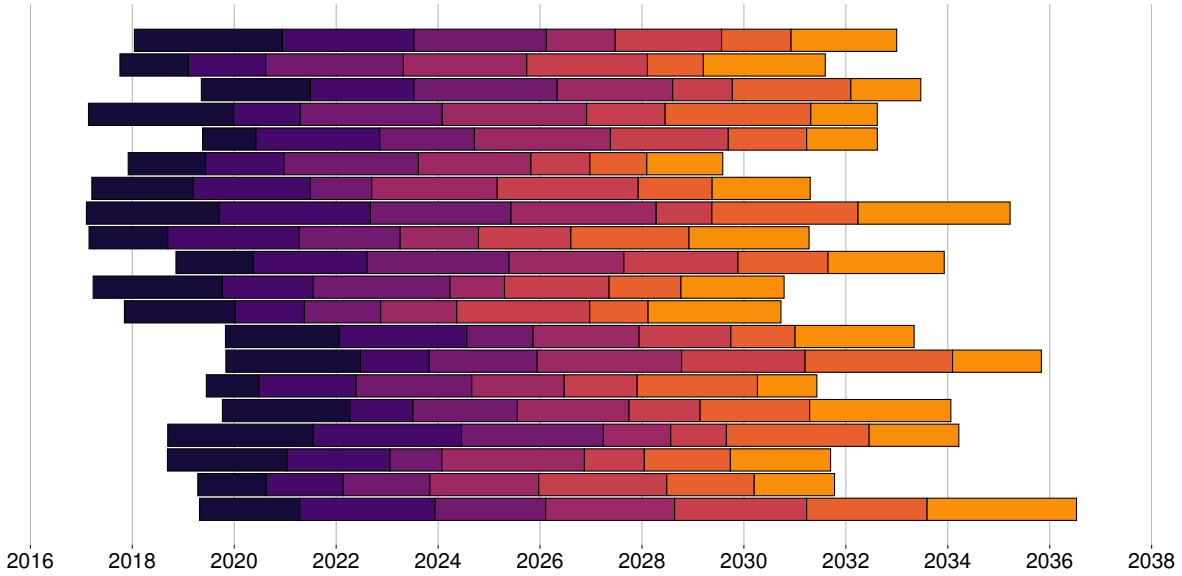

Stochastic Optimization: Pharmaceutical Portfolios

Decision-making under uncertainty

Master's thesis in Mathematical Optimization

NILS CARLSSON

DMITRII SERGEJEV

Department of Mathematical Sciences CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Stochastic Optimization: Pharmaceutical Portfolios

Decision-making under uncertainty

NILS CARLSSON
DMITRII SERGEJEV

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics
Chalmers University of Technology
Gothenburg, Sweden 2019

Stochastic Optimization: Pharmaceutical Portfolios
Decision-making under uncertainty
NILS CARLSSON
DMITRII SERGEJEV

Supervisor & Examiner: Michael Patriksson, Mathematical Sciences, Chalmers University of Technology

Master's Thesis 2019
Department of Mathematical Sciences
Division of Applied Mathematics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visual representation of a pharmaceutical portfolio, with projects and corresponding phases

Typeset in LaTeX
Printed by [Teknologtryck]
Gothenburg, Sweden 2019

Stochastic Optimization: Pharmaceutical Portfolios
Decision-making under uncertainty
NILS CARLSSON
DMITRII SERGEJEV
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

The process of developing pharmaceutical drugs is long and costly, with a low probability of an approved drug in the final stage. Given a portfolio of several different pharmaceutical projects, it is therefore highly important to select the ones that maximize the expected profit. This paper presents a mathematical optimization model given the rules of a pharmaceutical project. The model is initially fully deterministic but is later expanded to include stochastic constraints. A recourse view of the problem is also discussed, meaning optimization under the assumption that choices can be made based on the realization of stochastic variables. The deterministic model is linear and thus straightforward to solve, while the stochastic constraints introduce non-linearities that greatly increase the complexity of the problem. Possible approaches to reduce this complexity are discussed, such as approximations and linearizations, along with the best use of the models. The deterministic model is also applied to a test portfolio and the results, such as the revenue, cost, solution time and others are discussed in the light of combinatorial complexities and decisions under risk.

Keywords: optimization, Captario, pharmaceutical projects, linear optimization, MILP, stochastic constraint, recourse, AMPL.

Acknowledgements

We would like to thank our examiner, Michael Patriksson, and our supervisor, Magnus Ytterstad, for their help, guidance and patience during this thesis work. We would also like to thank Röda Rummet, Tronsalen, Kårrest and everything and everyone who helped us while we wrote this thesis.

Nils Carlsson & Dmitrii Sergejev, Gothenburg, August 2019

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

Pharmaceutical development is a process consisting of many different stages of production, research and testing. Whenever a new drug is researched and patented, it needs to pass through several stages of approval and testing to be deemed safe to release to the market [22, 12], with each stage costing several millions of dollars. However, few drugs actually make it through this process. Only a small fraction (approximately 1–9% [17, 15]) of pharmaceutical developments actually result in a marketable product. As these testing stages can take up to a decade or more to complete in total, combined with a high investment cost, it is clear that a robust investment strategy is of the utmost importance. More information about the pharmaceutical development process can readily be found in literature such as [17].

Captario is the developer of Captario SUM [14], *SUM: Strategic Uncertainty Management*, a tool that helps pharmaceutical developers assess and analyze the risks and profits in their portfolio of pharmaceutical projects. Projects in a given portfolio consist of a number of phases which must be completed in succession and are defined by parameters such as their costs, time length and revenues. These parameters may be deterministic but in practice they are described as a stochastic variable. SUM simulates tens of thousands realizations of a project, and produces an estimation of the final distribution for the stochastic variables for a particular project, and by extension, a portfolio.

Throughout this report, we aim to expand the capabilities of SUM by introducing a method of determining which particular projects will have the highest likelihood of being more profitable than others by using simulated data from SUM. Each project is defined by a set of specific parameters, such as current phase, phase lengths, costs, revenues, and a yearly budget. With this information it is possible to present a combination of projects that would result in a maximum profit. To create an optimal project schedule when all the parameters are known, integer linear programming (ILP) models are extensively used. Many examples can be found in literature, such as solving the problem of scheduling computing tasks optimally in grid computing systems, finding an optimal schedule for medical equipment maintenance and optimal scheduling and flight planning of unmanned aerial vehicles [8, 13, 7].

However, in real portfolios, the given parameters might not always be fixed, but can only be estimated with the use of a probability distribution. These stochastic factors may result in non-linear terms, which generally increases the complexity of the project and requires a different approach to be used. Along with the increased

complexity, a stochastic problem also requires a decision to be made regarding the level of acceptable risk/reward for any project. Furthermore, some stochastic parameters impact the problem in a larger scope than others, for example the success or failure of a specific phase (also known as *technical risk*).

Throughout this text, we will present and explain several different models intended to capture the scope of the problem, and expand upon possible solutions under certain constraints. The different models can be divided into two parts - the deterministic and the stochastic models. In the deterministic models, the parameters are assumed to be fixed and known beforehand, while in the stochastic models it is assumed that one or more parameters and variables are stochastic. The complexity and stochastic nature of this model opens the way for many different approaches and possible solutions, and requires us to determine the best possible decision to make at each realization of the stochastic variables. This leads into what is called *recourse* theory, where the possibility to adjust the choice of projects as more information becomes available is introduced. Since the technological risk may drastically alter the optimal portfolio, recourse theory may prove be a powerful tool to mitigate that uncertainty.

1.2 Problem statements

1.2.1 Deterministic form

Given a set of pharmaceutical projects, all of which are guaranteed to succeed, along with associated deterministic phase, revenue and cost parameters, construct a set of mathematical variables and constraints that models the projects as closely as possible. The model should be constructed with the intention of being used in mathematical optimization in order to find the most profitable set of projects.

1.2.2 Stochastic form

Given a set of pharmaceutical projects, all of which are guaranteed to succeed, along with associated stochastic phase, revenue and cost parameters,with one or more stochastic parameters, construct a set of mathematical variables and constraints that models the projects as closely as possible. The model should be constructed with the intention of being used in mathematical optimization in order to find the most profitable set of projects.

1.2.3 Recourse form

Given a set of pharmaceutical projects, each with a certain probability to succeed, along with associated stochastic phase, revenue and cost parameters,with one or more stochastic parameters, construct a set of mathematical variables and constraints that models the projects as closely as possible. The model should be constructed with the intention of being used in mathematical optimization in order to find the most profitable set of projects, and a strategy should formulated so maximizes the profits under the assumption that any given project might fail.

1.3 Scope

The main goal of this project is the development and explanation of the mathematical models that can be used to solve the pharmaceutical portfolio planning problem. Therefore the focus is on the theoretical aspects of the problem and how the construction of the models will adjust the solution of the problems. When possible we have discussed how the models are best utilized, and suggested possible expansions and methods of usage. However, given the complexity of many of the problems discussed, it has not been the goal to develop and more explicitly evaluate these expansions.

2
Theory

In order to fully understand the models and the decisions made when building them, some knowledge of pharmaceutical development is essential. This chapter is intended to explain the basic concepts of pharmaceutical development as modelled by Captario AB, along with mathematical optimization, and how the two parts relate to each other. As such, many technical details will be ignored in order to focus on what is most relevant for the mathematical models at hand.

2.1 Basics of pharmaceutical development

2.1.1 Projects

Each project starts with a patent date, corresponding to when the drug in question has been approved for patent. A patent holds for approximately 20 years [23, 21], and each project thus has a corresponding date when the patent expires, referred to as the *Significant Pattern Expiry*, or SPE. This date is the major limiting factor in the profitability of the project. After the SPE date, the pharmaceutical developer no longer holds the exclusive rights to the sale of the drug, and competitors are allowed to launch their generic alternatives. This leads to the market share going from exclusive to divided, and profits sink accordingly. Given the SPE being a ”deadline” of sorts, it is then imperative to develop a drug as quickly as possible. The actual development can be described by a set number of *phases*.

2.1.2 Phases

Each project consists of several sequential stages, called 'phases'. They can be simply seen as the different stages of testing that a drug must go through in order to be approved for sale. This is typically a very comprehensive process, starting with small, focused tests on a few test subjects, and moving towards more expensive, longer tests on large test groups. A phase is typically defined by its stage in the project, cost, revenue, length, possible starting dates, and probability to succeed. For a phase to be initiated the previous phase must have been successfully completed. If a phase fails, the whole project is deemed failed as well. The effect is essentially that each project consists of several separate phases that must all succeed in order to move onto the next stage. Typically a project only generates revenue when the final phase is completed and the drug is sold on the market.

2.1.3 Revenue

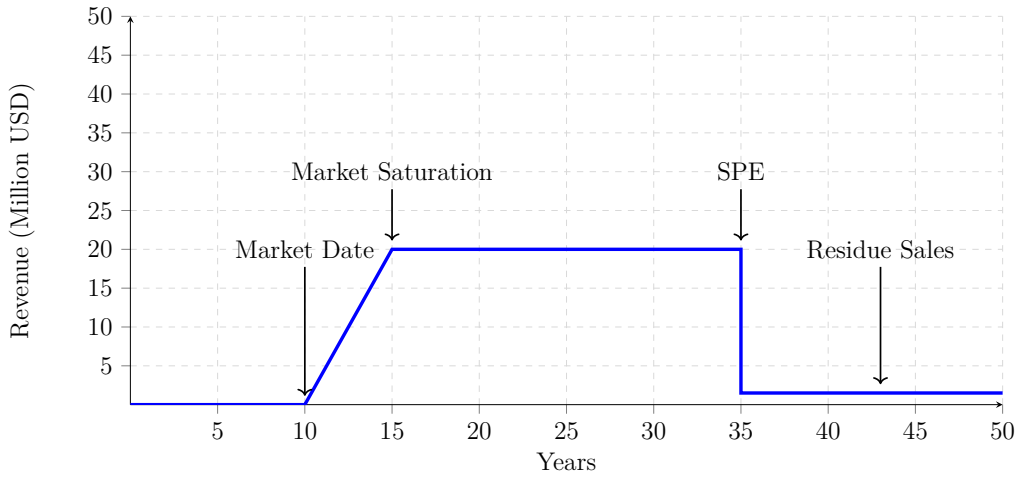

Figure 2.1: An example of the revenue generated by a project after completion of all phases.

If all phases in a pharmaceutical drug project are successful, the drug enters the selling phase in which revenue is generated. To calculate the revenue in the selling phase an approximative model developed by Captario is used, an example of which is depicted in Figure 2.1.

The revenue generation of the selling phase is most easily viewed as consisting of separate stages. Whenever a product enters the market, resources are allocated to promote it. Thus the profits slowly increase during *ramp* period, between the ”market date” and ”market saturation” in figure 2.1. When the sales have peaked and the product has reached full market saturation, the revenue reaches the *Peak Year Sales* period, called PYS. During this time the project generates the maximum possible revenue. The end of the PYS section comes at the SPE date, which means that competitors have launched their generic drugs, and the revenue will then typically fall to a fraction of the PYS section. This is called the ’residue’ section. An overview of this sequence of events can be seen in Figure 2.1, where each date and section is clearly marked. This means that the revenue is highly dependent on when the final phase of the project begins, with an earlier sales date generating more years of PYS. The ramp and residue sections however, are less influenced by the starting date and can thus be approximated as remaining constant, with good accuracy.

2.1.4 Budget and costs

Each phase of each project corresponds to a cost, be it a cost for testing, research, or marketing. This is a lump cost that is divided throughout several time steps, typically over several years. To accommodate these costs, a yearly budget is available.

These financial constraints are the major limiting factor in choosing projects and maximizing revenue. The total cost of each project is also considered when calculating the total revenue from each project.

2.1.5 Probability and stochastic variation

In the real world, each phase carries with it a probability to succeed or fail. Parameters such as the length and cost of a phase are also uncertain, and can only be estimated beforehand. However, we will not concern ourselves with the specific reasons as to why a project might fail, and instead only view them as stochastic variables. Where numerical values are needed, they are gathered from simulations performed by Captario SUM.

2.2 Mathematical concepts

Before presenting the optimization models, it is necessary to lay down a framework of their mathematical characteristics, and how these impact the solution process. Some aspects of the model might change the requirements of the optimization methods, and others may result in different solution times, complexity or result reliability. Therefore, the purpose of this section is to present the basic mathematical concepts relating to the optimization models.

2.2.1 Optimization model

The mathematical models in this paper all follow the same basic structure: maximize a given function f given a decision vector $\mathbf{x}$, with the requirement that several separate constraints must hold. More specifically, it can be written as:

$$\begin{aligned}
\underset{x}{\text{maximize}} \quad & f(\mathbf{x}), && && \text{(1a)}\\
\text{subject to} \quad & g_i(\mathbf{x}) \leq 0, && i \in \{1, ..., m\}, && \text{(1b)}\\
& h_j(\mathbf{x}) = 0, && j \in \{1, ..., n\}, && \text{(1c)}\\
& \mathbf{x} \in \mathrm{R}^k, && && \text{(1d)}
\end{aligned}$$

where k is the number of decision variables, m and n are the number of constraints of each type and f, along with each g_i,h_j are real-valued functions.

An important characteristic for such optimization models is whether it is linear or non-linear. A linear model is defined to be an optimization model where the objective function and the constraints can be expressed by a polynomial with degree one or less with respect to the decision variables. In the same vein, a non-linear model is defined to be an optimization model in which at least one of the constraints or the objective function contains some term which can not be expressed as a polynomial of degree one or lower with respect to the decision variables.

Integer Linear Programming (ILP) is a variation of the linear model, where the decision vector may only take on integer values. Finding an optimal solution to ILP problems is proven to be NP-complete [2], which means that the time required to solve the problem grows very quickly as the number of variables increases.

Generally, it is desirable to find linear formulations of a problem as the space of feasible solutions $\mathbf{x}$ will be convex, allowing for more efficient optimization procedures such as the simplex method. For example, if the integrality requirement on the

binary decision variables is relaxed, one may find good bounds on the optimal value by solving the resulting linear program which can be done relatively quickly using for example the simplex method. Typically, non-linear models may not have a convex domain and thus it may be more computationally difficult to find a globally optimal solution.

Some exact solution procedures for ILP problems (such as cutting planes or branch and bound [1]), start by relaxing the integrality constraint on the decision variables, which highlights the importance of a linear formulation, even as the decision variables are restricted to integer values.

2.2.2 Stochastic optimization model

In some applications, the parameters can not be explicitly known and thus must be modelled with random variables, which leads to an extension of optimization models known as a *Stochastic optimization model*, formulated as follows:

$$\begin{aligned}
\underset{x}{\text{maximize}} \quad & \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x}, \boldsymbol{\xi})], && && \text{(2a)}\\
\text{subject to} \quad & g_i(\mathbf{x}, \boldsymbol{\xi}) \leq 0, && i \in \{1, ..., m\}, && \text{(2b)}\\
& h_j(\mathbf{x}, \boldsymbol{\xi}) = 0, && j \in \{1, ..., n\}, && \text{(2c)}\\
& \mathbf{x} \in X, && && \text{(2d)}\\
& \boldsymbol{\xi} : \Omega \to \mathbb{R}^n. && && \text{(2e)}
\end{aligned}$$

Here, $\boldsymbol{\xi}$ is a vector of random parameters in the model and thus the optimal choice of $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ depends on each realization of $\boldsymbol{\xi}$, while all other parameters are the same as in the previous section.

Similarly, the value of f may vary wildly depending on different realizations of $\boldsymbol{\xi}$ which are not known beforehand. As a consequence of this, single realizations of the objective function f are not of much interest for the decision-maker. Instead one may focus on studying the distribution or expected value of f.

One way to handle stochastic parameters is by using *chance-based constraints*, which transform a stochastic constraint into a deterministic form and instead lets the decision-maker specify the risk of the constraint being violated. A general treatise on chance-based constraints relating to energy management may be found in [10] and an application of joint chance-based constraints for hydro reservoir management may be found in [11].

More specifically for a set of probability levels α_i, a chance-based constraint formulation (for the inequality constraints g_i) for this model takes the form

$$\mathbb{P}\left[g_i(\mathbf{x}, \boldsymbol{\xi}) \leq 0\right] \geq \alpha_i, \qquad i \in \{1, ..., m\}. \tag{3}$$

The probability levels α_i are set to values which reflect the risk one is willing to take that the constraint is violated. In general, the decision maker wants to select a reasonably high value for α_i such as 0.95 or 0.99 to minimize the risk of violating the constraint.

However, this is also problem dependent and in order to make the best choice it is desirable to study the cumulative distribution function $F_{\boldsymbol{\xi}}(x, \xi)$ and determine

which scenarios are unacceptable for the decision maker. For small cases, this can be done by evaluating every possible combination of decision variables x but for larger cases this becomes computationally intractable. In that case, one may consider studying carefully selected, problem specific cases or to sample a subset of all x-values and analyze the cumulative distribution function with respect to those values only.

Depending on the distribution of $\boldsymbol{\xi}$, the constraint may become non-linear and non-convex, increasing the computational complexity further.

2.2.3 The complexity of the model

The speed with which an optimizer can deliver a solution depends on several factors, such as the method used by the optimizer, the type of problem solved, the quality of the optimizer itself, optimality conditions, and the space of all solutions that the optimizer must consider. Aspects relating to the optimizer is a matter of implementation, and so the more relevant factors are the ones inherent in the problem at hand.

The problem is one of combinatorics — which projects should be chosen to maximize the revenue. It is clear that the problem very much depends on how many options that are available to the solver — the *optimization space*. The size of this space is directly controlled by constraints, such as the budget. A low budget would reduce the number of choices to a few projects, and finding a solution would then be quite fast. Correspondingly, if the budget was high enough all projects could be chosen, and an optimal solution would be found quickly in this case as well. However, a less extreme budget would only allow for a few projects, meaning that there are many more combinations that must be evaluated before an optimal solution can be found. In addition, it is possible to allow the phases to start earlier or later than planned. This introduces additional complexity, since the optimization space is massively increased if each phase can be individually shifted.

Thus, it is important to be aware of how the complexity of the model behaves, along with how to best handle the problem. An approach could be that parts of the feasible optimization space are directly cut off with additional constraints, if it is known that no relevant solution can be found there. Another is to analyze the structure of the model and divide it into subproblems or apply some other, approximative, approach. In this report, however, the focus is on the core of the model and not its implementation, and therefore no concrete approaches are suggested.

2.3 Statistical concepts

In order to fully understand the constraints given by the stochastic variables, and how they translate into the mathematical model, basic knowledge of statistics and probability distributions is required. There is no one ”general” probability distribution used in portfolios, and as such we will not go into detail regarding specific distributions. The core concepts will revolve around the expected value, variance, and the cumulative distribution function.

A generic probability distribution defined over any given sample space S, returns the corresponding probability that a given random variable (or outcome) X from S would be randomly sampled from the distribution. S can be defined as a continuous space, a set of discrete values, or outcomes that are non-numerical, such as heads or tails from a coin flip.

The mean, μ, of the distribution, is also called the expected value and essentially gives the average of all the possible outcomes, weighted by the probability of each outcome. Thus, given a probability distribution and a large number of sampled values, the average of the sampled values will approach the mean in time.

The variance, σ^2, is instead a measure of how much the values are spread relative to the expected value — a low variance implies that the probability distribution is clustered around the mean, while a high variance corresponds to a more spread-out distribution of values.

In cases where different probability distributions contribute to the value of a variable or constraint, it might be relevant to look at the probability distribution of say, the sum or product of the separate stochastic variables. This is only possible to obtain in a closed-form solution in a few special cases, and in other cases approximations or numerical methods must be used.

2.4 Revenue management

Calculating the revenue R_i for project i is one of the most important aspects of the pharmaceutical portfolio model, since the goal is to maximize revenue. However, due to the fact that projects may fail with a high probability, using the revenue as calculated directly from the data set provided by Captario is misleading and does not incorporate the stochastic nature of the problem into the objective function.

To mitigate this, the fundamental approach is to work with expected values. The revenue is split into the three parts described in the figure 2.1 - ramp, PYS and residue.

Taking into account the probability of phases to fail, the expected revenue for the PYS, $R_{i,PYS}$ would then be given by

$$R_{i,PYS} = R_{i,PYS}^{total} \prod_{k=1}^{|\mathcal{K}|} p_{ik}, \tag{4}$$

where p_{ik} is the probability of phase k of project i to succeed. The same approach is used for the ramp and residue revenues.

2.5 Utility

One problem of calculating R_i in this manner is that the realization of the actual revenue is not reflected explicitly. In reality, the revenue is either 0 or R_i^{total}. One may argue that since all the project revenues are given this treatment, this is not problematic but for some investors it may not provide enough comfort.

Depending on the attitude of the investor, it may be more important to not lose their investment but perhaps lower the profitability of the overall portfolio while

others may want to take more of a risk for projects with higher revenues should they succeed.

To this end, a utility function u may be implemented to reflect the attitude of the investor in the model. Such a function would be highly dependent on the needs of the investor, and therefore no specific function is recommended, but instead left up to the investor to decide and design.

Given the cumulative probability for a project i to succeed as $p_i = \prod_{k=1}^{|\mathcal{K}|} p_{ik}$, a function $u(p_i, R_i^{tot})$ can map the probability of a project together with the total revenue given that the project succeeds to a value between 0 and R_i^{tot}, reflecting the attitude of the investor toward that particular combination to succeed. This value can also be used in place of the expected revenue R_i as a way to include the willingness to take a risk.

The advantage of using an utility function is that it may take on any form and thus can be fine-tuned to the desires of the investor.

A simple example of an utility function for a risk-averse investor is $u(p, R) = Rp^3$. Since $0 \leq p \leq 1$, a lower value of p will yield a utility function u with lower values compared to the original expected revenue R_i, while a larger p will close the gap between u and R_i.

Using the utility, the model is reformulated as a maximization of utility instead of expected revenue, which may interest the decision-maker more.

Special care must be taken when constructing the utility function, depending on the probabilities involved. Even if p may be close to one theoretically speaking, it is unlikely that such a project would ever be found in practical applications, given the circumstances of the pharmaceutical development process. Therefore, one might instead consider the range of p_i, which in turn depends on the probabilities that the different phases will succeed. This will enable the decision-maker to easier distinguish between probabilities that may be relatively close when $0 \leq p \leq 1$ but far enough away from each other to matter greatly on a lower scale.

To do this, it is sufficient to obtain p_i for all projects, and define the allowed interval of p as between the minimum and maximum of p_i.

Moreover, it is important to ensure that the utility function takes values which are larger than the cost of the corresponding project, since if it is not, then no project is profitable and the optimal solution will be to choose no project. If this happens the investor may need to reassess the willingness to take risks.

3
Modelling the problem

In this chapter the basics of the proposed model will be introduced. The models were constructed in an iterative fashion, with new additions being built on the old models. The core model is based on the deterministic approach and fulfills the basic requirements of the constraints, such as the budget, phase ordering, revenue and cost calculation, etc.

Improvements to the models include performance and accuracy increases through fractional modeling, and flexibility in the form of allowing the phases to shift start dates. These additions are explained in greater detail below. Care has been taken to keep the deterministic model fully linear. The stochastic model, however, is non-linear, and while the theoretical complexity is only slightly higher than the deterministic counterpart, the practical complexity of the model is far higher. Finally, the recourse approach does not yield a new model, but is discussed with regards to the problem statement and the model.

3.1 Discrete deterministic model

In the discrete deterministic model, no stochastic factors are taken into account. Because of this, the problem is reduced to selecting the best projects and their starting times.

The time horizon over which the optimal portfolio is computed is split into discrete periods of time. These periods of time are denoted by a set $\mathcal{T}$.

Let $\mathcal{I}$ be the set of all projects which can be included in the portfolio.

Likewise, the set $\mathcal{K}$ is defined to be the set of phases that each project is composed of. In this model the number of phases is assumed to be the same in each project. Therefore $\mathcal{K}$ is defined to be a set of integers starting at one and ending at the number of phases. For simplicity of notation, an additional set $\tilde{\mathcal{K}}$ is introduced, which includes all phases in $\mathcal{K}$ except the last.

Given these sets, we proceed to define the necessary variables and parameters for a complete LP formulation of the Discrete deterministic model in Table 3.1 and 3.2 respectively.

Variables	Definition
x_{ik}	Binary decision variable which is set to 1 if phase k of project i is included in the portfolio, 0 otherwise.
h_{ikj}	Binary tracker variable which is set to 1 if phase k of project i is active during time period j, 0 otherwise.
s_{ik}	Variable denoting at what point in time phase k of project i is started.
z_{ikj}	Variable which equals s_{ik} for the time indices j such that $h_{ikj} = 1$, 0 otherwise. Used to keep the model linear and calculated as $h_{ikj}s_{ik}$. Additional constraints are used to make this variable behave as $h_{ikj}s_{ik}$.

Table 3.1: Definition of the variables used in the discrete deterministic model.

Parameter	Definition
l_{ik}	Length of phase k of project i, expressed as a decimal number where a year corresponds to 1.0 time units.
c_{ik}	Cost incurred by phase k of project i, given on a per year basis.
t_i^{min}	The time at which project i is allowed to start. Corresponds to the "discovery" of a patented drug substance.
SPE_i	The date of significant pattern expiry for project i.
R_{is}	The revenue per time unit generated by project i in market stage s. As described in the previous revenue section, s can be either 'ramp', 'PYS' and 'residue'.
R_i	The total revenue achieved by completing project i. Calculated as $R_{i,\text{ramp}} + (SPE_i - s_{ik^{end}})R_{i,\text{PYS}} + (\lvert T\rvert - SPE_i)R_{i,\text{residue}})$
C_{ik}	The total cost incurred by phase k of project i.

Table 3.2: Definition of the parameters used in the discrete deterministic model.

Now that the variables and parameters are defined, the discrete deterministic model follows:

$$\underset{x}{\text{maximize}} \quad \sum_{i=1}^{|\mathcal{I}|} x_{ik^{end}} \left(R_i - \sum_{k=1}^{|\mathcal{K}|} c_{ik} l_{ik} \right), \tag{5a}$$

$$\text{subject to} \quad \sum_{i=1}^{|\mathcal{I}|} \sum_{k=1}^{|\mathcal{K}|} h_{ikj} c_{ik} \leq b_j, \qquad j \in \mathcal{T}, \tag{5b}$$

$$x_{ik} \geq x_{i(k+1)}, \qquad i \in \mathcal{I}, k \in \tilde{\mathcal{K}}, \tag{5c}$$

$$s_{ik} + l_{ik} \leq s_{i(k+1)}, \qquad i \in \mathcal{I}, k \in \tilde{\mathcal{K}}, \tag{5d}$$

$$s_{i1} \geq t_i^{min}, \qquad i \in \mathcal{I}, \tag{5e}$$

$$s_{ik^{end}} \leq SPE_i, \qquad i \in \mathcal{I}, \tag{5f}$$

$$\sum_{\forall j} h_{ikj} = x_{ik} l_{ik}, \qquad i \in \mathcal{I}, k \in \mathcal{K}, \tag{5g}$$

$$j h_{ikj} + 1 \leq s_{ik} + l_{ik}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{5h}$$

$$j h_{ikj} \geq z_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{5i}$$

$$(SPE_i - l_{ik}) h_{ikj} \geq z_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{5j}$$

$$z_{ikj} \geq t_{min} h_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{5k}$$

$$s_{ik} - t_{min}(1 - h_{ikj}) \geq z_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{5l}$$

$$s_{ik} - (SPE_i - l_{ik})(1 - h_{ikj}) \leq z_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{5m}$$

$$x_{ik}, h_{ikj} \in \{0, 1\}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{5n}$$

$$s_{ik} \in \mathcal{T}, \qquad i \in \mathcal{I}, k \in \mathcal{K}. \tag{5o}$$

The objective function (5a) represents the net profit of the projects included in the portfolio. By construction, the financial contribution of phase k of project i only changes the objective value if it is chosen to be included in the optimal portfolio, or equivalently if $x_{ik^{end}} = 1$.

The budget limitations are enforced through constraint (5b), which states that the sum of all phase costs currently in progress should not exceed the total budget for each year.

A phase cannot be started if the previous phase hasn't concluded, and is enforced through constraint (5c). If phase k of project i is not chosen to start, then the corresponding binary decision variable x_{ik} will be zero and thus the decision variables for the following phases can not be set to one.

Similarly, the time $s_{i(k+1)}$ when phase $k+1$ of project i starts can not be earlier than the starting time of the previous phase s_{ik} and the time required to complete phase k of project i. This is enforced with constraint (5d).

A project cannot start before the patent discovery time t_i^{min} and it is also important to finish the final phase of a project before significant pattern expiry, which is enforced by constraints (5e) and (5f), respectively.

The constraints (5g), (5h), and (5i) serve to limit the tracker variable h_{ikj} to behave as described previously.

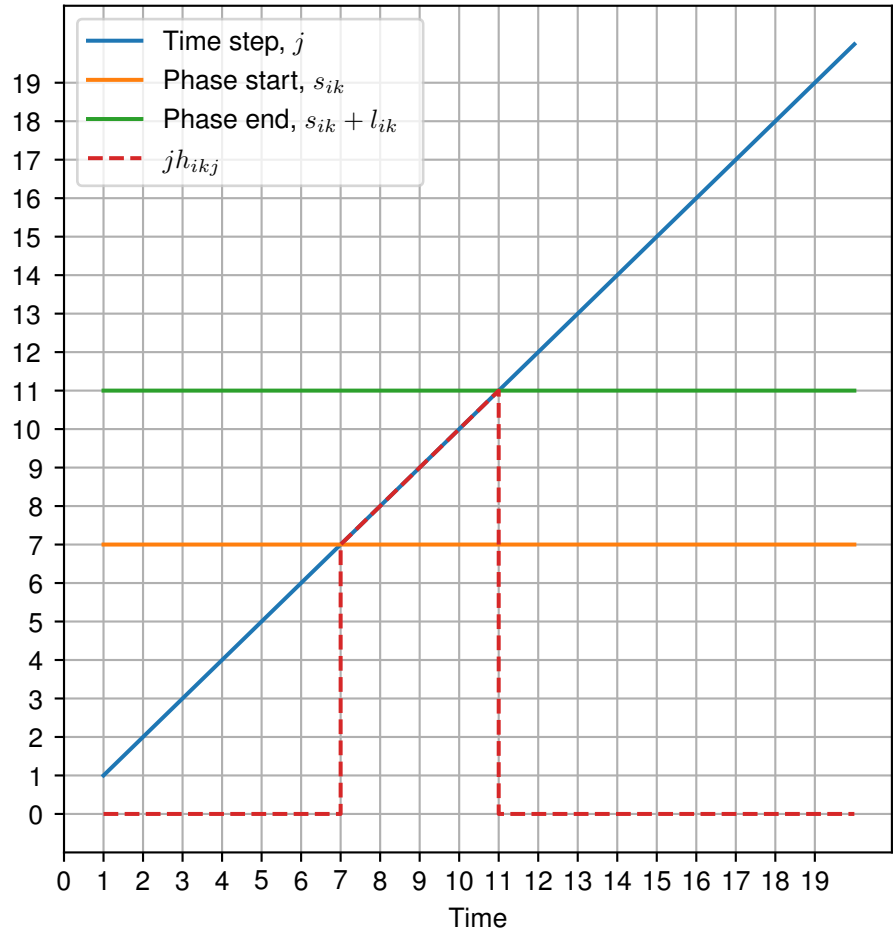

Figure 3.1: Illustration of the product jh_{ikj} bounded by the two constraints (5h) and (5i).

More specifically, constraint (5g) ensures that the total sum of the tracker variables for each phase k in a project i is equal to the length of the time-steps required for that phase.

Constraint (5h) ensures that all the tracker variables h_{ikj} are equal to zero after the last time index j in which phase k of project i is active. The right hand side corresponds to the point in time where this particular project phase ends, which means that the tracker variable must be equal to zero for all j which are larger than the phase time length for the inequality to hold. Similarly (5i) expresses that all tracker variables which correspond to a time earlier than the starting time for the particular project phase must be zero. The constraints on the tracker variables is best illustrated by figure 3.1. This shows the product jh_{ikj} bounded by the two constraints (5h) and (5i). In this example the phase starts at year 7 and has a length of 4. Forcing the tracking variable to zero outside the two constraints ensures that the variable will not track a phase outside of the allowed time-span, while constraint (5g) also ensures that the model cannot ”skip” any time steps and thereby lowering the cost of the project. The effect is that the tracking variable is 1 for all time-steps required to cover the year 7 to 11, and zero everywhere else.

As for the z_{ikj}-variable, it is of interest to force it to be equal to the starting date of phase k of project i at the same set of time indices j where $h_{ikj} = 1$. For this purpose the constraints (5i), (5j), (5k), (5l) and (5m) are used.

Constraint (5j) bounds z from above to ensure that the starting time of any given phase in a given project does not end before the SPE date, ensuring that at least some time is spent in the most profitable PYS phase.

The set of time indices j for which $z_{ikj} = s_{ik}$ is also bounded from below by the inequality (5k), by ensuring that it is at least equal to the t^i_{min} value at the indices j for which $h_{ikj} = 1$.

If $h_{ikj} = 0$, then z_{ikj} is also forced to be equal to 0 by (5j) and (5k). Constraint (5l) is then positive and constraint (5m) takes a negative value.

However, if $h_{ikj} = 1$, constraint (5m) ensures that $z_{ikj} = s_{ik}$ since the left-hand side is equal to 0 in that case.

3.1.1 Considerations when modelling project revenue

As previously mentioned in the introduction, a major component of the pharmaceutical portfolio problem is the technical risk meaning that projects may either fail or succeed. In the deterministic model we assume that every project will succeed and instead attempt to model the technical risk by using the expected value of the revenue, as seen in Section 2.4.

In the model, the revenue from a project is multiplied with the probability to succeed which implies that a project will always succeed and when solving the model, the objective function does not actually reflect the real profit of the portfolio, but rather just the expected value of it.

Because of the stochastic nature of the technical risk and the varying scenarios the outcome of projects introduces, it is hard to model the profit exactly before the outcome is known. Therefore, the decision maker plays an important role in analyzing the results and may have to consider using an utility function to model the revenue instead, if it is desired to include risk management in the model.

The actual value of an example project can be seen in Figure 3.2, while in the current model, the value of the project would be the same if the project fails, but the profit after success would be multiplied by the probability of success.

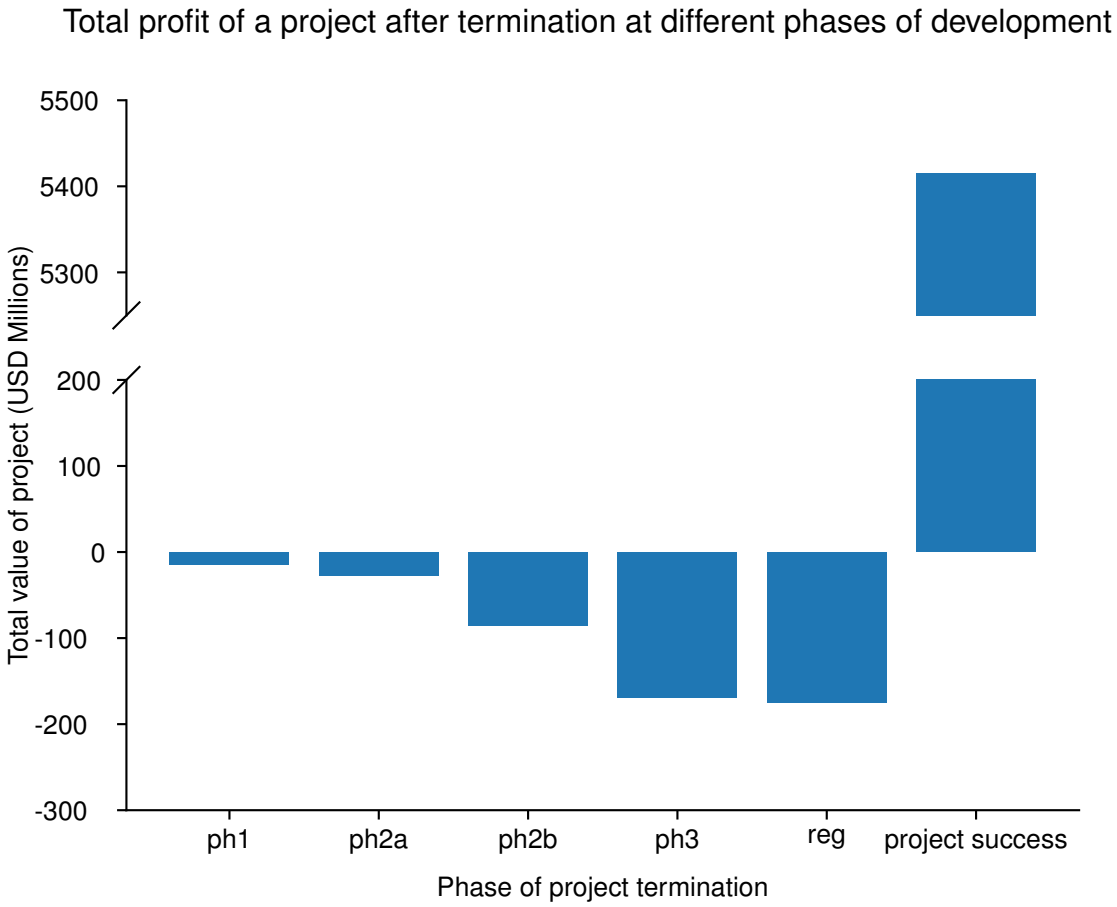

Figure 3.2: Example of the net value of a pharmaceutical project when terminating it in different stages of development. We see that the project is a net loss until the product enters the market, at which point it becomes a magnitude more profitable. It is important to keep in mind that modelling project revenue using the expected value of a project does not necessarily capture this binary aspect of the revenue model and thus care must be taken on behalf of the decision maker.

3.2 Fractional deterministic model

The resolution of the time span is a significant limiting factor regarding the accuracy of the integer deterministic model. The typical phase lengths mean that anything but time-steps in the order of a month or smaller results in quite poor approximations.

At the same time, changing the time-steps from years to months in a model with a time span of 40 years would result in the model optimizing over 40 time-steps to 480 time-steps. Adding the freedom of choice for phase movements quickly results in a model that is too cumbersome to use in any practical sense.

The solution to the issue of time resolution is to allow *fractional* phase lengths. Fraction phase lengths allows us to set the phase length to any given accuracy, while still keeping a smaller number of time-steps to optimize over.

However, the fractional approach requires even more careful tracking of phase costs than previously. In the integer model, the phase cost is equal for each time-step over the phase length. In the fractional model however, two phases can be active during the same time-step, and a phase can be split unevenly over time-steps. For example, a phase can start halfway into one year and continue into a third of the next year, when the next phase will start. As such, it is important that the phase length is split over the years, and the phase cost will instead be defined as the cost for the whole phase divided by the phase length.

The change is made possible by the implementation of a second tracker variable, h_{ikj}^{frac}. It fulfills the same basic function as the original tracker variable, but can instead take on fractional values as opposed to only binary values. Many of the constraints on this tracker variable are the same: if h_{ikj} is 0, then h_{ikj}^{frac} is also zero, and if h_{ikj} is 1, then h_{ikj}^{frac} is higher than 0, but no bigger than 1. The same constraints as before ensure a sequential line of non-zero elements, with additional constraints to only allow the first and last elements to be smaller than 1. If a phase covers three years, the middle year must be fully covered in order for the model to be correct. The constraints enforcing the fractional tracker variable is illustrated in figure 3.3, where the discrete tracker variable h_{ikj} is used as a base, slightly modified from the discrete model in order to cover the correct number of years. Two linear constraints enforce that the edges of the fractional variable are limited so as the phase does not cover more of the year than what is dictated by the phase start and phase end. The phase here starts at 7.5, with a phase length of 5.2. The upper bounds correctly limit the fractional tracker variable in the same way as the two bounds in figure 3.1.

These changes are modelled by additional constraints which together with the integer deterministic model make up the fractional deterministic model;

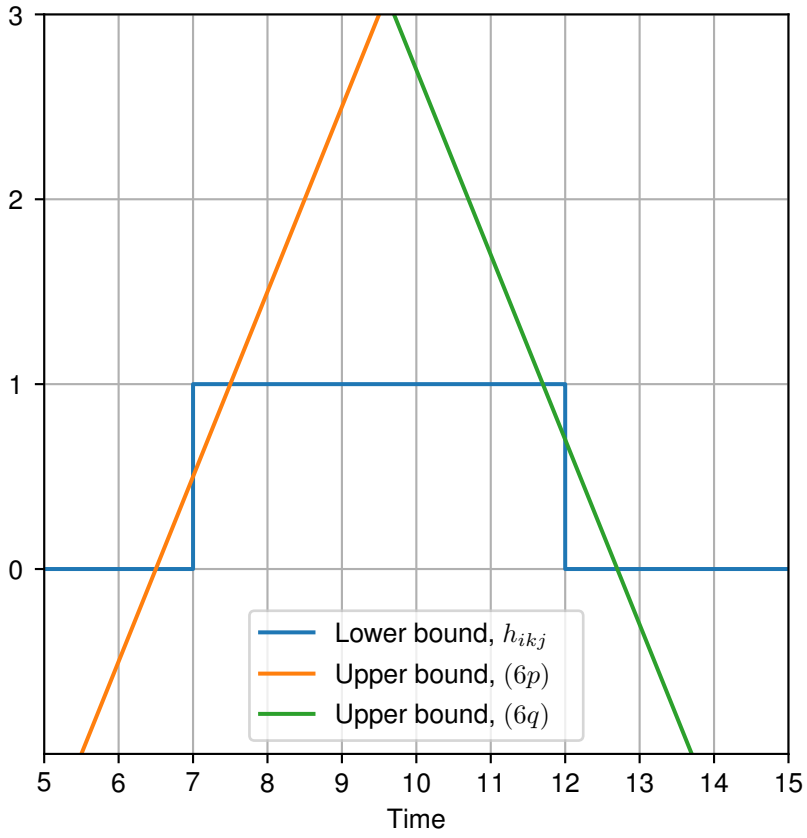

Figure 3.3: Illustration of the discrete tracker variable h_{ikj} bounded by two linear constraints. The values bounded by these lines is the allowed values for the fractional tracker variable h_{ikj}^{frac}.

$$\underset{x}{\text{maximize}} \quad \sum_{i=1}^{|\mathcal{I}|} x_{ik^{end}} \left(R_i - \sum_{k=1}^{|\mathcal{K}|} c_{ik} l_{ik} \right), \tag{6a}$$

$$\text{subject to} \quad \sum_{i=1}^{|\mathcal{I}|} \sum_{k=1}^{|\mathcal{K}|} h_{ikj}^{\text{frac}} c_{ik} \leq b_j, \qquad j \in \mathcal{T}, \tag{6b}$$

$$x_{ik} \geq x_{i(k+1)}, \qquad k \in \tilde{\mathcal{K}}, \tag{6c}$$

$$s_{ik} + l_{ik} \leq s_{i(k+1)}, \qquad i \in \mathcal{I}, k \in \tilde{\mathcal{K}}, \tag{6d}$$

$$s_{i1} \geq t_i^{min}, \qquad i \in \mathcal{I}, \tag{6e}$$

$$s_{ik^{end}} + l_{ik^{end}} \leq SPE_i, \qquad i \in \mathcal{I}, \tag{6f}$$

$$h_{ikj}^{\text{frac}} \leq h_{ikj}, \qquad i \in \mathcal{I}, \tag{6g}$$

$$\sum_{\forall j} h_{ikj}^{\text{frac}} = x_{ik} l_{ik}, \qquad i \in \mathcal{I}, k \in \mathcal{K}, \tag{6h}$$

$$l_{ik} \leq \sum_{\forall j} h_{ikj} \leq 2 + l_{ik}, \qquad i \in \mathcal{I}, k \in \mathcal{K}, \tag{6i}$$

$$j h_{ikj} \leq s_{ik} + l_{ik}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6j}$$

$$j h_{ikj} \geq z_{ikj} - 1, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6k}$$

$$(SPE_i - l_{ik}) h_{ikj} \geq z_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6l}$$

$$z_{ikj} \geq t_{min} h_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6m}$$

$$s_{ik} - t_{min}(1 - h_{ikj}) \geq z_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6n}$$

$$s_{ik} - (SPE_i - l_{ik})(1 - h_{ikj}) \leq z_{ikj}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6o}$$

$$j h_{ikj} + 1 - z_{ikj} \geq h_{ikj}^{\text{frac}}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6p}$$

$$z_{ikj} - j h_{ikj} + l_{ik} \geq h_{ikj}^{\text{frac}}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6q}$$

$$x_{ik}, h_{ikj} \in \{0, 1\}, \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6r}$$

$$h_{ikj}^{\text{frac}} \in [0, 1], \qquad j \in \mathcal{T}, i \in \mathcal{I}, k \in \mathcal{K}, \tag{6s}$$

$$s_{ik} \in \mathcal{T}, \qquad i \in \mathcal{I}, k \in \mathcal{K}. \tag{6t}$$

The objective function (6a) remains the same as in the integer deterministic model, but with one important change. Since the start of a phase is now a variable instead of a parameter, the objective function is no longer linear. However, this issue is circumvented as before by introducing yet another linearization variable, g_{ik} and four new constraints, and using it in place of the product $x_{ik}s_{ik}$.

The budget constraint (6b) remains in the same form as well, but the fractional tracker variable is used instead to include fractions of whole years.

Constraint (6g) states that the fractional tracker variable h_{ikj}^{frac} may not be larger than the corresponding binary integer tracker variable h_{ikj}. The consequence of this is that the fractional tracker variable is zero if the integer tracker variable is zero, while it may take any value between zero and one while h_{ikj} is equal to one.

(6i) is used to make sure that the amount of integer tracker variables marks the appropriate number of years to be set to something else than zero in the fractional tracker variable. To illustrate with an example, if the length of a phase is $l_{ik} = 4.3$,

then in total at least five years need to be "marked" with some value other than zero in the corresponding fractional tracker variable.

As in the integer deterministic model, constraints (6j) and (6k) represent the upper and lower bound respectively on which time periods are supposed to be marked by the tracker variable h_{ikj} as active (set to equal one). However, a small shift is made here compared to the integer deterministic model where each constraint is subtracted by one. Assume for example that we have a phase which starts at year 2015.5 and has length 2.75. Then the first time h_{ikj} is set to one is when $j = 2015$ due to (6k), rather than 2016. Similarly, assume that $j = 2018$ while $s_{ik} + l_{ik} = 2015.5 + 2.75 = 2018.25$. Then, according to constraint (6j) all years below 2019 are marked with ones.

Finally, the only remaining added constraints are (6p) and (6q), which exist to make sure that the starting and ending h^{frac}_{ikj} corresponds to the fraction of that year that is occupied by a phase. Note that these constraints are not the limiting factor for j:s for which h is zero, but are only used when $h = 1$.

Mainly the adjustments made to this model serve to reflect the fact that phases may begin and end at arbitrary times, even if the set $\mathcal{T}$ is made of discrete time intervals (such as years).

3.3 Phase shifting

One interesting aspect of the pharmaceutical portfolio problem is "Phase Shifting", namely the possibility to move the start of a project to a later point in time instead of forcing the project to start at the exact moment of discovery. For example, there may be an abundance of projects that are likely to succeed one year and a set of projects that are less likely to succeed in the following years. Then it might be more valuable to delay the start of the better projects instead.

Recall the objective function (5a) where the decision variable x_i for project i is multiplied with the revenue R_i of project i.

If the activities in a project have a set starting and ending time, the objective function is linear since s_{ik} effectively becomes a constant. However if the starting time s_{ik} of phase k in project i is not fixed, then neither is the finish time and s_{ik} becomes a variable instead. In effect, this makes the objective function non-linear.

Consider once again the revenue term in the objective function (5a) which can be written as

$$x_{ik^{end}} R_i = x_{ik^{end}} \left(R_{i,ramp} + (\text{SPE}_i - s_{ik^{end}}) R_{i,\text{PYS}} + (|T| - \text{SPE}_i) R_{i,\text{residue}} \right).$$

Luckily, only one term involves both variables and thus we can focus only on that term, while keeping the rest of the objective function as is.

To linearize the product $x_{ik^{end}} s_{ik^{end}}$, let us introduce another variable $A_{ik^{end}}$ which behaves in the same way as the product, which is enforced by additional constraints.

These resemble the constraints used to linearize the variable z_{ikj} which is the product of the tracker variable h and starting time variable s.

The constraints $A_{ik^{end}} \leq (SPE_i - l_{ik}) x_{ik}$ and $A_{ik^{end}} \geq t^i_{min} x_{ik}$ ensure that $A_{ik^{end}}$ is within the allowed interval for a project to start given that a phase in the project has been started ($x_{ik} = 1$) and zero otherwise.

We also introduce the constraints $A_{ik^{end}} \leq s_{ik} - t^i_{min}(1 - x_{ik})$ and $A_{ik^{end}} \geq s_{ik} - (\text{SPE}_i - l_{ik})(1 - x_{ik})$. If phase k in the project i is chosen then $A_{ik^{end}}$ is forced to be equal to the starting time of that phase. If the project is not selected then $A_{ik^{end}}$ is forced to be zero.

With this linearization in mind the revenue part of the objective function (5a) may be restated as

$$x_{ik^{end}} R_i = x_{ik^{end}}(R_{i,ramp} + (\text{SPE}_i)R_{i,\text{PYS}} + (|T| - \text{SPE}_i)R_{i,\text{residue}}) - A_{ik^{end}} R_{i,\text{PYS}}.$$

The last step is adding a constraint $s^{param}_{ik} - t_{shift} \leq s_{ik} \leq s^{param}_{ik} + t_{shift}$, where the s^{param}_{ik} is a parameter signifying the original start date, and t_{shift} is a parameter modelling how far forward and back in time the phase can be shifted. t_{shift} can be modified to only allow shifting forward in time, or contain shifts specific for each phase and project. For simplicity and generality, the model described here shifts allows the same magnitude of shift forwards and backwards in time for all phases and projects. With this, phase shifting may be introduced into the model while still keeping the objective function linear, at the cost of extra variables and constraints. Introducing these constraints essentially allows the optimization process more flexibility when selecting projects. As an example there might exist two projects with a combined cost just above the yearly budget. Without phase shifting, the optimizer can only choose one of them, disregarding the other. This does not only result in potentially lower revenue, it also leaves a large portion of the yearly budget unused. The same situation with phase shifting would allow for one project to be moved slightly backwards or forwards in time, letting part of the cost be absorbed by the budget for another year, and allowing both projects to be chosen and the budget more efficiently used. It is important to remember that the phase shifting is only concerned with the yearly budget and the cost of the projects, no other aspects are considered in the process. Indeed, if the budget and time constraints allowed the optimizer would attempt to place all projects as early as possible, since this would automatically generate the highest revenue. This is of course infeasible in reality, and it is important the phase shifts are kept within an acceptable range. As each project can have its own individual parameter describing the phase shift, a practical use of phase shifting would likely be to only allow a select few projects deviate from their set start time by a small amount, either backwards or forwards. In that way the end user of the model can fine-tune the behaviour to correspond to a specific set of real-life circumstances.

3.4 Stochastic model

The deterministic model assumes that phase lengths, phase costs and phase revenues are all fixed and predetermined. However, in the real world, these can all vary due to unforeseen consequences and to model this a stochastic approach is required. While the changes are straightforward in terms of text and additional constraints, it is important to stress that the addition of stochastic elements fundamentally changes the model, resulting in non-linearity and a far higher computationally complex model.

In this model, the phase cost(c_{ik}), phase length (l_{ik}) and the revenue (R_{ik}) are all stochastic variables with an arbitrary stochastic distribution. In practice, the

distribution is typically chosen to be something like log-normal or triangular, but it is kept undefined in this sections in order to present more general results and not focus on details.

Beginning with the phase length, the stochastic addition means that scheduling the phases now becomes an issue. One can generally assume that the starting point for the phase, while being a variable, not stochastic and thus will not change once decided. This is typical for the real world scenarios, and a delay in a phase start can be shifted to an increase in the phase length with the same phase start. If one phase takes longer time than planned, it is possible for two phases to collide, resulting in an infeasible proposed solution. A possible approach could be to simply work with the expected value of the length, and in the event of two phases colliding in the real world, simply perform an additional run with updated parameters and variables.

However, this gives no measurable security when choosing projects, and any such collision must be assumed to incur a significant cost, both in time and money. An alternative is then to introduce a buffer variable q, and again work with chance-based constraints. A constraint of the form $P(s_{i,k} + l_{i,k} + q_{i,k} < s_{i,k+1}) > \alpha$ would then make sure that it is very unlikely that two phases collide. However, this comes at the cost of added nonlinear constraints, which might be very computationally expensive, particularly if the phase start variables $s_{i,k}$ are allowed to shift.

A final option is to utilize the distribution of the phase length variables and choose a percentile such that the constraints are not violated with a certain percentage of security. As a short example, imagine a project has eight phases, and one wants to make sure that the phases do not collide with a 95% probability. Using fixed phase starts but uncertain phase lengths, this effectively results in seven possible collisions between the phases. These are all stochastically independent from each other, which means that the total chance of all phases having less or equal the length is given by $p^7 = 0.95$, which solves for $p \approx 0.9927$. Thus, one can exchange the stochastic phase length for a length from the 99th percentile of the corresponding distribution, and have a 95 percent security that no phases will collide.

The great advantage of this is that it results in no added non-linear terms, since the percentile calculation can be done before the optimizer starts. However, when realized this method will most likely result in large gaps between the end and start of phases. While it is a very safe approach, it might not be a satisfactory solution in the real world. Thus, there is once again the question of the risk versus gain when choosing stochastic percentiles. It is important to remember here that the optimizer cannot make any decisions, as this leads into recourse theory. The optimizer can only deliver the most high-value solution that does not violate any constraints, but this does not necessarily translate to what happens in the real world scenario. Perhaps a project ends earlier than expected by the model, and if so the optimizer must the be run again, since the information it based the earlier run on has been altered.

The stochastic revenue on the other hand is straightforward to evaluate, since there are no constraints that take it into account, only the objective function. It is therefore enough to take the expected value of the revenue, as there are no choices the optimizer can make that impact the revenue beyond the objective function. Lastly, the cost is the one that most directly impacts the choice of other projects and the constraints. The phase length can be handled before the optimizer even

starts, and the revenue cannot be directly influenced in the model. The cost for each year, however, is a result of the project chosen by the optimizer, and is therefore a combined probability distribution from all the separate stochastic cost variables from the chosen projects in the model. The constraint for the budget used in the stochastic model is

$$\mathbb{P}\left(\sum_{i,k} h_{ikj} c_{ik} \leq b_j\right) \geq \alpha, \tag{7}$$

meaning that the budget for each year must not be violated with probability of α. Note that the constraint depends on the tracker variable h_{ikj}, which in turn depends on the projects which are chosen to be included in the portfolio. Because of this it is not possible to compute the cumulative distribution function F_Φ for the random variable $\Phi = h_{ikj} c_{ik}$ before optimization starts.

In addition, in many cases direct computation of the combined probability distribution is not possible, as it depends on the stochastic variables themselves. For example, the log-normal distribution is widely used in pharmaceutical developments, but a sum of log-normal variables has no closed-form probability distribution. This means one must approximate the distribution, which must also be done by the optimizer at run time. Regardless of the distribution, the budget constraint cannot be evaluated with stochastic variables without also including non-linear terms in the optimization model. The inclusion of these non-linear constraints requires a different solver than previously used, and fundamentally changes the model with regards to solution time and optimality.

To conclude this section, there is no one way to model the stochastic terms. As with all stochastic models, a choice has to be made regarding the acceptable risk versus gain. While some parts of the model can be dealt with beforehand, such as the phase lengths, the inclusions of stochastic variables such as the cost cannot be so easily avoided. Alternative way to handle these non-linear aspects will be discussed further in chapter 4.

3.5 Recourse background and theory

In stochastic programming, a major limiting factor is the fact that random variables are unknown at the point in time when a decision needs to be made. In the earlier sections, this uncertainty was treated in an indirect fashion by incorporating the probabilities of project success in the total revenue of a project.

Another approach which may offer better results is known as recourse modelling, in which the goal is to allow the decision-maker to alter their initial decision after some uncertain information becomes known. For the pharmaceutical portfolio problem where the risk of projects failing is high, a recourse model may provide better results than the ones developed in the previous sections of this paper.

In this section, we introduce some basic recourse theory and then proceed to summarize some of the previous work which has been done on this subject and give suggestions for further reading and implementation for Captario SUM.

3.5.1 The newsvendor problem

A typical example of recourse is the known "Newsvendor Problem", described in greater detail by Birge and Louveaux [3, pp. 15–17].

Consider a vendor who buys m magazines at the beginning of each day where m is bounded from above by some maximum amount of magazines U. The vendor wishes to maximize the profit from selling the magazines, given a uncertain demand ξ which is not known explicitly when the magazines are bought. However, from previous experience the vendor has approximated the demand with a cumulative distribution function F_ξ.

Each magazine costs c to order from the supplier, but may be returned to the supplier at the end of the day with a return price r, where $r < c$. The profit for selling a paper is q.

Let us define y as the amount of magazine sales and w as the number magazines returned to the supplier.

We may then formulate the problem, where dependence on the random variable ξ is denoted as an argument to a variable/parameter, as follows:

$$\begin{aligned} \text{maximize} \quad & f(m) := -cm + \mathbb{E}_\xi[Q(m,\xi)], && \text{(8a)} \\ \text{subject to} \quad & 0 \leq m \leq U. && \text{(8b)} \end{aligned}$$

The expression $Q(m,\xi)$ is the objective value describing the profit from selling y newspapers and returning w newspapers to the vendor at the reduced cost, modelled as a stochastic linear program,

$$\begin{aligned} \text{maximize} \quad & Q(m,\xi) = qy(\xi,m) + rw(\xi,m), && \text{(9a)} \\ \text{subject to} \quad & y(\xi) \leq \xi, && \text{(9b)} \\ & y(\xi) + w(\xi) \leq m, && \text{(9c)} \\ & y(\xi), w(\xi), m \geq 0. && \text{(9d)} \end{aligned}$$

Note that the actual demand ξ is not known until the end of the day and thus a decision needs to be made knowing only F_ξ. Therefore, the objective function is formulated as a sum of the value gained (or lost) by making a decision before the information is known and the expected value of the recourse decision after the uncertainty has been resolved, given the first decision.

3.5.1.1 Two-stage recourse program

As seen in the example, the notion of *stages* is central to a recourse model, which we define below.

Definition 3.5.1 *(First and second-stage decisions)*

Let ξ be a random variable. The decisions which have to be taken before ξ is realized are defined as first-stage decisions, while the decisions which are taken after the experiment are called second-stage decisions.

Let ω be a vector of random parameters and denote the set of first-stage decisions by x and the set of second-stage decisions by $y(\omega, x)$.

Then a two-stage program with simple recourse can be formulated as follows:

$$\begin{aligned}
\text{minimize} \quad & c^T x + \mathbb{E}_\omega[\min q(\omega)^T y(\omega, x)], && \text{(10a)}\\
\text{subject to} \quad & Ax = b, && \text{(10b)}\\
& T(\omega)x + Wy(\omega, x) = h(\omega), && \text{(10c)}\\
& x \geq 0, y(\omega, x) \geq 0. && \text{(10d)}
\end{aligned}$$

Here all terms which are dependent on the random variable ω are defined as a function of ω. $T(\omega)$ is a matrix of stochastic coefficients and W is a matrix of constant coefficients.

Note that given a realization of the random variables ω, the second-stage term simply becomes the solution to a linear program. This makes the computational difficulties of recourse programming evident, since to evaluate the expectation explicitly, one needs to solve a linear program for every outcome of ω which may take on infinitely many values in the general case.

3.5.1.2 Multistage programs with simple recourse

Unfortunately, two-stage programs are not sufficient to model the pipeline R&D problem since the random variables involved (specifically the technical risk) are revealed at different points in time and preferably one would like to be able to perform a recourse decision at every such point in time.

A natural extension of a two-stage program is a multistage program, where the purpose is to split the problem into many stages, in which one or more random variables are realized and a recourse decision has to be taken.

Let us define the different stages as the set $\mathcal{T} = 1, ..., H$. These correspond to the points in time where a decision is made based on the realization of some subset of the random vector of parameters ω. Note that these stages do not have to correspond to the actual time discretization that is made in a model, even though the notation is similar.

Let us define the set of decisions to be made at stage t to be the vector x_t and the information known about the random variables at stage t as ω_t. T is a constraint matrix which depends on ω in some fashion, whereas W does not.

Then we formulate the multistage recourse problem as a generalization of the two-stage model as follows:

$$\begin{aligned}
\underset{x_1}{\text{minimize}} \quad & c_1 x_1 + \Phi(\omega, x), && \text{(11a)}\\
\text{subject to} \quad & W_1 x_1 = h_1, && \text{(11b)}\\
& T_1(\omega_2)x_1 + W_2 x_2(\omega_2) = h_2(\omega), && \text{(11c)}\\
& ..., && \text{(11d)}\\
& T_{H-1}(\omega_H)x_{H-1}(\omega_{H-1}) + W_H x_H(\omega_H) = h_H(\omega), && \text{(11e)}\\
& x_1 \geq 0, x_t(\omega_t) \geq 0. && \text{(11f)}
\end{aligned}$$

The objective function is split in parts - the deterministic part (c_1x_1) and the recourse part $\Phi(\omega, x)$ which is defined as

$$\Phi(\omega, x) = \mathrm{E}_{\omega_2}\left[c_2x_2 + \mathrm{E}_{\omega_3|\omega_2}\left[\min_{x_3} + ... + \mathrm{E}_{\omega_T|\omega_1...\omega_{T-1}}\left[\min_{x_T} c_T x_T\right]\right]\right]. \quad (12)$$

The expression Φ describes the recursive relationship between stages, where the base case is when stage H is reached, since all information is known and we may solve an integer linear program.

The operator $\mathbb{E}_{\omega_3|\omega_2}[f(\omega_3)]$ can be read as the expectation of the stochastic function f w.r.t ω_3 given a particular realization of ω_2. This notation is used to emphasize the importance of the decisions made in the past stages, which in turn depend on the realizations of the random variables in each stage t.

When it comes to the constraints, a set of constraints is defined for each new stage, after new information is revealed. This is to ensure feasibility at each stage. Note that in this formulation, only the state of the previous stage is considered when making a decision in the current stage.

This is a reasonable assumption to make given the pipeline R&D problem, since the major recourse action for each project is to either do nothing (if the current phase succeeds) or to choose another viable project to replace a failed project with. Therefore the information required to make a decision only depends on knowing the status of the previous and current phase in every project.

As in the case with two stages, an explicit evaluation of this model requires a solution of a linear program for every possible realization of the random variables at different stages, which is computationally difficult when the model includes many projects.

3.5.2 A metaheuristic approach to recourse

The specific pharmaceutical portfolio model studied in this thesis is an example of so called stochastic Resource Constrained Project Scheduling Problem (RCPSP) and has been examined in several publications from different points of view, both in a more general sense [4, 9, 16] and through limiting the problem by only considering a subset of the uncertainties such as activity (phase) duration [18]. An interesting assessment of the uncertainty in the portfolio problem was studied by Subramanian et al [5] and then further elaborated on in [6].

The idea is to repeatedly simulate the project portfolio as it propagates in time and adjusting the decision vector as new information about the system is revealed. Every time an activity finishes, the state of the system is updated and a deterministic integer optimization problem is solved to determine the new optimal state of the system. Although this method is not guaranteed to find a globally optimal solution, the state of the portfolio is always known in every iteration of the simulation and thus a lot of useful data can be gathered and analyzed. Besides determining which projects are the best to pursue and how to distribute the resources between them, it is also possible to determine the risk profile of these choices,see which resources are limiting and more.

In this particular article, a generalized resource constrained project scheduling problem (RCPSP) is defined which is similar to the one studied in this report in many aspects. First a deterministic model is defined and then it is extended with a stochastic formulation.

The deterministic RCPSP problem is defined by a set of projects which are composed of activities, which corresponds to phases in our formulation. These activities are bound by precedence constraints which describe the order in which the activities must be completed. An activity may not start until one or several activities preceding it have been completed. Each activity consumes an amount of some limited resources. Finally, the projects also have some kind of measure of performance which determines how "good" a project is.

As in our formulation, each task is assumed to be successful with a fixed duration, cost and reward. The deterministic RCPSP is then extended by modelling the uncertainties in the problem by joint chance constraints. The problem is then to choose the optimal way to assign resources to the projects such that the measurement of performance is maximized. The computing architecture, Sim-opt, used to produce solutions has two main parts which communicate with each other: An optimizer and a discrete event simulator.

The discrete event simulator attempts to simulate the state of a specific portfolio realization given the probability distributions of the different parameters. For example, assume there are two projects currently active. Since length of an activity is probabilistic, we have no way of knowing when it ends for sure. Therefore, the system traverses forward in time and listens for events, such as an activity completion. As soon an activity is completed, the system is temporarily halted and the updated portfolio state is sent to the optimizer.

The optimizer receives this new state and updates the model accordingly. For example, when an activity fails, the optimizer is updated to no longer consider this activity. Then, the RCPSP is solved to find the new "optimal" state of the system considering the current state.

The solution corresponding to the updated state is sent back to the discrete event simulator which then continues to traverse forward in time until another event is triggered, upon which the optimizer is triggered again. This continues until the time horizon for the system is reached.

Parallel to this, as each event is triggered and a new state is found, it is recorded by an outer loop for further analysis.

Since the optimal choice of projects in a recourse model heavily depends on the realization of the stochastic variables involved, there is no single "best portfolio" in the sense that one set of projects is the best every time. Instead, the best *decision-making policy* is sought after. The decision-making policy can be thought of as an algorithm which automatically chooses a recourse action based on the state of the system. This policy can be as simple as just choosing the projects with the highest net revenue as soon as possible or it can be more complicated.

In the Sim-opt framework, instead of explicitly finding the optimal decision-making policy, the authors implement and evaluate different policies by using different optimization algorithms in the optimizer module. For example, one such policy is to solve a deterministic version of the stochastic RCPSP every time the portfolio state

is updated. Another policy may be to simply place the projects in a priority queue based on some metric and choosing the project with the highest priority every time the system allows, such as if a ongoing project fails or the budget becomes more accommodating.

4

Methods

4.1 Model building and AMPL

In order to build and test the models proposed in Chapter 3, the scripting language AMPL was utilized. AMPL is designed to model optimization problems in a mathematically approachable fashion, and has various syntax and functions tailor-made to this end.

An AMPL model consists of a model file and a data file. The model file fully describes the given mathematical model using indexed sets and coefficients, providing a clear overview of the full model from a single resource. The data file, in turn, contains the specific values for all parameters, such as cost, phase length and probabilities. This separation makes it straightforward to test the model with a different data set and allows for rapid adjustment and solution runs over several different data sets. Examples of these files for the different models can be seen in Appendix B. This allows for a swift evaluation of a model using different parameters. The actual optimization is then done by a *solver*, which is separate software that determines what algorithms that will actually be utilized when optimizing the problem.

The solvers used in this report are CPLEX (for linear integer programs), and COUENNE, for nonlinear integer programs. CPLEX is used to solve linear integer and quadratic programming optimization problems, and is widely used in the industry [20]. For general non-linear problems, the open-source solver COUENNE [19] was used. COUENNE is designed to find global optima even in large non-linear, non-convex problems. However, its use in this thesis came mainly from the fact that it is open source and thus freely available. If needed, AMPL can also be expanded with the GSL AMPL library, which provides common mathematical functions for use in AMPL, such as probability distributions. This is advantageous when handling stochastic constraints and other nonlinear functions.

The focus of this thesis has not been on the solvers but on building the mathematical models. The rest of the chapter is instead intended to give a better overview of what aspects of the models that require a more specialized solution method. Performance will be benchmarked against variations of the models, and not an external standard of some other modeling problem.

4.2 Data generation

Simple data sets are easily written directly in AMPL and used to test and confirm that the mathematical constraints implement the restrictions as intended. The portfolio data is obtained through Captario SUM, where a portfolio of projects in different phases has been generated. The portfolio consists of 51 projects in varying stages of completion, meaning that some phases are already done while some projects have not yet started. The portfolio contains all relevant information about the projects, such as phase length, current phase, the cost and revenue corresponding to each date, the SPE, etc. All stochastic data, such as probability distributions, are also defined in SUM and exported to AMPL.

4.3 Handling of stochastic and non-linear terms

One of the biggest obstacles in the stochastic model is the handling of the stochastic costs. The model requires the budget to not be exceeded with some given probability, which results in the need to find the appropriate joint probability distribution and the corresponding cumulative distribution function for the given set of chosen projects. As mentioned before, this often involves non-linear terms, but a significant obstacle is the joint probability distribution itself. Since the distribution depends on the chosen projects, and these projects are not known before starting the computation, this distribution must then be computed during the run time of the computation. This is not only a non-trivial computation to perform, but is in many cases not possible to compute in closed-form.

For example, given that each variable is normally distributed, the probability distribution of their sum will itself be normally distributed, with a mean and variance equal to the sum of the mean and variance of the original distributions. This is nonlinear, but the computation of the probability distribution is straightforward to find.

However, this is a special case and many distributions do not allow solutions of this kind. The sum of log-normally distributed variables is not itself log-normally distributed and can not be expressed in a closed form. Various approximations instead have to be used at different parts of the distribution. Going one step further, there is often a mix of different distributions used, such as normal, triangular, log-normal, and others. The computation of the resulting probability distribution must be obtained using integration or other numerical methods, which is computationally taxing.

An alternative is to approximate each distribution with a suitable discrete probability distribution, by using a series of rectangular approximations. An example can be seen in Figure 4.1, using a normal distribution. This approximation can be done to any desired level of accuracy, again at the cost of computational complexity - a higher accuracy corresponding to more rectangles. The joint probability distribution is then gained from a combinatorial problem determining all possible outcomes. Assuming that a joint probability distribution could be obtained, further simplifications could be made by replacing the non-linear terms with some linear approximation. The

main advantage would then be that if the distribution can be efficiently generated, then the rest of the problem will be stated in linear terms.

All of the discussed solutions above are cumbersome at best to implement in most optimization routines, given the need to generate a new distribution in each step. The addition of any approximation also means that a measure of the accuracy of the implementation, and by extension the solution obtained, must be developed.

A more computationally viable approach would be to treat the set of variables as alternatively fixed parameters and free variables. For example, one might run a simulation with all stochastic values set to their expected value (risk-averse, if needed), and observe which projects are the most profitable ones. Then, one could focus only on this subset of problems, determine the necessary probability distribution outside of the optimization run, and then run the program again, only this time using the stochastic constraint regarding the budget.

The same approach could be used in the deterministic model when looking at the available span to shift projects — a large span results in a significant combinatorial space, which is countered by the smaller set of projects under consideration. However, applying methods such as these then results in that the optimization run is no longer guaranteed to find an optimal solution, and it is important to be aware of what simplifications are made and how they might impact the solution at hand. The advantage is that projects of this kind are unlikely to be overly sensitive to small changes. This follows from the fact that the objective function is linear, and a project being moved back n months will simply result in n months more revenue for the given project. The cost would also be split continuously between the revenue years, meaning that except for some rare corner cases, the solver would not be able to choose one whole additional project. And if one such corner case is suspected, it would be possible to simply raise the budget slightly to see if the solver is at the border of a new solution. An important note is that this reasoning holds only for the optimization model, and there is no guarantee that the real-life project does not have some dependency that is not present in the simplified mathematical model.

In closing, it is important to note that the specifics of the approximations implemented requires a solid knowledge of the behaviour of the model, and more exact solution methods are outside of the scope of this report, and best left up to the each specific implementation of the models.

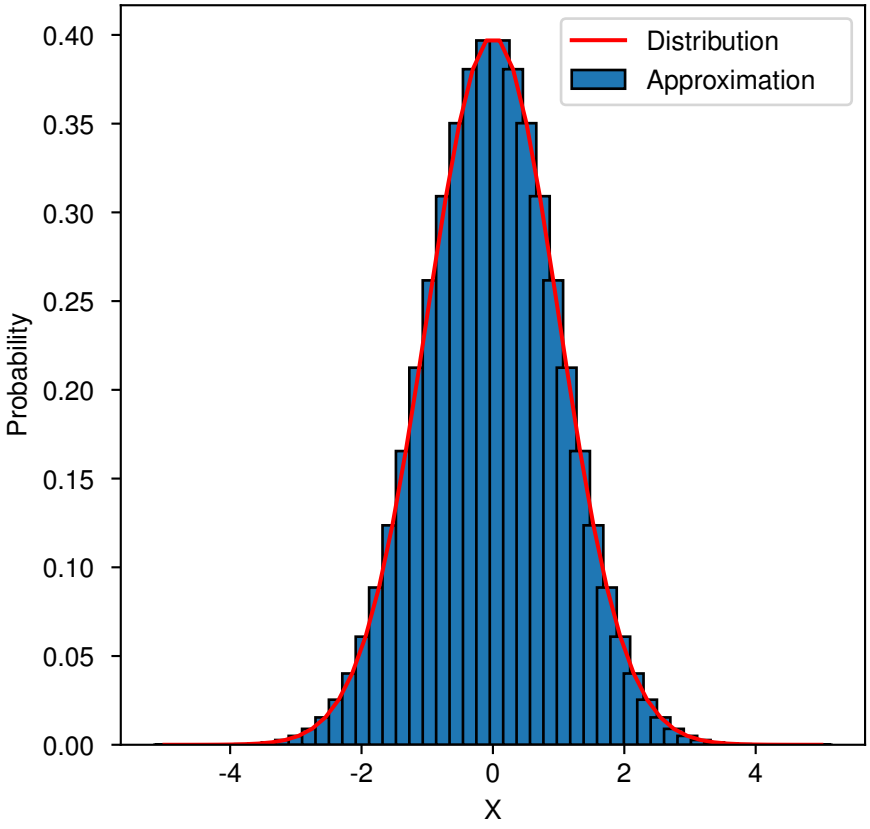

Figure 4.1: A plot of the standard normal distribution along with a discrete approximation.

5
Results

5.1 Proof of concept

Since an actual portfolio may contain over 50 projects, with corresponding lengths, costs and other parameters, the size of the problem makes it infeasible to try to solve manually. However, it is important to verify that the solver actually finds the optimal solution and that all the constraints are enforced as intended. To that end, several sets of dummy portfolios were made, intended to confirm that the various constraints dictating phase starts, costs, phase shift and so on were not violated. The dummy portfolios were of limited size, only three projects, making it easy to construct a solution by hand and verify that the solver generates the same solution. The full test model given from Captario consists of 51 projects, and the full set of parameters for this problem is given in Appendix A.

5.1.1 Sample problems

This section is intended to confirm that separate constraints are enforced correctly. Aspects such as integrity of phases, minimum dates and selection of budget must work as intended. The base dummy portfolio can be seen in Table 5.1. The goal is to verify that given a set of three projects, the model chooses the most profitable set, but does not violate the structure of the phases and project in doing so. The sample parameters are chosen as integer values and care is taken to avoid any splitting over time units, in order to make the results more comprehensible for the reader. Table 5.1 is chosen as the base, and each given test case modifies the base with additional parameters and settings, specified in additional tables. The first set of test runs ensures that the model chooses the most profitable projects. For simplicity's sake each number is given in the unit of USD. In this run the budget is set to \$5 for each year, and the model does not allow for any projects to be shifted forward or backwards in time. Thus, the model does not allow for all projects to be selected. Given the parameters presented in the Table 5.1, the budget allows either project 1 and 2 to be chosen simultaneously, or only project 3. The deciding factor in what projects are chosen should the be the combined revenue and cost for all projects.

The portfolio is designed so that the total revenue, costs included, for Proj1 and 2 is very close to the total revenue for Proj3. Thus, if given a limited budget, then the optimal solution generated should be to disregard Proj1 and 2 and to only select Proj3. By varying the revenue for Proj3 we can confirm that the model indeed follows the expected behaviour, as seen in Table 5.2. Here the revenue is varied, and it is apparent that the chosen projects varies with it.

	Minimum date		
	Phase 1	Phase 2	Phase 3
Proj1	1	2	3
Proj2	1	2	3
Proj3	1	2	3
	Length		
	Phase 1	Phase 2	Phase 3
Proj1	1	1	0
Proj2	1	1	0
Proj3	1	1	0
	Cost		
	Phase 1	Phase 2	Phase 3
Proj1	2	2	0
Proj2	3	3	0
Proj3	5	5	0
	SPE date		
	Proj 1	Proj 2	Proj 3
	2030	2030	2030
	Revenue		
	Ramp	PYS	Residue
Proj1	0	10	0
Proj2	0	10	0
Proj3	0	9	0
	Budget for all years		
	5		

Table 5.1: Set of parameters used for testing the functionality of the fractional model.

The next test run is made by forcibly setting one phase to zero from the start, to confirm that no other phase will be chosen in the model. The same parameters in Table 5.1 are used, but phase two of Proj3 is now set to zero. The results can be seen in Table 5.3, confirming that the project is not chosen even if it would increase the profitability of the portfolio, and that the structure of the phases is not violated. Additionally we can test the phases by setting the budget for the second year to zero, as seen in Table 5.4. Here the model cannot choose any projects, since the phase starts cannot be violated, but there is no budget for choosing a project. However, in Table 5.5, the budget for year two is kept at zero but we allow the phases to start within a span of a year before or after their specified date. In order to illustrate the fractional nature of the model more clearly the costs for the projects have been altered slightly. Here all projects are chosen, and the model also selects all phases as early as possible, in order to maximize the profitability, causing the phases to be split over the years.

	Revenue of Proj3 PYS		
	9	10	11
Proj1	1	0	0
Proj2	1	0	0
Proj3	0	1	1

Table 5.2: Projects chosen when varying the revenue of Proj3, for the portfolio described in Table 5.1.

	Phase 1	Phase 2	Phase 3
Proj1	1	1	1
Proj2	0	0	0
Proj3	0	0	0

Table 5.3: Selected phases for the parameters in Table 5.1, but with the second phase in Proj2 forcibly set to zero, and it's revenue set to $100 USD. As can be seen this results in Proj2 not being chosen, even though it is by far the most profitable one.

	Phase 1	Phase 2	Phase 3
Proj1	0	0	0
Proj2	0	0	0
Proj3	0	0	0

Table 5.4: Selected phases for the parameters in Table 5.1, but with the budget of the first year set to 0. This results in no projects being chosen, since the model cannot violate the phase start of the projects.

	Cost		
	Cost phase 1	Cost phase 2	Cost phase 3
Proj1	2	1	0
Proj2	3	3	0
Proj3	5	3	0
	Start of phase 1	Start of phase 2	Start of phase 3
Proj1	1	3	4
Proj2	1	3	4
Proj3	3.8	4.8	5.8

Table 5.5: Selected projects and corresponding phase starts when allowing for two years of phase shift. The budget for the second year is set to zero, but the phase shifting still allows all projects to be selected. Notice the fractional values of the phase starts when the model is allowed to shift them, along with the fact that the model moves them as early as possible to maximize the revenue.

5.2 Full portfolio – deterministic fractional

The final result of the optimized portfolio is heavily influenced by two fundamental parameters - the yearly budget, and how far the phases are allowed to be shifted in time. In the deterministic model, a higher budget always results in a greater profit and more projects chosen, since all projects are assumed to succeed (and be profitable). An increase in the phase-shift similarly increases the profit by allowing phases to be placed earlier, along with the possibility of shifting projects around so that it is possible to circumvent a limited budget. However, the main drawback is an exponential increase in computational time, since adjustment of these parameters directly influence the space of possible project combinations. The solution time typically grows from a few CPU seconds to upwards of several thousand CPU seconds when going from no phase shift to the maximum phase shift of a year and a half.

5.2.1 Fixed dates and varying budget

In this scenario, the budget started at $25M. This is low enough that only a single project can be chosen. Each next run increases the budget by $125M in 20 steps, until the budget is $3000M per year, which allows all projects to be chosen. Along with the total profit other aspects were measured, such as number of projects chosen, the optimization time and revenue relative the cost. This allows for a closer inspection of the model behaviour.

The results related to the total revenue can be seen in Figure 5.1. As expected, the revenue steadily increases as the budget does. Corresponding to this is the results in Figure 5.2, which displays how many projects have been chosen for each run. The two figures show a close correlation, as is expected. However, these show nothing groundbreaking, since each project is profitable — thus it is always beneficial to choose as many projects as possible.

More interesting is Figure 5.4, which takes into account the total cost for each project. The ratio between revenue and cost shows that the most profitable project is clearly chosen first, and that the relative profit steadily declines with an increased budget. Given that the return on investment is lowered as the budget increases, it would be wise to limit the investment to the most profitable projects. Furthermore, taking into account that in the real-world scenario, many projects would fail, it is clear that the deterministic model still has some usefulness in supporting decision-making.

The last result is Figure 5.3, displaying the time in CPU seconds, taken by the optimizer for each budget run. The plot is noticeably irregular, but the process is clearly faster at the lowest and highest budgets, corresponding to very few options being available, or simply being able to choose all projects. The rest of the data shows a few spikes, most noticeably at around $600M. Comparing to Figure 5.2, this is where around half of all projects are chosen and half discarded. This is then in the midpoint of the two extremes mentioned before, and thus it requires the most computational time.

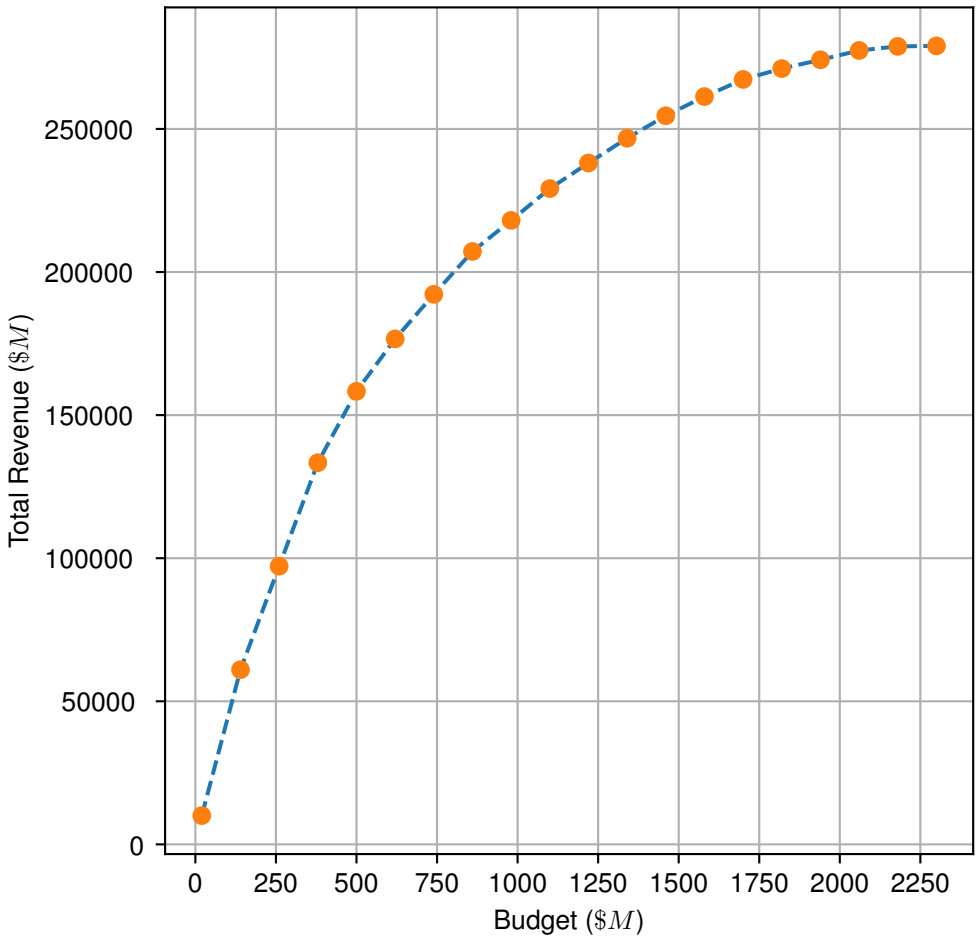

Figure 5.1: Optimal solution from the deterministic fractional model with fixed phase dates. Displayed as the total revenue (i.e, the objective function) against the yearly budget.

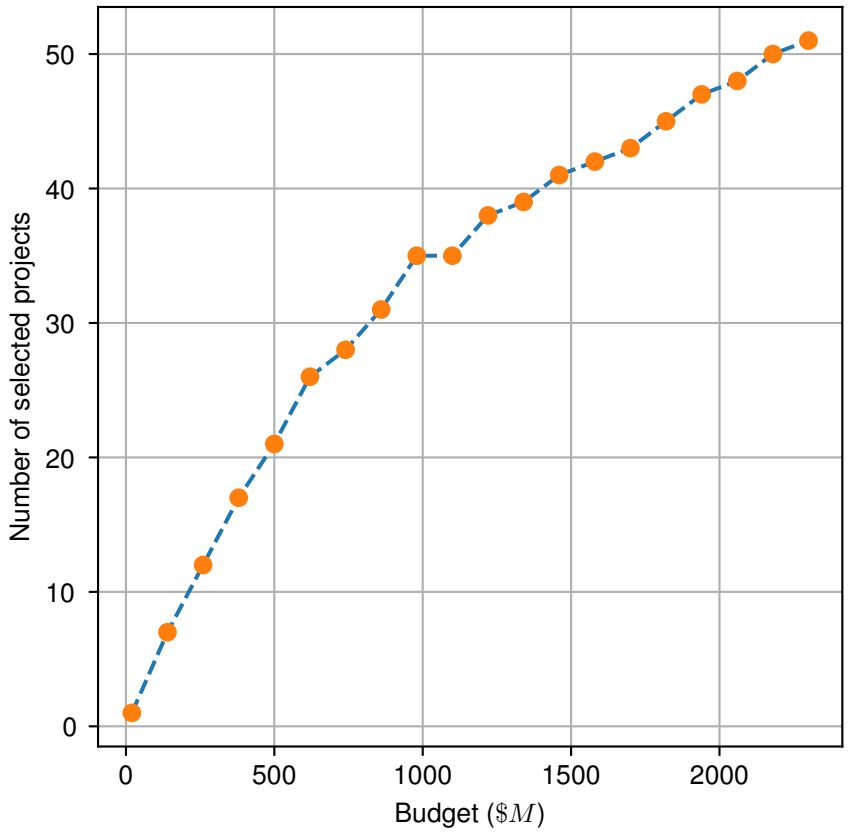

Figure 5.2: Optimizer results from the deterministic fractional model with fixed phase dates. The results are presented as the number of projects chosen against the yearly budget.

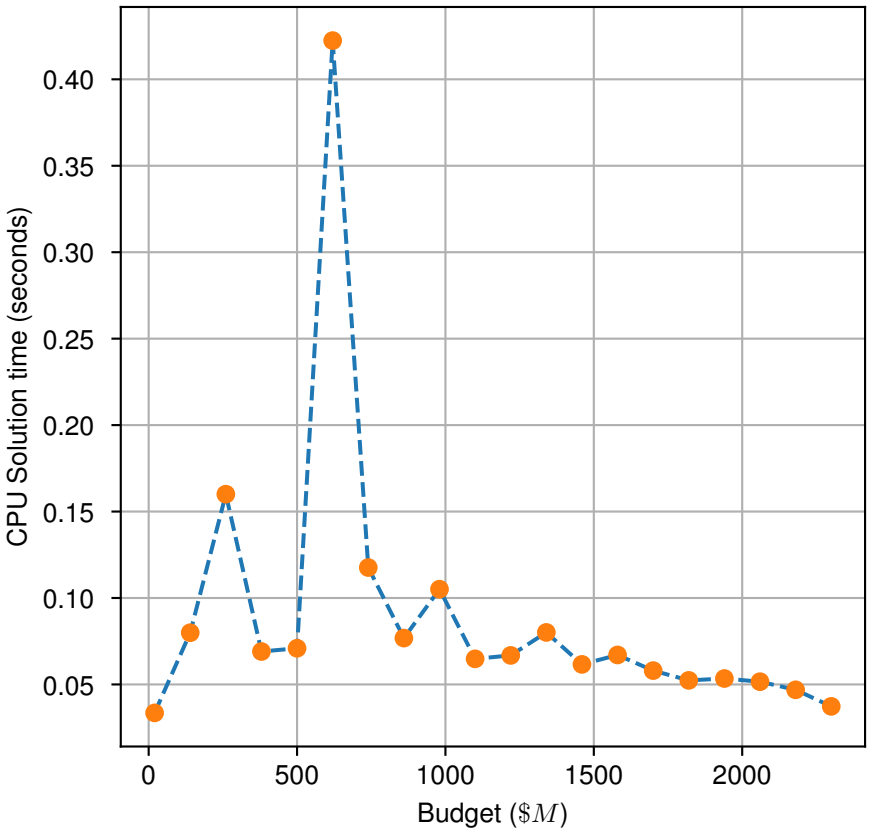

Figure 5.3: Optimizer results from the deterministic fractional model with fixed phase dates. The results are presented as the solution time for each optimization run against the yearly budget.

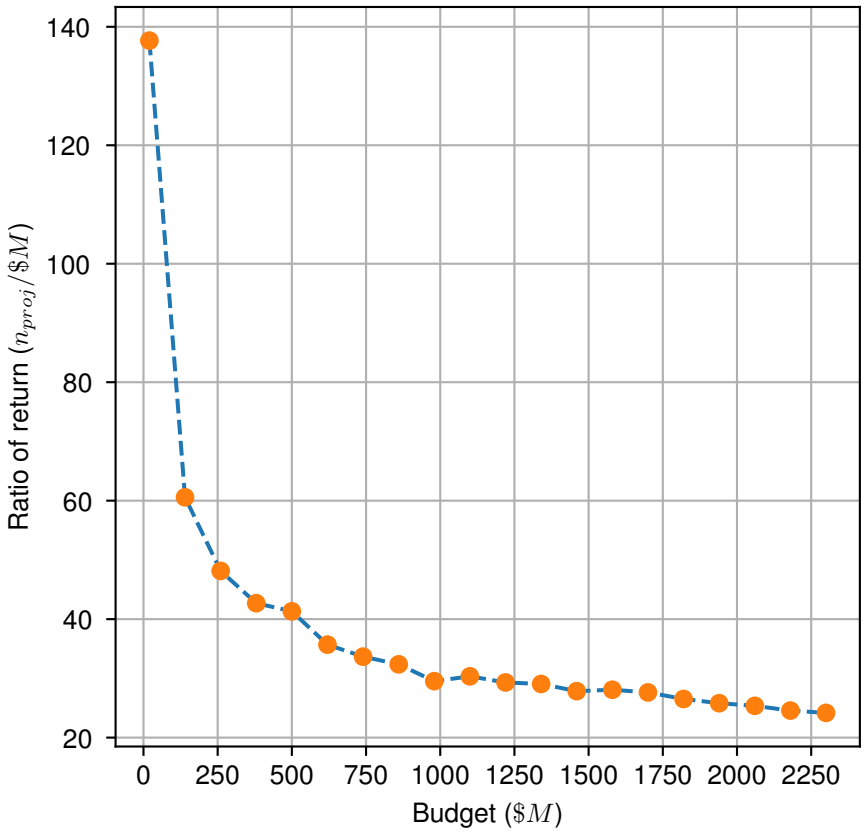

Figure 5.4: Optimizer results from the deterministic fractional model with fixed phase dates. The results are presented as the ratio of total revenue divided by the total cost, against the yearly budget.

5.2.2 Fixed budgets and shifting phases

A few representative budgets were chosen from the previous run, and the process was repeated with a fixed budget. However, the start dates of the phases were now allowed to shift a limited amount of time. A run started at a phase shift of 0 years and then incremented by 0.25 years both forward and backwards in time, until it allowed for a shift of 1.5 years backwards or forwards in time. The same measurements were taken as in the fixed phase scenario, and all runs of different budgets are displayed in the same figure.

Looking at the total revenue, Figure 5.7, an increase in the phase shift generally allows more projects to be chosen. However, the main limiting factor is still the budget. In the $20M budget run, no amount of phase shift allows the model to choose more than one project.

Similarly in the $100M budget run, the number of projects chosen stagnate after one and a half years. Only the $500M budget run show consistent increases, but also has clear plateaus where no additional projects can be chosen. Further analyzing the revenue in Figure 5.8, increasing the phase shift seems to result in a fairly linear increase of the revenue. Even the $20M budget run increases in revenue, even though only one project is chosen. The reason for this is that the phase shift allows for the revenue accumulation to start earlier, and thus one still sees some modest increase in revenue. The ratio of the revenue and cost, seen in Figure 5.6, displays a slightly more irregular behaviour.

As before, the ratio is higher for the runs with the more limited budget. An interesting aspect is that for the $500M budget, and sections of the $100M budget run, the ratio actually decreases with the magnitude of the phase shift. This is due to the inclusion of more projects, and it is clear that the $100M run has the highest ratios at the plateaus of the number of projects. Simply put, the plateaus then correspond to the best configuration of the set number of projects, right before the inclusion of new ones. These figures clearly show that the phase shifting allows for more flexibility and profit for the model, although at a cost. Figure 5.5 shows the time taken by the solver to generate these results. The figure is plotted in logarithmic scale due to the massive differences in magnitude of the solution times. It is clear that the solution time increases exponentially with the magnitude of the phase shift. Interestingly, the solution time does not completely correspond with the previous result in the fixed-phase runs. In the fixed-phase runs the budget with the largest solution time was around the $600M budget, while the largest solution time for the phase shift model was for the budget of $1100M budget. This indicates that the solution space of the model is more complex than initially surmised. A possible explanation is that at the $1100M budget contains many edge cases and boundary issues regarding the project that leads to a larger selection of possible options.

5.3 Stochastic model

The stochastic model was not used for any results. This was partly because the approximations used in the building of the model were not of a high enough quality to run any serious tests on. Another problematic aspect was the optimization time.

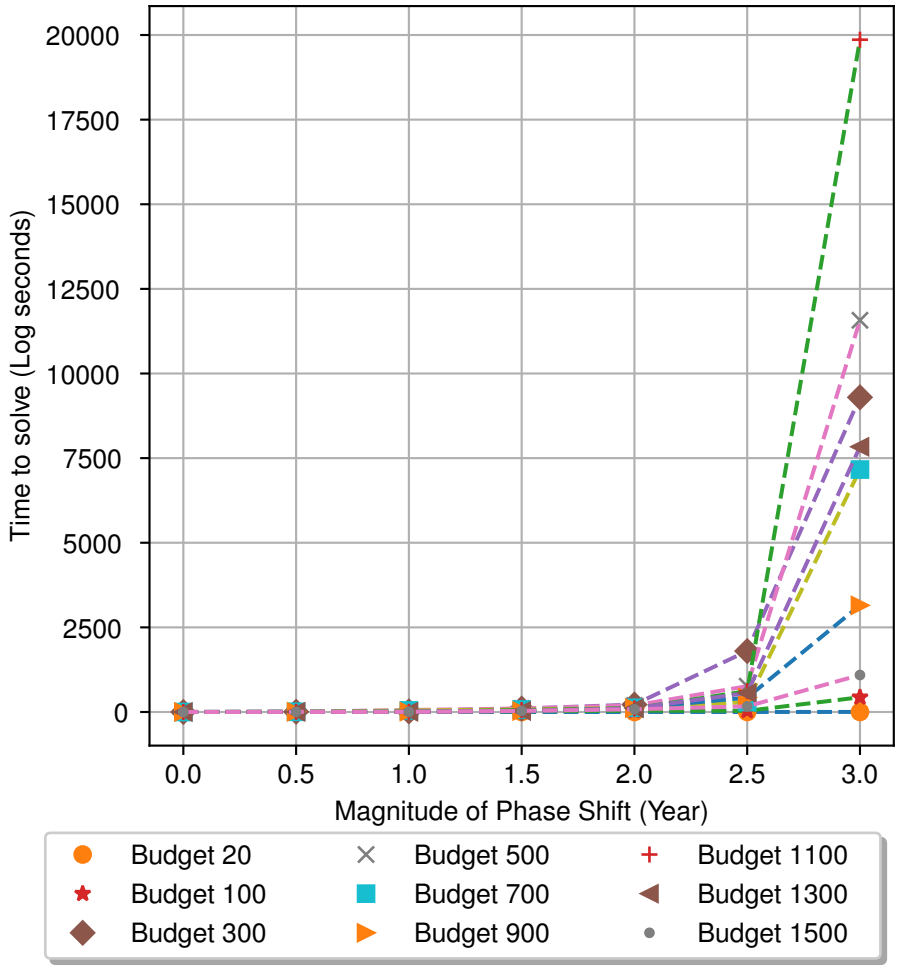

Figure 5.5: Solution time for the optimizer with increasing phase shift over numerous yearly budgets. Time is plotted in log-scale, due to the difference in magnitude. The effect of the combinatorics on the solution time is apparent.

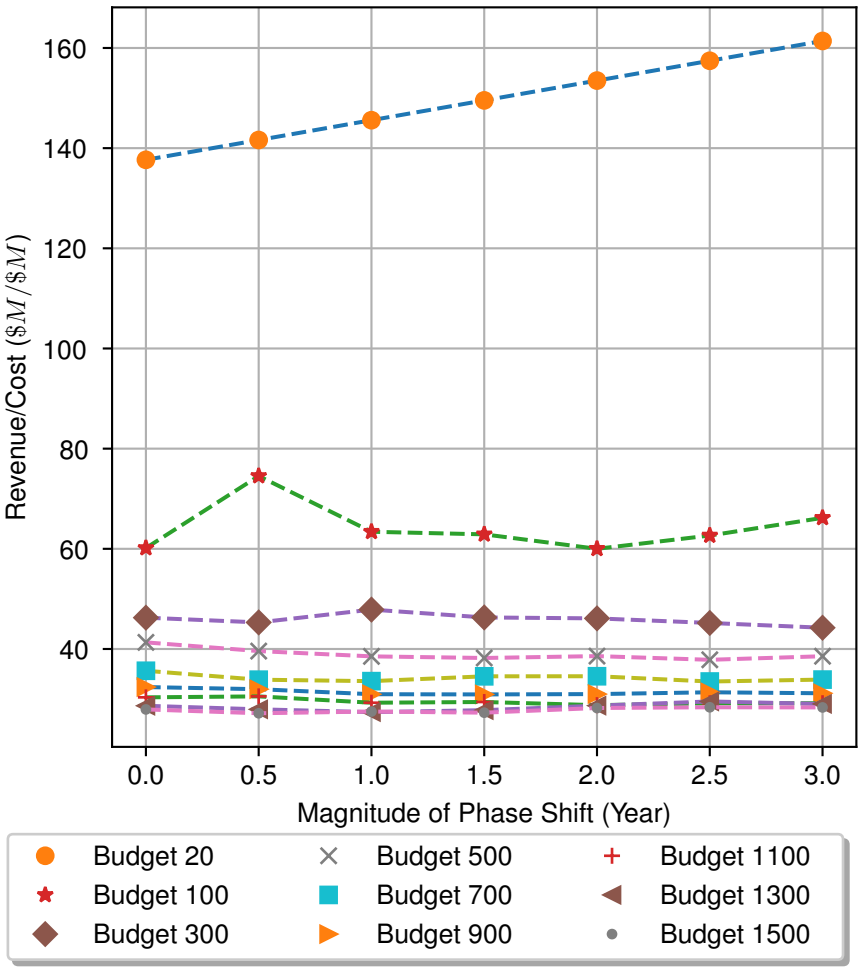

Figure 5.6: Ratio of the revenue and cost from the optimizer with increasing phase shift over numerous yearly budgets.

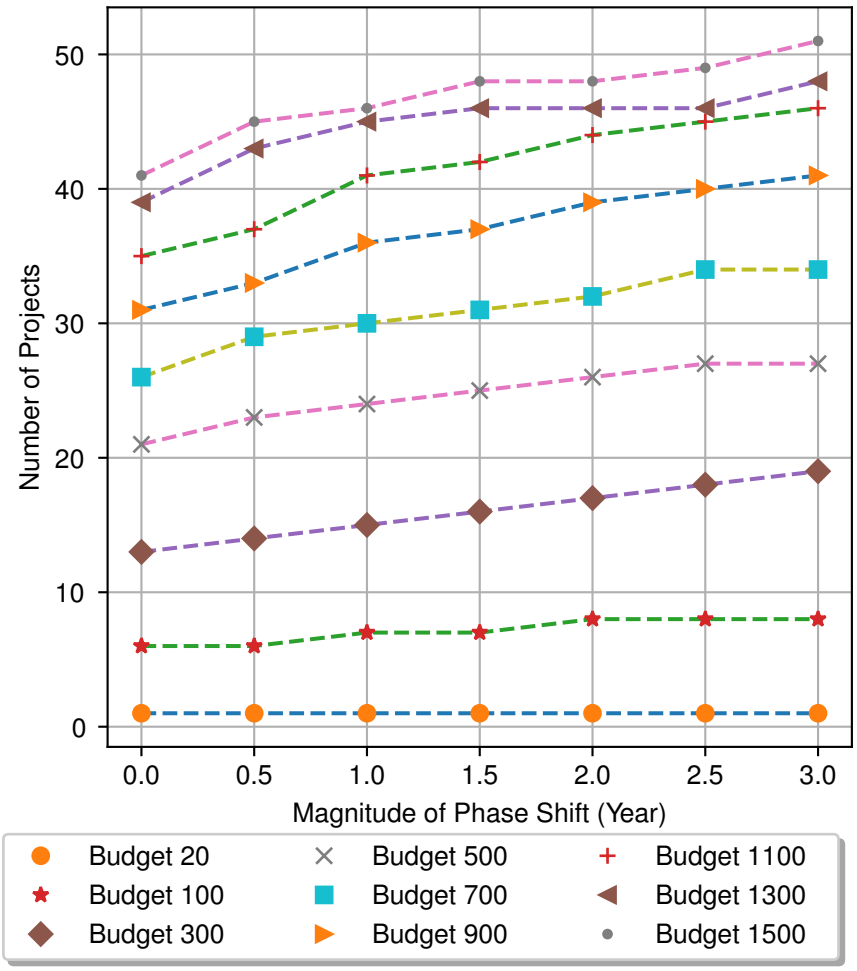

Figure 5.7: Number of projects chosen by the optimizer with increasing phase shift over numerous yearly budgets.

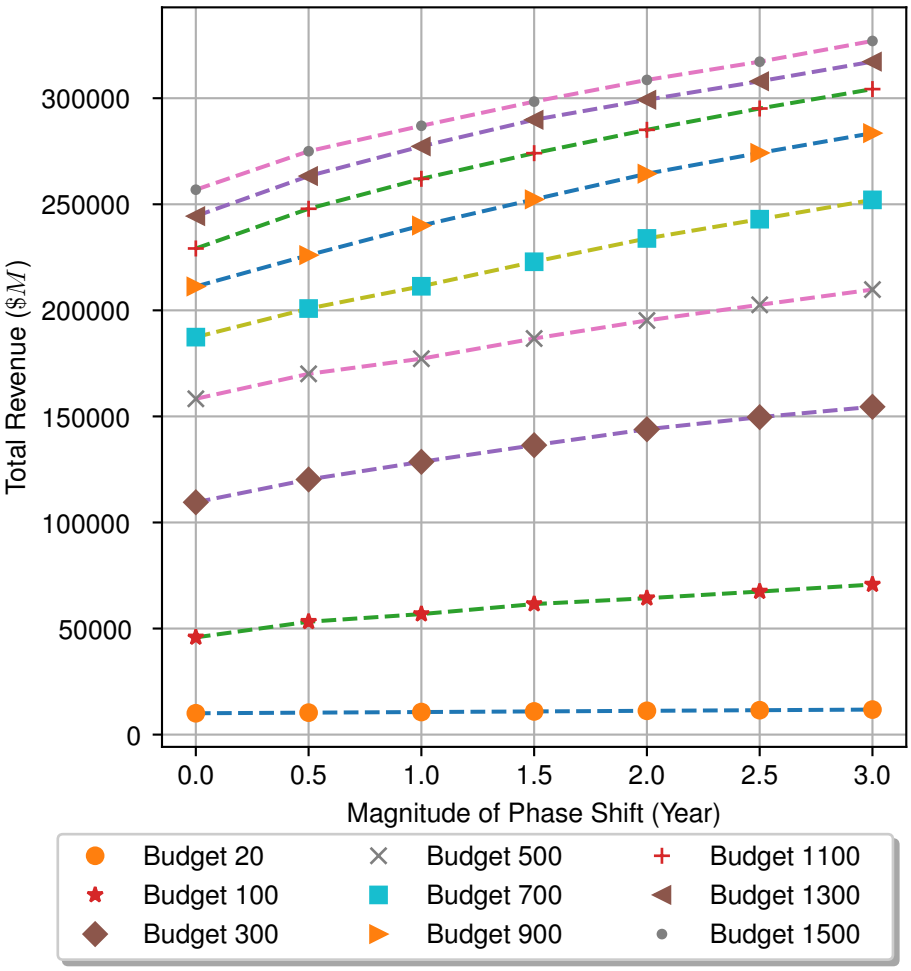

Figure 5.8: Total revenue from the optimizer with increasing phase shift over numerous yearly budgets.

The solver took a significant amount of time even on the simpler test cases, and as such any run of a full portfolio was infeasible, and would likely not give any results reliable enough to analyze.

6
Discussion

6.1 Model scope and capabilities

The deterministic model yields an optimal solution given enough time, influenced by factors such as the budget, number of potential projects and size of the phase shift. In this case, a standard ILP solver is sufficient. However, the aforementioned factors still present an issue regarding the time complexity of the problem, and given large spans on the variables (such as start date), the combinatorial possibilities of the problem can grow very large.

To mitigate the growth in complexity, the decision-maker may consider sacrificing optimality of the solution for more reasonable solution times. For example, one may group the projects into subsets and solve the deterministic problem for each of these. Another option is to reduce the total amount of time steps by increasing their length, such as tracking by two years instead of one.

However, the deterministic case is a very idealized version of reality and even if care is taken to minimize risk by considering utility instead of pure financial cost, its usefulness in the real world would likely be limited to being used as an initial heuristic to find and compare the most valuable projects. Each project is more or less self-contained, with the only interaction between two projects being budgetary. Furthermore, the gain and value of a project is quite stable, as there are no large fluctuations in the objective function, and it is always preferable to finish a project early.

So while the optimal solution might take some time to reach, there are many options in pruning the number of choices in a portfolio, down to the most valuable projects. One such approach might be to start with a very high budget and successively lower it, in order to get a grasp of which projects are the most valuable ones. Another could be to run the model with tight bounds on the starting date for all phases, and then evaluate if there are years that nearly allow for another project. Then some slight leeway in the starting date might allow for a more valuable solution.

The main limitation of the model is, as can be expected, the lack of any advanced stochastic estimations. It is possible, once all projects have been selected/discarded, to evaluate the risk and probabilities for each constraint, with the main limitation that these probabilistic constraints were not under consideration when the solution was found. Therefore the chosen projects might violate the probabilistic constraints. The full stochastic model, on the other hand, is complete in theory, but the complexity of a solution of any reasonable size renders it unsuited for practical use.

The base stochastic model is therefore only relevant when treating a very small

set of variables, such as when a previous heuristic has been employed, or when the set of projects under consideration is small enough. Approximations such as linearizations are also possible, but leaves one with the issue of finding the magnitude of the gap between the optimal and the approximated results.

Essentially, it is clear that none of the proposed models will be able to generate an optimal strategy without some human agent providing risk estimations and judgement. Even if this had been the case, use of the model still requires very carefully selected risk margins.

The most important takeaway for any user is that the model under no circumstances can determine and select which projects succeed and fail: it can only select projects that are the most likely to bring in revenue.

6.2 Recourse

In theory, the recourse statement of the problem could be solved by expanding the tree of all possible solutions and choosing the most valuable path in the tree. However, the scope of the problem is such that in a portfolio with 50 projects, the number of different scenarios at the "top" of the tree would be around $2^{50} > 10^{15}$. In addition, this is not considering any other stochastic variables such as length, cost or revenue, which may have continuous probability distributions instead of discrete, which increases the scope of the scenarios to a practically uncountable amount. With this in mind we believe it neither practically feasible nor useful to devote time to investigate these scenarios beforehand.

Instead, the recourse problem decision should not be to make the best possible decision for all possible outcomes, but instead to make the best possible choice with the given information, such as in the Sim-opt architecture. The main advantage of this is that it neatly reduces itself to the previous problem solved by the model. The model calculates the optimal set of projects to fund based on current knowledge. In the recourse problem, that knowledge is updated whenever a project fail or succeeds, or whenever any stochastic variable is realized. At such a time, the stochastic variable is exchanged for a given fixed parameter, and the new model is of the same form as the old one, but with strictly less free variables, thereby being a simpler problem to solve.

The main disadvantage of this method is the fact that it gives no information about the future: only the current best choice is known. However, by alternatively fixing and freeing some stochastic variables, their impact on the final result can be seen.

In general, we believe that this is the most effective use of the model, as a first step to weigh the impact and value of events and projects. Therefore, when using models described in this text to evaluate a portfolio of projects, it is important for the end user to be mindful of the stochastic nature of the problem and to comfortably be able to weigh stochastic scenarios against each other.

6.3 Expansions and further work

The deterministic model provides an excellent foundation for expansion and refinement. The core features regarding the project and phase rules are in place, and many further rules can be implemented simply by adding new constraints and variables. For example, the model can be expanded to include a budget injection, allowing the budget to be slightly higher during a given number of years if it corresponds to a higher revenue. Projects can also be sorted into groups, if one does not wish for two drugs of the same type to be developed parallel with each other.

Expansions of this kind are not expected to significantly change the scope or complexity of the model, and instead should allow for more flexibility and fine-tuning of the choices made. Larger expansions would likely involve the use of approximations in the stochastic model, either with a procedure to approximate the stochastic distributions, or a way to split the problem into master and sub-problems, allowing for some sort of iterative solution to be implemented.

However, the iterative approach requires a solid foundation of problem analysis before being used in practice. A way to find acceptable upper and lower bounds on the problem value must be constructed, and the acceptable gap between them must be determined. This would likely entail determining which data points have the largest impact on the problem, and what data points could be ignored in favor of others. It is clear that much of the ground work would be qualitative, researching what tolerance levels are both computationally feasible and practically use-able. These are factors that very much depend on the methods employed, the computational resources and subjective aspects of the projects and the targets for the decision-maker.

Considering the scope of the problem and the numerous different approaches that are available, we do not recommend any specific path forward regarding this expansion. It will likely be best handled by a more focused study into this matter, possibly in a separate thesis.

7
Conclusion

While a model concerning the deterministic case was developed and was shown to be reasonably efficient given the size of the problem, the intricacies of the stochastic model proved too complex to solve without employing some outside approximations and simplifications. There is large room for development and refinement of the stochastic model, and we suggest further studies be made on this topic.

Regarding the deterministic model, we strongly advise to develop a risk policy that generates expected values that align with the interest of the end user of Captario SUM, along with a suitable ”search” process, that evaluates and runs through several different cases after which they can be compared to each other. In their current state, the models developed in this paper should only be used in a guiding capacity, and to explore the most likely scenarios along with the variables that impact the end result the most. When considering the further development of the stochastic model, the most fruitful approach would likely be the development of numerical methods to discretize and combine probability distributions, combined with some iterative method and division of sub-problems that would generate an approximate solution. However, this approach requires careful study and research to ensure that it both aligns with the goals of the end user and generates results that are provable to be within accepted boundaries. And as with all stochastic problems, a firm grasp of how to best utilize and understand risk, probabilities and expected values is central to the use of any model within this paper.

As with the base stochastic model, the recourse model can be considered too impractical to solve optimally. However, metaheuristic methods such as Sim-opt architecture may provide valuable information about the portfolio, and yield better solutions than a human planner which is valuable in and of itself. And while it is clear that all models presented have many options for expansion, the purpose of this report is not to expand on all of them. The models presented all correctly translate the fundamentals of pharmaceutical development while also allowing the user to make the choice between higher accuracy or lower solution time. It falls on the end user to make the appropriate choices regarding accuracy, solution time and risk level. In the end the models all hinge on the level of risk a user is willing to take, and while no model can predict the future, we hope that the ideas presented in this report will function as a first step towards making the most informed choice possible.

Bibliography

[1] S.P. Bradley, A.C. Hax, and T.L. Magnanti. *Applied Mathematical Programming*. Reading, MA: Addison-Wesley Publishing Company, 1977. ISBN: 9780201004649.

[2] A Schrijver. *Theory of Linear and Integer Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1986. ISBN: 0-471-90854-1.

[3] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. New York, NY, USA: Springer-Verlag, 1997.

[4] P Brucker, A Drexl, R Mohring, K Neumann, and E Pesch. "Resource-constrained project scheduling: Notation, classification, models, and methods". *European Journal of Operational Research* 112.1 (1999), pp. 3–41.

[5] D Subramanian, J F. Pekny, and G Reklaitis. "A simulation-optimization framework for addressing research and development pipeline management". *AIChE Journal* 47 (Oct. 2001), pp. 2226–2242.

[6] D Subramanian, J F. Pekny, G V. Reklaitis, and G E. Blau. "Simulation-optimization framework for stochastic optimization of R&D pipeline management". *AIChE Journal* 49.1 (Apr. 2004), pp. 96–112.

[7] B. Alidaee, H. Wang, and F. Landram. "A note on integer programming formulations of the real-time optimal scheduling and flight path selection of UAVs". *IEEE Transactions on Control Systems Technology* 17.4 (July 2009), pp. 839–843.

[8] H. Song, S. Yang, B. Wu, X. Liu, and L. Guo. "An optimal algorithm for scheduling tasks within deadline and budget constraints". *2009 Fifth International Joint Conference on INC, IMS and IDC*. Aug. 2009, pp. 62–65.

[9] S Hartmann and D Briskorn. "A survey of variants and extensions of the resource-constrained project scheduling problem". *European Journal of Operational Research* 207.1 (Nov. 2010), pp. 1–14.

[10] Wim van Ackooij, Riadh Zorgati, René Henrion, and Andris Möller. "Chance Constrained Programming and Its Applications to Energy Management". *Stochastic Optimization*. Ed. by Ioannis Dritsas. Rijeka, Croatia: IntechOpen, 2011. Chap. 13.

[11] Wim van Ackooij, René Henrion, Andris Möller, and Riadh Zorgati. "Joint chance constrained programming for hydro reservoir management". *Optimization and Engineering* 15.2 (June 2014), pp. 509–531.

[12] A A. Ciociola, L B. Cohen, P Kulkarni, and the FDA-Related Matters Committee of the American College of Gastroenterology. "How drugs are developed and approved by the FDA: current process and future directions". *The American Journal Of Gastroenterology* 109 (May 2014). The Red Section, p. 620.

[13] A. Khalafa, K. Djouania, Y. Hamama, and Y. Alaylid. "Mixed-integer linear programming (MILP) for optimisation of medical equipment maintenance schedules". *2015 22nd Iranian Conference on Biomedical Engineering (ICBME)*. Nov. 2015, pp. 205–209.

[14] *Captario AB*. 2016. URL: https://captario.com/ (accessed 09/22/2018).

[15] *Clinical development success rates 2006-2015*. 2016. URL: https://www.bio.org/sites/default/files/Clinical%5C%20Development%5C%20Success%5C%20Rates%5C%202006-2015%5C%20-%5C%20BIO,%5C%20Biomedtracker,%5C%20Amplion%5C%202016.pdf (accessed 09/22/2018).

[16] S Kreter, J Rieck, and J Zimmermann. "Models and solution procedures for the resource-constrained project scheduling problem with general temporal constraints and calendars". *European Journal of Operational Research* 251.2 (June 2016), pp. 387–403.

[17] David Taylor. "The Pharmaceutical Industry and the Future of Drug Development". *Pharmaceuticals in the Environment* 2016 (Jan. 2016), pp. 1–33.

[18] R K. Chakrabortty, R A. Sarker, and D L. Essam. "Resource constrained project scheduling with uncertain activity durations". *Computers & Industrial Engineering* 112 (Oct. 2017), pp. 537–550.

[19] *Couenne, an exact solver for nonconvex MINLPs*. 2018. URL: https://projects.coin-or.org/Couenne (accessed 10/13/2018).

[20] *CPLEX for AMPL*. 2018. URL: https://ampl.com/products/solvers/solvers-we-sell/cplex/ (accessed 08/13/2018).

[21] *Frequently asked questions on patents and exclusivity*. 2018. URL: https://www.fda.gov/drugs/developmentapprovalprocess/ucm079031.htm#howlongpatentterm (accessed 10/13/2018).

[22] *The drug development process*. 2018. URL: https://www.fda.gov/ForPatients/Approvals/Drugs/default.htm (accessed 05/28/2018).

[23] *Upprätthåll patentskyddet*. 2018. URL: https://www.prv.se/sv/patent/forvalta-dina-patentintressen/uppratthall-patentskyddet/ (accessed 10/13/2018).

A

Project data tables

This appendix contains the full Project portfolio obtained from Captario SUM, and has been used to generate the majority of the results in this paper.

Table A.1: Minimum date .

	Minimum date						
Phases	ph1	ph1a	ph2a	ph2b	ph3	reg	market
Proj1	1.00	1.00	1.00	1.00	1.42	4.5	5.25
Proj2	1.00	1.00	1.00	1.00	3.17	6.33	8.0
Proj3	1.00	1.00	1.00	2.42	4.83	7.0	8.0
Proj4	1.00	1.00	1.00	2.5	4.92	7.83	8.67
Proj5	1.00	1.00	1.00	2.08	4.42	8.0	9.17
Proj6	1.00	1.00	1.00	1.92	4.83	6.83	8.08
Proj7	1.00	3.25	5.83	5.83	7.5	11.17	12.33
Proj8	1.00	2.92	5.0	5.0	7.25	10.92	12.08
Proj9	1.00	3.25	5.25	5.25	7.25	10.25	11.33
Proj10	1.00	1.00	1.00	3.67	5.92	9.17	10.5
Proj11	1.00	1.00	1.00	2.0	3.67	7.0	8.5
Proj12	1.00	3.17	5.75	5.75	7.75	10.08	11.25
Proj13	3.08	4.5	4.5	5.5	7.75	10.92	12.42
Proj14	3.17	4.92	4.92	5.33	7.0	9.75	10.75
Proj15	3.25	4.42	4.42	5.58	7.42	10.67	12.0
Proj16	3.25	4.25	4.25	5.33	7.0	9.83	10.67
Proj17	2.92	5.17	5.17	6.08	8.67	11.17	12.25
Proj18	3.25	4.5	4.5	5.08	7.25	10.58	11.75
Proj19	3.25	4.42	4.42	5.33	6.92	9.5	10.67
Proj20	2.92	4.67	4.67	5.75	8.83	12.83	14.25
Proj21	3.25	4.67	4.67	5.83	8.0	11.58	12.83
Proj22	1.00	2.33	4.42	4.42	6.83	9.83	10.75
Proj23	1.00	1.08	4.58	4.58	5.67	8.67	9.83
Proj24	1.00	2.33	4.75	4.75	6.0	8.42	9.42
Proj25	1.00	1.00	4.5	4.42	5.33	7.67	9.0
Proj26	1.00	1.67	4.25	4.25	5.92	8.5	9.58
Proj27	1.00	2.83	4.83	4.83	4.92	7.75	8.83
Proj28	1.00	2.08	4.33	4.33	5.42	8.17	9.75
Proj29	1.00	2.33	5.0	5.0	7.0	9.58	10.92

	Continuation of Table A.1						
Phases	ph1	ph1a	ph2a	ph2b	ph3	reg	market
Proj30	1.00	3.08	5.33	5.33	6.58	10.25	11.58
Proj31	1.00	2.0	4.83	4.83	5.83	8.0	9.58
Proj32	1.00	1.08	4.67	4.67	5.75	7.25	8.75
Proj33	1.00	1.00	1.00	3.33	4.58	6.08	7.33
Proj34	1.00	1.00	1.00	2.83	4.42	5.83	7.33
Proj35	1.00	1.00	1.00	3.75	4.75	6.33	7.33
Proj36	1.00	1.00	1.00	2.83	4.33	6.92	8.17
Proj37	1.00	1.00	1.00	2.42	5.75	8.58	10.0
Proj38	1.00	1.00	1.00	3.08	5.17	7.33	8.25
Proj39	1.00	1.00	1.00	1.58	5.33	7.33	8.83
Proj40	1.00	1.00	1.00	2.25	4.75	6.25	7.67
Proj41	1.00	1.00	1.00	2.08	4.08	6.42	7.33
Proj42	1.00	1.00	1.00	2.33	4.33	5.33	6.83
Proj43	1.00	1.00	1.00	1.08	5.08	8.33	8.67
Proj44	1.00	1.00	1.00	1.00	2.08	4.75	6.33
Proj45	1.00	1.00	1.00	1.00	2.83	4.92	5.5
Proj46	1.00	1.00	1.00	1.00	2.42	4.67	6.0
Proj47	1.00	1.00	1.00	1.00	1.08	5.0	6.0
Proj48	1.00	1.00	1.00	1.00	1.58	4.75	6.25
Proj49	1.00	1.00	1.00	1.00	3.25	6.67	8.0
Proj50	1.00	1.00	1.00	1.00	2.08	5.5	6.92
Proj51	1.00	1.00	1.00	1.00	2.33	4.83	5.25
	End of Table A.1						

Table A.2: Phase lengths .

	Phase length						
Phases	ph1	ph1a	ph2a	ph2b	ph3	reg	market
Proj1	0.0	0.0	0.0	0.0	3.08	0.75	0.0
Proj2	0.0	0.0	0.0	0.0	3.17	1.67	0.0
Proj3	0.0	0.0	0.0	2.42	2.17	1.00	0.0
Proj4	0.0	0.0	0.0	2.42	2.92	0.83	0.0
Proj5	0.0	0.0	0.0	2.33	3.58	1.17	0.0
Proj6	0.0	0.0	0.0	2.92	2.0	1.25	0.0
Proj7	0.0	2.58	0.0	1.67	3.67	1.17	0.0
Proj8	0.0	2.08	0.0	2.25	3.67	1.17	0.0
Proj9	0.0	2.0	0.0	2.0	3.0	1.08	0.0
Proj10	0.0	0.0	0.0	2.25	3.25	1.33	0.0
Proj11	0.0	0.0	0.0	1.67	3.33	1.5	0.0
Proj12	0.0	2.58	0.0	2.0	2.33	1.17	0.0
Proj13	1.42	0.0	1.00	2.25	3.17	1.5	0.0
Proj14	1.75	0.0	0.42	1.67	2.75	1.00	0.0
Proj15	1.17	0.0	1.17	1.83	3.25	1.33	0.0

	Continuation of Table A.2						
Phases	ph1	ph1a	ph2a	ph2b	ph3	reg	market
Proj16	1.00	0.0	1.08	1.67	2.83	0.83	0.0
Proj17	2.25	0.0	0.92	2.58	2.5	1.08	0.0
Proj18	1.25	0.0	0.58	2.17	3.33	1.17	0.0
Proj19	1.17	0.0	0.92	1.58	2.58	1.17	0.0
Proj20	1.75	0.0	1.08	3.08	4.0	1.42	0.0
Proj21	1.42	0.0	1.17	2.17	3.58	1.25	0.0
Proj22	0.0	2.08	0.0	2.42	3.0	0.92	0.0
Proj23	0.0	3.5	0.0	1.08	3.0	1.17	0.0
Proj24	0.0	2.42	0.0	1.25	2.42	1.00	0.0
Proj25	0.0	3.5	0.0	0.92	2.33	1.33	0.0
Proj26	0.0	2.58	0.0	1.67	2.58	1.08	0.0
Proj27	0.0	2.0	0.0	0.08	2.83	1.08	0.0
Proj28	0.0	2.25	0.0	1.08	2.75	1.58	0.0
Proj29	0.0	2.67	0.0	2.0	2.58	1.33	0.0
Proj30	0.0	2.25	0.0	1.25	3.67	1.33	0.0
Proj31	0.0	2.83	0.0	1.00	2.17	1.58	0.0
Proj32	0.0	3.58	0.0	1.08	1.5	1.5	0.0
Proj33	0.0	0.0	0.0	1.25	1.5	1.25	0.0
Proj34	0.0	0.0	0.0	1.58	1.42	1.5	0.0
Proj35	0.0	0.0	0.0	1.00	1.58	1.00	0.0
Proj36	0.0	0.0	0.0	1.5	2.58	1.25	0.0
Proj37	0.0	0.0	0.0	3.33	2.83	1.42	0.0
Proj38	0.0	0.0	0.0	2.08	2.17	0.92	0.0
Proj39	0.0	0.0	0.0	3.75	2.0	1.5	0.0
Proj40	0.0	0.0	0.0	2.5	1.5	1.42	0.0
Proj41	0.0	0.0	0.0	2.0	2.33	0.92	0.0
Proj42	0.0	0.0	0.0	2.0	1.00	1.5	0.0
Proj43	0.0	0.0	0.0	4.0	3.25	0.33	0.0
Proj44	0.0	0.0	0.0	0.0	2.67	1.58	0.0
Proj45	0.0	0.0	0.0	0.0	2.08	0.58	0.0
Proj46	0.0	0.0	0.0	0.0	2.25	1.33	0.0
Proj47	0.0	0.0	0.0	0.0	3.92	1.00	0.0
Proj48	0.0	0.0	0.0	0.0	3.17	1.5	0.0
Proj49	0.0	0.0	0.0	0.0	3.42	1.33	0.0
Proj50	0.0	0.0	0.0	0.0	3.42	1.42	0.0
Proj51	0.0	0.0	0.0	0.0	2.5	0.42	0.0
	End of Table A.2						

Table A.3: Project revenue .

	Project revenue		
Phases	ramp	PYS	residue
Proj1	0.33	70.83	0.0

	Continuation of Table A.3		
Phases	ramp	PYS	residue
Proj2	0.17	64.58	0.0
Proj3	0.25	91.67	0.0
Proj4	0.35	38.87	0.0
Proj5	0.3	25.77	0.01
Proj6	0.32	13.98	0.01
Proj7	0.36	24.57	0.01
Proj8	0.44	74.76	0.01
Proj9	0.29	90.55	0.01
Proj10	0.3	81.22	0.0
Proj11	0.34	32.55	0.01
Proj12	0.29	67.76	0.0
Proj13	0.34	28.78	0.01
Proj14	0.3	34.55	0.01
Proj15	0.37	33.93	0.01
Proj16	0.33	46.51	0.01
Proj17	0.29	29.3	0.01
Proj18	0.42	96.2	0.0
Proj19	0.26	19.1	0.0
Proj20	0.28	16.72	0.01
Proj21	0.31	36.0	0.01
Proj22	0.25	69.0	0.0
Proj23	0.25	87.17	0.0
Proj24	0.42	86.67	0.0
Proj25	0.33	75.0	0.0
Proj26	0.25	87.58	0.0
Proj27	0.42	92.25	0.0
Proj28	0.42	86.33	0.0
Proj29	0.42	94.75	0.0
Proj30	0.33	95.67	0.0
Proj31	0.25	96.83	0.0
Proj32	0.33	79.92	0.0
Proj33	0.42	69.42	0.0
Proj34	0.25	93.67	0.0
Proj35	0.33	69.0	0.0
Proj36	0.33	72.08	0.0
Proj37	0.42	97.42	0.0
Proj38	0.42	72.08	0.0
Proj39	0.25	90.83	0.0
Proj40	0.33	93.08	0.0
Proj41	0.25	78.58	0.0
Proj42	0.33	67.75	0.0
Proj43	0.42	85.67	0.0
Proj44	0.33	86.42	0.0

	Continuation of Table A.3		
Phases	ramp	PYS	residue
Proj45	0.42	74.17	0.0
Proj46	0.42	79.42	0.0
Proj47	0.33	80.25	0.0
Proj48	0.33	81.0	0.0
Proj49	0.42	91.5	0.0
Proj50	0.25	96.33	0.0
Proj51	0.42	73.75	0.0
	End of Table A.3		

Table A.4: Phase costs .

	Phase cost						
Phases	ph1	ph1a	ph2a	ph2b	ph3	reg	market
Proj1	0.0	0.0	0.0	0.0	56.76	13.33	0.0
Proj2	0.0	0.0	0.0	0.0	52.42	4.2	0.0
Proj3	0.0	0.0	0.0	14.48	87.69	8.0	0.0
Proj4	0.0	0.0	0.0	23.62	51.06	4.12	0.0
Proj5	0.0	0.0	0.0	22.5	28.5	5.54	0.0
Proj6	0.0	0.0	0.0	16.46	60.94	5.15	0.0
Proj7	0.0	12.77	0.0	40.17	30.01	4.35	0.0
Proj8	0.0	16.32	0.0	22.04	31.51	4.44	0.0
Proj9	0.0	18.5	0.0	30.94	45.73	5.38	0.0
Proj10	0.0	0.0	0.0	22.09	27.72	4.67	0.0
Proj11	0.0	0.0	0.0	25.44	38.44	2.53	0.0
Proj12	0.0	6.97	0.0	27.08	42.37	3.29	0.0
Proj13	7.06	0.0	11.23	18.95	30.68	3.3	0.0
Proj14	6.29	0.0	32.15	31.84	79.45	5.01	0.0
Proj15	12.0	0.0	11.39	31.84	25.56	4.82	0.0
Proj16	9.0	0.0	11.99	31.26	48.35	6.6	0.0
Proj17	3.11	0.0	14.34	15.53	47.47	3.12	0.0
Proj18	7.2	0.0	19.48	22.76	36.38	2.95	0.0
Proj19	9.43	0.0	11.16	26.57	58.18	4.05	0.0
Proj20	6.86	0.0	12.45	15.0	38.57	3.19	0.0
Proj21	8.47	0.0	9.13	24.78	61.37	5.6	0.0
Proj22	0.0	9.12	0.0	24.51	91.51	10.91	0.0
Proj23	0.0	4.0	0.0	41.8	93.58	8.57	0.0
Proj24	0.0	4.97	0.0	45.69	46.17	10.0	0.0
Proj25	0.0	3.71	0.0	79.29	85.38	7.5	0.0
Proj26	0.0	5.42	0.0	40.42	34.61	9.23	0.0
Proj27	0.0	4.0	0.0	699.55	90.28	9.23	0.0
Proj28	0.0	4.0	0.0	67.23	105.66	6.32	0.0
Proj29	0.0	5.25	0.0	22.92	108.14	7.5	0.0
Proj30	0.0	4.89	0.0	44.89	59.9	7.5	0.0

	Continuation of Table A.4						
Phases	ph1	ph1a	ph2a	ph2b	ph3	reg	market
Proj31	0.0	7.41	0.0	56.04	74.21	6.32	0.0
Proj32	0.0	3.63	0.0	68.36	57.22	6.67	0.0
Proj33	0.0	0.0	0.0	44.52	162.02	8.0	0.0
Proj34	0.0	0.0	0.0	22.47	108.87	6.67	0.0
Proj35	0.0	0.0	0.0	70.95	54.73	10.0	0.0
Proj36	0.0	0.0	0.0	37.22	71.74	8.0	0.0
Proj37	0.0	0.0	0.0	18.54	41.47	7.06	0.0
Proj38	0.0	0.0	0.0	31.37	136.11	10.91	0.0
Proj39	0.0	0.0	0.0	11.1	31.66	6.67	0.0
Proj40	0.0	0.0	0.0	12.34	124.96	7.06	0.0
Proj41	0.0	0.0	0.0	21.83	112.69	10.91	0.0
Proj42	0.0	0.0	0.0	26.75	291.47	6.67	0.0
Proj43	0.0	0.0	0.0	16.88	30.23	30.0	0.0
Proj44	0.0	0.0	0.0	0.0	31.9	6.32	0.0
Proj45	0.0	0.0	0.0	0.0	94.28	17.14	0.0
Proj46	0.0	0.0	0.0	0.0	108.32	7.5	0.0
Proj47	0.0	0.0	0.0	0.0	56.37	10.0	0.0
Proj48	0.0	0.0	0.0	0.0	79.22	6.67	0.0
Proj49	0.0	0.0	0.0	0.0	23.86	7.5	0.0
Proj50	0.0	0.0	0.0	0.0	18.51	7.06	0.0
Proj51	0.0	0.0	0.0	0.0	85.64	24.0	0.0
	End of Table A.4						

Table A.5: Patent expiry date (SPE) .

	Patent expiry date (SPE)
Project	Year
Proj1	14.25
Proj2	17.0
Proj3	19.0
Proj4	19.25
Proj5	21.67
Proj6	15.33
Proj7	21.25
Proj8	20.17
Proj9	20.42
Proj10	23.42
Proj11	18.17
Proj12	19.58
Proj13	23.92
Proj14	20.67
Proj15	24.58
Proj16	19.67

	Continuation of Table A.5
Project	Year
Proj17	21.83
Proj18	20.25
Proj19	18.0
Proj20	21.75
Proj21	20.25
Proj22	19.0
Proj23	14.42
Proj24	10.75
Proj25	18.42
Proj26	19.5
Proj27	12.42
Proj28	19.5
Proj29	21.5
Proj30	15.92
Proj31	11.5
Proj32	13.67
Proj33	13.92
Proj34	14.83
Proj35	9.25
Proj36	13.67
Proj37	16.5
Proj38	19.67
Proj39	16.75
Proj40	14.92
Proj41	17.67
Proj42	16.5
Proj43	16.75
Proj44	12.08
Proj45	8.67
Proj46	11.92
Proj47	16.0
Proj48	16.58
Proj49	18.92
Proj50	16.67
Proj51	13.33

B

AMPL model files

This appendix contains the AMPL model and script files used in order to run the optimizer and generate all results in this paper.

```
set PROJ;
set TIME ordered = 1..30;
set PHASE ordered;
set RATES;
param phaseTest:=0;
param minDate{PROJ,PHASE};
param phaseProb{PROJ, PHASE}>=0;
param revenue {PROJ,TIME}>=0;
param cost {PROJ, TIME}>=0;
#param budget_min {j in TIME} >=0;
#param budget_max {j in TIME} >=budget_min[j];
param budget_min>=0;
param budget_max>=budget_min;
param SPE {PROJ} >= 0;
var z {j in TIME, i in PROJ, k in PHASE};
var Project {i in PROJ,k in PHASE} binary;
var phaseStart {i in PROJ, k in PHASE} >=first(TIME);
var helper{j in TIME,i in PROJ, k in PHASE} binary;
var helper2{j in TIME,i in PROJ, k in PHASE};
param phaseLength{i in PROJ, k in PHASE};
param phaseRevenue{i in PROJ, j in RATES};
param phaseCost{i in PROJ, k in PHASE}>=0;
maximize total_revenue:
sum {i in PROJ}
(Project[i,last(PHASE)]*(phaseRevenue[i,'ramp']*2 +
phaseRevenue[i,'PYS']*(SPE[i]-phaseStart[i,last(PHASE)]) +
phaseRevenue[i,'residue']*(last(TIME)-SPE[i]))) -
(sum {i in PROJ,k in PHASE}
Project[i,k]*phaseCost[i,k]*phaseLength[i,k]);

subject to numberOfProjects:
sum {i in PROJ} Project[i, last(PHASE)] <= 50000000;
subject to totBudget:
sum {j in TIME,i in PROJ,k in PHASE} helper2[j,i,k]*phaseCost[i,k] <=30000000;
```

```
subject to Budget {j in TIME}:
sum {i in PROJ,k in PHASE} helper2[j,i,k]*phaseCost[i,k] <=budget_max;

subject to helperConstraint1 {i in PROJ,k in PHASE,j in TIME}:
helper[j,i,k]*j>=z[j,i,k]-1;

subject to helperConstraint2 {i in PROJ,k in PHASE,j in TIME}:
helper[j,i,k]*j<=phaseStart[i,k]+phaseLength[i,k];

subject to helperConstraint3 {i in PROJ,k in PHASE}:
(phaseLength[i,k])<=sum {j in TIME} helper[j,i,k] <=2+ (phaseLength[i,k]);

subject to helperConstraint4 {i in PROJ,k in PHASE}:
sum {j in TIME} helper2[j,i,k] = Project[i,k]*phaseLength[i,k];

subject to helper2Limit {j in TIME, i in PROJ, k in PHASE}:
helper2[j,i,k]<=helper[j,i,k]*j+1 -z[j,i,k];

subject to helper2Limit1{j in TIME, i in PROJ, k in PHASE}:
helper2[j,i,k] <=-helper[j,i,k]*j+z[j,i,k]+phaseLength[i,k];

subject to helper2Limit2{j in TIME, i in PROJ, k in PHASE}:
0 <= helper2[j,i,k] <=1;

subject to helper2Limit3{j in TIME, i in PROJ, k in PHASE}:
helper2[j,i,k] <=helper[j,i,k];

subject to collisionCheck {i in PROJ,k in PHASE: ord(k) > 1}:
phaseStart[i,prev(k)]+phaseLength[i,prev(k)]<=phaseStart[i,k];

subject to startDate {i in PROJ, k in PHASE}:
minDate[i,k] <= phaseStart[i,k] <= minDate[i,k]+phaseTest;
subject to endDate {i in PROJ}:
phaseStart[i,last(PHASE)]+phaseLength[i,last(PHASE)]<=SPE[i];

subject to phaseCheck {i in PROJ, k in PHASE: ord(k) >1}:
Project[i,prev(k)]>=Project[i,k];

subject to  linear1 {j in TIME,i in PROJ, k in PHASE}:
z[j,i,k]<= (SPE[i]-phaseLength[i,k])*helper[j,i,k];
subject to  linear2 {j in TIME,i in PROJ, k in PHASE}:
z[j,i,k]>= first(TIME)*helper[j,i,k];
subject to  linear3 {j in TIME,i in PROJ, k in PHASE}:
z[j,i,k]<= phaseStart[i,k]-first(TIME)*(1-helper[j,i,k]);
subject to  linear4 {j in TIME,i in PROJ, k in PHASE}:
z[j,i,k]>=phaseStart[i,k]-(SPE[i]-phaseLength[i,k])*(1-helper[j,i,k]);
```

```
reset;
model phaseShiftFrac.mod
data phaseShiftFrac.dat

option solver cplex;
option cplex_options 'absmipgap=1e-3 mipgap=1e-3';
option presolve_eps 2.13e-10;

reset data;
data phaseShiftFrac.dat

for {0..19} {
solve;
print budget_max > budgetRuns/budgetColl.txt;
print numberOfProjects.body > budgetRuns/nProjectsColl.txt;
print totBudget.body > budgetRuns/totBudgetColl.txt;
print total_revenue > budgetRuns/totRevColl.txt;
print _solve_system_time > budgetRuns/sysTimeColl.txt;
print _solve_user_time > budgetRuns/userTimeColl.txt;
print _solve_time > budgetRuns/solveTimeColl.txt;


display "............................" > budgetRuns/collectedBudgetRuns.txt;
display budget_max > budgetRuns/collectedBudgetRuns.txt;
display numberOfProjects.body > budgetRuns/collectedBudgetRuns.txt;
display totBudget.body > budgetRuns/collectedBudgetRuns.txt;
display total_revenue > budgetRuns/collectedBudgetRuns.txt;

display _solve_system_time > ("budgetRuns/budget" & budget_max & ".txt");
display _solve_system_time > ("budgetRuns/budget" & budget_max & ".txt");
display _solve_user_time > ("budgetRuns/budget" & budget_max & ".txt");
display _solve_time > ("budgetRuns/budget" & budget_max & ".txt");
display total_revenue >("budgetRuns/budget" & budget_max & ".txt");
display Project > ("budgetRuns/budget" & budget_max & ".txt");
option display_1col 0;
display phaseStart > ("budgetRuns/budget" & budget_max & ".txt");
display Budget.ub > ("budgetRuns/budget" & budget_max & ".txt");
display Budget.body > ("budgetRuns/budget" & budget_max & ".txt");
display helper2 > ("budgetRuns/budget" & budget_max & ".txt");
let budget_max := budget_max + 120;
}
```