Pharmaceutical R & D Pipeline Management under Trial Duration Uncertainty

Elvan Gökalp*[a], Juergen Branke[b]

[a] *WMG, The University of Warwick, Coventry, CV4 7AL, UK*
[b] *Warwick Business School, The University of Warwick, Coventry, CV4 7AL, UK.*

Abstract

We consider a pharmaceutical Research & Development (R & D) pipeline management problem under two significant uncertainties: the outcomes of clinical trials and their durations. We present an Approximate Dynamic Programming (ADP) approach to solve the problem efficiently. Given an initial list of potential drug candidates, ADP derives a policy that suggests the trials to be performed at each decision point and state. For the classical R&D pipeline planning problem with deterministic trial durations, we compare our ADP approach with other methods from the literature, and find that it can find better solutions more quickly in particular for larger problem instances. For the case with stochastic trial durations, we compare the ADP algorithm with a myopic approach and show that the expected net profit obtained by the derived ADP policy is higher (almost 20% for a 10-drug portfolio).

Keywords: Dynamic Programming, Pharmaceutical R&D Pipeline Management, Heuristics, Approximate Dynamic Programming, Project Scheduling

1. Introduction

Most pharmaceutical companies develop several molecules and drugs simultaneously. Drug development (from discovery to market launch) can take up to 15 years, and the average cost of a new drug is about \$1.3–2 billion of which 50% are spent on clinical trials (Lainez et al., 2012). The candidate drugs go through four main stages of development: pre-clinical, Phase 1, 2, and 3. The last three stages are highly regulated and systematized processes to determine the appropriate dosage and whether compounds are effective or not for humans. The drugs that successfully completed all stages go to the approval process and then to the market if approved. However, many drug candidates fail during the clinical trials (see Figure 1). On the other hand, the profits earned from a drug on the market can be quite substantial, e.g. around \$ 16 billion per year (Lines, 2012).

*Corresponding author
Email addresses: elvan.gokalp@warwick.ac.uk (Elvan Gökalp*), juergen.branke@wbs.ac.uk (Juergen Branke)

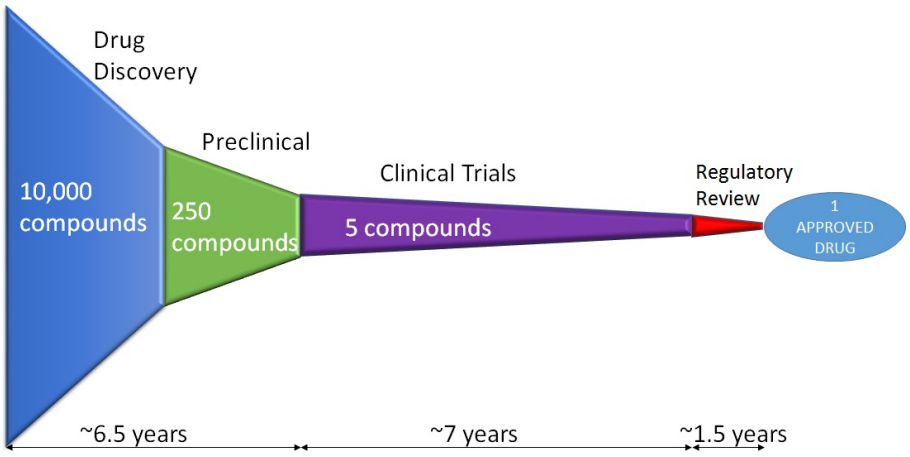

Figure 1: Drug discovery and development timeline (S Raghavendra et al., 2012)

The selection of different drug candidates and resource allocation to the related development activities are known as drug (R & D) pipeline management. Conducting a clinical trial constitutes many activities such as site recruitment, site monitoring, site retention, patient recruitment, lab activities, and protocol approvals. These activities involve several critical resources that are usually limited and not easy to expand at short notice. Thus, although several drug candidates enter the clinical trial stages, not all of them may be conducted at the same time.

Drug pipeline management requires to make interdependent and dynamic decisions. These include advancing a drug candidate to the next stage when it successfully passed a clinical phase, or freezing it due to resource and financial limitations. There is usually a fixed budget to be used for the drug development activities including clinical trials. Since each trial takes up to 6 years, the decision of allocating resources to any project may result in a lack of resources for another project during this period. Thus, the decisions taken today affect possible actions in the future which makes the problem time-dependent.

Drug development activies are affected by several uncertainties. The most significant uncertainty is observed in the outcomes of clinical trials. Once the design of a trial has been approved by the regulatory agencies, a targeted patient sample is recruited. However, there may not be enough patients to recruit or several companies may be competing for the same pool of patients. Thus, the completion time of a trial may be delayed and cannot be known in advance. In case portfolio planning decisions are made at fixed time intervals, some drugs that already completed a phase may need to wait until the next decision point. On the other hand, time-to-market (the time spent in the pipeline) is a crucial factor for the profits to be earned from a drug due to a fixed patent life (Jekunen, 2014). Thus, delaying a candidate should be avoided as much as possible and when trial durations are stochastic, the timing of decisions within a portfolio planning process may be stochastic as well.

Drug pipeline management is crucial for pharmaceutical companies considering the associated impact on profits and costs. Moreover, it is a highly complex process with a dynamic nature involving several uncertainties. However, the industry generally manages the pipelines by ad-hoc policies rather than analytical methods (Skrepnek et al., 2007). In this paper, we propose a stochastic dynamic programming formulation for the drug pipeline management problem. Then, we develop an ADP approach based on value iteration that can account for the complexities of the underlying problem and allows to solve larger portfolio planning problems than other state-of-art methods which is important since the size of portfolios is expected to increase as the industry moves toward more personalized medicine. We also compare the ADP approach with a greedy approach applied in practice that ranks the drug candidates according to ENPV.

The rest of this paper is organized as follows. The next section presents the current literature related to portfolio planning in pharmaceutical companies. Section 3 presents the stochastic dynamic programming formulation. The ADP approach is explained in Section 4. Section 5 consists of the computational experiments and the analysis of the results. Finally, Section 6 summarizes the paper and outlines avenues for future research.

2. Literature Review

Due to the stochastic and dynamic nature of pipeline management problems, a solution should recommend an action at each possible state and time point. Such a mapping from state to action is called a *policy*. However, as the problem size grows, the number of possible states increases quickly, making it harder to obtain the optimal policy and requiring some approximation tools. The modelling and solution approaches in the literature can be divided into mathematical programming and simulation-based approaches. To solve the pharmaceutical portolfio planning problem, we propose ADP that combines simulation and dynamic programming to learn an approximate policy. ADP has been used for many complex operations research problems such as data resource planning (Li et al., 2014), project scheduling (Li and Womer, 2015), or patient scheduling (Saure et al., 2012; Lu et al., 2018). To the best of our knowledge, it has not been applied to the pharmaceutical portfolio planning problem before.

All papers in the literature consider the uncertainty in trial outcomes while trial duration uncertainty is considered only in few simulation-based papers (Perez-Escobedo et al, 2012; Blau et al., 2004). The most frequently used performance measure considered for portfolio planning is expected net present value (ENPV), which takes into account the uncertainty of the outcomes as well as their discounted value with time.

2.1. Mathematical Programming-based Approaches

The early papers focusing on drug pipeline management use two-stage stochastic programming. In this framework, the first stage decisions are drug selection and capacity investments. The uncertainty in trial outcomes is represented by a large number of scenarios. As the trials are completed, the additional information is used to make the second-stage decisions that consist of capacity re-allocation or stopping some trials in case of lack of resources. However, due to the required large number of scenarios, the resulting models are hard to solve optimally. Thus, several heuristic approaches have been developed to solve real-size instances.

Rotstein et al. (1999) developed a two-stage stochastic programming model for simultaneous drug selection and capacity planning for a single pharmaceutical production site. Since solving real-size instances to optimality is computationally burdensome, they used a heuristic cut-off procedure. In this method, the scenarios with higher probabilities are selected until the overall probability across these scenarios is above a threshold level, and then the problem is solved in this reduced scenario space. Their computational results indicate that when the threshold level is around 0.5, the results are satisfactory while the computational effort drops significantly. Gatica et al. (2003) proposed to define four possible outcomes (high, target, low and failure) for clinical trials. Considering the integer capacity expansion variables as well, the model becomes a large MILP that can be solved only for very small number of drugs. Thus, the cases solved in this paper only consist of four drugs. Papageorgiou et al. (2001) developed a deterministic capacity planning model with multi-sites and multi-periods. The model determines the promising drugs, when and where to produce them and the allocation and expansion of the capacity. There is no uncertainty in clinical trial outcome and demand. Instead, they consider the transfer pricing and the taxation framework, as well as manufacturing details, including setup, scale-up, qualification and manufacturing suite structure.

Patel et al. (2013) considered portfolio planning by incorporating the trial design decisions (sample sizes) and scheduling of trials simultaneously into the cost and duration parameters. They used integer programming and considered budget constraints. They calculated the probability of success of a trial by combining frequentist and Bayesian approaches assuming that a prior distribution is known for efficacy. The decision variables are binary, indicating whether a trial for a drug is started with a certain type of design at a particular time t. They also developed a stochastic integer model (Patel and Ankolekar, 2015) that provides solutions for possible Phase 3 outcomes.

Colvin and Maravelias (2008) developed a multi-stage stochastic programming model to find the trials to be performed for a given portfolio for each planning period. They reduce the number of scenarios and non-anticipativity constraints by employing reduction techniques and could solve up to 5 drug instances. Later,

Colvin and Maravelias (2010) extended their work by developing a novel branch and cut algorithm which further improves the computational performance: they could solve up to 7 drug instances. Based on the model presented in this paper, Solak et al. (2010) present a multi-stage stochastic model where the uncertainties are required investment levels, updated return estimates and final return levels. They propose a sample average approximation algorithm to find the solution.

To solve the same portfolio planning problem, Christian and Cremaschi (2015) proposed three heuristics: shrinking horizon, multiple two-stage stochastic programming decomposition algorithm and a knapsack decomposition algorithm. The first one decomposes the problem into smaller ones that are solved whenever resources become available. The knapsack approach decomposes the original problem into several knapsack problems, which are solved at decision points using a rolling horizon procedure. The results obtained for up to 7 drugs are promising in terms of the optimality gap and computation time. In a more recent paper, Christian and Cremaschi (2017) extended their knapsack decomposition algorithm by changing the time points where the knapsack problems are produced. The modified algorithm results in a smaller optimality gap (around 1% for the 6-drug case) but comparatively larger computation times. They also extended their branch-and-cut algorithm by a new branching method and combine it with the knapsack decompostion approach. Although the memory requirement of the modified algorithm is much smaller than the exact branch-and-bound algorithm, the computation times are very large (around 643 hours for the 5 drug case).

Table 1 summarizes the important attributes of the related (both mathematical and simulation-based) papers. The table shows that the trial outcome is the main uncertainty considered in almost all papers while trial durations are only studied in two papers (Blau et al., 2004; Perez-Escobedo et al., 2012). Heuristics are slightly more prevalent than exact methods.

Table 1: A review of the literature on drug portfolio planning

Papers	No. of stages			Solution Approach			Uncertainties	
	Single	Two	Multi	Exact	Heuristic	Simulation	Outcome	Duration
Rotstein et al. (1999)		✓			✓		✓	
Blau et al. (2000)			✓		✓	✓	✓	
Papageorgiou et al. (2001)	✓			✓				
Gatica et al. (2003)			✓	✓			✓	
Blau et al. (2004)			✓		✓	✓	✓	✓
Choi et al. (2004)			✓		✓	✓	✓	✓
Rajapakse et al. (2005), (2006)			✓		✓		✓	
Varma et al. (2008)	✓			✓				
Zapata et al. (2008)			✓		✓	✓	✓	
Colvin and Maravelias (2008), (2010)			✓	✓			✓	
Solak et al. (2010)			✓			✓	✓	
Perez-Escobedo et al. (2012)			✓		✓	✓	✓	✓
Patel and Ankolekar (2015)	✓			✓			✓	
Christian and Cremaschi (2015), (2017)			✓	✓			✓	

A related stream of the literature to R&D pipeline management studies the project scheduling problem. The project scheduling problem deals with finding the starting times of a number of tasks that constitute a project and brings a discounted cash flow after all activities are completed successfully. The tasks preserve some precedence relationship, i.e. need to be performed in order to achieve the profit. Creemers et al. (2010) and Hermans and Leus (2018) aim to find the optimum starting time for each task in a project to maximize total discounted cash flow. They assume that the tasks have exponentially distributed durations. As different from Creemers et al. (2010), Hermans and Leus (2018) allow a task to be interrupted (preemption). They model the problem as a continuous-time Markov Chain and suggests a near-optimal algorithm.

Wiesemann et al. (2010) study the project scheduling problem but suggest to find the optimum target processing time for the activities instead of starting times. In this setting, an activity should start as early as possible but never before its target processing time. With this change, they formulate the project scheduling problem as a global optimization problem and solve with branch-and-bound algorithm with instances up to 30 activities. De Reyck and Leus (2008) study the project scheduling problem with deterministic activity durations, unlimited resources and probability of activity failures. They show that the problem is NP-hard but small instances can be solved with branch-and-bound to optimality. Different from De Reyck and Leus (2008), Creemers et al. (2009) consider stochastic activity durations and activity failures. They assume that there are several modules that need to be completed for a project to be successful and bring profit. Each module contains several activities and at least one of these activities should be completed successfully for the module to be successful. Similarly, Creemers et al. (2015) develop a stochastic dynamic programming model for project activities with uncertain task durations and outcomes. Not all modules have precendence relationships. The problem studied in these papers presents a different structure than ours which has several projects and therefore several different profits/final modules. Finally, Choi et al. (2004) study the resource constrained project scheduling problem in the presence of task duration, cost and outcome uncertainties. They propose to overcome the computational complexity by limiting the state space heuristically. Specifically, they simulate different scenarios using three simple decision heuristics and only consider the states visited by these heuristics. They use Bellman iteration to find the actions for each of these states by using the estimated state values by the heuristics. However, the Bellman iteration steps as well as combining different state spaces of the heuristics are computationally burdensome. Besides, the heuristics may lead to non-optimum decisions because ignoring states not visited by any of the heuristics may remove the optimal policy from the search space. In a later study, Choi et al. (2007) apply Q-learning to expedite the computation of the state transitions while the rest of the algorithm in Choi et al. (2004) stays the same. They also incorporate the arrival of new projects. Although

Q-learning is useful as a model-free algorithm, it also requires to consider all state action pairs, instead of states only. Therefore, it is more efficient when the state transition probabilities are unknown or the set of future possible states is quite large (Powell, 2011).

2.2. Simulation-based Approaches

Mathematical programming-based approaches usually suffer from an exponential increase in the number of variables as the instance size is increased. Simulation-based methods can solve large size problems more efficiently, but need to be combined with detailed experiments to examine the quality of a solution. This section summarizes previous research on pharmaceutical portfolio planning by utilizing simulation-based approaches. We focus on two uncertainties affecting portfolio planning, namely trial outcomes and trial durations. First, we summarize the studies with deterministic trial durations.

Blau et al. (2000) focus on selection and sequencing of drug candidates. They consider uncertainties in clinical trial outcomes. First, they compared the candidates in terms of development capital cost, possible sales, and success probabilities. After a portfolio has been selected based on this comparison, they applied a simple heuristic utilizing Monte Carlo simulation to sequence the portfolio in the pipeline. They compared different sequences in terms of their ENPVs. However their method cannot explicitly enforce the resource constraints. Instead, they suggest to keep track of resource constraint violations.

Rajapakse et al. (2005) used Monte Carlo simulation to model the uncertainty in trial outcomes and develop a decision tool to examine different portfolio management strategies. They consider different planning aspects such as resource management, manufacturing activities and clinical trials. Their tool can be used to evaluate several performance outcomes of a given portfolio (no optimization). They show the practicability of the tool in a case study. Later, they extended this work, (Rajapakse et al., 2006), by generating an efficient frontier for ENPV and risk (standard deviation of the NPV distribution) of each possible candidate. Each portfolio in the efficient frontier is considered as a possible solution.

The trial durations may be affected by uncertainties or resources allocated to them. For resource-dependent trial durations, Zapata et al. (2008) developed a simulation-optimization based decision tool to aid the resource management and scheduling of activities within a drug development pipeline. Similarly, Varma et al. (2008) assume that resource allocations affect activity durations. They developed a decision tool that can be used to assess the impact of various scheduling and resource allocation policies on several strategic metris such as ENPV, risk and average time to market. Simulation optimization packages such as Sim-Opt has been utilized for the research and development pipeline management by Subramanian et al. (2003), Subramanian et al. (2001).

Uncertainty in trial durations may affect the performance of a portfolio significantly. Thus, several authors

have considered stochastic trial durations in their modelling. Blau et al. (2004) consider several dependencies among resources, manufacturing costs, financial returns and technical successes of drug candidates. They used simulation to obtain the distribution of NPV for a portfolio and a genetic algorithm to optimize the drug selection and sequencing. This combined method results in 28% increase in ENPV compared to a bubble chart approach that visualizes the drug candidates according to their expected revenues and shows the pareto frontier. However, it is also computationally burdensome; it takes about 60 hours for the algorithm to find a solution.

Extending the work of Blau et al. (2004), Perez-Escobedo et al. (2012) additionally consider multiple objectives: NPV and risk of money loss. They combined simulation and a genetic algorithm to model the uncertainties affecting the portfolio planning, such as activity durations and trial outcomes, by using intervals rather than point-wise estimates.

2.2.1. Literature Gaps and Contribution

Our review shows that there are several gaps in the literature for the drug pipeline management problem:

- Few authors study the case where the trial durations are stochastic. Most of the time, the decisions need to be taken as soon as information is obtained (such as the completion of a clinical trial). However, trial durations cannot be known exactly beforehand due to the inherent uncertainties in the patient recruitment process.
- The stochastic arrival of new candidates into the pipeline is not considered. The authors assume a fixed set of drug candidates throughout the planning horizon.
- The number of drugs considered is at most 10, while the pipeline of top 25 pharma companies range from 66 to 251 drugs with an average of 141 drugs (Informa UK, 2017).

To adress these gaps in the literature, we propose a stochastic dynamic programming model and an ADP algorithm. The proposed solution framework is flexible enough to incorporate the features not considered in the current literature. Meanwhile, our approach still produces near-optimal policies for the instances up to 10 drugs, within a reasonable solution time, especially as the portfolio size increases.

3. Stochastic Dynamic Programming Formulation

In this section, we present a stochastic dynamic programming framework for the drug pipeline management problem. We follow the model of Christian & Cremaschi (2015, 2017) that is in line with most of the models presented in the literature. Later, this model is extended with uncertainty in trial durations. We assume that

the revenue for a drug is obtained as a lump sum once all corresponding trials have been completed successfully. In practice, pharmaceutical companies collect drug revenues over a long time horizon. However, this is assumed to be certain and thus the revenues can be combined into a single net present value. Finally, we do not allow preemption, i.e. a trial is not stopped before it is completed. The clinical trials are conducted based on a protocol that has been approved by regulatory agencies beforehand.

Sets:

- The planning horizon is assumed to be finite and divided into discrete time periods (decision points), represented as $t = 1, \cdots, T$ at which the decisions are made. Note that if a trial is still continuing for a drug candidate, no decision is made for this candidate until the current trial is completed. The duration between two decision points is assumed to be fixed, e.g. 3 months.
- The number of drug candidates in the pipeline is P, while each candidate is represented with $p = 1, \cdots, P$.
- We only consider the drugs in clinical stages that are denoted by $s = 1, 2, 3$.

Parameters:

- Duration of stage s for drug p: d_{ps}.
- Cost of stage s of drug p incurred at the start of the stage: q_{ps}.
- Probability to complete stage s of drug p successfully: ϕ_{ps}.
- Maximum level of resource 1 available at any time: R^1_{max}.
- Maximum level of resource 2 available at any time: R^2_{max}.
- Resource 1 and 2 usage at stage s of drug p in every time period: r^1_{ps} and r^2_{ps}.
- Expected revenue to be obtained from drug p, if completed successfully: m_p.
- Late launch penalty rate for drug p due to a shorter patent life: ρ_p.
- The fraction of diminished revenue due to opportunity cost of lost market share for drug p per time period: γ_p.

State Variables:

- The current status of drug p at time t is denoted by:

$$f_{pt} = \begin{cases} 2, & \text{if drug } p \text{ has completed all stages successfully,} \\ 1, & \text{if the current stage of drug } p \text{ has finished with success,} \\ 0, & \text{if the current stage of drug } p \text{ is still continuing,} \\ -1, & \text{if the current stage of drug } p \text{ has finished with failure,} \\ -2, & \text{if drug } p \text{ is frozen.} \end{cases}$$

- The current stage of drug p at time t is denoted by ψ_{pt}. If the project has not started yet or has failed, then $\psi_{pt} = 0$.

- The number of periods that have passed since the current stage has started for drug p at time t is denoted by h_{pt}.

State of the system at time t is denoted by $\boldsymbol{S_t} = [\boldsymbol{f_t}, \boldsymbol{\psi_t}, \boldsymbol{h_t}]$, where $\boldsymbol{f_t} = \{f_{pt}, p = 1, \cdots, P\}$ and $\boldsymbol{\psi_t}$ and $\boldsymbol{h_t}$ are defined similarly.

Uncertainties: Note that the success probabilities are only effective when the drug completes a trial. We define ϕ'_{pt} as the (effective) success probability for drug p at time t. In other words, ϕ'_{pt} shows the success probability of drug p for its current stage. Note that we need to update the duration passed for a trial. If $h_{pt} + 1 < d_{p,\psi_{pt}}$,

$$h_{p,t+1} = h_{p,t} + 1, \forall p,\ t = 1, \cdots, T-1,$$

otherwise,

$$h_{p,t+1} = 0,\ \phi'_{pt} = \phi_{p,\psi_{pt}}, \forall p,\ t = 1, \cdots, T-1.$$

For the stochastic trial duration case, we define additional probabilities $\zeta_{pt}(h_{pt})$ showing the probability of trial completion in the next period given that h_{pt} periods have passed for trial t of drug p. This additional uncertainty increases the number of possible future states, $\boldsymbol{S}_{t+1}$, at each time period.

Decision Variables:

- We define y_{pt} as 1 if project p is advanced to the next stage at time t, and 0, if it is frozen (put on hold). If a drug is frozen, it means that the next clinical trial does not start until an advancing decision (if any) is made in a future period.

Feasible Actions: The action that can be taken for a drug at time t depends on the status of the drug;

- If the current stage of a drug ended successfully, then,
  - If it was the last stage ($\psi_{pt} = 3$), no action is taken, profit occurs and f_{pt} is set to 2.
  - Else, the decision maker should either advance the project to the next stage or freeze it.
- If the project was frozen ($f_{pt} = -2$), again a decision has to be made about advancing or freezing the project.
- If the current stage is continuing or failed ($f_{pt} = 0, -1$), or the drug had completed all stages succesfully ($f_{pt} = 2$), then no action is taken ($y_{pt} = 0$).

Note that the set of feasible actions is constrained by the available resources. The set of feasible actions available at time t can be formulated as:

$$\mathcal{A}_t = \left\{ \boldsymbol{y}_t \middle| R^i_{max} - \sum_{p|f_{pt}=0} r^i_{p,\psi_{pt}} \geq \sum_p y_{pt} r^i_{p,\psi_{pt}},\ y_{pt} = 0 \ \forall\ p|f_{pt} = 0, -1, 2, i = 1, 2 \right\}. \tag{1}$$

This formulation can be extended to consider more complex resource requirements. For example, if the resource requirement at each time period is different, then $r^i_{p,\psi_{pt}}$ can be replaced with $r^{it}_{p,\psi_{pt}}$ and the feasibility within several periods ahead would be considered. This extension is investigated in the computational experiments. Note that two types of resources are used only for the comparison purposes with Christian and Cremaschi (2015), in the rest of the experiments a single resource type is assumed.

Update the state after the action:

- If drug p is advanced to the next stage ($y_{pt} = 1$), then the current stage and the drug status are updated: $\psi_{p,t+1} = \psi_{pt} + 1$, $f_{p,t+1} = 0$.
- Else if the drug is frozen ($y_{pt} = 0$): $\psi_{p,t+1} = \psi_{p,t}$, $f_{p,t+1} = -2$.
- Otherwise (the drug has failed), $\psi_{p,t+1} = 0$.

Figure 2 presents the possible transitions between different drug states. The drug can be in any of 5 possible states at any time. Depending on the action, the stage of the drug and the transition probabilities, it can move to any of 5 states. Note that the states where the drug is successful or the trial (and equivalently drug) has failed are absorbing states. The state of a successful drug does not change as it is completed. After a *trial*

success, if that is the last stage, then the system moves into the *drug success* state. If it is not the last stage, then it can be either frozen or advanced which correspond to moving into *trial continue* or *drug frozen* states. A continuing trial can continue, fail or become successful depending on the probabilities. A frozen drug can be advanced and the system moves into *trial continue* state or kept frozen.

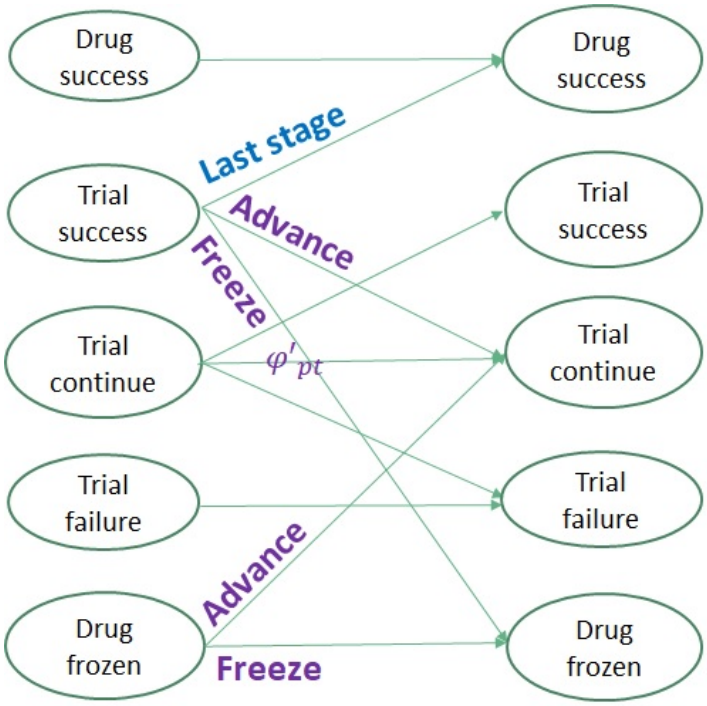

Figure 2: State transition diagram for drug pipeline management problem

One-step Cost Function: We assume that there is no cost of freezing while the cost of advancing is the lump sum cost of the corresponding stage, incurred at the beginning of that stage. One step cost of action $\boldsymbol{y_t}$ is denoted by $c(\boldsymbol{y_t})$ and formulated as follows:

$$c(\boldsymbol{y}_t) = \sum_p (1 - 0.025t)(q_{p,\psi_{pt}+1}) y_{pt},\ t = 1, \cdots, T-1, \tag{2}$$

that is linearly depreciated with rate 2.5% for each time period (Christian and Cremaschi (2015)).

Value Function: Note that the cost of clinical trial mainly depends on the sample size of the trial that is computed by the company based on statistical considerations. Once the trial protocol has been approved by the regulatory agencies, the company recruits the patients which mostly affects the duration of the trial. Therefore, the cost of the trial is not dependent on the trial duration but rather the sample size, i.e. we are assuming a stochastic recruitment rate. If all clinical trials of drug p are completed successfully at time t ($\psi_{pt} = 3$, $f_{pt} = 1$),

then the present value of the profit

$$m'_{pt} = m_p - \rho_p(t - \sum_s d_{ps}) + \gamma_{pt}$$

occurs. Note that the late launch penalty depends on the total time periods that the project has stayed idle. Let's represent total profit obtained from all projects finished at time t with $M_t(\boldsymbol{S}_t) = \sum_{p|\psi_{pt}=3, f_{pt}=1} m'_{pt}$.

The objective of dynamic programming formulation is to minimize the ENPV of the planning period. Let's represent the objective function as $V_0(S_0)$ which is also labelled as value function of the initial state S_0 in $t = 0$.

The value function (Bellman equation) can be formulated as:

$$V_t(\boldsymbol{S}_t) = \max_{\boldsymbol{y}_t \in \mathcal{A}_t} \Big\{ M_t(\boldsymbol{S}_t) - c(\boldsymbol{y}_t) + E[V_{t+1}(\boldsymbol{S}_{t+1}|\boldsymbol{S}_t, \boldsymbol{y_t})] \Big\}, \ \forall \boldsymbol{S}_t, \ \ t = 1, \cdots, T-1, \tag{3}$$

Note that all possible future states $\boldsymbol{S}_{t+1}$ are generated based on the current state $\boldsymbol{S}_t$ and action $\boldsymbol{y_t}$. Then, the expected value of each possible future state is multiplied with the probability of that corresponding state which depends on both probabilities of trial success and the completion, $\zeta_{pt}(h_{pt})$ and ϕ'_{pt}, in the stochastic duration case.

The value obtained at the end of the planning horizon consists of the expected profits from frozen and continuing projects, shown with m_p^{open} and m_p^{cont} and are formulated as

$$rev_p^{cont} = m_p - \gamma_p \left(T + \sum_{s=\psi_{pT}}^{3} d_{ps} - h_{pT} \right) - \sum_{s=\psi_{p,T+1}}^{3} q_{ps}, \ \ \forall p, \tag{4}$$

$$rev_p^{open} = m_p - \gamma_p \left(T + \sum_{s=\psi_{pT}+1}^{3} d_{ps} \right) - \sum_{s=\psi_{pT}+1}^{3} q_{ps}, \ \ \forall p, \tag{5}$$

$$m_p^{open} = rev_p^{open} \prod_{s=\psi_{pT}+1}^{3} \phi_{p,s}, \ \ \forall p, \tag{6}$$

$$m_p^{cont} = rev_p^{cont} \prod_{s=\psi_{pT}}^{3} \phi_{p,s}, \ \ \forall p. \tag{7}$$

The late launch penalties ρ_p and γ_p ensure that the profits computed for the open projects always have a smaller expected revenue than completing them within the planning horizon. Therefore, it is penalized to delay the projects. Note that Christian and Cremaschi (2015) use the following formulations for open and continuing

projects:

$$rev_p^{cont} = m_p - \gamma_p\Big(T + \sum_{s=\psi_{pT}}^{3} d_{ps} - h_{pT}\Big), \qquad \forall p, \tag{8}$$

$$rev_p^{open} = m_p - \gamma_p\Big(T + \sum_{s=\psi_{pT}+1}^{3} d_{ps}\Big), \qquad \forall p, \tag{9}$$

$$\epsilon_p = 0.9\left[\frac{m_p - \gamma_p T - \sum_{s=\psi_{pt}}^{3} q_{ps}}{m_p - \gamma_p T}\right], \qquad \forall p, \tag{10}$$

$$m_p^{open} = rev_p^{open}\epsilon_p, \qquad \forall p, \tag{11}$$

$$m_p^{cont} = rev_p^{cont}\epsilon_p, \qquad \forall p \tag{12}$$

that are used when comparing our algorithm with theirs. For both cases, the value at the end of planning horizon is formulated as

$$V_T(\boldsymbol{S_T}) = \sum_{p|f_{pT}=-2} m_p^{open} + \sum_{p|f_{pT}=0} m_p^{cont}.$$

The motivation behind different revenue formulations is that we consider the probability of failure in the coming trials explicitly rather than implicitly as in their formulation. Also note that the profit obtained in the previous time steps are then added backwards to the the value at the end of the planning horizon.

4. Solution Method: Approximate Dynamic Programming

The portfolio planning problem outlined in the previous section is computationally expensive to solve due to the large state space. For example, in a small instance with 3 drugs, and at most 6 time periods for a trial, the state space may be as large as $90^3 = 729,000$.

Note that there are only two possible actions for a frozen drug or a drug that successfully completes a trial. Considering that the probability of a successful trial is small, few drugs require an action at any decision point. Besides, the resource constraint limits the number of drugs that can be advanced at the same time. Thus, we use enumeration to find the optimal action at a decision point.

Simulation-based ADP is very suitable to solve large stochastic dynamic programming problems (Powell, 2009). To solve the pharmaceutical portfolio planning model, we develop a simulation-based ADP algorithm with double-pass (Powell, 2009). We apply both a lookup table and basis function approximation. In this section, we provide the details of the proposed ADP algorithm with the lookup table approach. A linear programming

based ADP is not applied since the value function (3) is complex (Powell, 2009). We implement a value iteration based algorithm instead of policy iteration since the problem has a large state space and a comparatively small action set (Sun and Li, 2013).

ADP is a forward pass algorithm by default, i.e. the algorithm moves forward in time at each step. However, the costs or revenues realized in the later time periods should be transferred to the previous time periods. With the default single pass version, this transfer may take many iterations. To overcome this problem, a double pass is suggested (Powell, 2007) which employs an additional backward pass updating the value function estimations by moving backwards in time in the trajectory. Note that most of the revenues are collected towards the end of the planning horizon (including for unfinished projects). Thus, we prefer a double pass, rather than a single-pass approach, to update the state values at each iteration. In the double-pass approach, at the end of an iteration, the value of each state in the chain is propagated backward from the last state through to the initial one.

In order to increase the number of explored states, a random feasible action is selected with probability Γ. Otherwise, the action is selected randomly among the optimum actions computed that have the same optimal value. However, this strategy may result in suboptimal policies and decreases the exploitation (Powell, 2009). Therefore, we only apply it for the first half of the iterations, i.e. for $n = 1, \cdots, N/2$, where n and N denote the iteration counter and the maximum number of iterations set by the modeller, respectively.

An initial state $\mathbf{S}^0$, as well as the probability distributions for trial outcomes are given as the inputs to the algorithm. As we run the algorithm, each visited state and its approximate value are inserted into a (lookup) table. At each iteration of the algorithm, we simulate a possible scenario over the entire planning horizon. The decisions implemented in each scenario are computed based on the Bellman equation (3) and the estimated values of the states computed in the previous iterations. Next, the algorithm employs a backward pass, i.e. following the scenario back in time and updating the values of the earlier states. Algorithm 1 shows the pseudo-code of the ADP algorithm with value iteration, lookup table and double pass. The algorithm is divided into four main steps. After initialization, the algorithm enters a loop consisting of carrying over the values computed in the previous iteration, and then executing a forward and backward pass. During the forward pass, it simulates a scenario and chooses actions based on Bellman's equation or randomly based on some probability in the first half of the run. During the backward pass, it updates the state value estimates based on the information gained during the forward pass. Once the maximum number of iterations has been reached, it returns the computed values and states, i.e. the lookup table.

The initial state, with all drug candidates frozen, is the same in each iteration, and its value ($\mathbf{S}_1^0$) is initialised as zero. Based on the probability distributions of trial outcomes, at each iteration n and time t, the outcomes of

completed trials, and thus the drug status vector $\boldsymbol{f}_t^n$, are generated. If $n \leq N/2$ and a generated random number $\omega \in [0,1]$ is smaller than Γ, then an action $\boldsymbol{y}_t^n \in \mathcal{A}$ is chosen randomly among feasible actions. Otherwise, it finds the optimum action $\boldsymbol{y}_t^n \in \mathcal{A}$ as well as the state value, represented with $v_t^n(\mathbf{S}_t^n)$, by using the Bellman equation (3) and the approximate values stored in the lookup table. If the value of a state required in the calculation of the selected action has not been visited by the algorithm before, then it is estimated as the summation of the ENPVs of the continuing projects which is a lower bound. The ENPV of a continuing project is computed by multiplying the success probabilities of the remaining clinical trials of the project with its discounted profit. $\mathbf{S}_{t+1}^n$ can be computed based on the selected action $\boldsymbol{y}_t^n$, and the state $\mathbf{S}_t^n$. In other words, for the ongoing projects in $\mathbf{S}_t^n$ and the advanced drugs $\boldsymbol{y}_t^n$, a new drug status vector $\boldsymbol{f}_{t+1}^n$ is generated based on the success probabilities. The other state variables are also updated as explained in Section 3. The value stored in the lookup table for state $\mathbf{S}_t^n$ is denoted by $\overline{V}_t^n(\mathbf{S}_t^n)$ for $n = 1, \cdots, N$ and $t = 1, \cdots, T$.

In each iteration $n = 1, \cdots, N$, after all states in the sample path have been visited, the algorithm goes backward in time and recursively reflects the values of the future states (in the sample path) into v_t^n for $t = T-1, \cdots, 1$. If a state $\mathbf{S}_t^n$ is visited for the first time by the algorithm, then its computed value v_t^n is directly added to the lookup table, i.e. $\overline{V}_t^n(\mathbf{S}_t^n) = v_t^n(\mathbf{S}_t^n)$. Otherwise, $\overline{V}_t^n(\mathbf{S}_t^n)$ is computed by summing $v_t^n(\mathbf{S}_t^n)$ and the value of the state most recently stored in the lookup table, $\overline{V}_t^{n-1}(\mathbf{S}_t^n)$, after weighting them by a smoothing parameter, α_n: $\overline{V}_t^n(\mathbf{S}_t^n) = \alpha_n \overline{V}_t^{n-1}(\mathbf{S}_t^n) + (1-\alpha_n) v_t^n(\mathbf{S}_t^n)$. Since the state values are expected to approach their exact values through iterations, α_n is formulated as a linearly increasing function of n: $\alpha_n = a + b\alpha_n$ where a and b are parameters. The linear form is selected because it is simple and also converges eventually (Powell, 2007). Finally, the values stored in the lookup table for all states visited until iteration n are carried over to the next iteration; $\overline{V}_t^{n+1}(\mathbf{S}_t^k) = \overline{V}_t^n(\mathbf{S}_t^k)$ for $k = 1, \cdots, n$ and $t = 1, \cdots, T$.

Pseudo-code of the ADP algorithm

Initialization: Set maximum number of iterations N, Γ and $n = 1$. Initialize the value of the initial state $\overline{V}_1^0(\mathbf{S}_1^k)$ as 0 for $k = 1, \cdots, n-1$, $n = 1, \cdots, N$.

for $n = 1, 2, \cdots, N$, **do**

Carry over: For $k = 1, \cdots, n$ and $t = 1, \cdots, T$, set $\overline{V}_t^n(\mathbf{S}_t^k) = \overline{V}_t^{n-1}(\mathbf{S}_t^k)$.

Forward Pass:

for $t = 1, 2, \cdots, T-1$, **do**

– Generate $\boldsymbol{f}_t^n$ based on $\mathbf{S}_{t-1}^n$.

– Generate a random number ω and,

if $n \leq N/2$ and $\omega \leq \Gamma$ **then**

Randomly select $\boldsymbol{y}_t^n$ among the feasible action set $\mathcal{A}$, and compute $v_t^n(\mathbf{S}_t^n)$ by using (3).

else

– Find the action $\boldsymbol{y}_t^n$ and $v_t^n(\mathbf{S}_t^n)$ by solving (3) based on the state values stored in the lookup table.

– If a state value does not exist in the lookup table, then its value is assumed to be the sum of ENPVs of all frozen and continuing projects.

end if

– Update state variables based on the action $\boldsymbol{y}_t^n$ and $\boldsymbol{S}_t^n$: $\mathbf{S}_{t+1}^n = \mathbf{S}_{t+1}(\mathbf{S}_t^n, \boldsymbol{y}_t^n)$.

end for

Backward Pass:

for $t = T-1, \cdots, 1$ **do**

– Compute $v_t^n(\mathbf{S}_t^n) = v_{t+1}^n(\mathbf{S}_{t+1}^n) - c(\mathbf{y}_t^n)$, where $c(\mathbf{y}_t^n)$ is defined as in (2).

if state $(\mathbf{S}_t^n)$ exists in the lookup table, **then**

– Update $\overline{V}_t^n(\mathbf{S}_t^n) = (\alpha^{n-1})\overline{V}_t^{n-1}(\mathbf{S}_t^n) + (1-\alpha^{n-1})v_t^n(\mathbf{S}_t^n)$,

else

– Set $\overline{V}_t^n(\mathbf{S}_t^n) = v_t^n(\mathbf{S}_t^n)$.

end if

end for

end for

Return all value function approximations ($\overline{V}_t^N$, i.e. lookup table) for $t = 1, \cdots, T$.

A high level description of the algorithm can be seen in Figure 3.

4.1. Extension to Base Formulation

The portfolio planning model presented in Section 3 allows us to compare the ADP algorithm with other solution methods proposed in the literature. However, this formulation still lacks important aspects of the problem such as emergence of new drug candidates or stochastic trial durations. To incorporate the emergence of new drug candidates into the ADP algorithm, at each time period a new drug is added to the list like a frozen

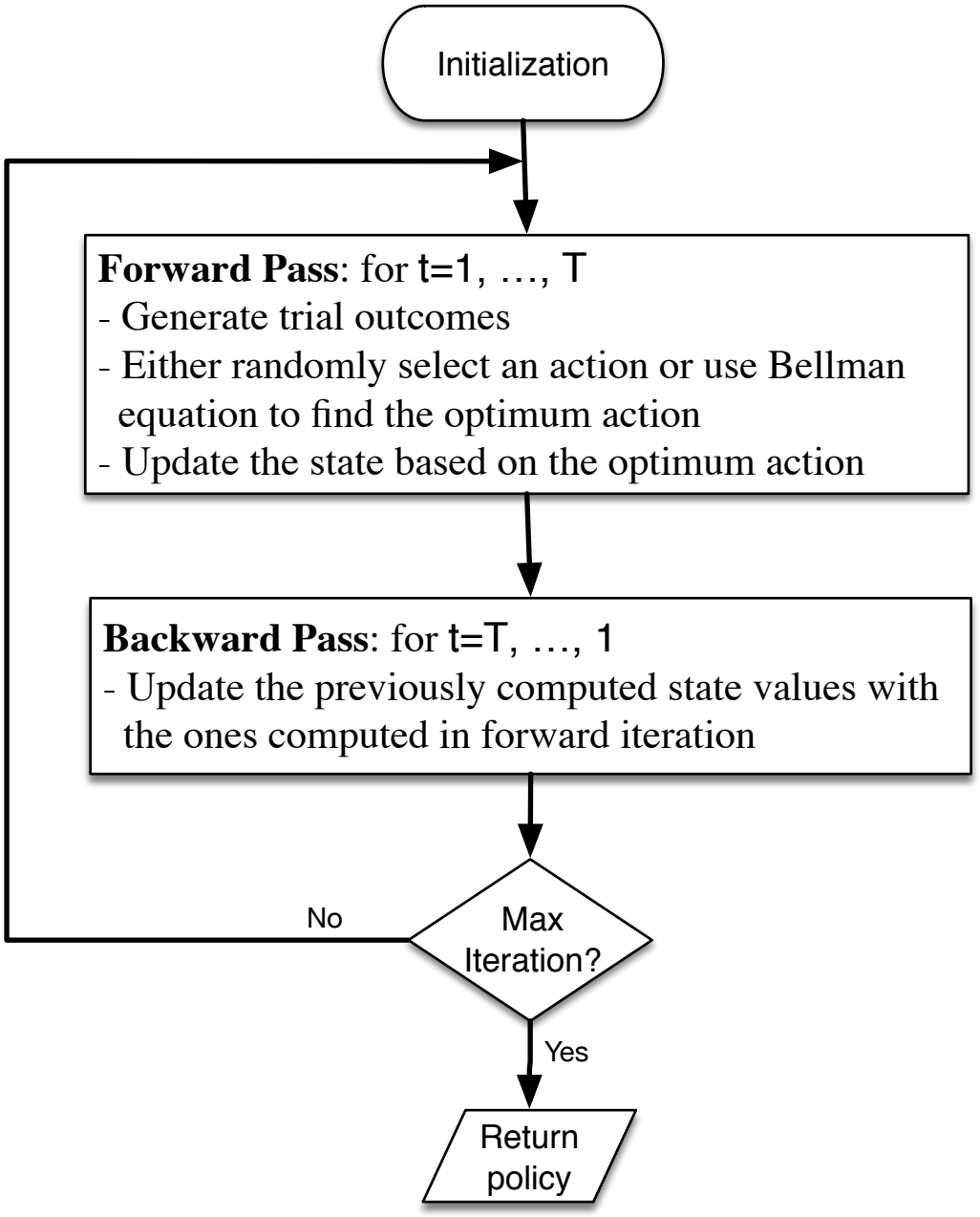

Figure 3: High level flowchart of the ADP algorithm

drug: $f_{pt} = -2, \psi_{pt} = 0, h_{pt} = 0$ with a small probability, 0.1, based on expert opinion. Note that since at most one drug can be added at each time period, the number of new drugs is finite. Therefore, the drug list can be initialized with maximum number of possible drugs in the portfolio during the planning horizon, whereas only really existing drugs are considered.

Another possible extension is to include drug dependencies. Pharmaceutical companies can target one disease with several drug candidates. In this case, the technical or financial attributes of these candidates would be affected by the failure or success of other similar candidates (Blau et al., 2004). If one of the drugs in a similar group of drugs fails (succeeds) in a clinical stage, then the probability of failure (success) for the other drugs in the group increases. Also, we assume that if a drug in a group launches successfully, the expected revenues of the other drug candidates in the same group decrease. A scenario with drug dependencies is investigated in the computational experiments.

Other possible extensions considered are *variable resource requirements during a trial* and *speeding the trials by putting more resources*. The first extension requires to update the formulation of feasible actions 1. The other extension requires to double the action space by defining the resource levels of each continued trial as decision variables. The applicability of ADP on these extensions is presented in the computational experiments for a small instance.

4.2. Basis Functions and Value Function Approximation

The lookup table ADP does not guarantee to provide a value for each possible state, even though it covers most of the state space. Thus, an alternative approach based on basis functions is also implemented which ensures to provide a value function approximation for all possible states. In this approach, the state values are approximated by a basis function instead of reading them from a lookup table. Usually, the basis functions are selected as a linear combination of the state variables and based on trial-and-error (Powell, 2007). To find a good basis function, we apply regression on the approximate state values computed by the lookup table approach. After trial and error, the best fit for approximate state values is achieved by:

$$\overline{V}(S_t) = w_1 \sum_p \theta_{pt} + w_2 \left(\left(\sum_{p|f_{pt} \neq -1,2} \mathbb{1} \right) \sum_p (m_p/P) \right), \tag{13}$$

where,

$$\theta_{pt} = \begin{cases} 2\psi_{pt} m_p, & \text{if } f_{pt} = -2||1, \\ 0.5\psi_{pt} m_p & \text{elseif } f_{pt} = 0, \\ 0, & \text{otherwise,} \end{cases}$$

and w_1 and w_2 are the weights of the corresponding basis functions. The second summation in (13) computes the number of projects that are not completed yet. The values for these weights are updated as the algorithm runs for each particular instance, i.e. they are dependent on the particular instance. The approximate state values obtained by the proposed basis function converge to the ones computed by the lookup table approach. This indicates that this structure can be used to find an approximate policy (Powell, 2007). The basis function structure suggests that the expected profit of a drug has a positive effect on advancing that drug. Also, it is preferred to advance the drug that has completed more phases. Our experiments with different instances showed that the R-square levels and coefficients do not differ substantially for different instances of the problem.

5. Computational Experiments

The computational experiments consist of three parts. The first part simply asserts the stability of the policy derived by ADP. The second part examines the performance of ADP under a variety of problem settings. Because we are not aware of any other published approach that is capable of handling the complexities considered, we compare the results to a myopic heuristic often used in industry based on our discussions with industry partners. For these experiments, we use test instances based on a dataset that is partially collected from industry (BIO et al., 2016) and shown in Table 8 in the Appendix. Trial durations for Phase 2 & 3 are assumed to take $\{\bar{x}-1, \bar{x}, \bar{x}+1\}$ with equal probabilities, where $\bar{x}$ is given in Table 8 under the trial duration column. The durations of Phase 1 are assumed to be deterministic. Maximum available capacities are 15, 6 and 3 for 20, 10 and 5-drug instances, respectively. The final part compares the performance of ADP with a state-of-the-art solution method from the literature on simplified problem instances with deterministic trial durations. We assume that decisions are taken every 3 months and there are 14 time periods.

5.1. The Stability of the Derived ADP Policy

The policy resulting from an ADP run depends on the scenarios generated during the run. Therefore, a different ADP policy may be obtained at every time the algorithm is run. We expect that the policies obtained from different runs would be similar to each other. To validate this assumption, we generate 30 ADP policies

(from 30 runs) by using the same set-up (1000 iterations). Then, we apply these policies on 1000 randomly generated pathways and obtain the ENPVs corresponding to each policy for a 5-drug instance.

To investigate the source of variation in ADP runs, we conduct an ANOVA analysis on the ENPV values obtained by different ADP runs. The p-values, shown in Table 2, suggest that the variance is caused by in-sample simulation variation, while the variation due to different ADP policies is not significant (the second row). The following results will thus be based on a single policy.

Table 2: ANOVA results for source of variation

Source of Variation	*F-value*	*P-value*	*F crit*
Scenario	6.501	5.4E-275	1.113
Policy	0.538	0.847	1.881

Note that the stability of the ADP algorithm is independent of the particular instance due to this characteristic being inherent to the nature of the algorithm.

5.2. Performance Comparison of ADP Algorithm with a Myopic Approach

In this section, we use the drug pipeline management model with uncertain durations. As there is no algorithm in the literature to solve the full model defined in Section 3, we resort to a greedy (myopic) approach that is also applied by pharmaceutical companies in practice (based on expert opinion). In this greedy approach, the available projects in the pipeline are ordered according to decreasing ratio of expected revenue/resource requirements over all stages ahead. Then, starting from the top of this list, the projects are advanced until there are not enough resources left to advance another project.

In this section, we compare the performance of such a greedy approach and our ADP algorithm considering the uncertainties in trial duration and outcome. We obtain an ADP policy (a lookup table) by running ADP algorithm for 1000 iterations and apply the myopic approach and the ADP policy on 1000 scenarios. The average ENPVs and standard errors reported throughout this section are obtained from applying these two algorithms on these scenarios. Note that when a state does not exist in the lookup table of the ADP policy, we use the basis function approximation to find the optimum action (cf. Section 4.2). The average ENPVs obtained by ADP and myopic approaches, along with standard errors and % improvement of ADP over myopic in terms of ENPV are presented in Table 3 for 3 different instances.

Table 3: Average, ± standard error and % improvement of ENPVs obtained by myopic and ADP policies

	20-drug		**10-drug**		**5-drug**	
ADP	601 ± 10.17	13%	300 ± 7.13	21%	77 ± 4.78	30%
Myopic	528 ± 3.26		247 ± 3.24		59 ± 2.52	

The ADP algorithm results in around 13, 21 and 30% larger ENPV than the myopic approach for 20, 10 and 5-drug instances, respectively. These results indicate that as the portfolio size is smaller, the advantage of ADP increases.

Variable Resource Requirements during a Trial: In this extension, we assume that the resource requirements are not fixed throughout a clinical trial but vary over time. The formulation that computes the feasible action space, (1), is extended to consider the new requirements. In this case, r^i_{ps} is replaced with $r^{it'}_{p,s}$, where t' represents the time passed in the current trial of drug p (i.e. for $t' > d_{p,\psi_{pt}}$, $r^{it'}_{p,s} = 0$), and the rest of formulation (1) becomes:

$$\mathcal{A}_t = \left\{ \boldsymbol{y}_t \middle| R^i_{max} - \sum_{p|f_{pt}=0} r^{i,t'+h_{pt}}_{p,\psi_{pt}} \geq \sum_p y_{pt}\left(r^{it'}_{p,\psi_{pt}+1}\right),\ i=1,2,\ t'=1,\cdots,T-t,\ y_{pt}=0\ \forall\ p|f_{pt}=0,-1,2, \right\},$$

which checks whether the action is still feasible for the upcoming periods. Since the resource requirements may change each time period, actions that were not feasible in the previous time period may become feasible later. Therefore, in the following period, the algorithm needs to check again whether the frozen drugs can be advanced, if there are any. This leads to a longer computation time.

To examine the algorithm's performance in this extended model, we solve the 5-drug case of Christian and Cremaschi (2015) (Table 7) assuming that the resource requirements increase by one unit from their original levels in the 3rd and 4th time periods of a trial and decreases by one unit in the 5th and 6th time periods. In this case, we have to increase the maximum available resource level of 2nd resource type to 4, otherwise none of the drugs could be completed. The ENPV changes to 295 $\pm$ 5.47, whereas the ENPV of the original model, solved with the increased maximum resource as the extended one, is 303 $\pm$ 5.6, showing no significant difference. The computation time (50 seconds), as well as the policy obtained, do not show a significant difference to the original model. The myopic policy provides a significantly smaller ENPV in both cases, as shown in Figure 4.

Choosing Resource Requirements to Speed up Trials: In practice, it is often possible to speed up a trial by allocating additional resources. This case requires to expand the action space, as the resource level for each advanced trial becomes a decision variable. For example, a trial can be executed with more resource use and shorter duration in addition to the default resource use and duration. In this case, there would be 2 resource categories available. Assuming that X_{ps} resource categories are available for stage s of drug p, the resource category for drug p in period t is defined as $x_{pt} \in \{0, 1, \cdots, X_{p,\psi_{pt}+1}\}$, as the additional decision variable, where 0 corresponds to not advancing the drug. An integer variable, $x'_{pt} \in \{0, 1, \cdots, X_{p,\psi_{pt}}\}$,

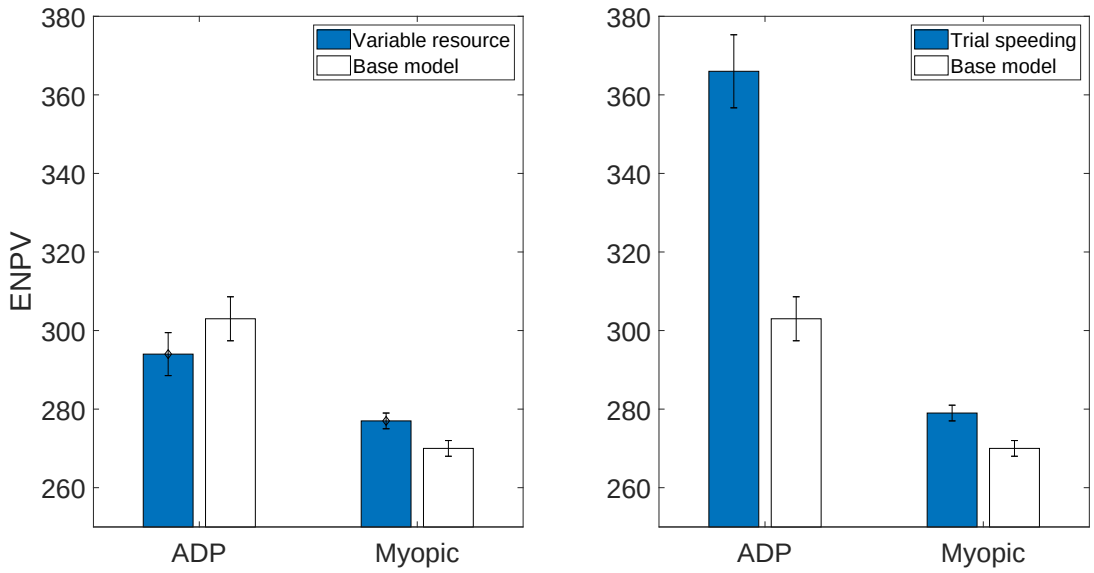

Figure 4: ENPVs obtained by ADP and myopic approach in different model extensions

is added to the state space indicating the resource category used for the execution of the current trial of drug p. The resource levels and trial durations corresponding to each resource category are also defined as additional parameters: d_{ps}^{j} and r_{ps}^{j}, where $j \in \{1, \cdots, X_{p,s}\}$, which replace d_{ps} and r_{ps} in the original model.

Since the action space is doubled, the computation time of the algorithm is expected to rise. To examine its performance, we solve this extended model for the 5-drug case of Christian and Cremaschi (2015), assuming one more resource category for Phase 2 and Phase 3 trials of each drug (the duration of Phase 1 trials are short anyway), in addition to the original levels. The additional resource category reduces the (original) duration of a trial by two periods with a one unit increase in the resource use. Due to the increased resource use in Phase 3 trials, we again need to set the maximum available resource to 4. To compare the result of this extension, we solve the original model again with the increased maximum available resource. As shown in Figure 4, the ENPV increases to 366 ± 9.4, which is significantly larger than the original model, while the computation time increases to 192 seconds. The ADP policy usually speeds up Phase 3 trial while this option is used much less frequently in Phase 2 trials. The myopic policy still provides a significantly smaller ENPV compared to the ADP.

Capacity Tightness: The resource capacity may affect the performance of the solution algorithms. To investigate the impact of capacity, we solve the 20-drug instance with 10 (low) and 20 (high) units of capacity. In addition, to emphasize the relevance of resource restriction, the probability of a new drug appearing

during a time period is set to 0.2 in all runs. When there are new drugs appearing, the capacity usage becomes important as the resource requirements increase.

Figure 5 shows the ENPVs and their standard errors obtained by the ADP and myopic algorithms in the standard (base), high and low capacity cases along with the percentage difference between the ENPVs obtained by two methods. The results show that ENPV obtained by ADP is higher than that of the myopic approach by 14% and 18% in high and low capacity cases, respectively, for the standard (base) portfolio. This indicates that ADP performs even better relative to the myopic approach when the capacity is tighter.

Portfolio Compositions: A pharmaceutical pipeline may target different therapeutic areas such as oncology or hypertension. Due to large patient populations of some therapeutic areas, corresponding drug candidates can have comparatively larger resource requirements and expected revenues. On the other hand, stratified medicines may have comparatively less resource requirements and revenues. These different pipeline compositions may affect the performances of the solution methods. In the current test, we apply ADP and myopic algorithms for two different types of portfolios both with 20 drugs. The first one (labelled as 'diverse portfolio') includes at least 6 comparatively large projects while the rest are composed from medium and small size projects. In the second portfolio (labelled as 'harmonized portfolio'), the projects have similar (medium level) resource requirements and expected revenues.

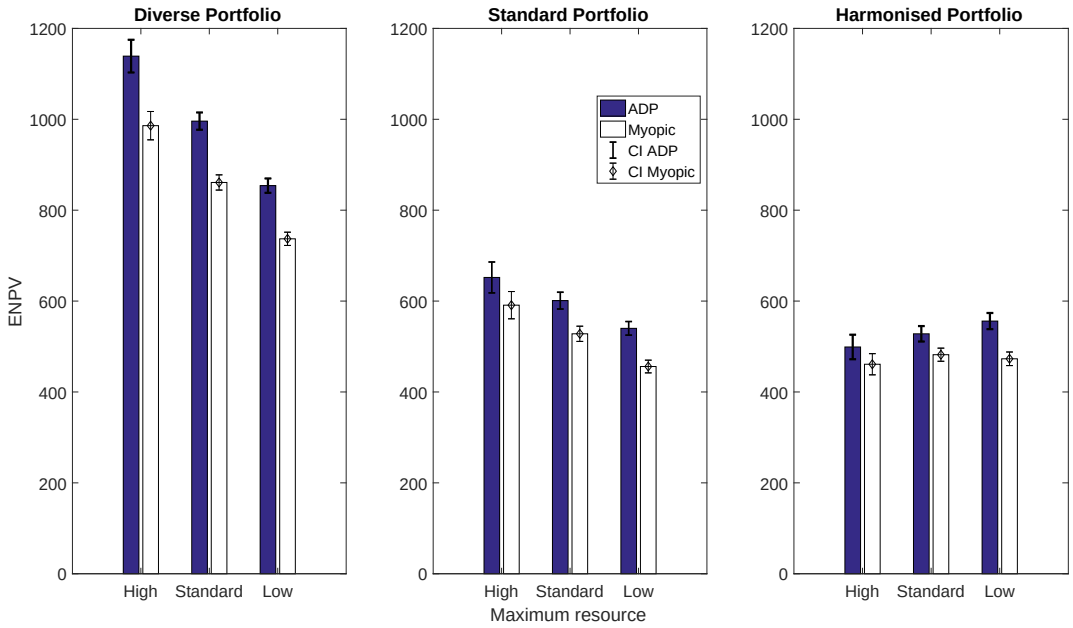

Figure 5: ENPVs obtained by ADP and myopic approach in different available resource and portfolio compositions

Table 4: ENPVs obtained by ADP and myopic approaches and % improvement of ADP over myopic in different decision timing strategies

	ADP		Myopic	
Flexible (3 months)	601 ± 19	20%	528 ± 17.8	22%
Fixed (6 months)	500± 16.74		412 ± 14.6	

Figure 5 shows the average ENPV ± its standard error for different portfolios and capacity levels. The results indicate that improvement of ADP over the myopic approach is higher as the portfolio gets more diverse. We also observe that the capacity change has a more significant impact on the harmonized portfolio.

Flexible vs. Fixed Decision Intervals: So far, we assumed that the decision-maker takes an action as soon as a trial has been completed. An alternative assumption is that the decisions are made at fixed time intervals such as every 6 months. However, this strategy may cause drugs to be released later than they could, which results in less profit. We implement both strategies (flexible with 3 months and fixed with 6 months intervals) and show ENPVs obtained by ADP and myopic policies for the 20-drug case (standard portfolio with 15 units of capacity) in Table 4. In other words, the flexible approach allows to make decisions once in 3 month which is much shorter than that in the fixed interval setting.

By allowing for flexible decision intervals, ENPVs increase by 20% and 22% for ADP and myopic approaches, respectively. It should be noted that flexible decision timings may be burdensome and not very practical for senior management in the real world. However, due to its significant effect on ENPV, it should be considered. Finally, a smaller time period such as 3 months is not computationally more expensive, as no decision is made in some time periods.

Drug Dependencies: We investigate the effect of drug dependencies on the performance of solution approaches and assume that the drug candidates are divided into several groups. If the first one of the drugs in a group fails (succeeds) in a clinical stage, the probability of failure (success) for the other drugs in the group increases by 50% for the same stage. Also, once the first drug in a group launched, the expected revenues of the other drug candidates in the same group drop by 50%. The increase/decrease rates are chosen based on expert opinion.

We divide the drugs into the following groups: {1, 2}, {3, 4, 5}, {6, 7, 8, 9, 10}, {11, 12, 13, 14, 15, 16}, and {17, 18, 19, 20}. To investigate the effects of two types of dependencies (success probability and revenues) separately, we present the ENPVs for three cases: (i) without any dependencies, (ii) with success probability dependencies only, and (iii) with both success and revenue dependencies together. Table 5

shows the average and standard error of ENPVs obtained by ADP and myopic approaches as well as the percentage improvement between those averages for three cases.

Table 5: Effect of success probability and revenue dependencies on ENPVs obtained by ADP and myopic approaches

No dependency	ADP Myopic	601 ± 19 528 ± 16.8	13%
Probability dependency	ADP Myopic	560 ± 36 492± 28	13 %
Both dependencies	ADP Myopic	505 ± 41.2 431 ± 29.7	17 %

When there is only the success probability dependency, ENPVs obtained by myopic and ADP policies drop by around 7%. The profits drop by around 19% for ADP policy and 22% for the myopic policy when both dependencies are present. The results indicate that ADP can handle drug dependencies better than the myopic approach.

5.3. Performance Comparison of the ADP Algorithm with Other Approaches from Literature

The main motivation behind the experiments of this section is to show that the proposed ADP algorithm can be used to solve larger problems than the methods proposed in the literature, and still within acceptable computation times. For this purpose, we select Christian and Cremaschi (2015) as the benchmark study. It is a recent paper focusing on pharmaceutical pipeline management and also with the biggest instance (10 drugs) solved so far, therefore considered as state-of-the-art. We used the same modelling assumptions and dataset as Christian and Cremaschi (2015) which is replicated in Table 7 in the Appendix for 10 drugs. For 3 and 5-drug instances, we used the information in this table for the first 3 and 5 drugs, respectively. The 20-drug instance comprises of two copies of 10-drug instance shown in Table 7. Similarly, the 40-drug instance comprises of the four copies of the 10-drug instance. To investigate the effect of the number of iterations, we run the ADP algorithm for 200 and 1000 iterations for each instance. The exact solutions reported in Christian and Cremaschi (2015), are shown in Table 6 for different numbers of drugs. They have used CPLEX to find the exact solutions of the dynamic model. The table also presents the computation times of the (Knapsack) algorithm proposed therein and obtained on our computers using the code kindly provided by the authors. Note that the 40 and 20-drug instances are not solved in Christian and Cremaschi (2015), probably due to the computational intractability.

We also report the ENPV values in terms of mean ± standard error obtained by ADP algorithm over 20 separate tests with 1000 scenarios in each run. The results indicate that the ADP algorithm results in reasonable

Table 6: Computational performance of the ADP algorithm and the benchmark solution method for different number of iterations and instances

Instance	Sol. Method	No. of iterations	Solver time (sec.)	ENPV	Optimal ENPV
3-drug	ADP	200	1.27	116 ± 0.75	118.9
		1000	19	118.31 ± 0.75	
	C&C (2015)	-	<1	118.9	
5-drug	ADP	200	3.66	208 ± 1.3	208.3
		1000	43.87	207.63 ± 1.3	
	C&C (2015)	-	1	204.3	
10-drug	ADP	200	29.9	372 ± 2.25	-
		1000	526	441 ± 2.25	
	C&C (2015)	-	1296	406	
20-drug	ADP	200	109	809 ± 5.73	-
		1000	2254	911 ± 6.37	
40-drug	ADP	200	3.5 hrs	1866 ± 13.49	-
		1000	20 hrs	2186 ± 15.45	

computation times across all instances and higher ENPV's than the Knapsack algorithm for instances except the 3-drug instance where the difference is not statistically significant in 1000 iterations. Although the computation time is 20 hours for the 40-drug instance, consider that this policy is generated for more than a 5-year planning horizon and it may be reduced by using faster computers and a different programming language. Also note that Knapsack algorithm has been coded in Python whereas the ADP algorithm is coded in Matlab. Therefore, the comparison of computation times is only indicative. All computational experiments are carried out on a PC with Windows 10 Enterprise operating system, CPU 4GHz Intel Core i7 and 32GB of RAM.

Finally, we also tested the algorithm proposed in Choi et al. (2004) on the 5-drug case. This algorithm needs heuristics to determine the reachable states. Because our problem setting is somewhat different from theirs, we added our myopic heuristic to their set of heuristics. Still, the algorithm only reached an ENPV of 191.15 (8% from optimum) and required significantly longer running times with 1000 iterations for heuristics.

6. Conclusion

Effective drug pipeline management is crucial for the performance of a pharmaceutical company. However, it is a complex task due to substantial uncertainties and the dynamic nature of the problem. In this paper, we present a new solution approach to the pipeline management problem as well as model some unexplored features such as uncertain trial durations and expansion of the pipeline. With the proposed ADP solution approach, we are able to solve problems of size larger than what has been solved in the literature before, within reasonable computational times. ADP provides comparable performance to the state-of-art methods for smaller simple

instances.

We compare the performances of ADP and a myopic heuristic on more complex problems involving uncertain trial durations, new drugs appearing, and dependencies between drugs. Our computational experiments show that ADP provides at least 10% increase in ENPV compared to the myopic heuristic. The relative advantage of ADP increases as the portfolio size gets smaller. We also investigate the impact of several technical factors such as the level of capacity, different portfolio structures, timing of decisions and drug dependencies on the performance of ADP. These experiments reveal that ADP provides a fairly robust performance against changes in the environmental factors in terms of its advantage over the myopic heuristic.

Future studies can consider more complexities regarding the problem such as the outsourcing the execution of trials as an additional decision. Another possible extension is considering a parallel companion diagnostic development process along with the drug development. A successful companion diagnostic development could decrease the cost and duration of the trials.

Acknowledgements

We would like to thank Selen Cremaschi and Zuo Zen for making available their Knapsack Decomposition heuristic and Xin Fei for his comments on an initial draft of this paper. We also gratefully acknowledge support through the Engineering and Physical Sciences Research Council, grant EP/P006485/1.

References

BIO, Biomedtracker, Amplion, 2016. Clinical Development Success Rates. Technical Report.

Blau, G., Mehta, B., Bose, S., Pekny, J., Sinclair, G., Keunker, K., Bunch, P., 2000. Risk management in the development of new products in highly regulated industries. Comput. Chem. Eng. 24, 659–664. doi:10.1016/S0098-1354(00)00388-4.

Blau, G.E., Pekny, J.F., Varma, V.A., Bunch, P.R., 2004. Managing a portfolio of interdependent new product candidates in the pharmaceutical industry. J. Prod. Innov. Manag. 21, 227–245. doi:10.1111/j.0737-6782.2004.00075.x.

Choi, J., Realff, M.J., Lee, J.H., 2004. Dynamic programming in a heuristically confined state space: a stochastic resource-constrained project scheduling application. Comput. Chem. Eng. 28, 1039–1058. URL: http://www.sciencedirect.com/science/article/pii/S0098135403002412, doi:https://doi.org/10.1016/j.compchemeng.2003.09.024.

Choi, J., Realff, M.J., Lee, J.H., 2007. A Q-Learning-based method applied to stochastic resource constrained project scheduling with new project arrivals. Int. J. Robust Nonlinear Control IFAC-Affiliated J. 17, 1214–1231.

Christian, B., Cremaschi, S., 2015. Heuristic solution approaches to the pharmaceutical R&D pipeline management problem. Comput. Chem. Eng. 74, 34–47. URL: http://dx.doi.org/10.1016/j.compchemeng.2014.12.014, doi:10.1016/j.compchemeng.2014.12.014.

Christian, B., Cremaschi, S., 2017. Variants to a knapsack decomposition heuristic for solving R&D pipeline management problems. Comput. Chem. Eng. 96, 18–32. URL: http://www.sciencedirect.com/science/article/pii/S0098135416303271, doi:https://doi.org/10.1016/j.compchemeng.2016.10.011.

Colvin, M., Maravelias, C.T., 2008. A stochastic programming approach for clinical trial planning in new drug development. Comput. Chem. Eng. 32, 2626–2642. URL: http://linkinghub.elsevier.com/retrieve/pii/S0098135407002876, doi:10.1016/j.compchemeng.2007.11.010.

Colvin, M., Maravelias, C.T., 2010. Modeling methods and a branch and cut algorithm for pharmaceutical clinical trial planning using stochastic programming. Eur. J. Oper. Res. 203, 205–215. URL: http://dx.doi.org/10.1016/j.ejor.2009.07.022, doi:10.1016/j.ejor.2009.07.022.

Creemers, S., De Reyck, B., Lambrecht, M., Leus, R., 2009. Project scheduling with alternative technologies and stochastic activity durations, in: Proc. 4th Multidiscip. Int. Sched. Conf. Theory Appl. (MISTA 2009), Citeseer.

Creemers, S., De Reyck, B., Leus, R., 2015. Project planning with alternative technologies in uncertain environments. Eur. J. Oper. Res. 242, 465–476.

Creemers, S., Leus, R., Lambrecht, M., 2010. Scheduling Markovian PERT networks to maximize the net present value. Oper. Res. Lett. 38, 51–56.

De Reyck, B., Leus, R., 2008. R&D project scheduling when activities may fail. IIE Trans. 40, 367–384. URL: https://doi.org/10.1080/07408170701413944, doi:10.1080/07408170701413944.

Gatica, G., Papageorgiou, L., Shah, N., 2003. Capacity Planning Under Uncertainty for the Pharmaceutical Industry. Chem. Eng. Res. Des. 81, 665–678. URL: http://linkinghub.elsevier.com/retrieve/pii/S0263876203723533, doi:10.1205/026387603322150516.

Hermans, B., Leus, R., 2018. Scheduling Markovian PERT networks to maximize the net present value: New results. Oper. Res. Lett. 46, 240–244.

Informa UK, 2017. Pharma R&D Annual Review 2017. Technical Report.

Jekunen, A., 2014. Decision-making in product portfolios of pharmaceutical research and development. Drug Des. Devel. Ther. 8, 2009–2016. doi:10.2147/DDDT.S68579.

Laínez, J.M., Schaefer, E., Reklaitis, G.V., 2012. Challenges and opportunities in enterprise-wide optimization in the pharmaceutical industry. Comput. Chem. Eng. 47, 19–28. URL: http://dx.doi.org/10.1016/j.compchemeng.2012.07.002, doi:10.1016/j.compchemeng.2012.07.002.

Li, H., Womer, N.K., 2015. Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming. Eur. J. Oper. Res. 246, 20–33.

Li, X., Nie, L., Chen, S., 2014. Approximate dynamic programming based data center resource dynamic scheduling for energy optimization, in: Internet Things (iThings), 2014 IEEE Int. Conf. on, Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Soc. Comput. (CPSCom), IEEE, IEEE. pp. 494–501.

Lines, S.A., 2012. Drug patent expirations and the "patent cliff.". US Pharm 37, 12–20.

Lu, Y., Xie, X., Jiang, Z., 2018. Dynamic appointment scheduling with wait-dependent abandonment. Eur. J. Oper. Res. 265, 975–984.

Papageorgiou, L.G., Rotstein, G.E., Shah, N., 2001. Strategic supply chain optimization for the pharmaceutical industries. Ind. Eng. Chem. Res. 40, 275–286.

Patel, N.R., Ankolekar, S., 2015. Dynamically optimizing budget allocation for phase 3 drug development portfolios incorporating uncertainty in the pipeline, in: Optim. Pharm. R&D Programs Portfolios. Springer, pp. 181–200.

Patel, N.R., Ankolekar, S., Antonijevic, Z., Rajicic, N., 2013. A mathematical model for maximizing the value of phase 3 drug development portfolios incorporating budget constraints and risk. Stat. Med. 32, 1763–1777.

Perez-Escobedo, J.L., Azzaro-Pantel, C., Pibouleau, L., 2012. Multiobjective strategies for New Product Development in the pharmaceutical industry. Comput. Chem. Eng. 37, 278–296.

Powell, W.B., 2007. Approximate Dynamic Programming: Solving the curses of dimensionality. volume 703. John Wiley & Sons.

Powell, W.B., 2009. What you should know about approximate dynamic programming. Nav. Res. Logist. 56, 239–249. doi:10.1002/nav.20347.

Powell, W.B., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality. volume 842. John Wiley & Sons.

Rajapakse, A., Titchener-Hooker, N.J., Farid, S.S., 2005. Modelling of the biopharmaceutical drug development pathway and portfolio management. Comput. Chem. Eng. 29, 1357–1368. doi:10.1016/j.compchemeng.2005.02.010.

Rajapakse, A., Titchener-Hooker, N.J., Farid, S.S., 2006. Integrated approach to improving the value potential of biopharmaceutical R&D portfolios while mitigating risk. J. Chem. Technol. Biotechnol. 81, 1705–1714. URL: http://dx.doi.org/10.1002/jctb.1595, doi:10.1002/jctb.1595.

Rotstein, G., Papageorgiou, L., Shah, N., Murphy, D., Mustafa, R., 1999. A product portfolio approach in the pharmaceutical industry. Comput. Chem. Eng. 23, S883–S886. doi:10.1016/S0098-1354(99)80217-8.

S Raghavendra, M., Raj, J., Seetharaman, A., Rudolph Raj, J., Seetharaman, A., 2012. A study of decrease in R&D spending in the pharmaceutical industry during post-recession. volume 4. doi:10.7813/2075-4124.2012/4-5/B.6.

Saure, A., Patrick, J., Tyldesley, S., Puterman, M.L., 2012. Dynamic multi-appointment patient scheduling for radiation therapy. Eur. J. Oper. Res. 223, 573–584.

Skrepnek, G.H., Sarnowski, J.J., 2007. Decision-making associated with drug candidates in the biotechnology research and development (R&D) pipeline. J. Commer. Biotechnol. 13, 99–110. URL: https://doi.org/10.1057/palgrave.jcb.3050040, doi:10.1057/palgrave.jcb.3050040.

Solak, S., Clarke, J.P.B., Johnson, E.L., Barnes, E.R., 2010. Optimization of R&D project portfolios under endogenous uncertainty. Eur. J. Oper. Res. 207, 420–433. URL: http://dx.doi.org/10.1016/j.ejor.2010.04.032, doi:10.1016/j.ejor.2010.04.032.

Subramanian, D., Pekny, J.F., Reklaitis, G.V., 2001. A simulation-optimization framework for research and development pipeline management. AIChE J. 47, 2226–2242.

Subramanian, D., Pekny, J.F., Reklaitis, G.V., Blau, G.E., 2003. Simulation-optimization framework for stochastic optimization of R&D pipeline management. AIChE J. 49, 96–112.

Sun, Y., Li, X., 2013. Response surface optimisation of surgery start times in a single operating room using designed simulation experiments. Int. J. Healthc. Technol. Manag. 14, 61–72.

Varma, V.A., Pekny, J.F., Blau, G.E., Reklaitis, G.V., 2008. A framework for addressing stochastic and combinatorial aspects of scheduling and resource allocation in pharmaceutical R&D pipelines. Comput. Chem. Eng. 32, 1000–1015. doi:10.1016/j.compchemeng.2007.05.006.

Wiesemann, W., Kuhn, D., Rustem, B., 2010. Maximizing the net present value of a project under uncertainty. Eur. J. Oper. Res. 202, 356–367.

Zapata, J.C., Varma, V.A., Reklaitis, G.V., 2008. Impact of tactical and operational policies in the selection of a new product portfolio. Comput. Chem. Eng. 32, 307–319. doi:10.1016/j.compchemeng.2007.03.024.

7. Appendix

Table 7: Data obtained from (Christian and Cremaschi, 2015) for comparison purposes

	Duration			**Probability**			**Trial cost** ($)			**Res. 1 (max=4)**			**Res. 2 (max=3)**			**Revenue ($)**	γ	ρ
	P1	**P2**	**P3**	**P1**	**P2**	**P3**	**P1**	**P2**	**P3**	**P1**	**P2**	**P3**	**P1**	**P2**	**P3**			
1	2	2	6	0.3	0.5	0.8	10	90	220	1	1	2	1	2	3	3100	22	19.2
2	2	4	4	0.4	0.6	0.8	10	80	200	1	2	2	1	1	3	3250	28	19.6
3	2	2	6	0.3	0.6	0.9	10	90	180	1	1	2	1	1	3	3300	26	20
4	2	4	4	0.4	0.6	0.8	10	100	170	1	1	2	1	2	3	3000	24	19.4
5	2	4	6	0.35	0.5	0.9	10	70	210	1	1	2	1	1	3	3150	24	19.6
6	2	4	6	0.45	0.45	0.8	10	85	195	1	2	2	2	1	3	3050	25	19
7	2	2	6	0.45	0.55	0.85	10	95	180	1	1	2	1	2	3	3200	27	19.7
8	2	2	6	0.4	0.6	0.75	20	70	210	1	1	2	1	2	3	3100	22	19.6
9	2	4	4	0.35	0.55	0.8	10	80	195	1	2	2	1	1	3	3200	24	19.4
10	2	2	6	0.25	0.6	0.8	20	80	200	1	1	2	1	1	3	3350	25	19.2

Note: The 20-drug instance is a combination of two 10-drug instance data together.

Table 8: Hypothetical data used for simulation experiments (standard portfolio)

	Duration			Probability			Trial cost			Resource					
Drugs	**P1**	**P2**	**P3**	**P1**	**P2**	**P3**	**P1**	**P2**	**P3**	**P1**	**P2**	**P3**	**Revenue**	γ	ρ
1	2	2	4	0.62	0.3	0.6	10	100	220	1	1	2	2870	22	19.2
2	1	2	3	0.7	0.3	0.7	10	50	150	0.5	1	2	1200	28	19.6
3	2	4	6	0.6	0.3	0.5	10	130	270	1	2	3	3800	26	20
4	2	4	4	0.6	0.3	0.5	10	70	130	1	2	3	1600	24	19.4
5	2	2	6	0.7	0.3	0.7	10	100	200	1	2	3	2860	24	19.6
6	2	2	4	0.65	0.28	0.5	10	100	220	1	1	2	2870	25	19
7	1	2	3	0.6	0.32	0.6	10	80	200	1	2	2	2500	27	19.7
8	2	2	4	0.6	0.31	0.7	10	70	210	1	1	2	1500	22	19.6
9	1	2	3	0.62	0.32	0.5	10	70	210	1	2	3	2720	24	19.4
10	2	4	6	0.7	0.35	0.5	10	100	220	1	2	3	2000	25	19.2
11	2	4	4	0.6	0.3	0.6	10	50	150	1	1	2	2870	22	19.2
12	2	2	6	0.6	0.3	0.7	10	130	270	1	2	2	2500	28	19.2
13	2	2	4	0.7	0.3	0.5	10	70	210	1	1	2	2000	26	20
14	1	2	3	0.65	0.3	0.5	10	100	220	1	2	3	2720	24	19.4
15	2	2	4	0.6	0.3	0.6	10	50	150	1	2	3	2860	24	19.6
16	1	2	3	0.6	0.3	0.6	10	70	180	1	2	2	2700	25	19
17	2	4	6	0.6	0.3	0.6	10	130	270	1	2	2	2780	27	19.7
18	2	4	4	0.65	0.3	0.6	10	100	270	1	2.5	5	8500	22	19.6
19	2	2	6	0.65	0.3	0.6	10	100	170	1	2.5	5	8500	24	19.4
20	2	2	4	0.65	0.3	0.6	10	70	210	1	2.5	5	8500	25	19.2