Journal Pre-proof

Dynamic pharmaceutical product portfolio management with flexible resource profiles

Xin Fei, Jürgen Branke, Nalân Gülpınar

PII: S0377-2217(25)00036-0
DOI: https://doi.org/10.1016/j.ejor.2025.01.011
Reference: EOR 19373

To appear in: *European Journal of Operational Research*

Received date : 15 December 2023
Accepted date : 12 January 2025

Please cite this article as: X. Fei, J. Branke and N. Gülpınar, Dynamic pharmaceutical product portfolio management with flexible resource profiles. *European Journal of Operational Research* (2025), doi: https://doi.org/10.1016/j.ejor.2025.01.011.

Highlights

Dynamic Pharmaceutical Product Portfolio Management with Flexible Resource Profiles

Xin Fei, Jürgen Branke, Nalân Gülpınar

- Formulated the pharmaceutical portfolio management problem as a Markov decision process.
- Determined resource allocation and trial scheduling under uncertain trial outcomes.
- Proposed Monte Carlo tree search and statistical racing approach.
- Achieved policy quality and computational efficiency over the existing methods.

Dynamic Pharmaceutical Product Portfolio Management with Flexible Resource Profiles

Xin Fei[a,*], Jürgen Branke[b], Nalân Gülpınar[b,c]

[a]*Business School, The University of Edinburgh, Edinburgh, EH8 9JS, The United Kingdom*
[b]*Warwick Business School, The University of Warwick, Coventry, CV4 7AL, The United Kingdom*
[c]*Durham University Business School, Durham University, Durham, DH1 3LB, The United Kingdom*

Abstract

The pharmaceutical industry faces growing pressure to develop innovative, affordable products faster. Completing clinical trials on time is crucial, as revenue strongly depends on the finite patent protection. In this paper, we consider dynamic resource allocation for pharmaceutical product portfolio management and clinical trial scheduling, proposing a modelling framework, where resource profiles for ongoing clinical trials are flexible, with the possibility to add additional resources, thereby accelerating the completion of a clinical trial and enhancing pipeline profitability. Specifically, we treat both resource profiles and clinical trial scheduling as decision variables in the management of multiple pharmaceutical products to maximise the expected discounted profit, accounting for uncertainty in clinical trial outcomes. We formulate this problem as a Markov decision process and design a Monte Carlo tree search approach that can identify the best decision for each state by utilising a base policy to estimate value functions. We significantly improve the algorithm efficiency by proposing a statistical racing procedure using correlated sampling (common random numbers) and Bernstein's inequality. We demonstrate the effectiveness of the proposed approach on a pharmaceutical drug development pipeline problem, finding that the proposed modelling framework with flexible resource profiles improves the resource efficiency and profitability, and the proposed Monte Carlo tree search algorithm outperforms existing approaches in terms of efficiency and solution quality.

Keywords: Dynamic programming, Product scheduling, Flexibility, Sampling

1. Introduction

The pharmaceutical sector faces growing challenges in sustaining innovation. Research and development (R&D) expenses by medium-to-large pharmaceutical firms have risen by an average of 10% annually over the past decade (Forman et al., 2021). However, returns on investment have declined, dropping from 6.4% to 2.7% between 2014 and 2020 (Colin and Neil, 2022). Factors include a focus on rare diseases, the shift to precision medicine, advanced biotech methods, longer development timelines, and higher failure rates. Some firms have raised drug prices to offset costs, but excessive prices risk making treatments unaffordable, particularly for those without comprehensive insurance coverage or access to public healthcare. A key

*Corresponding author
Email address: xfei@ed.ac.uk (Xin Fei)

reason for rising R&D costs and declining returns is the lengthy, expensive, and risky clinical trial process required for pharmaceutical product development. This involves several stages, including discovery, preclinical trials, clinical trials, and market approval, as shown in Figure 1. These stages exhibit finish-to-start dependencies. Among them, clinical trials are particularly challenging, requiring substantial time and resources while determining product safety and efficacy. A clinical trial consists of three phases that a product must complete before regulatory submission. Phase I tests the product on healthy volunteers to assess adverse effects and pharmacokinetics. Phases II and III evaluate optimal dosage, benefits, risks, and comparative effectiveness on patients. Failure in any phase removes the product from the pipeline. After Phase III, regulatory authorities decide whether to grant market exclusivity. This period is critical for pharmaceutical companies to recover costs and earn revenue, as it allows them to sell products without generic competition. However, patent expiration eventually allows generics, causing a substantial drop in price and market share, a phenomenon known as the 'patent cliff' that threatens profitability. Consequently, companies aim to speed up development to maximise revenue. For products addressing unmet medical needs or showing great promise, regulatory agencies may allow phase-skipping or concurrent Phases II and III. This study considers a clinical trial setting with pre-determined Phase I, II, and III designs and patient numbers. This structured approach is more widely used in practice than adaptive trial designs, allowing us to focus on strategic resource allocation rather than trial design optimisation.

Managing a single pharmaceutical product involves complex decision-making, and managing a portfolio intensifies these challenges. For medium-to-large pharmaceutical companies, R&D pipelines comprise multiple products, requiring intricate scheduling and careful allocation of resources (e.g., research sites, staff, and nurses). Pharmaceutical product portfolio management can be regarded as a type of R&D product scheduling problem. Prescriptive analytics provides potential solutions by identifying (near-)optimal management policies. While past studies assumed resource profiles of clinical trial phases to remain constant over time, our study allows adding further resources to speed up the completion of any phase of a clinical trial, which often occurs in practice in the form of additional test sites. By optimising the resource profile, decision-makers can distribute resources in a more targeted way to avoid bottlenecks and ensure efficient scheduling. We formulate the pharmaceutical product portfolio management as a discrete Markov decision process and develop an efficient Monte Carlo tree search (MCTS) approach to identify a (near-)optimal decision for each state. The approach uses a base policy with scenario modelling and mathematical programming to estimate state-action

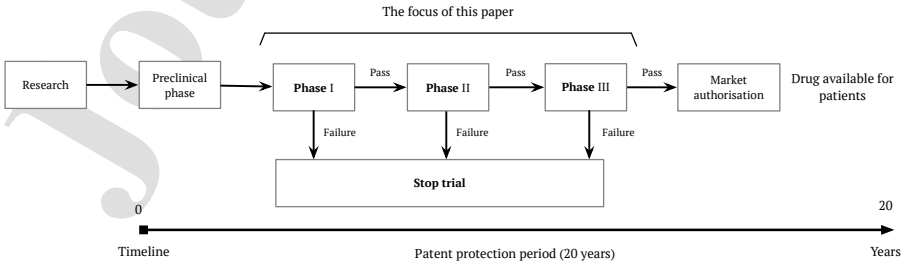


Figure 1: The development stages of a single pharmaceutical product.

value functions, quantifying the expected cumulative reward of each decision. The proposed base policy satisfies sequential consistency and improvement, guaranteeing policy improvement. A statistical racing procedure is then used to efficiently compare and select the best decision, utilising Bernstein's inequality and correlated sampling to quantify decision performance and reduce variance when assessing differences between potential decisions. Efficiency comes from iteratively eliminating inferior decisions during simulation and focusing on promising ones. We show that the procedure outperforms other sampling approaches and reliably identifies the best.

This paper is structured as follows: Section 2 reviews prior studies. Section 3 formulates the pharmaceutical product portfolio management as a Markov decision process, highlighting its complexity and the motivation for approximate dynamic programming. Section 4 details the base policy, Q-function estimation, and MCTS implementation. Section 5 presents our novel contribution through accelerating simulations using statistical racing and correlated sampling to enhance MCTS efficiency. Section 6 presents the numerical experiments and results, demonstrating the efficacy of the proposed approach. Finally, Section 7 summarises our findings, discusses implications, and outlines future research directions.

2. Literature Review

Table 1 offers an overview of prior studies in pharmaceutical product portfolio management. The early studies such as Schmidt and Grossmann (1996) and Jain and Grossmann (1999) primarily focused on the offline task scheduling problem under uncertainties in both activity duration and clinical trial outcomes for completing trials for a single product. Their objective was to maximise the expected discounted profit, which is negatively correlated with market exclusivity duration. The authors modelled the uncertain parameters using pre-specified distributions, generated sample sets from these distributions, and approximated the stochastic programming formulation with a deterministic counterpart solvable through cutting plane methods. De Reyck and Leus (2008) studied a similar problem through the lens of probabilistic graphical models. While these early studies provided valuable insights, the field has evolved to address more complex challenges. Over time, the emphasis in literature has shifted to optimising entire pharmaceutical product portfolios. Rogers et al. (2002) and Gupta and Maranas (2004) drew parallels between investing in real options and pharmaceutical products. They formulated the product portfolio management problem using the discrete-time Black-Scholes model to maximise market value and to determine optimal investment policies. Another approach in the literature is the multi-stage stochastic programming. This method delineates decisions over time under a scenario tree to maximise the expected discounted profit of a product portfolio. A limitation is that the scenario tree must be generated prior to modelling, posing challenges due to decision-dependent trial timelines and uncertainties. While lacking stability and optimality guarantees, various tree generation strategies have been proposed, with models solved through branch-and-bound (Colvin and Maravelias, 2008; Apap and Grossmann, 2017) or heuristic methods (Verderame et al., 2010; Christian and Cremaschi, 2015). Moreover, Markov decision processes have been used to formulate the product portfolio management problem without requiring scenario tree generation. However, due to the curse of dimensionality, exact algorithms are generally not applicable. To address this, Choi et al. (2004) proposed rollout

Table 1: Approaches for modelling and resolving pharmaceutical portfolio management.

References	Modelling Framework	Solution Approach	Resource Profile
Schmidt and Grossmann (1996); Jain and Grossmann (1999)	Two-stage stochastic programming	Scenario generation and cutting plane	Fixed
De Reyck and Leus (2008)	Probabilistic model	Branch-and-bound	Fixed
Rogers et al. (2002); Gupta and Maranas (2004)	Black-Scholes model	Branch-and-bound	Fixed
Colvin and Maravelias (2008); Apap and Grossmann (2017)	Multi-stage stochastic programming	Scenario tree generation and branch-and-bound	Fixed
Verderame et al. (2010); Christian and Cremaschi (2015)	Multi-stage stochastic programming	Scenario tree generation and heuristics	Fixed
Choi et al. (2004)	Markov decision process	State reduction	Fixed
Gökalp and Branke (2020)	Markov decision process	Value function approximation	Fixed
This work	Markov decision process	MCTS with statistical racing and correlated sampling	Flexible

algorithms with heuristic-based state reduction to derive (near-)optimal policies. Gökalp and Branke (2020) introduced a double-pass value function approximation algorithm for the same purpose. Our study builds upon the Markov decision process framework but differs from previous work by employing MCTS with an efficient sampling procedure to obtain (near-)optimal policies. More broadly, this work is also related to the stochastic resource-constrained project scheduling problem (SRCPSP), as both involve optimising the scheduling of multiple projects under resource constraints. While similarities exist, SRCPSP studies typically aim to maximise resource efficiency (Krüger and Scholl, 2010; Issa et al., 2021), the total profit (Satic et al., 2024), or minimise makespan or delays over finite time horizons (Pérez et al., 2016; Van Den Eeckhout et al., 2021). The SRCPSP studies often consider uncertainties in aspects such as duration, project arrival, or resource availability (Li and Womer, 2015; Xie et al., 2021; Satic et al., 2024). However, they typically do not account for the success probability of a project, which is crucial in pharmaceutical products. Recent SRCPSP studies (e.g., Naber (2017); Kogan et al. (2024)), allow for dynamic adjustment of resource profiles within pre-specified ranges. To the best of our knowledge, incorporating flexible resource usage has not been considered in pharmaceutical portfolio management, and our work aims to bridge this gap.

SRCPSP problems have been solved using a variety of approaches including dynamic programming, linear programming, constraint programming, and heuristics; for a comprehensive review, see Sánchez et al. (2023). We study the problem of maximising the expected discounted profit of a pharmaceutical product portfolio with flexible resource profiles using the Markov decision process. The main challenge is the curse of dimensionality and high-dimensional states. This makes exact solution methods (e.g., backward induction) intractable for large-scale problems. Therefore, approximate algorithms have been developed to balance optimality and computational efficiency. Powell (2016) provided reviews of approximate dynamic programming approaches. One such approximate approach is the rollout algorithm, which leverages one or multiple base policies to explore the state space and sequentially improve the policy through the policy improvement principle (Goodson et al., 2013). This method was originally proposed

by Bertsekas et al. (1997) for deterministic dynamic programming with numerous states, where (near-)optimal policies were produced by estimating state-value functions from a state to the terminal or a limited number of states using base policies, under sequential improvement and consistency conditions. Secomandi (2001) then extended the rollout approach to stochastic environment. The versatility of the rollout algorithm is evident in its successful application to various SRCPSP variants, as demonstrated in the works of Li and Womer (2015) and Xie et al. (2021). Despite its advantages, a key limitation of the rollout is that for every feasible decision, all state-trajectories must be evaluated, becoming computationally burdensome when the number of decisions and trajectories is large. One such technique is MCTS (also known as simulation-based rollout algorithm) which uses efficient tree policy instead of evaluating all possible state trajectories, to balance the simulation of promising decisions with sufficient exploration (Bertsekas, 2019). For over a decade, MCTS has received significant attention in game playing settings (Silver et al., 2017). More recently, its applications have expanded into operations research gaining popularity (Bertsimas et al., 2017; Świechowski et al., 2023). In this work, we propose an MCTS algorithm to address the pharmaceutical project management problem and develop an efficient sampling procedure to balance exploration and exploitation, to identify the best action at each state with fewer simulations.

In the MCTS literature, a sampling procedure (also known as tree policy) is the strategy used to select which action to explore during the selection phase. The objective of a sampling procedure is to maximise the probability of finding the optimal decision or minimise the expected difference between the optimal and selected decision's performance. The development of sampling procedures has a close connection with multi-armed bandit and ranking and selection studies. For example, Kocsis and Szepesvári (2006) and Silver et al. (2017) implemented an upper confidence bound (UCB) algorithm in MCTS. Under a mild assumption of bounded random variables, UCB uses Hoeffding's inequality to estimate the range of expected cumulative reward for each decision. It balances exploration and exploitation by favouring decisions with high potential reward, as discussed by Krafft and Schmitz (1969). Alternatively, Li et al. (2021) integrated optimal computing budget allocation into the MCTS framework. These sampling approaches assume the prior and posterior distributions of cumulative rewards are well-defined and conjugate. They allocate the simulation budget to decisions based on the expected improvement. In this paper, we integrate MCTS with a statistical racing procedure using correlated sampling and empirical Bernstein's inequality to improve efficiency. Maron and Moore (1997) first proposed a statistical racing procedure that used Hoeffding's inequality to compare model performance and eliminate poor models during simulation. In the context of pharmaceutical product portfolio management, revenues can vary greatly under different scenarios. Revenues may be negative with only development costs in failure cases but very high in success cases. Compared to Hoeffding's inequality, Bernstein's inequality, by incorporating variance, can provide a more accurate bound on estimation errors. Moreover, as discussed by Fu et al. (2007), variance reduction techniques like correlated sampling can significantly reduce the samples required for differentiation. By combining Bernstein's inequality and correlated sampling within MCTS, the pharmaceutical product portfolio management problem can be efficiently resolved as discussed in Section 6.

3. Problem Description and Model Formulation

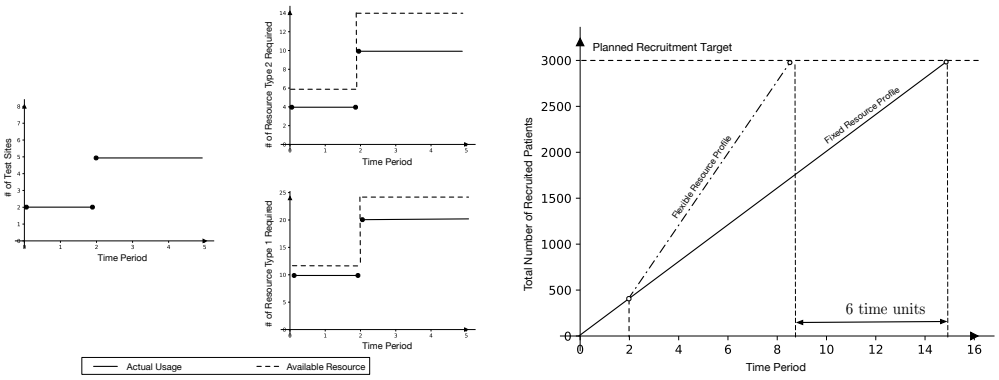
In this section, we introduce a pharmaceutical R&D pipeline management formulation that incorporates flexible resource profiles to better reflect real-world dynamics. We then formulate this problem as a Markov decision process and discuss the computational challenges involved in identifying the optimal policy. Appendix A summarises the notation used in the model formulation, providing a quick reference to facilitate understanding.

3.1. Flexible Resource Profiles in Pharmaceutical R&D Pipeline Management

A pharmaceutical R&D pipeline consists of a set of pharmaceutical products, denoted $\mathcal{I} = \{1, \ldots, I\}$. Before a marketing authorisation application can be submitted, regulations require that each product successfully completes three clinical trial phases, denoted $\mathcal{J} = \{1, 2, 3\}$. Let 2-tuple $(i, j) \in \mathcal{I} \times \mathcal{J}$ represent Phase j of pharmaceutical product i. Conducting any clinical trial phase incurs substantial expenses for activities like patient recruitment and data analysis. Patient recruitment involves identifying and enrolling suitable participants, gathering initial data, and informing participants. Data analysis involves systematically examining, interpreting, and statistically assessing the collected data to derive conclusions about the study outcomes.

We assume the designs of clinical trials in the R&D pipeline are fixed. Let $c_{i,j}^{\text{Recr}}$ and $c_{i,j}^{\text{Data}}$ represent the costs to complete patient recruitment and data analysis, respectively. We assume that the total time taken to complete any phase of a clinical trial is the sum of time spent on recruitment and data analysis. Efficient resource allocations can reduce recruitment time, which depends on the number of active test sites and the estimated recruitment rate per site. The patient recruitment activity for (i, j) can be executed at a variable number of test sites. The number of test sites ranges from a minimum of $h_{i,j}^{\text{Min}}$ to a maximum of $h_{i,j}^{\text{Max}}$. The total target for patient recruitment is denoted by $q_{i,j}^{\text{Target}}$, with an expected average recruitment rate per site of $\rho_{i,j}^{\text{Site}}$. Given that the targeted number of patients is predetermined, the duration for data analysis activities related to (i, j) remains constant, represented by $\lambda_{i,j}$. One of the primary challenges in R&D pipeline development involves balancing recruitment speed against resource consumption, especially considering the competition for limited resources among multiple pharmaceutical projects. This process requires various types of resources, denoted by the set $\mathcal{K} = 1, 2, \ldots, K$. The resource requirements in pharmaceutical R&D can be broadly categorised into two main components: data analysis and patient recruitment. For (i, j), the data analysis activity requires an amount $r_{i,j,k}^{\text{Data}}$ of resource type $k \in \mathcal{K}$. Regarding patient recruitment, Kaitin and DiMasi (2011) found that increasing the number of operational test sites can accelerate patient recruitment, as wider geographical reach accesses a larger patient population. The resource requirement for patient recruitment is modelled as directly proportional to the number of test sites. Let $\beta_{i,j,k}$ denote the resource allocation coefficient, representing the amount of resource k required per test site for (i, j). In practice, this coefficient can be estimated using a combination of data analysis, historical information, and expert judgement.

Figures 2(a) and 2(b) show how resource flexibility affects patient recruitment in a clinical trial with two resource types. With fixed resources (10 and 4 units) over two periods, two test sites recruit 100 volunteers per period, taking 15 time units to reach 3,000 patients. When



(a) Resource profile, showing actual resource usage versus availability across time periods.

(b) Reduction in time required to reach recruitment targets due to resource profile adjustments.

Figure 2: Impact of resource profile adjustments on recruitment efficiency.

resources become adjustable after period two, enabling five test sites through reallocation, recruitment time decreases by 6 units, demonstrating improved efficiency through flexibility.

3.2. Decision Variables

During the planning horizon, defined as a set of discrete time steps $\mathcal{T} = \{1, 2, \ldots, T\}$, a sequence of scheduling and resource allocation decisions is required. At each epoch $t \in \mathcal{T}$, the company needs to make decisions regarding the initiation of patient recruitment, the allocation of test sites, and the commencement of data analysis. The scheduling decision, represented by the binary variable $x^{\text{Recr}}_{i,j,t}$, indicates whether patient recruitment for (i, j) starts at time t with the minimum number of test sites. Concurrently, the resource allocation decision, denoted by the integer variable $x^{\text{Site}}_{i,j,t}$, specifies the number of additional test sites allocated to (i, j). Additionally, the binary variable $x^{\text{Data}}_{i,j,t}$ represents the scheduling decision for starting data analysis for (i, j) at time t. We assume that the number of test sites for recruitment can only increase, and a clinical trial phase can begin only after the previous phase is completed. Importantly, a trial, once begun, must be brought to completion. If a drug product successfully passes phase III, it becomes eligible to apply for market authorisation. The decisions made at each epoch t are collectively denoted by the triple $X_t = (\mathbf{x}^{\text{Recr}}_t, \mathbf{x}^{\text{Site}}_t, \mathbf{x}^{\text{Data}}_t)$ where $\mathbf{x}^{\text{Recr}}_t$, $\mathbf{x}^{\text{Site}}_t$, and $\mathbf{x}^{\text{Data}}_t$ are aggregates of decisions $x^{\text{Recr}}_{i,j,t}$, $x^{\text{Site}}_{i,j,t}$ and $x^{\text{Data}}_{i,j,t}$, respectively.

3.3. State Variables

At epoch t, the R&D pipeline status is defined by the availability of various resource types and the progress status of different tasks. Specifically, $L^{\text{Recr}}_{i,j,t}$ represents the number of patients still required for trial phase (i, j) at epoch t, while $L^{\text{Data}}_{i,j,t}$ denotes the remaining duration for data analysis for phase (i, j). Additionally, $A^{\text{Recr}}_{i,j,t}$ indicates whether patient recruitment for (i, j) can be scheduled at epoch t, and $A^{\text{Data}}_{i,j,t}$ indicates whether data analysis for (i, j) can be scheduled at epoch t. $R^{\text{Site}}_{i,j,t}$ denotes the number of testing sites assigned to patient recruitment for (i, j). The total amount of resource type k available at epoch t is represented by $R_{k,t}$. Thus, the state of the R&D pipeline at epoch t is captured by the following 6-tuple:

$$\mathcal{S}_t = \left(\mathbf{L}^{\text{Recr}}_t, \mathbf{L}^{\text{Data}}_t, \mathbf{A}^{\text{Recr}}_t, \mathbf{A}^{\text{Data}}_t, \mathbf{R}^{\text{Site}}_t, \mathbf{R}_t\right),$$

where $\mathbf{L}_t^{\text{Recr}}$, $\mathbf{L}_t^{\text{Data}}$, $\mathbf{A}_t^{\text{Recr}}$, $\mathbf{A}_t^{\text{Data}}$, $\mathbf{R}_t^{\text{Site}}$ and $\mathbf{R}_t$ aggregate these state variables across all tasks. At epoch $t = 1$, Phase I for all drug products can be scheduled, such that $A_{i,1,1}^{\text{Recr}} = 1$ for all drug products $i \in \mathcal{I}$. The number of patients still required, $L_{i,j,1}^{\text{Recr}}$, for all (i, j) is initialised to $q_{i,j}^{\text{Target}}$, while the remaining time for completing data analysis, $L_{i,j,1}^{\text{Data}}$, for all (i, j) is set to $\lambda_{i,j}$. These values define the initial condition of the R&D pipeline.

3.4. Exogenous Information

The outcome of clinical trial phase (i, j) is modelled as a Bernoulli random variable $\Omega_{i,j}$, where the probability of success is $p_{i,j}$. Let $\omega_{i,j}$ represent an observed outcome of $\Omega_{i,j}$. The probability mass function for the outcome of a clinical trial phase is expressed as follows:

$$P(\Omega_{i,j} = \omega_{i,j}) = \begin{cases} p_{i,j}, & \text{if } \omega_{i,j} = 1 \quad \text{(successful)} \\ 1 - p_{i,j}, & \text{if } \omega_{i,j} = 0 \quad \text{(failure)}. \end{cases}$$

In practice, the success probability of a clinical trial phase can be estimated by analysing historical data from previous similar trials and early phase results. The set of trial outcomes at decision epoch t, denoted as W_t, is an exogenous observation revealed once data analysis is completed by the end of that epoch, $W_t = \left\{\omega_{i,j} \mid \quad L_{i,j,t}^{\text{Data}} = 1\right\}$.

3.5. Feasible Region

Given state $\mathcal{S}_t$, the feasible region $\mathcal{X}_t$ is defined by the following Constraints (1a) - (1e).

$$x_{i,j,t}^{\text{Recr}} \leq A_{i,j,t}^{\text{Recr}}, \quad \forall (i, j) \tag{1a}$$

$$x_{i,j,t}^{\text{Site}} \leq \begin{cases} x_{i,j,t}^{\text{Recr}}(h_{i,j}^{\text{Max}} - h_{i,j}^{\text{Min}}), & \forall (i, j) \in \mid A_{i,j,t}^{\text{Recr}} = 1 \\ h_{i,j}^{\text{Max}} - R_{i,j,t}^{\text{Site}}, & \forall (i, j) \in \mid 0 < L_{i,j,t}^{\text{Recr}} < q_{i,j}^{\text{Target}} \\ 0, & \text{otherwise} \end{cases} \tag{1b}$$

$$x_{i,j,t}^{\text{Data}} \leq A_{i,j,t}^{\text{Data}}, \quad \forall (i, j) \tag{1c}$$

$$R_{k,t} \geq \sum_{(i,j)} \left(\beta_{i,j,k}(x_{i,j,t}^{\text{Site}} + x_{i,j,t}^{\text{Recr}} h_{i,j}^{\text{Min}}) + r_{i,j,k} x_{i,j,t}^{\text{Data}}\right), \quad \forall k \tag{1d}$$

$$x_{i,j,t}^{\text{Recr}} \in \{0, 1\}, \quad x_{i,j,t}^{\text{Site}} \in \mathbb{Z}, \quad x_{i,j,t}^{\text{Data}} \in \{0, 1\}, \quad \forall (i, j). \tag{1e}$$

Constraint (1a) establishes that recruitment scheduling is restricted to eligible phases, ensuring proper sequencing of activities. Constraint (1b) governs the allocation of additional test sites, limiting the number based on both the maximum allowable sites and the satisfaction of recruitment prerequisites. Constraint (1c) enforces the sequential nature of clinical trials by permitting data analysis to commence only after patient recruitment has reached the required number. Constraint (1d) maintains operational feasibility by ensuring that the total resource consumption remains within available capacity at each time period.

3.6. Transition Function

The state transition function describes how the R&D pipeline evolves from one state to another. Let $f(\cdot)$ represent the transition function that generates the next state $\mathcal{S}_{t+1}$ based on the current state $\mathcal{S}_t$, decision X_t, and exogenous information W_t. This evolution of the system state is formally described by

$$f(\mathcal{S}_t, X_t, W_t) = \left(\mathbf{L}_{t+1}^{\text{Recr}}, \mathbf{L}_{t+1}^{\text{Data}}, \mathbf{A}_{t+1}^{\text{Recr}}, \mathbf{A}_{t+1}^{\text{Data}}, \mathbf{R}_{t+1}^{\text{Site}}, \mathbf{R}_{t+1}\right) \tag{2}$$

where

$$L^{\text{Recr}}_{i,j,t+1} = \max\left\{0, L^{\text{Recr}}_{i,j,t} - \rho^{\text{Site}}_{i,j}\left(h^{\text{Min}}_{i,j} x^{\text{Recr}}_{i,j,t} + x^{\text{Site}}_{i,j,t} + R^{\text{Site}}_{i,j,t}\right)\right\}, \quad \forall (i,j) \tag{3}$$

$$L^{\text{Data}}_{i,j,t+1} = \begin{cases} \max\left\{0, L^{\text{Data}}_{i,j,t} - 1\right\}, \forall (i,j) \mid 0 < L^{\text{Data}}_{i,j,t} < \lambda_{i,j}, \\ \max\left\{0, L^{\text{Data}}_{i,j,t} - x^{\text{Data}}_{i,j,t}\right\}, \quad \text{otherwise.} \end{cases} \tag{4}$$

$$A^{\text{Recr}}_{i,j,t+1} = \begin{cases} \omega_{i,j-1}, & \forall (i,j) \mid \omega_{i,j} \in W_t,\ j \in \mathcal{J} \setminus \{1\}, \\ 0, & \forall (i,j) \mid x^{\text{Recr}}_{i,j,t} = 1, \\ A^{\text{Recr}}_{i,j,t}, & \text{otherwise.} \end{cases} \tag{5}$$

$$A^{\text{Data}}_{i,j,t+1} = \begin{cases} 1, & \forall (i,j) \mid L^{\text{Recr}}_{i,j,t} > 0,\ L^{\text{Recr}}_{i,j,t+1} = 0, \\ 0, & \forall (i,j) \mid x^{\text{Data}}_{i,j,t} = 1, \\ A^{\text{Data}}_{i,j,t}, & \text{otherwise.} \end{cases} \tag{6}$$

$$R^{\text{Site}}_{i,j,t+1} = \begin{cases} h^{\text{Min}} x^{\text{Recr}}_{i,j,t} + x^{\text{Site}}_{i,j,t} + R^{\text{Site}}_{i,j,t}, & \forall (i,j) \mid 0 < L^{\text{Recr}}_{i,j,t} \le q^{\text{Target}}_{i,j}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

$$\begin{aligned} R_{k,t+1} = R_{k,t} - \sum_{(i,j)} \left(\beta_{i,j,k}\left(h^{\text{Min}}_{i,j} x^{\text{Recr}}_{i,j,t} + x^{\text{Site}}_{i,j,t}\right) + r_{i,j,k} x^{\text{Data}}_{i,j,t}\right) + \sum_{(i,j) \mid L^{\text{Data}}_{i,j,t} > 0,\ L^{\text{Data}}_{i,j,t+1} = 0} r_{i,j,k} \\ + \sum_{(i,j) \mid L^{\text{Recr}}_{i,j,t} > 0,\ L^{\text{Recr}}_{i,j,t+1} = 0} \beta_{i,j,k}\left(h^{\text{Min}}_{i,j} x^{\text{Recr}}_{i,j,t} + x^{\text{Site}}_{i,j,t} + R^{\text{Site}}_{i,j,t}\right), \quad \forall k. \end{aligned} \tag{8}$$

Equation (3) describes the remaining number of patients to recruit, which is updated based on the number of test sites multiplied by the recruitment rate when recruitment is scheduled or additional sites are assigned. Equation (4) defines the evolution of the remaining analysis duration, which decreases by one unit per epoch once analysis commences. Equation (5) specifies conditions for scheduling patient recruitment, which can proceed only after a successful outcome in the previous phase. Recruitment, once started, sets the corresponding state values to 0 to prevent repetition, while unscheduled states remain unchanged. Equation (6) describes the transition for data analysis, which starts only after recruitment completion. States are set to 0 if analysis begins or remain unchanged otherwise. Equation (7) updates test site allocations, resetting corresponding state values to 0 after recruitment completion. Equation (8) models resource usage, subtracting consumption from available resources and returning resources upon phase completion.

3.7. Bellman Equation

In the pharmaceutical industry, drug pricing transitions from monopolistic pricing during patent protection to competitive pricing upon patent expiration. This shift, characterised by the entry of lower-priced generic drugs, requires careful consideration of the post-patent revenue decline when calculating discounted profits. Our model captures this concept, with Γ_i representing the maximum projected revenue for product i during its patent life. This value is influenced by factors such as the size of the patient population and the availability of alternative treatments. In contrast, γ_i denotes the periodic revenue loss associated with a reduced patent life. We define the immediate reward after taking X_t at $\mathcal{S}_t$, given the realised exogenous

information W_t, as

$$u_t\left(\mathcal{S}_t, X_t, W_t\right) = \sum_{i:\omega_{i,3}\in W_t} \omega_{i,3}\big(\Gamma_i - \gamma_i(t+1)\big) - \sum_{(i,j)} c_{i,j}^{\text{Recr}} x_{i,j,t}^{\text{Recr}} - \sum_{(i,j)} c_{i,j}^{\text{Data}} x_{i,j,t}^{\text{Data}}. \tag{9}$$

The objective is to identify the optimal policy that generates the maximum expected discounted profit from the initial to final time step. The Bellman equation is given by:

$$V_t(\mathcal{S}_t) = \max_{X_t\in\mathcal{X}_t} \Big\{ u_t\left(\mathcal{S}_t, X_t, W_t\right) + \mathbb{E}\left\{V_{t+1}(\mathcal{S}_{t+1}) \mid \mathcal{S}_{t+1} = f(\mathcal{S}_t, X_t, W_t)\right\} \Big\}. \tag{10}$$

The optimal decision that maximises the expected reward can be obtained by solving the Bellman equation recursively. However, this equation presents computational challenges as it lacks a compact form, requiring recursive relationships between value functions across time steps. In our context, the state space complexity grows exponentially with the number of products and the length of planning horizon. This exponential growth renders backward induction intractable for large-scale problems. To address this computational complexity, we employ an approximate dynamic programming method known as Monte Carlo Tree Search (MCTS) to find (near-)optimal decisions.

4. Monte Carlo Tree Search

MCTS is a simulation-based rollout that is useful when dealing with large state space or when the optimal policy is challenging to compute. By leveraging the principle of 'policy iteration truncated to one step,' the rollout often yields (near-)optimal actions while maintaining reasonable computational effort (Bertsekas et al., 1997). At the core of this approach is the Q-function, which estimates the long-term value of taking a given action in a particular state. Specifically, Q-function $Q(S_t, X_t)$ calculates the expected discounted reward after taking X_t in $\mathcal{S}_t$ and then following the optimal policy thereafter. The Q-function is defined as:

$$Q(\mathcal{S}_t, X_t) = \mathbb{E}\Big\{ u_t\left(\mathcal{S}_t, X_t, W_t\right) + V_{t+1}(\mathcal{S}_{t+1}) \mid \mathcal{S}_{t+1} = f(\mathcal{S}_t, X_t, W_t) \Big\}. \tag{11}$$

Computing the Q-function poses the same level of complexity as solving (10). This is because it requires knowledge of the optimal policy to determine the future value $V_{t+1}(\cdot)$. The rollout algorithm addresses this challenge by using a base policy to guide state-action exploration. Future values are estimated by simulating the base policy rather than the optimal one.

4.1. Base Policy

The base policy plays a pivotal role in the rollout. It is a function, denoted as $\Pi(\cdot)$, that maps state $\mathcal{S}_\tau$ to decision X_τ for $\tau = t+1, \ldots, T-1$: $\Pi(\mathcal{S}_\tau) : \mathcal{S}_\tau \to X_\tau, \quad \forall \tau \in \{t+1, \ldots, T-1\}$. According to the studies by Bertsekas et al. (1997), a base policy could take various forms such as a heuristic rule, a mathematical program, or a search method.

Definition 1. Sequential Consistency: A base policy is sequentially consistent if it produces a state-trajectory $\left\{\mathcal{S}_t, \mathcal{S}_{t+1}, \mathcal{S}_{t+2}, \ldots, \mathcal{S}_T\right\}$ from state $\mathcal{S}_t$, and also generates the same subsequent state-trajectory $\left\{\mathcal{S}_{t+1}, \mathcal{S}_{t+2}, \ldots, \mathcal{S}_T\right\}$ from state $\mathcal{S}_{t+1}$.

The sequential consistency ensures that the rollout algorithm, which simulates multiple potential outcomes based on the base policy, yields value estimates that are optimal or at least as good as directly applying the base policy.

This study introduces a base policy, developed through mathematical programming and scenario modelling, to effectively navigate potential future states. Let us consider estimating the future value $V_{t+1}(\mathcal{S}_{t+1})$ which requires decisions from the base policy across states $\mathcal{S}_\tau, \tau = t+1, t+2, \ldots, T$. Modelling R&D pipeline dynamics with mathematical programming involves estimating the conditional success probability for each product in future states. The conditional success probability for product i in state $\mathcal{S}_\tau$ is denoted by $P_i(w_{i,3} = 1 \mid \mathcal{S}_\tau)$. For a pharmaceutical product i, failure results in the cessation of its development, thereby resulting in its conditional success probability being $P_i(w_{i,3} = 1 \mid \mathcal{S}_\tau) = 0$. Conversely, if the product has successfully passed j-th phase in state $\mathcal{S}_\tau$, its conditional probability can be computed as

$$P_i(w_{i,3} = 1 \mid \mathcal{S}_\tau) = \prod_{j':j'>j} p_{i,j'}, \quad \forall j.$$

Our base policy determines the duration required to complete the remaining developmental activities for clinical trial phases, accounting for factors such as scheduling and resource allocation. To simplify this while ensuring that base policy decisions maintain the feasibility of state $\mathcal{S}_\tau$, we relax the resource capacity constraints for states $\mathcal{S}_{\tau'}, \tau' \geq \tau + 2$. To model the system dynamics, we introduce auxiliary decision variables $\mathbf{x}^{\text{Base}} = \{x^{\text{Base}} \mid i \in \mathcal{I}, j \in \mathcal{J}\}$ representing the duration to complete remaining clinical trial phases. Let $M^{\text{Recr}}(\mathcal{S}_\tau)$ denote the set of phases where patient recruitment is either ready or underway and has not failed in state $\mathcal{S}_\tau$:

$$M^{\text{Recr}}(\mathcal{S}_\tau) = \{(i,j) \mid 0 < L^{\text{Recr}}_{i,j,\tau} \leq q^{\text{Target}}_{i,j}\}.$$

$M^{\text{Data}}(\mathcal{S}_\tau)$, on the other hand, represents the set of trials where data analysis is in progress and has not encountered any failures:

$$M^{\text{Data}}(\mathcal{S}_\tau) = \{(i,j) \mid 0 < L^{\text{Data}}_{i,j,\tau} \leq \lambda_{i,j}\}.$$

Lastly, $M^{\text{Future}}(\mathcal{S}_\tau)$ denotes future trials after those in $M^{\text{Recr}}(\mathcal{S}_\tau)$ and $M^{\text{Data}}(\mathcal{S}_\tau)$ are completed:

$$M^{\text{Future}}(\mathcal{S}_\tau) = \Big\{(i,j) \mid i = i',\, j > j',\, \Big(0 < L^{\text{Recr}}_{i',j',\tau} \leq q^{\text{Target}}_{i',j'} \text{ or } 0 < L^{\text{Data}}_{i',j',\tau} \leq \lambda_{i',j'}\Big)\Big\}.$$

The base policy $\Pi(\mathcal{S}_\tau)$ solves the following mathematical program and is iteratively applied along a sampled path ($\tau = t+1, \ldots, T$) to inform decisions $\mathbf{x}^{\text{Recr}}_\tau, \mathbf{x}^{\text{Site}}_\tau$, and $\mathbf{x}^{\text{Data}}_\tau$:

$$\max_{\substack{\mathbf{x}^{\text{Base}}, \mathbf{x}^{\text{Recr}}_\tau \\ \mathbf{x}^{\text{Site}}_\tau, \mathbf{x}^{\text{Data}}_\tau}} \sum_{i \in \mathcal{I}} P_i(w_{i,3} = 1 \mid \mathcal{S}_\tau)\Big(\Gamma_i - \gamma_i\big(1 + \sum_{(i,j)} x^{\text{Base}}_{i,j}\big) - \sum_{(i,j)} \big(c^{Recr}_{i,j} + c^{Data}_{i,j}\big)\Big) \tag{12a}$$

$$s.t.\;\; x^{\text{Base}}_{i,j} \geq \frac{L^{\text{Recr}}_{i,j,t+1} - \big(h^{\text{Min}}_{i,j} x^{\text{Recr}}_{i,j,\tau} + x^{\text{Site}}_{i,j,\tau} + R^{\text{Site}}_{i,j,\tau}\big)\rho^{\text{Site}}_{i,j}}{\rho^{\text{Site}}_{i,j}} + \lambda_{i,j}, \quad \forall (i,j) \in M^{\text{Recr}}(\mathcal{S}_\tau) \tag{12b}$$

$$x^{\text{Base}}_{i,j} \geq L^{\text{Data}}_{i,j} + (1 - x^{\text{Data}}_{i,j,\tau}) A^{\text{Data}}_{i,j,\tau}, \quad \forall (i,j) \in M^{\text{Data}}(\mathcal{S}_\tau) \tag{12c}$$

$$x^{\text{Base}}_{i,j} \geq \frac{q^{\text{Target}}_{i,j}}{\rho^{\text{Site}}_{i,j} h^{\text{Max}}_{i,j}} + \lambda_{i,j}, \quad \forall (i,j) \in M^{\text{Future}}(\mathcal{S}_\tau) \tag{12d}$$

$$x^{\text{Base}}_{i,j} \geq 0, \quad \text{constraints (1a) - (1e) for } x^{\text{Recr}}_{i,j,\tau}, x^{\text{Site}}_{i,j,\tau}, x^{\text{Data}}_{i,j,\tau}, \quad \forall (i,j) \in \mathcal{I} \times \mathcal{J}. \tag{12e}$$

The objective function in (12a) calculates the total expected discounted profit across all phar-

maceutical products using the base policy decisions. Constraints (12b) and (12c) describe the completion time for trials where patient recruitment or data analysis is ready to start or already underway in state $\mathcal{S}_\tau$. Constraint (12d) relates the completion time for future trials. Constraint (12e) ensures that the base policy's scheduling and resource allocation decisions remain operationally feasible within the constraints of state $\mathcal{S}_\tau$.

Proposition 1. *The base policy in* (12a) - (12e) *satisfies the property of sequential consistency.*

Proof. The proof is provided in Appendix B. □

4.2. Q-Function Estimation

The base policy, as detailed in (12a) - (12e), estimates the Q-function value for a feasible action taken at a specific state by guiding decisions in future states and evaluating cumulative rewards along sampled trajectories. One such trajectory can be written as:

$$\Big\{(\mathcal{S}_t, \mathcal{S}_{t+1}, \ldots, \mathcal{S}_T) \mid \mathcal{S}_{t+1} \in f(\mathcal{S}_t, X_t, W_t), \mathcal{S}_{\tau+1} \in f(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau), W_\tau) \text{ for } \tau = t+1, \ldots, T-1\Big\}.$$

The trajectory often is not uniqueand varies with different trial outcomes, resulting in numerous possible realisations. By enumerating all trajectories and computing their cumulative rewards, we can estimate the future value $V_{t+1}(\mathcal{S}_{t+1})$ as

$$V_{t+1}(\mathcal{S}_{t+1}) \geq \mathbb{E}\Big\{\sum_{\tau=t+1}^{T} u_\tau\big(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau, W_\tau)\big)\Big\}, \tag{13}$$

where $\mathcal{S}_{t+1} = f(\mathcal{S}_t, X_t, W_t)$ and for $\tau > t+1$, $\mathcal{S}_\tau = f(\mathcal{S}_{\tau-1}, \Pi(\mathcal{S}_{\tau-1}), W_{\tau-1})$. Thus, the rollout algorithm identifies the best decision from a set of feasible options $\mathcal{X}_t$ by comparing their estimated Q-functions:

$$\begin{aligned}
&\max_{X_t \in \mathcal{X}_t} Q(\mathcal{S}_t, X_t) \\
&\geq \max_{X_t \in \mathcal{X}_t} \mathbb{E}\Bigg\{u_t\left(\mathcal{S}_t, X_t, W_t\right) + \sum_{\tau=t+1}^{T} u_\tau\left(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau), W_\tau\right) \Bigg| \begin{matrix} \mathcal{S}_{t+1} = f(\mathcal{S}_t, X_t, W_t), \\ \mathcal{S}_\tau = f(\mathcal{S}_{\tau-1}, \Pi(\mathcal{S}_{\tau-1}), W_{\tau-1}), \\ \tau = t+2, \ldots, T \end{matrix}\Bigg\}.
\end{aligned} \tag{14}$$

In practical applications, both rollout methods (including MCTS) and value function approximation methods offer distinct advantages, with neither approach demonstrating clear superiority across all scenarios. The optimal choice depends largely on specific implementation requirements and problem characteristics. For pharmaceutical product portfolio management, we chose MCTS because we identified a heuristic that is sequentially consistent, making it an effective base policy for the rollout algorithm. While value function approximation approaches can be highly effective in certain scenarios, they face difficulties in accurately approximating the value function in discrete decision spaces with nonlinear response surfaces, as demonstrated in Section 6. MCTS allows us to leverage the problem-specific base policy and our efficient sampling procedure to solve the problem without explicitly approximating the value function.

4.3. Implementation of Monte Carlo Tree Search

The rollout algorithm requires enumerating all possible trajectories to compute Q-function values, which becomes computationally intractable for large-scale pharmaceutical product portfolios. MCTS addresses this challenge by limiting the number of simulations for each decision,

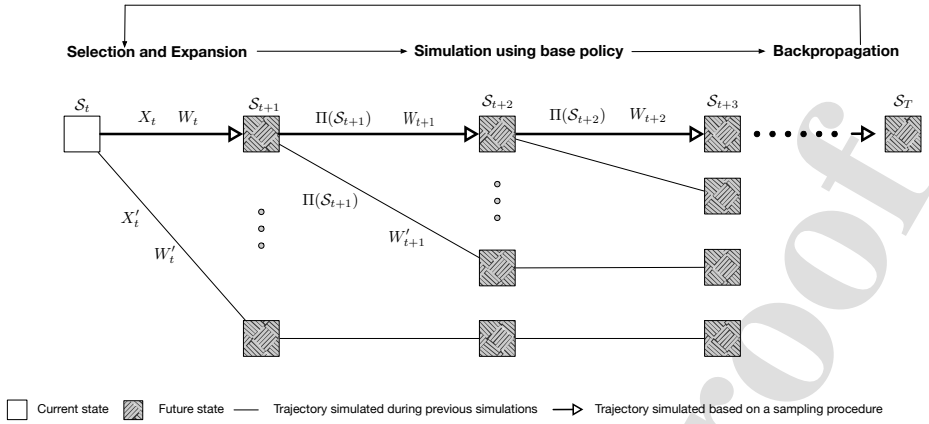


Figure 3: Illustration of the MCTS process with key steps such as selection, expansion, simulation, and backpropagation over search tree.

ensuring computational feasibility. The strategy for distributing the computational budget is defined by the tree policy or sampling procedure.

Algorithm 1 outlines how the base policy (12a - 12e) is integrated into MCTS to identify the best decision. Starting from the current state $\mathcal{S}_t$, MCTS selects a decision $X_t \in \mathcal{X}_t$ for simulation using a sampling procedure. The decision is then executed, exogenous information is observed, the immediate reward is calculated, and the state is updated via the transition function. From epoch $t+1$ until T, the base policy guides subsequent decisions while the state and rewards are updated with realised information. After completing a simulation, the backpropagation phase updates Q-function estimates for state-action pairs. Our MCTS implementation focuses on updating the Q-function values at the current state. This design choice is motivated by the high dimensionality of the state space, the rarity of overlapping states across sampled paths, and the design of the sampling procedure described in Section 5. Let $N_{(\mathcal{S}_t,X_t)}$ represent the number of simulations performed for state-action pair $(\mathcal{S}_t, X_t)$, and let $\hat{Q}^N(\mathcal{S}_t, X_t)$ denote the corresponding Q-function estimate based on these simulations, computed as:

$$\hat{Q}^N(\mathcal{S}_t, X_t) = \frac{1}{N_{(\mathcal{S}_t,X_t)}} \sum_{n=1}^{N_{(\mathcal{S}_t,X_t)}} \left\{ u_t\left(\mathcal{S}_t, X_t, W_t^n\right) + \sum_{\tau=t+1}^{T} u_\tau\left(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau), W_\tau^n\right) \middle| \begin{array}{l} \mathcal{S}_{t+1} = f(\mathcal{S}_t, X_t, W_t^n) \\ \mathcal{S}_\tau = f(\mathcal{S}_{\tau-1}, \Pi(\mathcal{S}_{\tau-1}), W_{\tau-1}^n) \\ \tau = t+2, \ldots, T \end{array} \right\} \quad (15)$$

After each simulation, the Q-function estimate for each simulated decision is updated based on the cumulative rewards observed along the sampled path, as described in (15). Figure 3 provides a visual representation of this iterative process, illustrating the four key phases: selection, expansion, simulation, and backpropagation. The procedure continues until the pre-determined computational budget reaches its limit. Upon completion, the algorithm selects and implements the decision associated with the highest estimated Q-function value in the current state. However, the constrained number of rollout simulations creates a significant challenge in accurately identifying the optimal action. To address this limitation, we present in the next section an efficient sampling procedure that strategically allocates computational resources whilst ensuring a high probability of selecting the correct decision.

Algorithm 1: Monte Carlo tree search to determine the best decision

Input : Feasible decisions $\mathcal{X}_t$ in state $\mathcal{S}_t$, base policy $\Pi(\cdot)$, simulation budget N
Output: Best decision in state $\mathcal{S}_t$
Initialise $N_{(\mathcal{S}_t,X_t)} \leftarrow 0$
while *simulation budget N is not exhausted* **do**
  Selection: Select $X_t \in \mathcal{X}_t$ for simulation based on the sampling procedure
  Expansion: Implement X_t, observe W_t, and compute $u_t(\mathcal{S}_t, X_t, W_t)$
  Simulation: Update $\mathcal{S}_{t+1} = f(\mathcal{S}_t, X_t, W_t)$
  for $\tau \in \{t+1, \ldots, T\}$ **do**
    Use base policy (12a)-(12e) to generate a decision $\Pi(\mathcal{S}_\tau)$
    Observe W_τ, compute $u_\tau(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau), W_\tau)$, and update $\mathcal{S}_{\tau+1} = f(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau), W_\tau)$
  end
  Increment $N_{(\mathcal{S}_t,X_t)} \leftarrow N_{(\mathcal{S}_t,X_t)} + 1$
  Backpropagation: Update $\hat{Q}^N(\mathcal{S}_t, X_t)$ using (15)
end
Return the decision with the highest Q-function estimate

5. Accelerating Simulations with Statistical Racing and Correlated Sampling

We propose a statistical racing procedure to identify the best decision with high probability. This procedure combines two complementary elements: correlated sampling, which reduces the variance of Q-function estimates, and Bernstein's inequality, which provides a statistical bound on the difference between two Q-function estimates. Through this combination, the procedure enables early elimination of suboptimal decisions, thereby allowing computational resources to focus on more promising alternatives. Algorithm 2 details how this statistical racing procedure integrates with MCTS to enhance simulation efficiency. For implementation, consider feasible decisions $X_t \in \mathcal{X}_t$, each with unknown Q-function values. Let $d(X_t, X_t')$ denote the difference between Q-values obtained by implementing X_t and X_t' in state $\mathcal{S}_t$, which is defined as

$$d(X_t, X_t') = \mathbb{E}\left\{ u_t(\mathcal{S}_t, X_t, W_t) - u_t(\mathcal{S}_t, X_t', W_t) + \sum_{\tau=t+1}^{T} u_\tau(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau), W_\tau) - \sum_{\tau=t+1}^{T} u_\tau(\mathcal{S}_\tau', \Pi(\mathcal{S}_\tau'), W_\tau') \right\},$$

where $\mathcal{S}_{t+1} = f(\mathcal{S}_t, X_t, W_t)$, $\mathcal{S}_{t+1}' = f(\mathcal{S}_t, X_t', W_t)$, $\mathcal{S}_\tau = f(\mathcal{S}_{\tau-1}, \Pi(\mathcal{S}_{\tau-1}), W_{\tau-1})$ and $\mathcal{S}_\tau' = f(\mathcal{S}_{\tau-1}', \Pi(\mathcal{S}_{\tau-1}'), W_{\tau-1}')$. The process begins with an initial sampling stage, evaluating the cumulative reward for each decision over a small set of state-trajectories. Let us assume X_t and X_t', evaluated using the same number of simulations. Let $\bar{N}_{(X_t,X_t')}$ denote this common number, i.e., $N_{(\mathcal{S}_t,X_t)} = N_{(\mathcal{S}_t,X_t')} = \bar{N}_{(X_t,X_t')}$. For analytical purposes, we introduce two statistics: $\bar{d}(X_t, X_t')$, denoting the sample mean of the pairwise difference between decisions X_t and X_t', and $\mathrm{Var}(X_t, X_t')$, representing its sample variance. We employ Bernstein's inequality to bound the sampling error, i.e., the difference between the population mean and sample mean given $\bar{N}_{(X_t,X_t')}$ sampled paths. This concentration inequality is particularly effective when $\bar{N}_{(X_t,X_t')}$ is small and the underlying population is highly skewed (Audibert et al., 2009), as it incorporates variance information. Proposition 2 establishes that all pairwise Q-value differences remain bounded, thereby satisfying the fundamental assumptions required for applying Bernstein's inequality.

Proposition 2. *For any two decisions X_t and X_t', their Q-value absolute difference $|d(X_t, X_t')|$ is bounded by a non-negative constant $\theta(X_t, X_t')$.*

Proof. The proof can be found in Appendix C. □

Definition 2. Bernstein's Inequality for Bounded Random Variables. Assuming that Proposition 2 holds, and decisions X_t and X'_t are evaluated using the same number of rollout simulations $\bar{N}_{(X_t,X'_t)}$, Bernstein's inequality for a given significance level α states:

$$|d(X_t, X'_t) - \bar{d}(X_t, X'_t)| \leq \sqrt{\frac{2\text{Var}(X_t, X'_t)\log(1/\alpha)}{\bar{N}_{(X_t,X'_t)}}} + \frac{2\theta(X_t, X'_t)\log(1/\alpha)}{3\bar{N}_{(X_t,X'_t)}}. \tag{16}$$

Bernstein's inequality provides a probabilistic bound on the difference between the true mean and the sample mean based on the sample variance and the number of sampled paths. As the of evaluations increases, the bound becomes tighter, implying our estimates are more likely to be close to the true values. Reducing sample variances can further tighten the Bernstein's bound. To achieve this, we implement correlated sampling, which introduces controlled dependence between random variables. This technique, also known as common random numbers, enables more precise differentiation between Q-function values by ensuring that different decisions are evaluated using identical sets of Bernoulli trial outcomes.

To further improve the sampling efficiency, Algorithm 2 dynamically allocates samples to the most promising decisions while eliminating inferior ones. The procedure compares cumulative reward estimates to select the decision maximising Q-value. Let X_t^* denote the current best decision based on sample averages. Another decision X_t stops sampling in the next iteration if its reward gap with X_t^* satisfies

$$\bar{d}(X_t, X_t^*) < -\sqrt{\frac{2\text{Var}(X_t, X_t^*)\log(1/\alpha)}{\bar{N}_{(X_t,X'_t)}}} - \frac{2\theta(X_t, X_t^*)\log(1/\alpha)}{3\bar{N}_{(X_t,X'_t)}}. \tag{17}$$

This elimination rule is derived from Definition 2. An upper bound on $d(X_t, X_t^*)$ can be obtained as:

$$d(X_t, X_t^*) \leq \bar{d}(X_t X_t^*) + \sqrt{\frac{2\text{Var}(X_t, X_t^*)\log(1/\alpha)}{\bar{N}_{(X_t,X'_t)}}} + \frac{2\theta(X_t, X_t^*)\log(1/\alpha)}{3\bar{N}_{(X_t,X'_t)}}.$$

If $d(X_t, X_t^*)$ is negative, X_t has a lower expected cumulative reward than X_t^* and need not be explored further. The sampling should focus on decisions that can potentially outperform X_t^*. Since $d(X_t, X_t^*)$ is unknown, we stop exploring X_t when its upper bound becomes negative. Setting this upper bound less than zero yields the elimination condition. By comparing the cumulative rewards of various decisions using the same trial outcomes, we can iteratively eliminate inferior decisions. Racing terminates when the simulation budget N is exhausted, ultimately returning the decision with the highest empirical mean reward over the sampled paths.

To assess the performance of the statistical racing method, we employ the family-wise error rate (FWER), which quantifies the probability of observing at least one false positive among multiple comparisons (Ryan, 1959). Controlling the FWER at a desired level is essential for improving the likelihood of correctly identifying the best decision. Loose or overly strict bounds can result in incorrect conclusions about pairwise differences, adversely affecting the results of multiple comparisons. We can show that the FWER of the statistical racing method is bounded above by the Bonferroni correction. Specifically, if the confidence level $1 - \alpha$ is the same for N Bernstein's bounds, the FWER satisfies FWER $\leq N\alpha$. This result follows directly from the properties of Bernstein's bounds. The probability of a single bound being exceeded is α, and

Algorithm 2: MCTS using statistical racing and correlated sampling

Input : Feasible decisions $\mathcal{X}_t$ in state $\mathcal{S}_t$, base policy $\Pi(\cdot)$, type I error α, budget for initial sampling N_0, total simulation budget N
Output: Best decision in state $\mathcal{S}_t$
Initialise $N_{(\mathcal{S}_t,X_t)} \leftarrow 0$
while *simulation budget N is not exhausted* **do**
  Selection: If $\forall X_t \in \mathcal{X}_t, N_{(\mathcal{S}_t,X_t)} \leq N_0$, select all feasible decisions for initial estimation; Otherwise, select from the set of non-eliminated decisions
  Correlated sampling: Generate trial outcomes for the sampled path
  for *each selected X_t* **do**
    Expansion: Implement X_t, observe W_t, and compute $u_t(\mathcal{S}_t, X_t, W_t)$
    Simulation: Update $\mathcal{S}_{t+1} = f(\mathcal{S}_t, X_t, W_t)$
    for $\tau \in \{t+1, \ldots, T\}$ **do**
      Use base policy (12a)-(12e) to generate decision $\Pi(\mathcal{S}_\tau)$
      Observe W_τ, compute $u_\tau(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau), W_\tau)$ and update $\mathcal{S}_{\tau+1} = f(\mathcal{S}_\tau, \Pi(\mathcal{S}_\tau), W_\tau)$
    end
    Increment $N_{(\mathcal{S}_t,X_t)} \leftarrow N_{(\mathcal{S}_t,X_t)} + 1$
    Backpropagation: Update $\hat{Q}^N(\mathcal{S}_t, X_t)$ using (15)
  end
  Statistical racing: **for** *each (X_t, X_t'), where $X_t \neq X_t'$, in the selected list* **do**
    Calculate $\bar{d}(X_t, X_t')$ and $\mathrm{Var}(X_t, X_t')$
    Estimate Bernstein's bound for $|d(X_t, X_t') - \bar{d}(X_t, X_t')|$ via (16) at confidence $1-\alpha$
  end
  Identify the decision with the highest average X_t^*, and eliminate X_t if (17) holds
end
Return the decision with the highest Q-function estimate

applying Boole's inequality to N such bounds yields the stated upper bound on the FWER. By maintaining this upper bound, the statistical racing method ensures that the FWER is controlled according to the chosen confidence level. This control enhances the reliability of the method for identifying optimal decisions in scenarios involving multiple comparisons.

6. Numerical Experiments

This section presents numerical experiments validating the proposed method for pharmaceutical pipeline management under uncertainty. The first experiment evaluates the comparative effectiveness of flexible versus fixed resource profiles in portfolio management. The second experiment benchmarks the proposed statistical racing procedure against state-of-the-art sampling methods in MCTS. The final experiment evaluates MCTS with statistical racing against other approximate dynamic programming approaches.

6.1. Pharmaceutical Pipeline Parameters and their Estimations

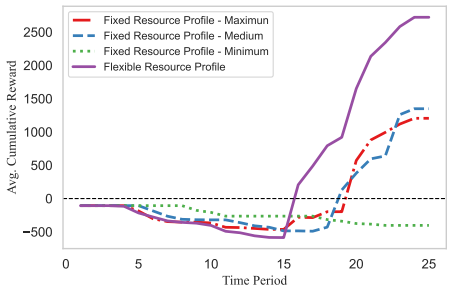
Our testbed simulates a pharmaceutical pipeline with 8 products across major therapeutic areas including oncology, endocrinology, central nervous system, anti-infective, and genitourinary system. Table D.4 presents revenue and clinical parameters derived from 2021 global sales data and established analyses by Wong et al. (2014) and Wong et al. (2019). Revenue projections incorporate patent life adjustments, calculated as half the annual sales every six months over a 20-year period. Clinical trial parameters, sourced from the *ClinicalTrials.gov* database and detailed in Table D.5, reflect therapeutic area characteristics. Oncology trials feature larger

patient populations and extended analysis periods in later phases, with lower recruitment rates. Anti-infective trials require more volunteers initially but achieve faster recruitment. Other therapeutic areas follow standardised parameters based on Wong et al. (2019)'s findings. Following the UK Health Research Authority guidelines (Authority, 2017), we focus on three essential roles: investigators ($k = 1$), nurses ($k = 2$), and statisticians ($k = 3$). The available resource pool comprises 50 investigators, 50 nurses, and 20 statisticians, with specific requirements for each trial phase detailed in Table D.6. The staffing model reflects phase-specific requirements. Phase I maintains consistent staffing across all drugs, requiring one investigator and one statistician, with research nurses excluded from data analysis tasks. Phase II requirements vary by therapeutic area, with anti-infective and oncology trials requiring additional statistical support whilst maintaining standard investigator levels. Phase III demands increased statistical resources across all products due to larger trial populations. Site administration follows a linear resource allocation model, with investigator and nurse requirements scaling proportionally with the number of sites. The numerical experiments were conducted using an Intel Xeon W-2133 CPU with 64GB memory, implemented in Python 3.10.9. The base policy calculations utilised the IBM CPLEX solver with a 0.005 relative gap tolerance.

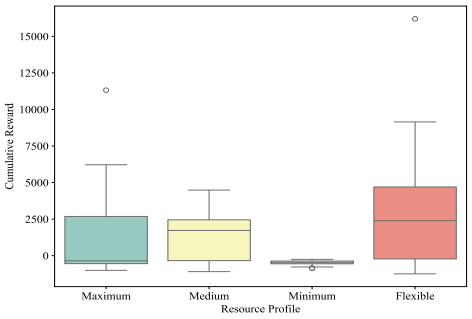
6.2. Profitability of Flexible and Fixed Resource Profiles

As our first contribution, we introduce flexible resource profiles into the pharmaceutical product portfolio management model, demonstrating improved profitability compared to fixed resource profiles. We establish three fixed profile benchmarks: maximum, medium, and minimum resource levels. The maximum resource level conducts patient recruitment at the highest allowable number of test sites. The medium level fixes patient recruitment at the midpoint between the maximum and minimum allowable numbers, while the minimum level sets recruitment at the lowest allowable number. We hypothesize that the flexible model, which adjusts resources based on project needs and economic benefits, will outperform these fixed benchmarks. Both fixed and flexible profiles were implemented in the 8-product testbed. The proposed MCTS approach was used to identify the (near-)optimal decisions. To ensure fair simulation allocation across states with varying numbers of decisions, we define "average evaluations per decision," which quantifies the average number of simulations performed per decision. For this, we set the value to 150, dynamically adjusting the total simulation budget based on the number of feasible decisions at each state. For example, if a state has 20 feasible decisions, the total simulation budget would be 3,000 (20×150) evaluations. This metric ensures states with more decisions receive a proportionally larger budget. Additionally, we set the family-wise error rate (FWER) during racing to 10%, maintaining statistical validity in the procedure's outputs.

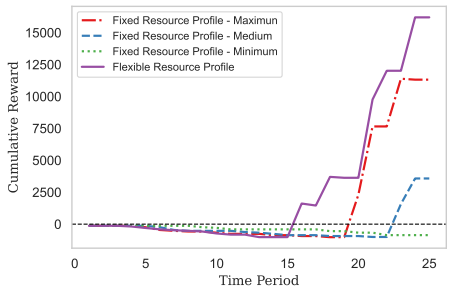
We assessed the performance of resource profiles by measuring average cumulative reward over 30 pipeline development scenarios, each with 25 epochs. The result presented by Figure 4(a) shows a V-shaped reward pattern. Flexible resource profiles provide a significant increase in cumulative reward around epoch 15, whilst fixed resource profiles increased around epoch 18. This suggests that certain drug products successfully completed Phase III, with costs only incurred before this stage. We find that the flexible resource profiles expedited trial completion and produced the highest rewards eventually. Fixed resource profiles at maximum and medium levels per trial yielded similar rewards. The minimum profile led to negative average cumulative



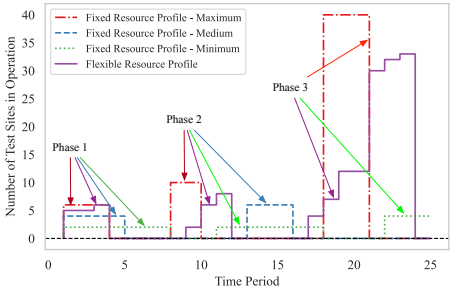
(a) Average cumulative reward across 30 scenarios, comparing fixed and flexible resource profiles.



(b) Box plot of cumulative rewards across 30 scenarios under different resource profiles.



(c) Cumulative reward trajectory for the best scenario.



(d) Resource allocation dynamics for product 2 in the best scenario.

Figure 4: Performance comparison of fixed and flexible resource profiles.

reward over the 30 simulation scenarios due to slow patient recruitment within the planning horizon. We analysed the distribution of cumulative rewards at the end of the planning horizon using box plots in Figure 4(b). Each box represents the interquartile range with the median line; whiskers extend to 1.5 times the interquartile range; points indicate outliers. Flexible resource profiles exhibit a higher median reward and greater variability, indicating potential for higher returns but with increased uncertainty. We also conduct a Kruskal-Wallis test and confirmed significant differences between resource profiles at significant level $p < 0.001$.

In a 'best-case' scenario where all products were approved, we analysed cumulative rewards across all products, and resource usage for product 2. As Figure 4(c) shows, flexible profiles consistently achieved the highest cumulative rewards. Resource allocation analysis for product 2 (Figure 4(d)) shows that fixed maximum profiles prioritised its development - due to its higher revenue potential - at the expense of other products, leading to resource bottlenecks. In contrast, flexible profiles dynamically adjusted resource allocation, maximising overall profitability by balancing the needs of all products. These findings emphasise the importance of optimising resource profiles, as fixed pre-specified profiles seldom achieve maximum profitability.

Finally, Table 2 compares drug approvals across resource profiles. A benchmark is the average number of successful drug products (1.83) achievable under optimal allocation with no constraint. This serves as an upper bound for performance evaluation. The flexible model achieved an average of 1.67 approvals, approaching the upper bound. The minimum fixed profile results in no approvals across all 30 developmental scenarios, aligning with the findings

Table 2: Impact of different resource profiles on pharmaceutical product approvals.

Resource Profile	**Flexible**	**Maximum**	**Medium**	**Minimum**	**Optimal***
Average number of products approved	1.67	0.80	1.07	0.00	1.83
Average Probability of Approval (%)					
Product 1	10.00	0.00	0.00	0.00	13.33
Product 2	0.00	0.00	0.00	0.00	6.67
Product 3	13.33	6.67	16.67	0.00	23.33
Product 4	23.33	0.00	6.67	0.00	26.67
Product 5	23.33	3.33	10.00	0.00	36.67
Product 6	23.33	10.00	0.00	0.00	36.67
Product 7	3.33	3.33	3.33	0.00	16.67
Product 8	13.33	0.00	0.00	0.00	23.33

* Optimal refers to optimal resource allocation under the unlimited resource scenario.

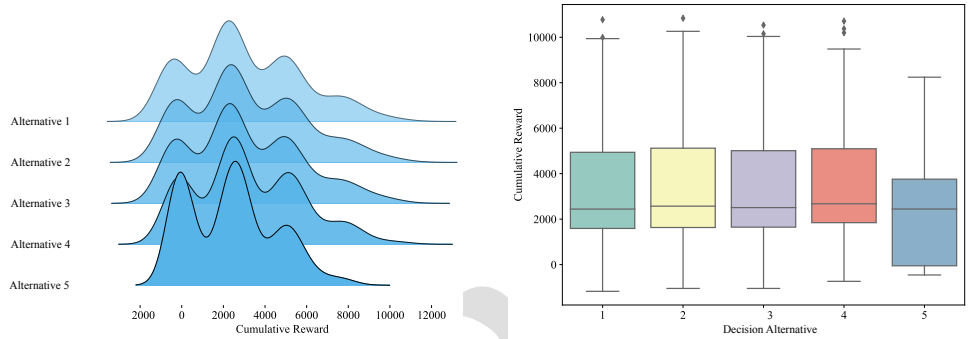
presented in Figure 4(a). This finding demonstrates the importance of adequate resource allocation, as insufficient resources severely impede patient recruitment and, consequently, product development progress. Examining the product-specific approval probabilities reveals further advantages of the flexible profile model. The flexible profile model demonstrates higher or equal approval probabilities compared to fixed profiles for most products, with product 3 being an exception where the medium fixed profile shows higher approval probability. Products 4, 5, and 6 exhibit the highest approval probabilities under the flexible profile model, each achieving a 23.33% chance of approval. When compared to the upper bounds, the flexible profile's performance confirms its ability to achieve near optimal allocations even under resource constraints, making it a valuable tool for pharmaceutical firms managing limited resources.

6.3. Budget Allocation and Efficacy Across Sampling Procedures

The second contribution of this paper is the development of a novel sampling method that efficiently allocates a limited sampling budget (as measured by the number of evaluations) to decision alternatives. We consider a case of just five feasible decisions at the initial state, and thereafter follow the base policy for subsequent states. To benchmark our proposed statistical racing, we have selected four state-of-the-art sampling methods from the literature. Table 3 summarises the main features of these methods in terms of the assumptions and hyper-parameter choices required. The UCB method (Chang et al., 2005) for Q-learning necessitates no hyper-parameters and only assumes that the reward distributions of decision alternatives have finite means and variances. The ϵ-greedy approach (Tokic and Palm, 2011) assumes the reward distributions to have finite means and variances and requires setting a hyper-parameter, ϵ, to balance exploration and exploitation. The expected value of information (EVI) approach (Chick et al., 2010) aims to maximise information gain from sampling. We assume the mean reward distribution for any decision alternative follows a normal distribution and the variance follows an inverse χ^2 distribution, so the marginal posterior distribution for the mean is a scaled and shifted t-distribution. It is worth noting that UCB, ϵ-greedy, and EVI approaches typically assume independence among the rewards of different arms. Implementing correlated sampling with these methods introduces additional complexities, as the induced correlations potentially cause deviations from their theoretical foundations. The indifference zone (IZ) approach, proposed by Malone et al. (2005), assumes that reward distributions are Gaussian.

Table 3: Description of various sampling approaches.

Sampling Approaches	Hyper-parameters	Assumptions	Correlated Sampling?
UCB	N/A	Distributions have finite means and variances	No
ϵ-Greedy	Change in randomly sampling a policy 10%	Distributions have finite means and variances	No
EVI	N/A	Prior and posterior distributions of rewards are conjugate pairs	No
IZ	1) Rate of incorrect selection 10% 2) Indifference zone parameter	1) Difference in reward between the best and the second best policies is known and non-zero 2) Rewards are normally distributed	Yes
Statistical Racing	FWER 10%	Distributions have finite means and variances	Yes



(a) Distributions of cumulative rewards for five decision alternatives.

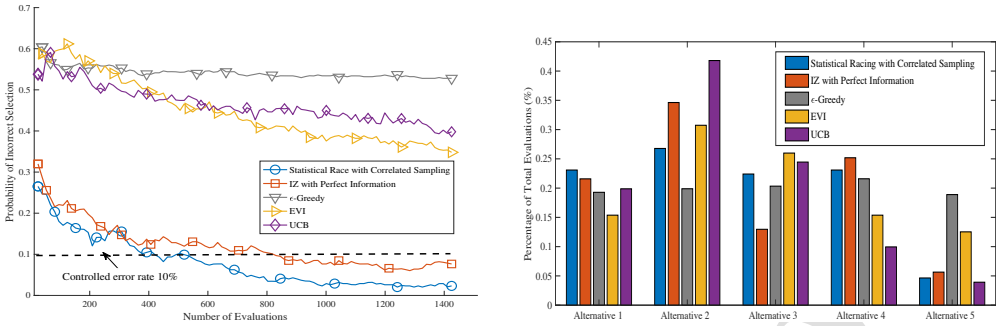
(b) Box plot of cumulative rewards by five decision alternatives.

Figure 5: Descriptive statistics comparing decision alternatives.

The IZ parameter represents the minimum difference between the best and second-best decisions that is considered practically significant. If the goal is to identify the best alternative, the choice for this parameter would be the actual difference between the best and second-best decisions. While the actual value is generally not known, we employ the perfect information assumption and use this true difference for this algorithm. The proposed statistical racing leverages the Bernstein's inequality and assumes that the reward distributions of decision alternatives have finite means and variances.

Figure 5(a) displays a ridgeline plot comparing the cumulative reward distributions for the five decisions across 1,000 developmental scenarios. The reward distributions do not exhibit a simple Gaussian shape but rather present three distinct peaks. The central tendency, dispersion, and range of these reward distributions are depicted using a box plot in Figure 5(b). The result shows that decision 5 has the lowest mean (2,386.05), whilst decisions 2 and 4 display the highest and second-highest sample means (3,188.59 and 3,175.74), respectively. These results suggest that an optimal sampling approach should dedicate the majority of the simulation budget to evaluating the performance of alternatives 2 and 4.

We present results comparing performance of sampling methods in Figure 6(a). The y-axis represents the average probability of incorrect selection over $1,000$ runs, whilst the x-axis shows the average number of allocated evaluations for each method. The UCB, ϵ-greedy, and EVI



(a) Probability of incorrect selection versus the number of evaluations, comparing the convergence performance of different sampling methods.

(b) Simulation budget allocation across decisions.

Figure 6: Performance comparison of sampling methods.

methods, which do not utilise correlated sampling, show a slower decline in their probability of incorrect selection under equivalent evaluation counts. The ϵ-greedy method demonstrates the weakest performance overall. Whilst the upper confidence bound algorithm outperforms the EVI until 220 evaluations, after which EVI becomes more effective, all three methods fail to reduce the probability of incorrect selection to the controlled level. The IZ and the statistical racing methods are compatible with correlated sampling. As shown, even after a few evaluations, their probability of incorrect selection is significantly lower than those of other approaches, suggesting that correlated sampling is a powerful tool for reducing comparison difficulty. It is worth noting that when perfect information on the difference between the best and the second-best is provided (although this assumption is unrealistic in practice), the IZ exhibits strong performance, quickly decreasing the probability of incorrect selection below the controlled level. The proposed statistical racing approach with correlated sampling employs a more reasonable assumption and achieves the best performance under the same number of evaluations. Overall, the findings indicate that the reward distribution of decision alternatives is often irregular in shape, which can affect the performance of sampling approaches that rely on assumptions about the reward distribution. Our statistical racing method is compatible with any sub-Gaussian distribution, offering a more general and robust approach compared to several well-known sampling methods. Additionally, for the pharmaceutical pipeline management model, correlated sampling effectively reduces variance and enhances the performance of sampling approaches, a factor that has been overlooked in the literature.

Figure 6(b) shows the average evaluations assigned to each decision, highlighting differences in sampling across methods. Optimal sampling should minimise evaluations for decision 5, which demonstrates the lowest expected reward. However, the ϵ-greedy method allocates evaluations more uniformly due to random sampling, explaining its weaker performance. The UCB algorithm focuses on the decision with the highest upper bound, heavily sampling decision 2, but, as shown in Figure 6(a), this allocation fails to sufficiently reduce the probability of incorrect selection. In contrast, both the EVI and IZ methods concentrate their evaluations on distinguishing between decisions 2 and 4, reflecting a more nuanced approach to sampling budget allocation. Our statistical racing method demonstrates sampling behaviour that aligns with its

theoretical foundations. Our method efficiently eliminates decision 5 due to its consistently low average reward, whilst allocating similar evaluation resources to the more competitive decisions 1, 3, and 4. As expected, decision 2 receives the largest simulation budget.

6.4. Performance Comparison of Various Approximate Dynamic Programming Techniques

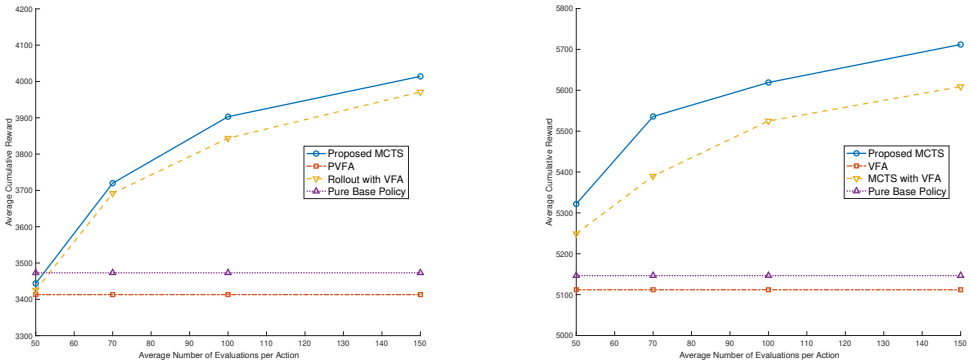
Lastly, we compare the performance of the proposed MCTS with statistical racing and correlated sampling to the following approximate dynamic programming techniques:

- *Parametric Value Function Approximation (PVFA)* combines forward and backward passes to approximate the value function using multivariate linear regression. This method employs state variables as independent variables and cumulative reward as the dependent variable. The regression coefficients are initialised to zero, with the algorithm executing up to 2,000 iterations using a temporal difference learning rate of 0.1. During the forward pass, the algorithm begins from the initial state, transitioning from random action selection in early iterations to Bellman equation-guided selections in later stages. The system evolves according to its dynamics, observes rewards, and updates state variables until reaching the final time horizon, thereby generating a complete trajectory. The backward pass refines the value function by updating regression coefficients through temporal difference learning, incorporating new data from the forward pass. This process iterates until reaching the specified limit. This approach aligns with Gökalp and Branke (2020), with our implementation differing in allowing for flexible resource profiles.

- *Rollout with PVFA* is derived from the Bellman equation solution using the PVFA after 2,000 iterations. At each state, the tree policy selects a decision randomly for simulation with the base policy, choosing the action with the highest Q-function value. Due to the policy improvement property, this approach may yield performance improvements compared to PVFA alone.

- *Pure Base Policy* utilises the mathematical programming model detailed in Section 4 to determine decisions directly for a given state, without employing MCTS. This serves as a baseline for evaluating the effectiveness of both MCTS and rollout approaches.

We evaluated the average cumulative reward across 30 scenarios, examining four levels of average evaluations per decision (50, 70, 100, and 150) to simulate varying time constraints for decision execution. To assess scalability and robustness, we extended the analysis to a larger pharmaceutical R&D pipeline scenario with 20 products, maintaining the same resource capacity to create a more constrained environment. The expanded scale aligns with industry practices (Citeline, 2024), representing a typical portfolio size for a mid-sized pharmaceutical company or a therapeutic area division in a large organisation. Pipelines typically range from 10 to 50 products. The planning horizon consists of 25 epochs (six months per epoch), consistent with standard portfolio decision-making cycles. Our model incorporates critical operational constraints, including limited skilled workforce, restricted clinical test sites, and varying patient recruitment rates across therapeutic areas. This configuration creates a tightly constrained setting for evaluating both proposed and benchmark methods under increased complexity. Detailed information about the additional 12 products and related parameters appears in Appendix E.



(a) 8-product R&D pipeline.

(b) 20-product R&D pipeline.

Figure 7: Average cumulative reward obtained by approximate solution techniques for pharmaceutical R&D pipelines of varying scales.

Figures 7(a) and 7(b) compare the average cumulative rewards across four approximate approaches for both the 8-product and 20-product scenarios. The pure base policy, which functions as a heuristic without requiring pre-training and operates independently of the evaluation number. In contrast, while the value function approximation offers rapid implementation after initial setup, it requires substantial upfront training time. The pure base policy achieves superior results compared to PVFA (trained with 2,000 iterations) in both scenarios, potentially due to the limitations of linear functions in capturing the complexity of the value function. Both MCTS and rollout with PVFA demonstrate progressive improvement as the number of evaluations per action increases. Under constrained computational conditions (50-55 evaluations per action), the pure base policy maintains a slight performance advantage over both MCTS and rollout with PVFA approaches, indicating the efficacy of heuristic methods when computational resources are limited. However, with increased evaluation capacity, both MCTS and rollout with PVFA show enhanced performance through improved Q-function estimation. The proposed MCTS approach emerges as the superior method, consistently outperforming rollout with PVFA across all evaluation ranges in both scenarios. This performance advantage maintains consistency between the 8-product and 20-product scenarios, demonstrating MCTS robust scalability to more complex problem environments.

7. Conclusions

We examined a pharmaceutical portfolio management problem focused on optimising clinical trial activity scheduling and feasible resource allocation. We formulated this as a Markov decision process and developed an MCTS approach that iteratively identifies near optimal decisions via Q-function value estimation and a base policy. The base policy demonstrates sequential consistency, enabling performance improvement. We proposed a statistical racing method using Bernstein's inequality and correlated sampling to improve MCTS efficiency and provide correct selection guarantees within a limited sampling budget. Numerical experiments demonstrated three key benefits: the advantages of flexible resource profiles in pharmaceutical product port-

folio management, the efficiency of statistical racing with correlated sampling for allocating computational budget, and the effectiveness of the proposed MCTS approach in identifying near optimal policies. Several promising directions exist for future research. The model could be extended to incorporate additional uncertainties, such as variable recruitment rates and task durations. Integration of adaptive trial designs could enable early termination when data indicates negative outcomes. Furthermore, combining MCTS with multivariate regression or Gaussian processes could yield more efficient Q-function approximation methods that eliminate the need to evaluate all decisions.

References

Apap, R. M. and Grossmann, I. E. (2017). Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. *Computers & Chemical Engineering*, 103:233–274.

Audibert, J.-Y., Munos, R., and Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902.

Authority, H. R. (2017). UK policy framework for health and social care research.

Bertsekas, D. (2019). *Reinforcement learning and optimal control*. Athena Scientific.

Bertsekas, D. P., Tsitsiklis, J. N., and Wu, C. (1997). Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3(3):245–262.

Bertsimas, D., Griffith, J. D., Gupta, V., Kochenderfer, M. J., and Mišić, V. V. (2017). A comparison of monte carlo tree search and rolling horizon optimization for large-scale dynamic resource allocation problems. *European Journal of Operational Research*, 263(2):664–678.

Chang, H. S., Fu, M. C., Hu, J., and Marcus, S. I. (2005). An adaptive sampling algorithm for solving Markov decision processes. *Operations Research*, 53(1):126–139.

Chick, S. E., Branke, J., and Schmidt, C. (2010). Sequential sampling to myopically maximize the expected value of information. *INFORMS Journal on Computing*, 22(1):71–80.

Choi, J., Realff, M. J., and Lee, J. H. (2004). Dynamic programming in a heuristically confined state space: a stochastic resource-constrained project scheduling application. *Computers & Chemical Engineering*, 28(6-7):1039–1058.

Christian, B. and Cremaschi, S. (2015). Heuristic solution approaches to the pharmaceutical R&D pipeline management problem. *Computers & Chemical Engineering*, 74:34–47.

Citeline (2024). Leading 15 pharmaceutical companies worldwide by size of R&D pipeline as of 2024. `https://www-statista-com.eux.idm.oclc.org/statistics/791306/top-pharma-companies-by-randd-pipeline-size/`.

Colin, T. and Neil, L. (2022). Measuring the return from pharmaceutical innovation.

Colvin, M. and Maravelias, C. T. (2008). A stochastic programming approach for clinical trial planning in new drug development. *Computers & Chemical Engineering*, 32(11):2626–2642.

De Reyck, B. and Leus, R. (2008). R&D project scheduling when activities may fail. *IIE Transactions*, 40(4):367–384.

Forman, R., Shah, S., Jeurissen, P., Jit, M., and Mossialos, E. (2021). COVID-19 vaccine challenges: What have we learned so far and what remains to be done? *Health Policy*.

Fu, M. C., Hu, J.-Q., Chen, C.-H., and Xiong, X. (2007). Simulation allocation for determining the best design in the presence of correlated sampling. *INFORMS Journal on Computing*, 19(1):101–111.

Gökalp, E. and Branke, J. (2020). Pharmaceutical R & D pipeline management under trial duration uncertainty. *Computers & Chemical Engineering*, 136:106782.

Goodson, J. C., Ohlmann, J. W., and Thomas, B. W. (2013). Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. *Operations Research*, 61(1):138–154.

Gupta, A. and Maranas, C. D. (2004). Real-options-based planning strategies under uncertainty. *Industrial & Engineering Chemistry Research*, 43(14):3870–3878.

Issa, S. B., Patterson, R. A., and Tu, Y. (2021). Solving resource-constrained multi-project environment under different activity assumptions. *International Journal of Production Economics*, 232:107936.

Jain, V. and Grossmann, I. E. (1999). Resource-constrained scheduling of tests in new product development. *Industrial & Engineering Chemistry Research*, 38(8):3013–3026.

Kaitin, K. I. and DiMasi, J. A. (2011). Pharmaceutical innovation in the 21st century: new drug approvals in the first decade, 2000–2009. *Clinical Pharmacology & Therapeutics*, 89(2):183–188.

Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

Kogan, B., Chernonog, T., and Herbon, A. (2024). Project scheduling to minimize the makespan under flexible resource profiles and marginal diminishing returns of the resource. *Computers & Operations Research*, 161:106440.

Krafft, O. and Schmitz, N. (1969). A note on Hoeffding's inequality. *Journal of the American Statistical Association*, 64(327):907–912.

Krüger, D. and Scholl, A. (2010). Managing and modelling general resource transfers in (multi-) project scheduling. *OR Spectrum*, 32(2):369–394.

Li, H. and Womer, N. K. (2015). Solving stochastic resource-constrained project scheduling problems by closed-loop approximate dynamic programming. *European Journal of Operational Research*, 246(1):20–33.

Li, Y., Fu, M. C., and Xu, J. (2021). An optimal computing budget allocation tree policy for monte carlo tree search. *IEEE Transactions on Automatic Control*, 67(6):2685–2699.

Malone, G., Kim, S.-H., Goldsman, D., and Batur, D. (2005). Performance of variance updating ranking and selection procedures. In *Proceedings of the Winter Simulation Conference, 2005.*, pages 825–83.

Maron, O. and Moore, A. W. (1997). The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11(1-5):193–225.

Naber, A. (2017). Resource-constrained project scheduling with flexible resource profiles in continuous time. *Computers & Operations Research*, 84:33–45.

Pérez, E., Posada, M., and Lorenzana, A. (2016). Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms. *Soft Computing*, 20:1879–1896.

Powell, W. B. (2016). Perspectives of approximate dynamic programming. *Annals of Operations Research*, 241(1):319–356.

Rogers, M. J., Gupta, A., and Maranas, C. D. (2002). Real options based analysis of optimal pharmaceutical research and development portfolios. *Industrial & Engineering Chemistry Research*, 41(25):6607–6620.

Ryan, T. A. (1959). Multiple comparison in psychological research. *Psychological Bulletin*, 56(1):26.

Sánchez, M. G., Lalla-Ruiz, E., Gil, A. F., Castro, C., and Voß, S. (2023). Resource-constrained multi-project scheduling problem: A survey. *European Journal of Operational Research*, 309(3):958–976.

Satic, U., Jacko, P., and Kirkbride, C. (2024). A simulation-based approximate dynamic programming approach to dynamic and stochastic resource-constrained multi-project scheduling problem. *European Journal of Operational Research*, 315(2):454–469.

Schmidt, C. W. and Grossmann, I. E. (1996). Optimization models for the scheduling of testing tasks in new product development. *Industrial & Engineering Chemistry Research*, 35(10):3498–3510.

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

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.

Świechowski, M., Godlewski, K., Sawicki, B., and Mańdziuk, J. (2023). Monte carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review*, 56(3):2497–2562.

Tokic, M. and Palm, G. (2011). Value-difference based exploration: adaptive control between epsilon-greedy and softmax. In *Annual Conference on Artificial Intelligence*, pages 335–346. Springer.

Van Den Eeckhout, M., Vanhoucke, M., and Maenhout, B. (2021). A column generation-based diving heuristic to solve the multi-project personnel staffing problem with calendar constraints and resource sharing. *Computers & Operations Research*, 128:105163.

Verderame, P. M., Elia, J. A., Li, J., and Floudas, C. A. (2010). Planning and scheduling under uncertainty: a review across multiple sectors. *Industrial & Engineering Chemistry Research*, 49(9):3993–4017.

Wong, C. H., Siah, K. W., and Lo, A. W. (2019). Estimation of clinical trial success rates and related parameters. *Biostatistics*, 20(2):273–286.

Wong, H.-H., Jessup, A., Sertkaya, A., Birkenbach, A., Berlind, A., and Eyraud, J. (2014). Examination of clinical trial costs and barriers for drug development final. *Office of the Assistant Secretary for Planning and Evaluation, US Department of Health & Human Services*, pages 1–92.

Xie, F., Li, H., and Xu, Z. (2021). An approximate dynamic programming approach to project scheduling with uncertain resource availabilities. *Applied Mathematical Modelling*, 97:226–243.

Appendix A. Notation and Definitions

Indices

- t: Decision epoch, $t \in \mathcal{T} = \{0, 1, \ldots, T\}$.
- i: Pharmaceutical product, $i \in \mathcal{I} = \{1, \ldots, I\}$.
- j: Phase of a clinical trial, $j \in \mathcal{J} = \{1 \text{ (Phase I)}, 2 \text{ (Phase II)}, 3 \text{ (Phase III)}\}$.
- k: Types of resources, $k \in \mathcal{K} = \{1, 2, \ldots, K\}$.

Parameters

- $c_{i,j}^{\text{Recr}}, c_{i,j}^{\text{Data}}$: Patient recruitment and data analysis costs for Phase j of product i.
- $q_{i,j}^{\text{Target}}$: Number of patient volunteers required for Phase j of product i.
- $\lambda_{i,j}$: Time periods required to complete data analysis for Phase j of product i.
- $r_{i,j,k}$: Amount of resource of type k required to complete analysis for Phase j of product i.
- $\rho_{i,j}^{\text{Site}}$: Average recruitment rate at each site for Phase j of product i.
- $\beta_{i,j,k}$: Resource allocation coefficient for Phase j of product i.
- $h_{i,j}^{\text{Max}}$, $h_{i,j}^{\text{Min}}$: Maximum and minimum numbers of test sites for Phase j of product i.
- Γ_i: Maximum revenue of product i if marketing authorisation is obtained.
- γ_i: Revenue loss due to the reduced period of exclusive marketing for product i.

Decisions & States

- X_t: Decision variables at epoch t.
- $x^{\text{Recr}}_{i,j,t}$: 1 if Phase j recruitment starts for product i at epoch t, 0 otherwise.
- $x^{\text{Site}}_{i,j,t}$: Number of additional test sites assigned to Phase j of product i at epoch t.
- $x^{\text{Data}}_{i,j,t}$: 1 if data analysis for Phase j of product i starts at epoch t, 0 otherwise.
- $\mathcal{S}_t$: State variable at decision epoch t.
- $L^{\text{Data}}_{i,j,t}$: Remaining time to complete analysis for Phase j of product i at epoch t.
- $L^{\text{Recr}}_{i,j,t}$: Remaining patients needed for Phase j of product i at epoch t.
- $A^{\text{Recr}}_{i,j,t}$: 1 if patient recruitment for Phase j of product i can be scheduled at epoch t.
- $A^{\text{Data}}_{i,j,t}$: 1 if data analysis for Phase j of product i can be scheduled at epoch t.
- $R^{\text{Site}}_{i,j,t}$: Number of test sites assigned to Phase j of product i at epoch t.
- $R_{k,t}$: Amount of resources of type k available at epoch t.

Uncertainties

- W_t: Exogenous information at epoch t.
- $\omega_{i,j}$: Outcome of Phase j of product i, following Bernoulli variable $\Omega_{i,j}$ with success probability $p_{i,j}$.

Reward

- $u_t\left(\mathcal{S}_t, X_t, W_t\right)$: Discounted profit of the R&D pipeline at decision epoch t.
- $V_t(\mathcal{S}_t)$: Maximum expected profit value when starting in state $\mathcal{S}_t$ and acting optimally thereafter.

Appendix B. Proof Proposition 1

The base policy decisions are obtained by solving the mathematical programming (12a) - (12e). If state $\mathcal{S}_\tau$ remains unchanged, the optimal decisions will be the same regardless of how many times the optimisation is solved. Consider applying the base policy to estimate future value. It informs decisions in state $\mathcal{S}_\tau, \tau = t+1, \ldots, T$, resulting in a state trajectory $\left\{\mathcal{S}_{t+1}, \mathcal{S}_{t+2}, \ldots, \mathcal{S}_T\right\}$ and exogenous information $\{W_{t+1}, \ldots, W_T\}$. If the outcomes in the exogenous information remain unchanged, re-solving the base policy optimisation will yield the same state trajectory. Therefore, we can conclude the base policy satisfies sequential consistency - applying it repeatedly in a given state will reproduce the same trajectory, as long as the exogenous information realisations remain fixed.

Appendix C. Proof Proposition 2

The cumulative rewards from time t to T depend on both the decisions made over the planning horizon, and the outcomes of pharmaceutical product development. Since the planning horizon is finite, there is a limit to the reward obtained from any single decision. Also, the reward of a decision can vary substantially based on whether products are approved or not. Therefore, we can define a constant $\theta(X_t, X'_t)$ that bounds the difference between Q-values achieved by any two decisions X_t and X'_t from time t to T. This constant can be estimated by comparing the best-case scenario (all products are approved) and the worst-case scenario (no product is approved) for each decision.

Appendix D. 8-Product Pharmaceutical Pipeline Parameters

Table D.4: Product success probabilities, work costs by phases (P.1, P.2 and P.3), max revenues, and patent loss.

Prod.	**Therap.**	**P.1**			**P.2**			**P.3**			Γ_i	γ_i
ID	**Areas**	%	(10^6)		%	(10^6)		%	(10^6)		(10^6)	(10^6) /6mo
		$p_{i,1}$	$c_{i,1}^{\text{Recr}}$	$c_{i,1}^{\text{Data}}$	$p_{i,2}$	$c_{i,2}^{\text{Recr}}$	$c_{i,2}^{\text{Data}}$	$p_{i,3}$	$c_{i,3}^{\text{Recr}}$	$c_{i,3}^{\text{Data}}$		
1	Oncol.	58	3	2	33	9	3	30	18	5	8,750	218
2	Oncol.	57	3	1	30	9	4	28	18	4	8,125	203
3	Endocrinol.	76	1	1	59	9	3	52	13	5	5,000	125
4	Cent. Nerv.	72	3	1	53	10	4	52	15	5	4,062	101
5	Cent. Nerv.	74	4	1	52	10	4	54	13	5	4,375	109
6	Anti-infect.	68	3	1	59	11	4	68	18	5	6,625	165
7	Anti-infect.	70	4	2	61	12	5	67	21	6	6,250	156
8	Genitourin.	67	2	1	46	11	4	64	13	5	1,625	40

Table D.5: Recruitment, data analysis time, and test site information by phases.

Prod.	$q_{i,j}^{\text{Target}}$			$\rho_{i,j}^{\text{Site}}$ *			$[h_{i,j}^{\text{Min}}, h_{i,j}^{\text{Max}}]$			$\lambda_{i,j}$ †		
ID	**P.1**	**P.2**	**P.3**	**P.1**	**P.2**	**P.3**	**P.1**	**P.2**	**P.3**	**P.1**	**P.2**	**P.3**
1	50	80	3,300	6	6	25	[2,6]	[2,10]	[4,40]	3	2	8
2	80	100	2,000	6	8	20	[2,6]	[2,10]	[4,40]	3	3	7
3	25	50	340	2	4	6	[2,6]	[2,6]	[10,40]	2	2	4
4	24	55	400	2	4	6	[2,6]	[2,6]	[10,50]	2	2	5
5	20	60	350	4	4	6	[2,6]	[2,6]	[10,50]	2	3	4
6	70	200	400	8	10	14	[2,6]	[2,6]	[2,6]	2	2	5
7	100	160	300	8	10	16	[2,6]	[2,6]	[2,6]	3	3	4
8	30	70	350	2	4	6	[2,6]	[2,6]	[10,50]	2	2	4

* Number of individuals recruited per 6 months.
† Duration is expressed in 6-month units.

Table D.6: Staffing requirements for data analysis and test site administration.

Prod.	**Data Analysis**							**Site Administration**		
ID	$r_{i,1,1}$	$r_{i,1,3}$	$r_{i,2,1}$	$r_{i,2,3}$	$r_{i,3,1}$	$r_{i,3,3}$	$r_{i,j,2}, \forall j$	$\beta_{i,j,1}, \forall j$	$\beta_{i,j,2}, \forall j$	$\beta_{i,j,3}, \forall j$
1	1	1	1	2	1	4	0	1	1	0
2	1	1	1	2	1	4	0	1	1	0
3	1	1	1	1	1	2	0	1	1	0
4	1	1	1	1	1	2	0	1	1	0
5	1	1	1	1	1	2	0	1	1	0
6	1	1	1	2	1	2	0	1	1	0
7	1	1	1	2	1	2	0	1	1	0
8	1	1	1	1	1	2	0	1	1	0

Appendix E. Expanded Pharmaceutical Pipeline Details

Table E.7: Expanded pipeline - Success probabilities, work costs by phases, max revenues, and patent loss

Prod.	**Therap.**	**P.1**			**P.2**			**P.3**			Γ_i	γ_i
ID	**Areas**	%	(10^6)		%	(10^6)		%	(10^6)		(10^6)	(10^6)
		$p_{i,1}$	$c_{i,1}^{\text{Recr}}$	$c_{i,1}^{\text{Data}}$	$p_{i,2}$	$c_{i,2}^{\text{Recr}}$	$c_{i,2}^{\text{Data}}$	$p_{i,3}$	$c_{i,3}^{\text{Recr}}$	$c_{i,3}^{\text{Data}}$		/6mo
9	Oncol.	59	3	2	35	10	3	32	18	4	8,500	212
10	Endocrinol.	74	1	1	57	9	4	50	13	5	5,250	131
11	Cent. Nerv.	70	3	2	54	11	4	53	15	5	4,200	105
12	Anti-inf.	69	3	2	60	12	4	70	18	6	6,500	162
13	Genitourin.	65	3	1	48	12	3	62	14	5	1,750	43
14	Oncol.	56	4	2	32	12	3	29	19	5	8,375	209
15	Endocrinol.	78	1	1	61	9	3	54	13	3	4,750	118
16	Cent. Nerv.	73	4	1	51	11	3	55	13	5	4,150	103
17	Anti-inf.	71	4	2	62	12	5	69	21	4	6,375	159
18	Genitourin.	68	2	1	47	11	4	65	13	5	1,500	37
19	Oncol.	60	3	1	34	9	3	31	17	4	8,625	215
20	Endocrinol.	75	1	1	58	9	3	51	12	5	5,125	128

Table E.8: Expanded pipeline - Recruitment, data analysis time, and test site information by phases.

Prod.	$q_{i,j}^{\text{Target}}$			$\rho_{i,j}^{\text{Site}}$			$[h_{i,j}^{\text{Min}}, h_{i,j}^{\text{Max}}]$			$\lambda_{i,j}$		
ID	**P.1**	**P.2**	**P.3**	**P.1**	**P.2**	**P.3**	**P.1**	**P.2**	**P.3**	**P.1**	**P.2**	**P.3**
9	80	90	2,800	5	7	22	[2,32]	[3,10]	[4,40]	3	3	8
10	30	55	360	2	4	6	[2,40]	[2,8]	[10,40]	3	3	4
11	22	58	380	2	4	6	[2,50]	[2,10]	[10,50]	3	3	5
12	85	180	350	8	10	15	[2,6]	[2,6]	[2,6]	3	3	5
13	35	75	370	2	4	6	[2,50]	[2,10]	[10,50]	3	2	5
14	70	95	3,300	5	7	22	[2,32]	[3,11]	[5,41]	3	3	8
15	28	52	340	2	4	6	[2,40]	[2,8]	[10,40]	2	3	4
16	23	57	390	2	4	6	[2,50]	[2,10]	[10,50]	2	3	4
17	90	190	375	8	10	15	[2,6]	[2,6]	[3,8]	3	3	4
18	32	72	365	2	4	6	[2,50]	[2,10]	[10,50]	2	3	4
19	65	88	3,100	5	7	24	[2,22]	[3,11]	[6,41]	3	3	8
20	27	53	350	2	4	6	[2,40]	[2,8]	[10,40]	2	3	4

Table E.9: Expanded pipeline - Staffing requirements for data analysis and test site administration.

Prod.	**Data Analysis**							**Site Administration**		
ID	$r_{i,1,1}$	$r_{i,1,3}$	$r_{i,2,1}$	$r_{i,2,3}$	$r_{i,3,1}$	$r_{i,3,3}$	$r_{i,j,2}, \forall j$	$\beta_{i,j,1}, \forall j$	$\beta_{i,j,2}, \forall j$	$\beta_{i,j,3}, \forall j$
9	1	1	1	1	1	4	0	1	1	0
10	1	1	1	2	1	2	0	1	1	0
11	1	1	1	1	1	2	0	1	1	0
12	1	1	1	1	1	2	0	1	1	0
13	1	1	1	2	1	2	0	1	1	0
14	1	1	1	1	1	4	0	1	1	0
15	1	1	1	2	1	2	0	1	1	0
16	1	1	1	1	1	2	0	1	1	0
17	1	1	1	2	1	2	0	1	1	0
18	1	1	1	1	1	2	0	1	1	0
19	1	1	1	2	1	4	0	1	1	0
20	1	1	1	1	1	2	0	1	1	0

Highlights

Dynamic Pharmaceutical Product Portfolio Management with Flexible Resource Profiles

Xin Fei, Jürgen Branke, Nalân Gülpınar

- Formulated the pharmaceutical portfolio management problem as a Markov decision process.
- Determined resource allocation and trial scheduling under uncertain trial outcomes.
- Proposed Monte Carlo tree search and statistical racing approach.
- Achieved policy quality and computational efficiency over the existing methods.