  Z1(H,I,J,8) <= (1-X12(H,I,J));
  Z1(H,I,J,8) <= (1-X22(H,I,J));
  Z1(H,I,J,8) <= (1-X32(H,I,J));
  X11 + X21 + X31 +  (1-X12(H,I,J)) + (1-X22(H,I,J)) + (1-X32(H,I,J)) - Z1(H,I,J,8) <= 5 );

Y(1) <= X11;
Y(1) <= X21;
Y(1) <= X31;
X11 + X21 + X31 - Y(1) <= 2;

Y(2) <= X11;
Y(2) <= X21;
Y(2) <= (1-X31);
X11 + X21 + (1-X31) - Y(2) <= 2;

@FOR(TRLMATRIX2F(I,J,K): Z2(I,J,K) <= Y(2));
@FOR(TRLMATRIX2(I,J): Z2(I,J,1) <= X12(I,J,5) );
@FOR(TRLMATRIX2(I,J): Z2(I,J,1) <= X22(I,J,5) );
@FOR(TRLMATRIX2(I,J): Z2(I,J,2) <= X12(I,J,5) );
@FOR(TRLMATRIX2(I,J): Z2(I,J,2) <= (1-X22(I,J,5)) );
@FOR(TRLMATRIX2(I,J): Z2(I,J,3) <= (1-X12(I,J,5)) );
@FOR(TRLMATRIX2(I,J): Z2(I,J,3) <= X22(I,J,5) );
@FOR(TRLMATRIX2(I,J): Z2(I,J,4) <= (1-X12(I,J,5)) );
@FOR(TRLMATRIX2(I,J): Z2(I,J,4) <= (1-X22(I,J,5)) );

Y(3) <= X11;
Y(3) <= (1-X21);
Y(3) <= X31;
X11 + X21 + X31 - Y(3) <= 2;

@FOR(TRLMATRIX2F(I,J,K): Z3(I,J,K) <= Y(3));
@FOR(TRLMATRIX2(I,J): Z3(I,J,1) <= X12(I,5,J) );
@FOR(TRLMATRIX2(I,J): Z3(I,J,1) <= X32(I,5,J) );
@FOR(TRLMATRIX2(I,J): Z3(I,J,2) <= X12(I,5,J) );
@FOR(TRLMATRIX2(I,J): Z3(I,J,2) <= (1-X32(I,5,J)) );
@FOR(TRLMATRIX2(I,J): Z3(I,J,3) <= (1-X12(I,5,J)) );
@FOR(TRLMATRIX2(I,J): Z3(I,J,3) <= X32(I,5,J) );
@FOR(TRLMATRIX2(I,J): Z3(I,J,4) <= (1-X12(I,5,J)) );
@FOR(TRLMATRIX2(I,J): Z3(I,J,4) <= (1-X32(I,5,J)) );
```

```
Y(4) <= (1-X11);
Y(4) <= X21;
Y(4) <= X31;
(1-X11) + X21 + X31 - Y(4) <= 2;

@FOR(TRLMATRIX2F(I,J,K): Z4(I,J,K) <= Y(4));
@FOR(TRLMATRIX2(I,J): Z4(I,J,1) <= X22(5,I,J) );
@FOR(TRLMATRIX2(I,J): Z4(I,J,1) <= X32(5,I,J) );
@FOR(TRLMATRIX2(I,J): Z4(I,J,2) <= X22(5,I,J) );
@FOR(TRLMATRIX2(I,J): Z4(I,J,2) <= (1-X32(5,I,J)) );
@FOR(TRLMATRIX2(I,J): Z4(I,J,3) <= (1-X22(5,I,J)) );
@FOR(TRLMATRIX2(I,J): Z4(I,J,3) <= X32(5,I,J) );
@FOR(TRLMATRIX2(I,J): Z4(I,J,4) <= (1-X22(5,I,J)) );
@FOR(TRLMATRIX2(I,J): Z4(I,J,4) <= (1-X32(5,I,J)) );

Y(5) <= X11;
Y(5) <= (1-X21);
Y(5) <= (1-X31);
X11 + (1-X21) + (1-X31) - Y(5) <= 2;

@FOR(TRLMATRIX1F(I,J): Z5(I,J) <= Y(5));
@FOR(TRL(I): Z5(I,1) <= X12(I,5,5) );
@FOR(TRL(I): Z5(I,2) <= (1-X12(I,5,5)) );

Y(6) <= (1-X11);
Y(6) <= X21;
Y(6) <= (1-X31);
(1-X11) + X21 + (1-X31) - Y(6) <= 2;

@FOR(TRLMATRIX1F(I,J): Z6(I,J) <= Y(6));
@FOR(TRL(I): Z6(I,1) <= X22(5,I,5) );
@FOR(TRL(I): Z6(I,2) <= (1-X22(5,I,5)) );

Y(7) <= (1-X11);
Y(7) <= (1-X21);
Y(7) <= X31;
(1-X11) + (1-X21) + X31 - Y(7) <= 2;

@FOR(TRLMATRIX1F(I,J): Z7(I,J) <= Y(7));
@FOR(TRL(I): Z7(I,1) <= X32(5,5,I) );
@FOR(TRL(I): Z7(I,2) <= (1-X32(5,5,I)) );


!A project can only be funded in the second period if it is funded in the first;
@FOR(TRLP1M(I,J,K): X11 >= X12(I,J,K));
@FOR(TRLP1M(I,J,K): X21 >= X22(I,J,K));
@FOR(TRLP1M(I,J,K): X31 >= X32(I,J,K));

!Budget constraints;
C1(1)*X11 + C1(2)*X21 + C1(3)*X31 <= B1;
@FOR(TRLMATRIX3(I,J,K): C2(1)*X12(I,J,K) + C2(2)*X22(I,J,K) +  C2(3)*X32(I,J,K) <= B2 );
@FOR(TRLMATRIX2(I,J): C2(2)*X22(5,I,J) + C2(3)*X32(5,I,J) <= B2 );
@FOR(TRLMATRIX2(I,J): C2(1)*X12(I,5,J) + C2(3)*X32(I,5,J) <= B2 );
@FOR(TRLMATRIX2(I,J): C2(1)*X12(I,J,5) + C2(2)*X22(I,J,5) <= B2 );

!Xs are binary variables;
@BIN(X11);
@BIN(X21);
@BIN(X31);
@FOR(TRLP1M: @BIN(X12));
@FOR(TRLP1M: @BIN(X22));
@FOR(TRLP1M: @BIN(X32));

END
```

Appendix E: Costs, Budgets, and Transition Probabilities for the Five Sample Problems in Section 4.2.3

This appendix includes all data for the five sample problems in Section 4.2.3 which are used to compare the results and run-times of the MINLP model with locally optimal solutions with linearized IP model with globally optimal solutions. All data are identical between the fixed and optimized budget allocations. The only difference is that for the optimized allocation, the sum of the two budgets is allocated in the optimal way; however, the total budgets are the same.

Table E1: Problem 1's Costs and Budgets

Project	Time Period 1	Time Period 2
Project 1	$ 4.7	$ 3.0
Project 2	$ 6.8	$ 4.2
Project 3	$ 5.8	$ 3.5
Budget	$ 15.0	$ 5.0

Table E2: Problem 2's Costs and Budgets

Project	Time Period 1	Time Period 2
Project 1	$ 4.7	$ 3.0
Project 2	$ 6.8	$ 4.2
Project 3	$ 5.8	$ 3.5
Budget	$ 12.0	$ 4.0

Table E3: Problem 3's Costs and Budgets

Project	Time Period 1	Time Period 2
Project 1	$ 4.7	$ 3.0
Project 2	$ 5.5	$ 4.8
Project 3	$ 5.8	$ 3.5
Budget	$ 13.0	$ 5.0

Table E4: Problem 4's Costs and Budgets

Project	Time Period 1	Time Period 2
Project 1	$ 5.1	$ 3.0
Project 2	$ 3.8	$ 2.6
Project 3	$ 4.5	$ 3.5
Budget	$ 13.0	$ 6.0

Table E5: Problem 5's Costs and Budgets

Project	Time Period 1	Time Period 2
Project 1	$ 5.1	$ 3.0
Project 2	$ 3.4	$ 2.6
Project 3	$ 4.5	$ 3.5
Budget	$ 12.0	$ 4.0

Table E6: First Stage Transition Probabilities for Problems 1, 2, and 3

Project	State	Prob
Project 1	1	0.30
	2	0.40
	3	0.25
	4	0.05
Project 2	1	0.40
	2	0.40
	3	0.15
	4	0.05
Project 2	1	0.35
	2	0.35
	3	0.25
	4	0.05

Table E7: First Stage Transition Probabilities for Problems 4 and 5

Project	State	Prob
Project 1	1	0.30
	2	0.30
	3	0.30
	4	0.10
Project 2	1	0.30
	2	0.50
	3	0.15
	4	0.05
Project 2	1	0.30
	2	0.40
	3	0.25
	4	0.05

Table E8: Second Stage Transition Probabilities for Problems 1 and 2

Project 1	State 1	State 2	State 3	State 4
State 1	0.30	0.20	0.50	0.00
State 2	0.00	0.20	0.70	0.10
State 3	0.00	0.00	0.35	0.65
State 4	0.00	0.00	0.00	1.00

Project 2	State 1	State 2	State 3	State 4
State 1	0.40	0.30	0.15	0.15
State 2	0.00	0.30	0.35	0.35
State 3	0.00	0.00	0.45	0.55
State 4	0.00	0.00	0.00	1.00

Project 3	State 1	State 2	State 3	State 4
State 1	0.35	0.35	0.15	0.15
State 2	0.00	0.30	0.40	0.30
State 3	0.00	0.00	0.60	0.40
State 4	0.00	0.00	0.00	1.00

Table E9: Second Stage Transition Probabilities for Problems 3 and 4

Project 1	State 1	State 2	State 3	State 4
State 1	0.40	0.50	0.10	0.00
State 2	0.00	0.20	0.66	0.14
State 3	0.00	0.00	0.40	0.60
State 4	0.00	0.00	0.00	1.00

Project 2	State 1	State 2	State 3	State 4
State 1	0.40	0.30	0.15	0.15
State 2	0.00	0.30	0.40	0.30
State 3	0.00	0.00	0.40	0.60
State 4	0.00	0.00	0.00	1.00

Project 3	State 1	State 2	State 3	State 4
State 1	0.35	0.35	0.15	0.15
State 2	0.00	0.30	0.40	0.30
State 3	0.00	0.00	0.65	0.35
State 4	0.00	0.00	0.00	1.00

Table E10: Second Stage Transition Probabilities for Problem 5

Project 1	State 1	State 2	State 3	State 4
State 1	0.40	0.40	0.05	0.15
State 2	0.00	0.20	0.60	0.20
State 3	0.00	0.00	0.40	0.60
State 4	0.00	0.00	0.00	1.00

Project 2	State 1	State 2	State 3	State 4
State 1	0.40	0.30	0.15	0.15
State 2	0.00	0.30	0.20	0.50
State 3	0.00	0.00	0.20	0.80
State 4	0.00	0.00	0.00	1.00

Project 3	State 1	State 2	State 3	State 4
State 1	0.35	0.35	0.15	0.15
State 2	0.00	0.30	0.40	0.30
State 3	0.00	0.00	0.65	0.35
State 4	0.00	0.00	0.00	1.00

Appendix F: State, Cost and Probability Data for an Oxyfuel CCS Project

This appendix contains cost and transition probabilities for the "Oxy 1" project from Chapter 5. The state structure follows the definitions shown in Figure 5.3.

Table F1: State, Cost and Transition Probability Data for "Oxy 1" for 2010 and 2012

				No spillover		Spillover	
Time Period	State	End Period	Third Party	Funding [mil Euro]	Probability	Funding [mil Euro]	Probability
2010	1,2,0,na,na -->2,2,na	2014	no	160	1.00	160	1.00
	1,2,0,na,na -->2,2,na	2014	no	320	1.00	320	1.00
	1,2,0,na,na --> 2,1,x,0,1	2014	no	160	0.00	160	0.00
	1,2,0,na,na --> 2,1,x,0,1	2014	no	320	0.00	320	0.00
	1,2,0,na,na --> 3,2,na,0	2014	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,2,na,0	2014	no	990	0.57	990	0.57
	1,2,0,na,na --> 3,1,x,0,0	2014	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,1,x,0,0	2014	no	990	0.43	990	0.43
2012	1,2,0,na,na -->2,2,na	2016	ps	160	1.00	160	1.00
	1,2,0,na,na -->2,2,na	2016	ps	320	1.00	320	1.00
	1,2,0,na,na --> 2,1,x,0,1	2016	ps	160	0.00	160	0.00
	1,2,0,na,na --> 2,1,x,0,1	2016	ps	320	0.00	320	0.00
	1,2,0,na,na --> 3,2,na,0	2016	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,2,na,0	2016	no	990	0.57	990	0.57
	1,2,0,na,na --> 3,1,x,0,0	2016	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,1,x,0,0	2016	no	990	0.43	990	0.43

Table F2: State, Cost and Transition Probability Data for "Oxy 1" for 2014

				No spillover		Spillover	
Time Period	State	End Period	Third Party	Funding [mil Euro]	Probability	Funding [mil Euro]	Probability
2014	1,2,0,na,na -->2,2,na	2018	no	160	1.00	160	1.00
	1,2,0,na,na -->2,2,na	2018	no	320	1.00	320	1.00
	1,2,0,na,na --> 2,1,x,0,1	2018	no	160	0.00	160	0.00
	1,2,0,na,na --> 2,1,x,0,1	2018	no	320	0.00	320	0.00
	1,2,2,na,na --> 2,2,na	2018	post	160	1.00	158	1.00
	1,2,2,na,na --> 2,2,na	2018	post	320	1.00	315	1.00
	1,2,2,na,na --> 2,1,x,0,1	2018	post	160	0.00	158	0.00
	1,2,2,na,na --> 2,1,x,0,1	2018	post	320	0.00	315	0.00
	1,2,0,na,na --> 3,2,na	2018	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,2,na	2018	no	990	0.57	990	0.57
	1,2,0,na,na --> 3,1,x,0,0	2018	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,1,x,0,0	2018	no	990	0.43	990	0.43
	1,2,2,na,na --> 3,2,na	2018	post	495	0.50	488	0.50
	1,2,2,na,na --> 3,2,na	2018	post	990	0.57	975	0.57
	1,2,2,na,na --> 3,1,x,0,0	2018	post	495	0.50	488	0.50
	1,2,2,na,na --> 3,1,x,0,0	2018	post	990	0.43	975	0.43
	2,2,0,na,na --> 3,2,na	2018	no	450	0.60	450	0.60
	2,2,0,na,na --> 3,2,na	2018	no	890	0.68	890	0.68
	2,2,0,na,na --> 3,1,x,0,1	2018	no	450	0.40	450	0.40
	2,2,0,na,na --> 3,1,x,0,1	2018	no	890	0.32	890	0.32
	2,1,0,0,1 --> 2,2,na	2016	no	160	0.86	160	0.86
	2,1,0,0,1 --> 2,2,na	2016	no	320	0.97	320	0.97
	2,1,2,0,1 --> 2,2,na	2016	post	160	0.86	158	0.86
	2,1,2,0,1 --> 2,2,na	2016	post	320	0.97	315	0.97
	2,1,0,0,1 --> 2,1,x,1,1	2016	no	160	0.14	160	0.14
	2,1,0,0,1 --> 2,1,x,1,1	2016	no	320	0.03	320	0.03
	2,1,2,0,1 --> 2,1,x,1,1	2016	post	160	0.14	158	0.14
	2,1,2,0,1 --> 2,1,x,1,1	2016	post	320	0.03	315	0.03
	2,1,0,0,na --> 3,2,na	2018	no	495	0.50	495	0.50
	2,1,0,0,na --> 3,2,na	2018	no	990	0.57	990	0.57
	2,1,1,0,na --> 3,2,na	2018	oxy	495	0.50	475	0.55
	2,1,1,0,na --> 3,2,na	2018	oxy	990	0.57	950	0.63
	2,1,2,0,na --> 3,2,na	2018	post	495	0.50	488	0.50
	2,1,2,0,na --> 3,2,na	2018	post	990	0.57	975	0.57
	2,1,0,0,na --> 3,1,x,0 ,0	2018	no	495	0.50	495	0.50
	2,1,0,0,na --> 3,1,x,0 ,0	2018	no	990	0.43	990	0.43
	2,1,1,0,na --> 3,1,x,0 ,0	2018	oxy	495	0.50	475	0.45
	2,1,1,0,na --> 3,1,x,0 ,0	2018	oxy	990	0.43	950	0.37
	2,1,2,0,na --> 3,1,x,0 ,0	2018	post	495	0.50	488	0.50
	2,1,2,0,na --> 3,1,x,0 ,0	2018	post	990	0.43	975	0.43
	3,1,0,0,0 --> 3,2,na	2016	no	495	0.50	495	0.50
	3,1,0,0,0 --> 3,2,na	2016	no	990	0.57	990	0.57
	3,1,2,0,0 --> 3,2,na	2016	post	480	0.50	473	0.50
	3,1,2,0,0 --> 3,2,na	2016	post	960	0.57	946	0.57
	3,1,0,0,0 --> 3,1,x,1,0	2016	no	495	0.50	495	0.50
	3,1,0,0,0 --> 3,1,x,1,0	2016	no	990	0.43	990	0.43
	3,1,2,0,0 --> 3,1,x,1,0	2016	post	480	0.50	473	0.50
	3,1,2,0,0 --> 3,1,x,1,0	2016	post	960	0.43	946	0.43

Table F3: State, Cost and Transition Probability Data for "Oxy 1" for 2016

				No spillover		Spillover	
Time Period	State	End Period	Third Party	Funding [mil Euro]	Probability	Funding [mil Euro]	Probability
2016	1,2,0,na,na --> 3,2,na	2020	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,2,na	2020	no	990	0.57	990	0.57
	1,2,0,na,na --> 3,1,x,0,0	2020	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,1,x,0,0	2020	no	990	0.43	990	0.43
	1,2,2,na,na --> 3,2,na	2020	post	495	0.50	488	0.50
	1,2,2,na,na --> 3,2,na	2020	post	990	0.57	975	0.57
	1,2,2,na,na --> 3,1,x,0,0	2020	post	495	0.50	488	0.50
	1,2,2,na,na --> 3,1,x,0,0	2020	post	990	0.43	975	0.43
	2,2,na --> 3,2,na	2020	ps	450	0.60	450	0.60
	2,2,na --> 3,2,na	2020	ps	890	0.68	890	0.68
	2,2,na --> 3,1,x,1,0	2020	ps	450	0.40	450	0.40
	2,2,na --> 3,1,x,1,0	2020	ps	890	0.32	890	0.32
	2,1,0,0,1 --> 2,2,na	2018	no	160	0.86	160	0.86
	2,1,0,0,1 --> 2,2,na	2018	no	320	0.97	320	0.97
	2,1,1,0,1 --> 2,2,na	2018	oxy-ps	160	0.86	160	0.86
	2,1,1,0,1 --> 2,2,na	2018	oxy-ps	320	0.97	320	0.97
	2,1,2,0,1 --> 2,2,na	2018	post	160	0.86	158	0.86
	2,1,2,0,1 --> 2,2,na	2018	post	320	0.97	315	0.97
	2,1,0,0,1 --> 2,1,x,1,1	2018	no	160	0.14	160	0.14
	2,1,0,0,1 --> 2,1,x,1,1	2018	no	320	0.03	320	0.03
	2,1,1,0,1 --> 2,1,x,1,1	2018	oxy-ps	160	0.14	160	0.14
	2,1,1,0,1 --> 2,1,x,1,1	2018	oxy-ps	320	0.03	320	0.03
	2,1,2,0,1 --> 2,1,x,1,1	2018	post	160	0.14	158	0.14
	2,1,2,0,1 --> 2,1,x,1,1	2018	post	320	0.03	315	0.03
	2,1,0,0,na --> 3,2,na	2020	no	495	0.50	495	0.50
	2,1,0,0,na --> 3,2,na	2020	no	990	0.57	990	0.57
	2,1,1,0,na --> 3,2,na	2020	oxy	495	0.50	475	0.55
	2,1,1,0,na --> 3,2,na	2020	oxy	990	0.57	950	0.63
	2,1,2,0,na --> 3,2,na	2020	post	495	0.50	488	0.50
	2,1,2,0,na --> 3,2,na	2020	post	990	0.57	975	0.57
	2,1,0,0,na --> 3,1,x,0 ,0	2020	no	495	0.50	495	0.50
	2,1,0,0,na --> 3,1,x,0 ,0	2020	no	990	0.43	990	0.43
	2,1,1,0,na --> 3,1,x,0,0	2020	oxy	495	0.50	475	0.45
	2,1,1,0,na --> 3,1,x,0,0	2020	oxy	990	0.43	950	0.37
	2,1,2,0,na --> 3,1,x,0 ,0	2020	post	495	0.50	488	0.50
	2,1,2,0,na -->32,1,x,0 ,0	2020	post	990	0.43	975	0.43
	3,1,0,0,0 --> 3,2,na	2018	no	495	0.50	495	0.50
	3,1,0,0,0 --> 3,2,na	2018	no	990	0.57	990	0.57
	3,1,2,0,0 --> 3,2,na	2018	post	480	0.50	473	0.50
	3,1,2,0,0 --> 3,2,na	2018	post	960	0.57	946	0.57
	3,1,0,0,0 --> 3,1,x,1,0	2018	no	495	0.50	495	0.50
	3,1,0,0,0 --> 3,1,x,1,0	2018	no	990	0.43	990	0.43
	3,1,2,0,0 --> 3,1,x,1,0	2018	post	480	0.50	473	0.50
	3,1,2,0,0 --> 3,1,x,1,0	2018	post	960	0.43	946	0.43
	3,1,0,1,0 --> 3,2,na	2018	no	495	0.50	495	0.50
	3,1,0,1,0 --> 3,2,na	2018	no	990	0.57	990	0.57
	3,1,2,1,0 --> 3,2,na	2018	post	480	0.50	473	0.50
	3,1,2,1,0 --> 3,2,na	2018	post	960	0.57	946	0.57
	3,1,0,1,0 -->3,1,x,2,na	2018	no	495	0.50	495	0.50
	3,1,0,1,0 -->3,1,x,2,na	2018	no	990	0.43	990	0.43
	3,1,2,1,0 --> 3,1,x,2,na	2018	post	480	0.50	473	0.50
	3,1,2,1,0 --> 3,1,x,2,na	2018	post	960	0.43	946	0.43

Table F4: State, Cost and Transition Probability Data for “Oxy 1” for 2018

				No spillover		Spillover	
Time Period	State	End Period	Third Party	Funding [mil Euro]	Probability	Funding [mil Euro]	Probability
2018	1,2,0,na,na --> 3,2,na	2022	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,2,na	2022	no	990	0.57	990	0.57
	1,2,0,na,na --> 3,1,x,0,0	2022	no	495	0.50	495	0.50
	1,2,0,na,na --> 3,1,x,0,0	2022	no	990	0.43	990	0.43
	1,2,2,na,na --> 3,2,na	2022	post	495	0.50	488	0.50
	1,2,2,na,na --> 3,2,na	2022	post	990	0.57	975	0.57
	1,2,2,na,na --> 3,1,x,0,0	2022	post	495	0.50	488	0.50
	1,2,2,na,na --> 3,1,x,0,0	2022	post	990	0.43	975	0.43
	2,2,na --> 3,2,na	2022	ps	450	0.60	450	0.60
	2,2,na --> 3,2,na	2022	ps	890	0.68	890	0.68
	2,2,na --> 3,1,x,0,1	2022	ps	450	0.40	450	0.40
	2,2,na --> 3,1,x,0,1	2022	ps	890	0.32	890	0.32
	2,1,0,0,1 --> 2,2,na	2020	no	160	0.86	160	0.86
	2,1,0,0,1 --> 2,2,na	2020	no	320	0.97	320	0.97
	2,1,1,0,1 --> 2,2,na	2020	oxy-ps	160	0.14	160	0.14
	2,1,1,0,1 --> 2,2,na	2020	oxy-ps	320	0.03	320	0.03
	2,1,2,0,1 --> 2,2,na	2020	post	160	0.86	158	0.86
	2,1,2,0,1 --> 2,2,na	2020	post	320	0.97	315	0.97
	2,1,0,0,1 --> 2,1,x,1,1	2020	no	160	0.14	160	0.14
	2,1,0,0,1 --> 2,1,x,1,1	2020	no	320	0.03	320	0.03
	2,1,1,0,1 --> 2,1,x,1,1	2020	oxy-ps	160	0.86	160	0.86
	2,1,1,0,1 --> 2,1,x,1,1	2020	oxy-ps	320	0.97	320	0.97
	2,1,2,0,1 --> 2,1,x,1,1	2020	post	160	0.14	158	0.14
	2,1,2,0,1 --> 2,1,x,1,1	2020	post	320	0.03	315	0.03
	2,1,0,0,na --> 3,2,na	2022	no	495	0.50	495	0.50
	2,1,0,0,na --> 3,2,na	2022	no	990	0.57	990	0.57
	2,1,1,0,na --> 3,2,na	2022	oxy	495	0.50	475	0.55
	2,1,1,0,na --> 3,2,na	2022	oxy	990	0.57	950	0.63
	2,1,2,0,na --> 3,2,na	2022	post	495	0.50	488	0.50
	2,1,2,0,na --> 3,2,na	2022	post	990	0.57	975	0.57
	2,1,0,0,na --> 3,1,x,0 ,0	2022	no	495	0.50	495	0.50
	2,1,0,0,na --> 3,1,x,0 ,0	2022	no	990	0.43	990	0.43
	2,1,1,0,na --> 3,1,x,0,0	2022	oxy	495	0.50	475	0.45
	2,1,1,0,na --> 3,1,x,0,0	2022	oxy	990	0.43	950	0.37
	2,1,2,0,na --> 2,1,x,0 ,0	2022	post	495	0.50	488	0.50
	2,1,2,0,na --> 2,1,x,0 ,0	2022	post	990	0.43	975	0.43
	3,1,0,0,0 --> 3,2,na	2020	no	495	0.50	495	0.50
	3,1,0,0,0 --> 3,2,na	2020	no	990	0.57	990	0.57
	3,1,2,0,0 --> 3,2,na	2020	post	495	0.50	488	0.50
	3,1,2,0,0 --> 3,2,na	2020	post	990	0.57	975	0.57
	3,1,0,0,0 --> 3,1,x,1,0	2020	no	495	0.50	495	0.50
	3,1,0,0,0 --> 3,1,x,1,0	2020	no	990	0.43	990	0.43
	3,1,2,0,0 --> 3,1,x,1,0	2020	post	495	0.50	488	0.50
	3,1,2,0,0 --> 3,1,x,1,0	2020	post	990	0.43	975	0.43
	3,1,0,1,0 --> 3,2,na	2020	no	495	0.50	495	0.50
	3,1,0,1,0 --> 3,2,na	2020	no	990	0.57	990	0.57
	3,1,2,1,0 --> 3,2,na	2020	post	495	0.50	488	0.50
	3,1,2,1,0 --> 3,2,na	2020	post	990	0.57	975	0.57
	3,1,0,1,0 -->3,1,x,2,na	2020	no	495	0.50	495	0.50
	3,1,0,1,0 -->3,1,x,2,na	2020	no	990	0.43	990	0.43
	3,1,2,1,0 --> 3,1,x,2,na	2020	post	495	0.50	488	0.50
	3,1,2,1,0 --> 3,1,x,2,na	2020	post	990	0.43	975	0.43
	3,1,0,0,1 --> 3,2,na	2020	no	446	0.60	446	0.60
	3,1,0,0,1 --> 3,2,na	2020	no	891	0.68	891	0.68
	3,1,2,0,1 --> 3,2,na	2020	post	446	0.60	439	0.60
	3,1,2,0,1 --> 3,2,na	2020	post	891	0.68	878	0.68
	3,1,0,0,1 -->3,1,x,1,1	2020	no	446	0.40	446	0.40
	3,1,0,0,1 -->3,1,x,1,1	2020	no	891	0.32	891	0.32
	3,1,2,0,1 --> 3,1,x,1,1	2020	post	446	0.40	439	0.40
	3,1,2,0,1 --> 3,1,x,1,1	2020	post	891	0.32	878	0.32

Table F5: State, Cost and Transition Probability Data for "Oxy 1" for 2020

				No spillover		Spillover	
Time Period	State	End Period	Third Party	Funding [mil Euro]	Probability	Funding [mil Euro]	Probability
2020	3,1,0,0,0 --> 3,2,na	2022	no	495	0.50	495	0.50
	3,1,0,0,0 --> 3,2,na	2022	no	990	0.57	990	0.57
	3,1,2,0,0 --> 3,2,na	2022	post	495	0.50	488	0.50
	3,1,2,0,0 --> 3,2,na	2022	post	990	0.57	975	0.57
	3,1,0,0,0 --> 3,1,x,1,0	2022	no	495	0.50	495	0.50
	3,1,0,0,0 --> 3,1,x,1,0	2022	no	990	0.43	990	0.43
	3,1,2,0,0 --> 3,1,x,1,0	2022	post	495	0.50	488	0.50
	3,1,2,0,0 --> 3,1,x,1,0	2022	post	990	0.43	975	0.43
	3,1,0,1,0 --> 3,2,na	2022	no	495	0.50	495	0.50
	3,1,0,1,0 --> 3,2,na	2022	no	990	0.57	990	0.57
	3,1,2,1,0 --> 3,2,na	2022	post	495	0.50	488	0.50
	3,1,2,1,0 --> 3,2,na	2022	post	990	0.57	975	0.57
	3,1,0,1,0 -->3,1,x,2,na	2022	no	495	0.50	495	0.50
	3,1,0,1,0 -->3,1,x,2,na	2022	no	990	0.43	990	0.43
	3,1,2,1,0 --> 3,1,x,2,na	2022	post	495	0.50	488	0.50
	3,1,2,1,0 --> 3,1,x,2,na	2022	post	990	0.43	975	0.43
	3,1,0,0,1--> 3,2,na	2022	no	446	0.60	446	0.60
	3,1,0,0,1--> 3,2,na	2022	no	891	0.68	891	0.68
	3,1,2,0,1 --> 3,2,na	2022	post	446	0.60	439	0.60
	3,1,2,0,1 --> 3,2,na	2022	post	891	0.68	878	0.68
	3,1,0,0,1-->3,1,x,1,1	2022	no	446	0.40	446	0.40
	3,1,0,0,1-->3,1,x,1,1	2022	no	891	0.32	891	0.32
	3,1,2,0,1 --> 3,1,x,1,1	2022	post	446	0.40	439	0.40
	3,1,2,0,1 --> 3,1,x,1,1	2022	post	891	0.32	878	0.32
	3,1,0,1,1--> 3,2,na	2022	no	446	0.60	446	0.60
	3,1,0,1,1--> 3,2,na	2022	no	891	0.68	891	0.68
	3,1,2,1,1 --> 3,2,na	2022	post	446	0.60	439	0.60
	3,1,2,1,1 --> 3,2,na	2022	post	891	0.68	878	0.68
	3,1,0,1,1--> 3,1,x,2,1	2022	no	446	0.40	446	0.40
	3,1,0,1,1--> 3,1,x,2,1	2022	no	891	0.32	891	0.32
	3,1,2,1,1 --> 3,2,x,2,1	2022	post	446	0.40	439	0.40
	3,1,2,1,1 --> 3,2,x,2,1	2022	post	891	0.32	878	0.32

Appendix G: Cost Coefficients Used for Each Budget Case for the Post-Combustion and Oxyfuel Projects

This appendix shows the cost coefficients used to construct the optimal *a priori* budgets for each funding case and time period for the post-combustion and oxyfuel projects. The reduction in combinations allows for the number of lower-level SDPs solve to remain manageable. The blank entries refer to those values that have been eliminated from the budget combination calculations for the corresponding the total budget case. The costs shown for the budgets of 600 and 1200 reflect the entire set of cost coefficients for each time period. Costs are in millions of Euros.

Table G1: Cost Coefficients Used for Each Budget Case and Time Period for the Oxyfuel and Post-Combustion Projects

	Budget Case											
Time Period	**600**		**1200**		**1800**		**2100**		**2400**		**4800**	
T = 1	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190
	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380
	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435
	Oxy 1	870	Oxy 1	870	Oxy 1	870	Oxy 1	870	Oxy 1	870	Oxy 1	870
T = 2	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190
	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380
	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435
	Oxy 1	870	Oxy 1	870	Oxy 1	870	Oxy 1	870	Oxy 1	870	Oxy 1	870
	Post 2	150	Post 2	150	Post 2	150	Post 2	150				
	Post 2	300	Post 2	300	Post 2	300	Post 2	300	Post 2	300	Post 2	300
	Post 2	380	Post 2	380								
	Post 2	760	Post 2	760	Post 2	760	Post 2	760	Post 2	760	Post 2	760
T = 3	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190
	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380
	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435				
	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450
	Oxy 1	480	Oxy 1	480	Oxy 1	480	Oxy 1	480	Oxy 1	480	Oxy 1	480
	Oxy 1	870	Oxy 1	870	Oxy 1	870						
	Oxy 1	890	Oxy 1	890	Oxy 1	890	Oxy 1	890	Oxy 1	890	Oxy 1	890
	Oxy 1	960	Oxy 1	960	Oxy 1	960	Oxy 1	960	Oxy 1	960	Oxy 1	960
	Post 2	150	Post 2	150	Post 2	150	Post 2	150				
	Post 2	300	Post 2	300	Post 2	300	Post 2	300	Post 2	300	Post 2	300
	Post 2	380	Post 2	380								
	Post 2	760	Post 2	760	Post 2	760	Post 2	760	Post 2	760	Post 2	760
T = 4	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190
	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380
	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435				
	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450
	Oxy 1	480	Oxy 1	480	Oxy 1	480	Oxy 1	480	Oxy 1	480	Oxy 1	480
	Oxy 1	870	Oxy 1	870	Oxy 1	870						
	Oxy 1	890	Oxy 1	890	Oxy 1	890	Oxy 1	890	Oxy 1	890	Oxy 1	890
	Oxy 1	960	Oxy 1	960	Oxy 1	960	Oxy 1	960	Oxy 1	960	Oxy 1	960
	Post 2	360	Post 2	360	Post 2	360	Post 2	360	Post 2	360	Post 2	360
	Post 2	380	Post 2	380								
	Post 2	720	Post 2	720	Post 2	720	Post 2	720	Post 2	720	Post 2	720
	Post 2	760	Post 2	760	Post 2	760	Post 2	760	Post 2	760	Post 2	760
T = 5	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190	Oxy 1	190
	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380	Oxy 1	380
	Oxy 1	435	Oxy 1	435	Oxy 1	435	Oxy 1	435				
	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450	Oxy 1	450
	Oxy 1	870	Oxy 1	870	Oxy 1	870						
	Oxy 1	891	Oxy 1	891	Oxy 1	891	Oxy 1	891	Oxy 1	891	Oxy 1	891
	Post 2	360	Post 2	360	Post 2	360	Post 2	360	Post 2	360	Post 2	360
	Post 2	380	Post 2	380								
	Post 2	720	Post 2	720	Post 2	720	Post 2	720	Post 2	720	Post 2	720
	Post 2	760	Post 2	760	Post 2	760	Post 2	760	Post 2	760	Post 2	760

Glossary: Acronyms

BARON	Branch and Reduce Optimization Navigator
CCS	Carbon Capture and Storage
CCTS	Carbon Capture, Transport and Storage
EEPR	European Energy Program for Recovery
ETS	Emission Trading Scheme
EU	European Union
FAA	Federal Aviation Administration
GCF	Greatest Common Factor
IEA	International Energy Agency
IGCC	Internal Gasification Combined Cycle
IP	Integer Program(ming)
IPCC	Intergovernmental Panel on Climate Change
IT	Information Technology
KKT	Karush-Kuhn-Tucker
kW	Kilowatt
LRIP	Low Rate of Initial Production
MINLP	Mixed-Integer Nonlinear Program
MPEC	Mathematical Programs with Equilibrium Constraints
MW	Megawatt
NASA	National Aeronautics and Space Administration
NPV	Net Present Value
NRO	National Reconnaissance Office
PCC	Pre-Combustion Capture
PMF	Probability Mass Function
R&D	Research and Development
SDP	Stochastic Dynamic Program(ming)
START	Strategic Assessment of Risk and Technology
TDRA	Technology Development Risk Assessment
TRL	Technology Readiness Level

Bibliography

Adams, W.P., and Forrester, R.J. (2005). "A Simple Recipe for Concise Mixed 0-1 Linearizations," *Operations Research Letters,* 33: 55-61.

Baker, E., Chon, H., and Keisler, J. (2009). "Carbon Capture and Storage: Combining Economic Analysis with Expert Elicitations to Inform Climate Policy," *Climate Change*, 96: 379-408.

BARON (2011). http://archimedes.cheme.cmu.edu/baron/baron.html. Referenced August 8, 2011

Bazaraa, M.S., Sherali, H.D., and Shetty, C.M. (1979). *Nonlinear Programming Theory and Algorithms*, Wiley, New York.

Bertsimas, D., and Sim, M. (2003). "Robust Discrete Optimization and Network Flows," *Mathematical Programming*, 98: 49–71.

Birge, J.R., and Louveaux, F. (1997). *Introduction to Stochastic Programming*, Springer, New York.

Brotcorne, L., Marcotte, P., and Savard, G. (2008). "Bilevel Programming: The Montreal School," *INFOR,* 46(4): 231-246.

Burchett, S., and Biswas, D. (2002). "Schemes of Emissions Management, Bottom up Approaches to Internalising the Cost of Sulphur Dioxide Emissions," *Real Options and Energy Management: Using Options Methodology to Enhance Capital Budgeting Decisions* (Ehud I. Ronn, ed.), London, Risk Books: 563-594.

Cao, Q., and Wang, Q. (2007). "Optimizing vendor selection in a two-stage outsourcing process," *Computers & Operations Research*, 34(12): 3757-3768.

Carson, R. (2007). *Contingent Valuation: A Comprehensive Bibliography and History,* Edward Elgar, Northampton, MA.

Ceylan, B.K., and Ford, D.N. (2002). "Using Options to Manage Dynamic Uncertainty in Acquisition Projects," *Acquisition Review Quarterly*, Fall: 241-258.

Chan, G., Anadon, L.D., Chan, M., and Lee, A. (2011). "Expert Elicitation of Cost, Performance, and R&D Budgets for Coal Power with CCS," *Energy Procedia*, 4: 2685-2692.

Cohon, J.L. (2003). *Multiobjective Programming and Planning*, Dover, Mineola, NY.

Costa, A., and Paixao, J.M.P. (2009). "An Approximate Solution Approach for a Scenario-Based Capital Budgeting Model," *Computational Management Science*, DOI: 10.1007/s10287-009-0117-4.

Cottle, R.W., Pang, J-S., and Stone, R.E. (1992). *The Linear Complementarity Problem*, Academic Press, San Diego.

Dalkey, N. (1969). *The Delphi Method: An Experimental Study of Group Opinion*, RAND Report, #RM-5888-PR.

Department of Defense (2001). *Mandatory Procedures for Major Defense Acquisition Programs*, Washington, DC: Department of Defense. Rep. Regulation 5000.2-R.

Department of the Navy (1998). *Top Eleven Ways to Manage Technical Risk*, Washington, DC: Office of the Assistant Secretary of the Navy (RD&A). Rep. NAVSO P-3686.

Deputy Under Secretary of Defense for Science and Technology (2005). *Technology Readiness Assessment (TRA) Deskbook*, Washington, DC.

Dixit, A., and Pindyck, R. (1994). *Investment under Uncertainty*, Princeton University Press, Princeton, NJ.

DoE (2011). "Fossil Energy: DoE's FutureGen 2.0 Initiative," U.S. Department of Energy, http://www.fossil.energy.gov/programs/powersystems/futuregen/, Referenced September 22, 2011.

Drezner, J., Jarvaise, J., Hess, R., Hough, P., and Norton, D. (1993). *An Analysis of Weapon System Cost Growth*, RAND Report: MR-291-AF.

Duong, M.H., Swart, R., Bernstein, L., and Petersen, A. (2007). "Uncertainty Management in the IPCC: Agreeing to Disagree," *Global Environmental Change*, 17(1): 8-11.

Eckhause, J.M., Gabriel, S.A., and Hughes, D.R. (2011). "An Integer Programming Approach for Evaluating R&D Funding Decisions with Optimal Budget Allocations," *IEEE Transactions on Engineering Management*, Pending Final Revisions.

Eckhause, J.M., and Herold, J. (2011). "Using Real Options to Evaluate Optimal Funding Strategies for Carbon Capture, Transport and Storage (CCS) Projects in the European Union," *Under Review*.

Eckhause, J.M., Hughes, D.R., and Gabriel, S.A. (2009). "Evaluating Real Options for Mitigating Technical Risk in Public Sector R&D Acquisitions," *International Journal of Project Management*, 27(4): 365-377.

Elfes, A., Weisbin, C., Manvi, R., Adumitoaie, V., Lincoln, W., and Shelton, K. (2006). "Extending the START Framework: Computation of Optimal Capability Development Portfolios Using a Decision Theory Approach," *Systems Engineering*, 9(4): 331-357.

Errais, E., and Sadowsky, J. (2008). "Valuing Pilot Projects in a Learning by Investing Framework: An Approximate Dynamic Programming Approach," *Computers & Operations Research*, 35(1): 90-112.

European Commission (2009). "Selection of Offshore Wind and Carbon Capture and Storage Projects for the European Energy Programme for Recovery," MEMO/09/543, Brussels, December.

Feller, W. (1968). *An Introduction to Probability Theory and its Applications, Vol. 1*, 3rd Edition, Wiley, New York.

Fischer, C. (2008). "Emission Pricing, Spillovers, and Public Investment in Environmentally Friendly Technologies," *Energy Economics*, 30: 487-502.

Fortuny-Amat, J., and McCarl, B. (1981). "A Representation and Economic Interpretation of a Two-Level Programming Problem," *Journal of the Operational Research Society*, 32(9): 783-792.

Gabriel, S.A., and Leuthold, F.U. (2010). "Solving Discretely-Constrained MPEC Problems with Applications in Electric Power Markets," *Energy Economics*, 32, 3-14.

Glover, F., and Woolsey, E. (1974). "Technical Note—Converting the 0-1 Polynomial Programming Problem to a 0-1 Linear Program," *Operations Research*, 22(1): 180-182.

Golabi, K., Kirkwood, C., and Sicherman, A. (1981). "Selecting a Portfolio of Solar Energy Projects Using Multiattribute Preference Theory," *Management Science*, 27(4): 174-189.

Hansson, A., and Bryngelsson, M. (2009). "Expert Opinions on Carbon Dioxide Capture and Storage – A Framing of Uncertainties and Possibilities," *Energy Policy*, 37: 2273-2282

Herold, J., Rüster, S., and von Hirschhausen, C. (2010a). "Carbon Capture, Transport and Storage in Europe: A Problematic Energy Bridge to Nowhere?," Working

Document No. 341, Center for European Policy Studies, November, ISBN 978-92-9079-6138-066-1.

Herold, J., Rüster, S., and von Hirschhausen, C. (2010b). "The Future of Coal in Europe," Project No. 213744, SECURE Work Package 5.3.5, http://www.feem-project.net/secure/plastore/Deliverables.

Huchzermeier, A., and Loch, C.H. (2001). "Project Management under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D." *Management Science*, 47(1): 85-101.

Hull, J.C. (1997). *Options, Futures and Other Derivatives*, Prentice Hall, Upper Saddle River, New Jersey.

IEA (2008). "CO2 Capture and Storage – A key carbon abatement option," *Energy Technology Analysis*, International Energy Agency. ISBN 978-92-64-04140-0.

IEA (2009). "Technology Roadmap: Carbon Capture and Storage," International Energy Agency. http://www.iea.org/papers/2009/CCS_Roadmap.pdf, Referenced August 22, 2011.

IEA (2011). R&D Statistics. http://wds.iea.org/WDS/Common/Login/login.aspx. International Energy Agency. Referenced August 23, 2011.

IPCC (2005). *IPCC Special Report on Carbon Dioxide Capture and Storage*, Technical Report, Intergovernmental Panel on Climate Change.

IPCC (2007). "Contribution of Working Group III to the Fourth Assessment Report," Intergovernmental Panel on Climate Change.

IZ-Klima (2010). Expert Meeting on CCS, Oral Communication, December 11, 2010, Berlin, Germany.

Jaffe, A.B. (2006). "The Importance of Spillovers in the Policy Mission of the Advanced Technology Program," *Journal of Technology Transfer*, 23(2): 11-19.

Kanniche, M., Gros-Bonnivard, R., Jaud, P., Valle-Marcos, J., Amann, J-M., and Bouallou, C. (2010): "Pre-Combustion, Post-Combustion and Oxy-Combustion in Thermal Power Plant for CO2 Capture" *Applied Thermal Engineering*, 30: 53-62.

Lichtenberg, F. (1988). "The Private R&D Investment Response to Federal Design and Technical Competitions," *American Economic Review*, 78(3): 550-559.

LINGO (2011). *LINGO User's Guide*, LINDO Systems, Chicago.

Linstone, H.A., and Turoff, M. editors. (1975). *The Delphi Method: Techniques and Applications*, Addison-Wesley, Reading, MA.

Luo, Z.Q., Pang, J.S., and Ralph, D. (1996). *Mathematical Programs with Equilibrium Constraints,* Cambridge University Press, Cambridge, United Kingdom.

Malerba, F. (2002). "Sectoral Systems of Innovation and Production," *Research Policy*, 31(2): 247-264.

Mathias, D., Goodsell, A., and Go, S. (2006). "Technology Development Risk Assessment for Space Transportation Systems," Proceedings of *8th Probabilistic Safety Assessment and Management Conference*, New Orleans, LA.

Meier, H., Christofides, N., and Salkin, G. (2001). "Capital Budgeting under Uncertainty—An Integrated Approach Using Contingent Claims Analysis and Integer Programming," *Operations Research*, 49(2): 196-206.

Mulvey, J.M., Vanderbei, R.J., and Zenios, S.A. (1995). "Robust Optimization of Large Scale Systems," *Operations Research*, 43(2): 264-281.

NASA (2011). "Definition Of Technology Readiness Levels," http://esto.nasa.gov/files/TRL_definitions.pdf, National Aeronautics and Space Administration. Referenced September 15, 2011.

Nemhauser, G.L., and Wolsey, L.A. (1988). *Integer and Combinatorial Optimization*, Wiley, New York.

NER300 (2010). "NER300 Passed with Amendments by Member States," http://www.ner300.com/?m=201002. Referenced August 22, 2011.

Newman, A., Yano, C., and Rubio, E. (2010). "Mining Above and Below Ground: Timing the Transition," Under Revision with *IIE Transactions*.

Nilim A., and El Ghaoui, L. (2005). "Robust Control of Markov Decision Processes with Uncertain Transition Matrices," *Operations Research*, 53(5): 780-798.

Panayi, S., and Trigeorgis, L. (1998). "Multi-Stage Real Options: The Cases of Information Technology Infrastructure and International Bank Expansion," *Quarterly Review of Economics and Finance*, 38: 675-692.

Perdue R., McAllister, W., King, P., and Berkey, B. (1999). "Valuation of R and D Projects Using Options Pricing and Decision Analysis Models." *Interfaces*, 29(6): 57-74.

Post, J., Bennett, M., and Knorr, D. (2004). "Real Options in Capital Investment Decision-Making: Data Link Case Study," Presented at the AIAA *4th Aviation Technology, Integration and Operations (ATIO) Forum*, AIAA 2004-6238.

Powell, W.B. (2009). "What You Should Know About Approximate Dynamic Programming," *Naval Research Logistics*, 56: 239-249.

Puterman, M. (1994). *Markov Decision Processes*, Wiley, New York.

PWC (2010). *Climate Change and Electricity: The European Carbon Factor - Comparison of CO2 Emissions of Europe's Leading Electricity Utilities*, PriceWaterhouseCoopers, November.

RECCS (2010). *RECCS Plus: Comparison of Renewable Energy Technologies with Carbon Dioxide Capture and Storage*, Study 0329967/07000285, Wuppertal Institute for Climate, Environment and Energy, Wuppertal, Germany, August.

Reuters (2009). "EU Recommends Winners of CCS Project Funding," http://www.reuters.com/article/idUKLG28841420091016, Referenced August 22, 2011.

Roberts, K., and Weitzman, W.L. (1981). "Funding Criteria for Research, Development, and Exploration Projects" *Econometrica*, 49(5): 1261-1288.

Rogerson, W. (1995). "Incentive Models of the Defense Procurement Process," in *Handbook of Defense Economics*, Vol. 1, Hartley, K., and Sandler, T., Eds. Elsevier Science, Amsterdam: 309-346.

Royden, H. L. (1988). *Real Analysis*, 3rd Edition, Prentice-Hall, Englewood Cliffs, NJ.

Rubin, E.S., Yeh, S., Antes, M., Berkenpas, M., and Davision, J. (2006). "Estimating Future Costs of CO2 Capture Systems Using Historical Experience Curves," Proceedings of *Eight International Conference on Greenhouse Gas Control Technologies (GHGT-8),* June 19-22, Trondheim, Norway.

Santiago, L., and Vakili, P. (2005). "On the Value of Flexibility in R&D Projects." *Management Science*, 51(8): 1206-1218.

Santiago, L.P., and Bifano, T.G. (2005). "Management of R&D projects under uncertainty: A multidimensional approach to managerial flexibility." *IEEE Transactions in Engineering Management*, 52(2): 269-280.

Snir, E., and Hitt, L. (2004). "Vendor Screening in Information Technology Contracting with a Pilot Project," *Journal of Organizational Computing and Electronic Commerce*, 14(1): 61-88.

Stern, N. (2007). Stern *Review on the Economics of Climate Change*, Cambridge University Press, Cambridge, United Kingdom.

Trigeorgis, L. (1996). *Real Options: Managerial Flexibility and Strategy in Resource Allocation*, MIT Press, Cambridge, MA.

Tseng, C.L., Lin, K.Y., and Sundararajan, S.K. (2005). "Managing Cost Overrun Risk in Project Funding Allocation," *Annals of Operations Research*, 135: 127-153.

Tzimas, E. (2009). *The Cost of Carbon Capture and Storage Demonstration Projects in Europe*, JRC Scientific and Technical Reports, European Commission.

Vallentin, D. (2007). *Inducing the International Diffusion of Carbon Capture and Storage Technologies in the Power Sector*, Wuppertal Institute for Climate, Environment and Energy, 162. ISSN 0949-5266.

Vonortas, N., and Hertzfeld, H. (1998). "Research and Development Project Selection in the Public Sector," *Journal of Policy Analysis and Management*, 17(4): 639-657.

Wang, J., and Hwang, W. (2008). "A Fuzzy Set Approach for R&D Portfolio Selection Using a Real Options Valuation Model," *Omega*, 36: 122-130.

Ward, A., Liker, J.K., Cristiano, J.J., and Sobek II, D.K. (1995). "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster," *Sloan Management Review*, Spring: 43–61.

Weisbin, C., Rodriguez, G., Elfes A., and Smith, J. (2004). "Toward a Systematic Approach for Selection of NASA Technology Portfolios," *Systems Engineering*, 7(4): 285-302.

Williams, H.P. (1999). *Model Building in Mathematical Programming (Fourth Edition)*, Wiley, New York.

Yeo, K.T., and Qiu, F. (2003). "The Value of Management Flexibility—a Real Option Approach to Investment Evaluation," *International Journal of Project Management*, 21(4): 243-250.

Yu, J., Liu, Y., and Yang, X. (2008). "Cluster-Based Optimization of Urban Transit Hub Locations: Methodology and Case Study in China," *Transportation Research Board Annual Meeting*, Paper ID 08-1744.

ZEP (2008). "EU Demonstration Programme for CO2 Capture and Storage (CCS)," European Technology Platform for Zero Emission Fossil Fuel Power Plants.

ZEP (2010). "CCS European Industrial Initiative Implementation Plan 2010-2012," European Technology Platform for Zero Emission Fossil Fuel power Plants.

Zerogen (2011). "Zerogen Project Overview," http://www.zerogen.com.au/project/overview.aspx, Referenced September 18, 2011.