PhD-FDEF-2023-035
The Faculty of Law, Economics and Finance

DISSERTATION

Defence held on 13/10/2023 in Luxembourg
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN SCIENCES ECONOMIQUE

by

Bonn Kleiford SERANILLA
Born on July 4, 1994, in Cagayan de Oro, Philippines

ON THE APPLICATIONS OF STOCHASTIC DUAL DYNAMIC PROGRAMMING

Dissertation Defence Committee

Prof. Dr. Nils Löhndorf, Dissertation Supervisor
Professor, Université du Luxembourg

Prof. Dr. Joachim Arts, Chair
Professor, Université du Luxembourg

Prof. Dr. Benny Mantin, Vice-Chair
Professor, Université du Luxembourg

Prof. Dr. Alexander Shapiro
Professor, Georgia Institute of Technology

Prof. Dr. Jörg Kalcsics
Professor, University of Edinburgh

Abstract

Multistage stochastic programming (MSP) problems belong to a class of problems that involve a sequence of decisions made over multiple time stages under uncertainty. Many real-world problems can be effectively represented using MSPs. However, MSPs pose challenges in optimization due to their inherent difficulty and complexity. In the literature, Stochastic Dual Dynamic Programming (SDDP) has emerged as a powerful and versatile methodology for solving MSPs. This thesis showcases the applications of SDDP in handling sequential decision-making under uncertainty across various domains.

We begin with a comprehensive introduction to MSPs, exploring their practical applications and various solution approaches. Additionally, we trace the historical development of SDDP from Benders' Decomposition to its modern enhancements.

In Chapter 2, we conduct a comprehensive survey of the diverse applications of SDDP in the literature. This includes an analysis of statistics on the prevalence of SDDP usage in various domains. Moreover, a substantial focus is placed the most common application of SDDP in the energy sector, particularly in hydro-thermal power production scheduling. The chapter outlines compelling arguments for the prominence of this specific application.

Chapter 3 introduces two valuable contributions: MSPLib, an open-source library of problems and MSPFormat, a standardized data format designed for benchmarking SDDP. MSPLib aims to facilitate the evaluation of computational performance among different SDDP implementations. It offers a wide array of instances, from real-world problems to synthetic variations with varying complexities. By incorporating MSPFormat into the library, a unified and consistent representation of MSPs is provided, further enhancing their usability and transferability.

In Chapter 4, we showcase an MSP application to the optimal location of COVID-19 vaccine facilities under the threat of natural disasters. We introduce a new algorithm, named *shadow price approximation* (SPA), which aims at approximating the shadow price of opening flood-prone vaccine facilities by tuning the parameters of a linear value function approximation which is present in the objective function of base optimization model. We also compare the performance of SPA against stochastic dual dynamic integer programming (SDDiP). The chapter closes with a detailed account of this model's application in two cities of a developing country.

Moving on to Chapter 5, we introduce a novel problem class named the multistage stochastic facility location problem under facility disruption uncertainty (MSFLPD).

This new class extends the classical stochastic *capacitated* facility location problem to handle uncertainty arising from facility disruptions. We then present and compare two solution algorithms tailored for addressing this problem: stochastic dual dynamic integer programming (SDDiP) and shadow price approximation (SPA).

Acknowledgments

I consider the last four years as a period of holistic growth. Nonetheless, it took a village for me to be where I am now. The individuals mentioned in this thank you note witnessed my growth, learning, enjoyment, parties, struggles, tears, and moments of feeling lost. I cannot find a way to repay all the love and support I have received.

First of all, my deepest gratitude and thanks go to my supervisor, Nils Löhndorf, for seeing the potential in me. I vividly remember the interview we had when I applied, and I honestly admitted that I had never taken any course on Stochastic Programming. I told you that I would do my best to learn as much as I could if given the chance - and you did give me that chance. Now, I find myself writing a dissertation I never thought I would in my life. Thank you for encouraging me not only to work hard but also to work smart, to go beyond what I had learned in class, and to go the extra mile. Your support was unwavering when I expressed my desire to work on the COVID-19 vaccine facility location problem, even though we initially thought it would not be part of this thesis. As a result, we received an international award for it. I am incredibly fortunate to have been your first PhD student in LCL. Thank you, Nils, for everything.

I would also like to extend my gratitude to all the professors in LCL. Joachim, thank you for all the valuable advice you provided to help me live my life as a PhD student to the fullest. I will always cherish our cold walks in Boston - with hot chocolate in hand and dilemmas to tackle. Benny, your unwavering commitment to excellence kept our team at the highest standard. Your cheer when I won the INFORMS award will forever be etched in my memory. Anne, thank you for consistently showing enthusiasm and belief in me from day one. I truly appreciated the coffee outing on my first day. Cagil, you are and will always be an inspiration to me. Thank you for being my Drag Race buddy in the office.

I might have gone insane if it weren't for my lovely and therapeutic time with you, Jackie. You always find a way to make me laugh and calm me down. Our minds seem to sync on the craziest wavelengths, and I always knew you would become my best friend in the office. Thank you for sharing Red with me - both wine and your adorable doggo. I cannot wait to spend more time with you.

Thanks to the first PhDs of LCL who have inspired me to follow in their footsteps. Melvin, Nicole, and You, thank you for setting the bar high for all of us. You guys are great inspirations. Roozbeh, my brother and all-around buddy in LCL, thank you for our cherished Starbucks walks. Tiffany, thank you for always checking on me and being a great friend. Special thanks as well to Sarah, Laurens, Matteo, Rishikesh, Neeraj, Poulad, Ranit, Nahid, Junxia, Mohammad, Farid, Raquel, Daria, Eleonore, and Carla for bringing life to our office.

My life outside LCL is also as colorful and fun because of my two best friends - Carolina and Samuele. Mami and Papi, you have no idea how lucky I feel to be a part of this trio. Thank you for visiting my home country, even though I haven't been to any of your hometowns yet. Also, with you, I met Luce, Adnan, Milos, Tessa, Nabina, Nitesh, Marielle, Sadeq, and Vitaliy. Thank you for your friendship. Additionally, I want to express my gratitude to my dance partner, Meghan, for always keeping me in the loop and providing opportunities to continue my love for dancing.

I also want to thank my *Pinoy* gang for being my anchor to home in Luxembourg. Shiela, my best friend and confidante, together with Micmic; Elke, my future roommate; Sofia and EJ, my go-to dinner and movie pals; and Monica and Guillaume, my fast-food drive-thru buddies. All my thanks go to Ena and Cheche for their friendship. You guys are family to me.

My research visit semester at Georgia Tech will always hold a special place in my heart because of the amazing people I met, who became an integral part of my life. Matteo, thank you for being you and for tolerating my craziness. My co-dependent family gang - Jaime, Himadri, Kevin, Alina, Bobak, and Amaya. Thanks for the colorful time we shared, and I eagerly anticipate more. Thanks also to Candice, Zixuan, Rosemary, Karina, Haoyun, Pouya, Mahya, Coco, and my roommates - Suehyun and Ryan. Of course, special thanks to Yixiao. You know how special you are to me. Thank you for sticking with me and sharing your life and time with me during my stay in Atlanta. I cannot wait for more adventures with you.

I extend my heartfelt gratitude to my best friends back home as well, starting with Kaoens Benz. Thanks for the unwavering love. To my international travel buddies - Monna and Nikka, I cannot wait for more adventures and trips with you. To Zenniel, my one-call-away buddy for everything when I am home, thank you. Florie, Reyna, Adam, Andy, and Xy - thanks for reminding me that I will always have a friend in all of you. Alyssa, Ibon, and Rupert, thanks for being the most supportive in everything I do. Of course, Kirsten and Angel, my dance buddies turned best friends, thank you for your support from different continents.

Lastly, my biggest gratitude goes to my biggest inspiration - my family. Mama and Papa, thank you for allowing me to pursue my dreams and for showing unwavering support and love. Thanks for encouraging my passion for learning and never doubting my plans. To my sisters - Ate Shang, Ate Kate, Ate Jay - you are the best sisters I could ever wish for. Thank you for having my back at all times. I wouldn't be who I am today if not for all of you. I am the luckiest uncle because of my niece, Yaz, and nephew, Don Z. Thanks for bringing color to our lives. Thank you as well to Kuya Raymond and Kuya Jonald for being a part of our family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human beings are inherent decision-makers. Daily, we make simple choices, such as determining the right bus time to ensure we arrive on time at work or school. Yet, we also deal with more complex problems, like discerning the ideal investment amount today to facilitate early retirement two or three decades down the line. The latter captures two key problem elements we deal with in this thesis - uncertainty and multiple stage planning. In the academic setting, multistage stochastic programming is commonly adopted to address multiple stage planning or sequential decision-making in the face of uncertainty, as it offers a structured framework for solving optimization problems where randomness plays a significant role. Some examples include energy production and scheduling problems, financial and investment management problems, operations and supply chain management problems, to name a few. In the literature, stochastic dual dynamic programming (SDDP) has emerged as a powerful and versatile methodology for solving multistage stochastic programs (MSPs). This thesis showcases the vast applications of SDDP in handling sequential decision-making under uncertainty across various domains. We begin in this chapter with a comprehensive introduction to MSPs, exploring their practical applications and various solution approaches. Additionally, we trace the historical development of SDDP from Benders Decomposition to its modern enhancements.

1.1 Multistage Stochastic Programs (MSPs)

Multistage stochastic programs (MSPs) are a set of programming problems dealing with sequential decisions where at least one data process is stochastic or random. MSPs, in their most general form, are formulated as

$$\begin{aligned} \min_{A_1x_1+C_1y_1\geq b_1} u_1^\top x_1 + v_1^\top y_1 + \mathbb{E}_{|\xi_1}[\min_{A_2x_2+B_2x_1+C_2y_2\geq b_2} u_2^\top x_2 + v_2^\top y_2 \\ + ... + \mathbb{E}_{|\xi_{[T-1]}}[\min_{A_Tx_T+B_Tx_{T-1}+C_Ty_T\geq b_T} u_T^\top x_T + v_T^\top y_T]] \end{aligned} \tag{1.1}$$

where $b_t = b_t(\xi_t)$ are the right-hand side vectors and $A_t = A_t(\xi_t)$, $B_t = B_t(\xi_t)$, and $C_t = C_t(\xi_t)$ are matrices of the appropriate dimensions, of which b_1, A_1, C_1 are of known deterministic values.

In this setting, the stochastic data process is revealed as time goes on. This stochastic data process $(\xi_1, \ldots, \xi_T)$ is modeled so that ξ_1 is deterministic and $\xi_2 ... \xi_T$ will be divulged over time. The history of the stochastic data process up to time t is represented by $\xi_{[t]} = (\xi_1, \ldots, \xi_t)$. The stochastic process is referred to as stage-wise independent if ξ_t is independent of $\xi_{[t-1]}$ for $t = 2, ..., T$. Otherwise, if the conditional distribution of ξ_t given $\xi_{[t-1]}$ is the same as ξ_t given ξ_{t-1} for $t = 2, ..., T$, then it is referred to as Markovian.

The decision variables in MSPs are categorized into two types: state variables x and control variables y. State variables are variables used to describe the mathematical 'state' of a dynamical system. Intuitively, the state of a system x_t at stage t sufficiently describes the system's characteristics to predict its future behaviour at time $t+1$, with respect to some uncertainty. Consequently, this state of the system flows from stage to stage until the end of the planning horizon T. Control variables y_t, on the other hand, are stage variables which the decision-maker decides upon and can control when uncertainty is revealed at stage t. Additionally, the objective function consists of the first stage deterministic function as well as nested expectation functions, or cost-to-go functions of the subsequent stages.

The original complex Problem (1.1) can be reformulated as a dynamic program to leverage the Bellman principle of optimality of the form:

$$V_t(x_{t-1}, \xi_t) = \min_{x_t, y_t} \{u_t^\top x_t + v_t^\top y_t + \mathcal{V}_{t+1}(x_t, \xi_t) : B_t x_{t-1} + A_t x_t + C_t y_t \geq b_t\}, \tag{1.2}$$

for $t = 1, \ldots, T$ where $\mathcal{V}_{t+1}(x_t, \xi_t)$ is the expected value cost-to-go function,

$$\mathcal{V}_{t+1}(x_t, \xi_t) := \mathbb{E}[V_{t+1}(x_t, \xi_{t+1}) | \xi_t], \tag{1.3}$$

with $\mathcal{V}_T \equiv 0$. The optimal value function at stage t, $V_t(x_{t-1}, \xi_t)$, is the optimal expected objective value given state (x_{t-1}, ξ_t), and assuming that optimal action will be taken at each stage t. In this setting, we assume ξ_t to be stage-wise independent or Markovian.

Finally, we define the optimal policy as

$$\pi^*(x_{t-1}, \xi_t) \in \operatorname{argmin}_{x_t, y_t} \{u_t^\top x_t + v_t^\top y_t + \mathcal{V}_{t+1}(x_t, \xi_t) : B_t x_{t-1} + A_t x_t + C_t y_t \geq b_t\} \tag{1.4}$$

for $t = 1, \ldots, T$ in set Π as the *policy* which specifies the decision to make for all possible states regardless of which state at stage t. The main goal is to solve Problem (1.1) identifying the optimal implementable policy π^* which minimizes the objective function.

1.1.1 Applications and Importance of MSPs in Real-world Problems

Real-world applications of MSPs span a wide spectrum of fields, ranging from energy (Gorenstin et al. 1992), finance (Dupačová & Kozmík 2015), operations (Nannicini et al. 2021), supply chain management (Tong et al. 2020), and natural disasters (Seranilla & Löhndorf 2023). In these domains, decisions are often subject to uncertain factors such as market and price fluctuations, demand deviations, and weather patterns, among others. The incorporation of stochastic elements in MSPs aids the exploration of multiple scenarios, capturing the exogenous uncertainties and enabling the assimilation of policies that perform well across a range of possible outcomes.

For instance, in the energy sector, MSPs have been used to solve complex problems related to renewable energy integration, specifically hydro-thermal power scheduling and generation, considering stochastic water inflows and fluctuating demand patterns. Additionally, in finance, MSPs have found significant application in portfolio optimization, asset allocation, and risk management. MSPs offer significant advantages in producing robust investment portfolios capable of withstanding market volatilities and sudden shocks.

In operations and supply chain management, where uncertainties are common, MSPs have proven instrumental in optimizing production planning, inventory control, and distribution schemes. Furthermore, in policy-making, MSPs have been exploited to undertake problems in the allocation of resources, climate change mitigation, and natural disaster preparedness.

Despite its effectiveness, the successful application of MSPs in real-world problems does encounter challenges, such as computational complexities and data requirements. Addressing these concerns remains a topic of continuing and growing research, with efforts to advance efficient algorithms and data-driven techniques.

In this thesis, we focus on an in-depth exploration of MSPs and their effectiveness in tackling difficult decision problems across various sectors through the well renowned solution method - stochastic dual dynamic programming (SDDP). With an understanding of its applications in different fields, we aim to shed light on the promising benefits and limitations of SDDP, thereby laying the groundwork for its future enhancements and broader implementation in real-world settings. Through this thesis, we endeavor to contribute to the growing body of literature surrounding SDDP, ultimately advocating for its wider adoption as a powerful methodology for addressing uncertainty and optimizing decisions in complex, dynamic environments.

1.2 Overview of Solution Techniques for Multistage Stochastic Programming Problems

In the literature, numerous methodologies have been attempted to successfully solve MSPs. Among them, the most prominent approaches include transforming MSPs into their deterministic equivalents, various decomposition strategies and scenario-tree techniques, and the current state-of-the-art method, SDDP. In the following sections, we delve into each of these methodologies to gain a better understanding of their capabilities and limitations.

1.2.1 The Deterministic Equivalent for Stochastic Programs

One common technique to solve a [multi-stage] stochastic program is to formulate and solve its *deterministic equivalent*. If the planning horizon T is aptly short and the number of scenarios S is not too large, the optimization Problem (1.1) can be solved as its deterministic equivalent, explicitly formulated as

$$\min \quad u_1^\top x_1 + v_1^\top y_1 + \sum_{t=2}^{T}\sum_{n=1}^{S} p_t^n (u_t^\top x_t^n + v_t^\top y_t^n) \tag{1.5}$$

$$\text{s.t.} \quad A_1 x_1 + C_1 y_1 = b_1 \tag{1.6}$$

$$A_t^n x_t^n + B_t^n x_{t-1}^n + C_t^n y_t^n \geq b_t^n, \qquad \forall n \in S, \forall t = 2, \ldots, T \tag{1.7}$$

$$x_t^n, y_t^n = x_t^z, y_t^z, \qquad \forall (n, z) \in S \tag{1.8}$$

$$x_t^n, y_t^n \geq 0, \qquad \forall n \in S, \forall t = 2, .., T \tag{1.9}$$

where we assume that the uncertain parameter ξ follows a (finite) discrete distribution or a discretized continuous distribution and that each scenario s at stage t occurs with probability $P(s) = p_t^s$, for all $s \in S$ and $\sum_{s=1}^{S} p_t^s = 1, \forall t \in T$. The objective function (1.5) includes first stage cost terms added with the following periods and stage scenarios with a given probability. Constraint (1.6) imposes the deterministic first stage decision. Constraint (1.7) imposes the technology and recourse matrices indexed by the sampled scenario s for every stage $t = 2, \ldots, T$. Constraint (1.8) imposes the non-anticipativity constraints - constraints that impose the condition that scenarios that share the same history until a particular stage should also make the same decisions, and do not anticipate future uncertainties. Constraint (1.9) define the domains of state variable x and control variable y.

Nonetheless, MSPs are designed to capture longer planning horizons, mirroring real-world problems. However, the size of the deterministic equivalent becomes exceedingly large and difficult to handle as it grows exponentially in the number of scenarios and the number of stages, leading to what is known as the *curse-of-dimensionality*. To address this challenge and find a computationally viable solution, researchers have utilized decomposition and scenario-tree approaches - techniques that offer a path towards converging to an optimal solution for Problem (1.1).

1.2.2 Discretization, Scenario Trees, and Decomposition Approaches

As highlighted in Section 1.1, the stochastic process ξ in the original Problem (1.1) can fall into two categories: stage-wise independent (i.e., ξ_t is independent of $\xi_{[t-1]}$ for $t = 2, ..., T$) or Markovian (i.e., the conditional distribution of ξ_t given $\xi_{[t-1]}$ is the same as the distribution of ξ_t given ξ_{t-1} for $t = 2, ..., T$). Nonetheless, it is vital to acknowledge that these stochastic processes can sometimes exhibit infinite dimensionality, especially when the underlying distributions are *continuous* or *(infinite) discrete*, rendering them intractable.

Discretization

Techniques to *discretize* stochastic processes with continuous distribution and stochastic processes with infinite discrete distribution are employed to address this intractability issue. Discretization methods transform these processes into finite-dimensional representations, enabling feasible solutions to the multistage stochastic programming problem. We can refer to a discretization of the true problem (1.1) as a *discretized problem*.

Stage-wise independent stochastic processes can be discretized by the Sample Average Approximation (SAA). SAA is a Monte Carlo simulation-based approach that generates independent identically distributed random samples from the true distribution, and the expected value function is approximated by the corresponding sample average function. Kleywegt et al. (2002) show that under some mild regularity condition, the optimal values and solutions of the SAA discretized problem converge with probability one to the true problem, when sample size N is large enough. It is important to note, however, that the required sample size N grows linearly with the dimension of the decision variables. Kleywegt et al. (2002) report that the convergence rate depends on the conditioning of the problem, which in turn tends to diminish with an increase in the number of decision variables.

For Markovian stochastic processes, two discretization approaches can be utilized - via Markov chain discretization (MC) and via autoregressive time series (TS). In MC discretization, we suppose that sample space of the Markovian random process ξ_t at stage t is Ω_t, with known ξ_1 and Ω_1 is a singleton. Furthermore, we suppose a separation of Ω_t, for $t > 1$, into a finite number of disjoint partitions K_t using optimal quantization and where the partition means $\mu_{tk}, k = 1, ..., K_t$ or quantizers are solutions of the problem

$$\min_{\mu_{t1},...,\mu K_t} \mathbb{E}_\xi \left\{ \sum_{t=2}^{T} \min_{k=1,...,K_t} ||\xi_t - \mu_{tk}||_p^p \right\}, \tag{1.10}$$

where $||\cdot||_p$ denotes p-norm with $p \geq 1$. Given solution $\mu_{t1}, ..., \mu_{K_t}$, we find a Voronoi partition of the space associated with the respective means $\mu_{tk}, k = 1, ..., K_t$, which serve as nodes to form a discrete probability lattice (a recombined scenario tree, explained further below). The objective is to create a lattice with minimal discretization error which depends on the stagewise Wasserstein distance between the Markovian data process ξ_t and its discretization. Three methods can then be used to solve 1.10 - SAA, stochastic approximation algorithm (Bally & Pagès 2003), and robust stochastic approximation approach (Nemirovski et al. 2009). Refer to Ding (2020) for further details.

Modelling Markovian stochastic processes as an autoregressive TS is also another approach. Suppose the uncertainty is only at the right hand side of the constraints, i.e. $\xi_t = b_t$. Further, suppose the process b_t is modelled as a first order autoregressive process, that is

$$\xi_t = \eta + \Phi\xi_{t-1} + \epsilon_t, \tag{1.11}$$

where vector η_t and matrix Φ_t are parameters of the first order autoregressive process, estimated from the data, and ϵ_t is the vector error process. Following the conditions above, we can view ϵ_t, now the underlying stochastic process, as stage-wise independent. Therefore, SAA can be used as the discretization method. Löhndorf & Shapiro (2019) present further details into discretizing stage-wise dependent stochastic processes and incorporating such techniques into SDDP.

Scenario Trees

Given the discretization techniques above, a common approach to represent a discretized problem is through a scenario tree - an oriented graph consisting of edges and nodes, where each node has a unique predecessor. By transforming the infinite-dimensional stochastic processes into finite-dimensional scenario trees, with appropriate discretization, we gain computational tractability and can efficiently explore the potential decision paths under uncertainty. Figure 1.1 shows an example of a scenario tree for $T = 3$ with two realizations. A special kind of scenario tree is a lattice.

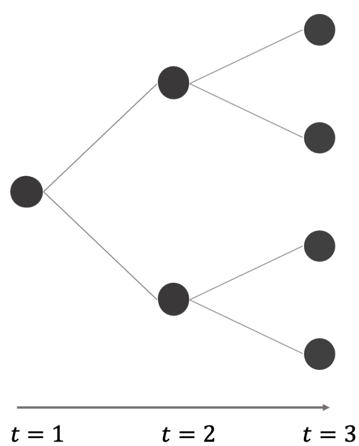


Figure 1.1: A scenario tree for $T = 3$ with two realizations per stage.

A lattice is a recombined scenario tree where each node can have multiple predecessor nodes. This is in contrast to a classic scenario tree which requires each node to have a unique predecessor. Gorski (2017) explores the performances of classic scenario trees and lattices in solving MSPs. Figure 1.2 shows an example of a scenario tree for $T = 3$ with two realizations.

The scenario tree structure lends itself well to formulation of a single large optimization problem. However, utilizing decomposition strategies, such as the dynamic programming reformulation in 1.2 and 1.3, remains a viable approach.

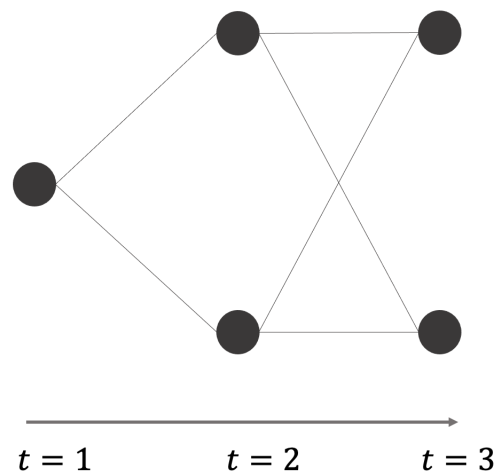


Figure 1.2: A lattice for $T = 3$ with two realizations per stage.

By breaking down the problem into smaller subproblems using the scenario tree, decomposition approaches can efficiently handle the exponential growth in the number of scenarios as the number of stages increases. The scenarios are now organized into decision stages and branches, allowing for more efficient solution methods and risk analysis, while retaining the ability to capture the essence of uncertainty in MSPs. In the literature, decomposition algorithms for solving MSPs are based on classical Lagrangian and Benders decomposition.

Decomposition Approaches

In Lagrangian decomposition (LD), the variables x and y are duplicated so that each scenario s has its own set of variables at each stage t. Non-anticipativity constraints are added, imposing that decisions at stage t may depend only on the information available until stage t. The scenario tree shown in Figure 1.1 can be modified as illustrated in Figure 1.3.

In the modified scenario tree, the root node at stage $t = 0$ in Figure 1.1 is duplicated for each scenario $s \in S$, and non-anticipativity constraints (depicted as red lines) are included to ensure that the decisions in stage $t = 0$ are consistent across all scenarios. Similarly, for stage $t = 1$, each node is duplicated once, and non-anticipativity constraints are introduced to ensure that scenarios with the same parent node in Figure 1.1 have identical decisions in stage $t = 1$. With this modified scenario tree, LD

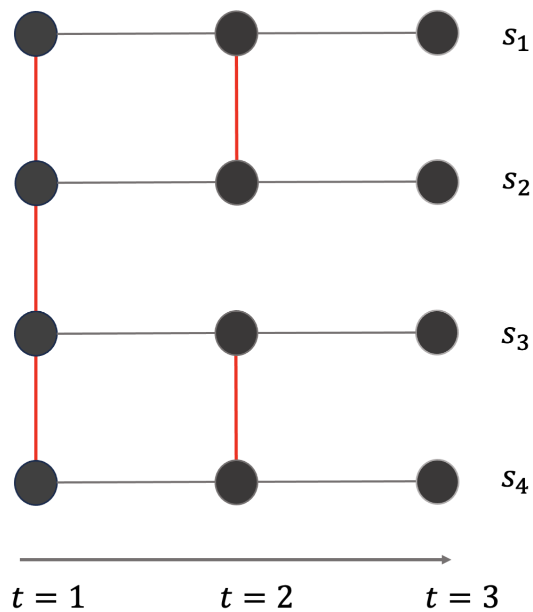


Figure 1.3: An alternative scenario tree for $T = 3$ with two realizations per stage for Lagrangian decomposition.

dualizes the non-anticipativity constraints and independently solve each scenario s. The progressive hedging (PH) algorithm, introduced in Rockafellar & Wets (1991), can be seen as a specific instance of the LD algorithm.

On the other hand, Benders decomposition (BD) techniques originated from the decomposition method developed in Benders (1962) for solving mixed-integer linear programming (MILP) problems. Subsequently, it was extended to solve two-stage SP problems, called the L-shaped method, as pioneered in Van Slyke & Wets (1969). Furthermore, this method was further extended to solve multistage stochastic programming problems, as presented in Dantzig & Glynn (1990), known as the nested Benders decomposition (NBD) algorithm.

However, like PH, NBD also has a drawback that necessitates solving an exponential number of leaf nodes in a fullspace problem. To address this limitation, SDDP was developed as a variant of NBD with scenario sampling and which allows cut sharing.

As this thesis is focused on the applications of SDDP, the entire Section 1.3 is dedicated to an in-depth discussion of the history and evolution of SDDP.

A notable distinction between LD and NBD, even though both utilize a full scenario tree, is their decomposition approach. The former decomposes MSPs by scenarios, while the latter does so by nodes. PH, on the other hand, stands out as an approach that is not paired with sampling strategies like SAA. Unlike NBD, which adds constraints to the subproblems at each node, PH replicates all scenarios. Consequently, NBD requires less memory upfront but may need more later on as there are can be a lot of additional constraints. This memory dynamic is not observed with PH. A detailed review and discussion on main types of decomposition algorithms for two-stage SPs and MSPs is presented in Escudero et al. (2017).

1.3 History and Evolution of Stochastic Dual Dynamic Programming (SDDP)

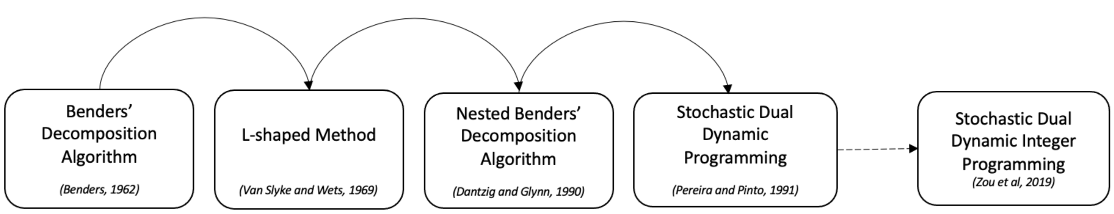


Figure 1.4: Evolution of SDDP throughout the years.

SDDP's origins can be traced back as far as 30 years before its conception in Pereira & Pinto (1991). It began with the groundbreaking Benders decomposition (BD) (Benders 1962), initially designed for mixed integer-linear programming (MILP) problems. Over time, it evolved and extended to handle two-stage stochastic programming problems through the L-shaped method (Van Slyke & Wets 1969). Subsequently, the nested Benders decomposition (NBD) (Dantzig & Glynn 1990) was introduced for multistage stochastic programming problems. These progressive developments culminated in the refinement of SDDP, offering significantly more efficient methods to solve MSPs.

Figure 1.4 shows the evolution of the SDDP methodology throughout the years. In the next sections, we take a closer look into every methodology contributing to the development of SDDP, its advancements, and its current enhancement, stochastic dual dynamic *integer* programming (SDDiP), which solves a more difficult class of problems - multistage stochastic *integer* programs (MSIP).

1.3.1 Benders Decomposition (BD)

The BD algorithm (also named as row-generation or divide-and-conquer technique) is a decomposition method developed in Benders (1962) for solving MILP problems of the form

$$\min_{x,y}\{u^\top x + v^\top y | Ax + Cy \geq b, x \geq 0, y \in \mathbb{N}_+\}, \tag{1.12}$$

where x and y are vectors of continuous and integer variables, respectively, A, C are matrices, and u, v, b are vectors having appropriate dimensions. The basic idea is to decompose the problem into two: a pure IP and a pure LP program, that may be solved successively to arrive to a solution for (1.12) by adding additional cuts (called *Benders cuts*). BD leverages the fact that two separate pure problems are more manageable to solve than the original MILP problem.

BD initiates by rewriting Problem (1.12) in terms of the y variables [master problem (MP)] as

$$\min_{y}\{v^\top y + f(y) | y \in \mathbb{N}_+\}, \tag{1.13}$$

where $f(y)$ is defined to be the optimal objective function value of

$$\min_{x}\{u^\top x | Ax = b - C\hat{y}, x \geq 0, \}, \tag{1.14}$$

for a fixed value of $\hat{y}$. Let us call (1.14) as the subproblem (SP). Now, let us associate π as the dual variables for SP. Then, the dual of SP [DSP] is

$$\max_{\pi}\{(b - Cy)^\top \pi | A^\top \pi \leq u, \pi \geq 0\}. \tag{1.15}$$

By strong duality, it holds that the objective value of SP is equal to that of the DSP.

Since the feasible region of DSP formulation does not depend on the value of y, and assuming this feasible region is not empty, it is then possible to enumerate all its extreme points $(\pi_p^1, ..., \pi_p^S)$ of the feasible region, where S are the numbers of extreme points and extreme rays of (1.15) constraints.

For a given $\hat{y}$, (1.15) can be solved by finding an extreme point π_p^n, which maximizes the value of the objective function $(b - Cy)(\pi_p^n)^\top$. Based on this, (1.15) can be reformulated as

$$\min_{\pi_p^n}\{\theta|\theta \geq (b - Cy)(\pi_p^n)^\top, z \in \mathbb{R}\}, \tag{1.16}$$

where $\theta \geq (b - Cy)(\pi_p^n)^\top$ are supporting hyperplanes called *Benders cuts*. Returning back to the reformulation of MP, (1.13) can now be reformulated in terms of θ and y variables by replacing $f(y)$ with θ as

$$\min_{y,\theta}\{v^\top y + \theta|\theta \geq (b - Cy)(\pi_p^n)^\top, y \in \mathbb{N}, \theta \in \mathbb{R}\}. \tag{1.17}$$

With all of decomposition and transformations above, the classical BD algorithm is as follows:

Step 0. Initialize a fixed value for y.

Step 1. Solve MP (1.17) and obtain a solution $(\hat{y}, \hat{\theta})$.

Step 2. Solve lower-bound $LB = v^\top \hat{y} + \hat{\theta}$.

Step 3. Solve DSP (1.15) with $\hat{y}$.

Step 4. Find an extreme point π_p^n.

Step 5. Solve upper-bound $UB = (b - Cy)(\pi_p^n)^\top$.

Step 6. If $UB \neq LB$, add *Benders cut*$\theta \geq (b - Cy)(\pi_p^n)^\top$ to MP (1.17)

Step 7. Repeat until $UB = LB$ or an iteration limit is met.

Step 8. Obtain optimal values of decision variables x, y.

1.3.2 L-Shaped Algorithm

The L-Shaped method, pioneered by Van Slyke & Wets (1969), is an extension of Benders decomposition to solve stochastic programming (SP) problems of the form

$$\min_{x\in X} u^\top x + \mathbb{E}_\xi \min\{v^\top y | T(\xi)x + Wy \geq h(\xi), y \geq 0\} \tag{1.18}$$

where ξ is a random variable having support $\Xi \subset \mathbb{R}$ and

$$X = \{x \geq 0 | Ax \geq b\}. \tag{1.19}$$

Assuming the random variable ξ is discretized with finite support $\Xi = \{\xi_1, ..., \xi_S\}$ and $P(\xi = \xi_n) = p_n$, then (1.18) is equivalent to

$$\begin{aligned} \min\ & u^\top x + \sum_{n=1}^{S} p_n v^\top y_n \\ s.t.\ & Ax \geq b \\ & T(\xi_n)x + Wy_n \geq h(\xi_n) \qquad \forall n = 1, \dots, S \\ & x \geq 0, y_n \geq 0 \qquad \forall n = 1, \dots, S \end{aligned} \tag{1.20}$$

where the constraints have an L-shaped form or a dual block-angular structure.

Furthermore, we can decompose (1.20) as a two-stage SP, where the first-stage problem is

$$\min_x \{u^\top x + \mathbb{E}[V(x)] | Ax \geq b, x \geq 0\} \tag{1.21}$$

and $V(x)$ is the optimal value of the second-stage problem

$$\min_y \{v^\top y_n | T(\xi_n)x + Wy_n \geq h(\xi_n), y_n \geq 0, \forall n = 1, \dots, S\} \tag{1.22}$$

and its dual of the form

$$\max_\pi \{\pi_n^\top (h(\xi_n) - T(\xi_n)x) | W\pi_n^\top \leq q, \pi_n \geq 0, \forall n = 1, \dots, S\} \tag{1.23}$$

with π as its dual variables.

By strong duality, it holds that the objective value of (1.22) is equal to that of (1.23). Furthermore, following Benders decomposition, we can formulate cuts (named as *optimality cuts*) from the optimal dual solution ($\hat{\pi}_n$) of (1.23) of the form

$$\theta \geq \sum_{n}^{S} p_n(\hat{\pi_n})^\top (h_n - T_n x), \tag{1.24}$$

suppressing dependence of h and T to ξ_n for simplicity. The L-shaped algorithm iterates between the first- and the second-stage problems discussed above, starting with a relaxed first-stage problem which contains no cuts. Generally, two types of cuts are added: optimality and feasbility cuts.

The classical L-shaped algorithm is as follows:

Step 0. Set $j = 0$ (v as iteration counter) and $o, f = 0$ (o, f as vectors containing feasibility and optimality cuts).

Step 1. Set $j = j + 1$. Solve the relaxation of (1.21) by removing the second-stage problem as

$$\begin{aligned}
\min \qquad & u^\top x + \theta & & (1.25)\\
\text{s.t.} \qquad & Ax = b & & (1.26)\\
& D_l x \geq d_l \qquad \forall l = 1, \dots, f & & (1.27)\\
& \theta \geq e_l - E_l x \qquad \forall l = 1, \dots, o & & (1.28)\\
& x \geq 0, \quad \theta \in \mathbb{R} & & (1.29)
\end{aligned}$$

where constraints (1.27) are feasibility cuts, constraints (1.28) are optimality cuts, and (x_j, θ_j) are optimal solution at iteration j. If no optimality cut is present, set $\theta_j = -\infty$ and do not consider in solving for x_j.

Step 2. Check feasibility of optimal solution x_j in the second-stage problem. If infeasible, for each discrete realization $\xi_n \in \Xi = \{\xi_1, \dots, \xi_S\}$, generate a feasibility cut by solving

$$\min \quad w' = e^\top (z_n^+ + z_n^-) \tag{1.30}$$

$$\text{s.t.} \quad W y_n + z_n^+ - z_n^- = h_n - T_n x_j \tag{1.31}$$

$$y_n \geq 0, \quad z_n^+ \geq 0, \quad z_n^- \geq 0 \tag{1.32}$$

where $e^\top = (1, \ldots, 1)$, until for some n, the optimal value $w' > 0$. In such a case, we let σ_j be the associated dual multipliers at iteration j and we generate a feasibility cut where

$$D_l = (\sigma_j)^\top T_n \qquad \text{and} \qquad d_l = (\sigma_j)^\top h_n. \tag{1.33}$$

Add the new feasibility cuts to (1.27) and return to Step 1. Otherwise, proceed to Step 3.

Step 3. Generate optimality cuts for feasible x_j. For each discrete realization $\xi_n \in \Xi = \{\xi_1, \ldots, \xi_S\}$, generate an optimality cut by solving

$$\min \quad w = v^\top y_n \tag{1.34}$$

$$\text{s.t.} \quad W y_n = h_n - T_n x_j \tag{1.35}$$

$$y_n \geq 0, \qquad \forall n = 1, \ldots, S \tag{1.36}$$

Let $\hat{\pi}_n$ be the dual multipliers associated with the optimal solution of second-stage problem n of type (1.35) and we generate an optimality cut where

$$E_l = \sum_{n=1}^{S} p_n \cdot (\hat{\pi}_n)^\top T_n \qquad \text{and} \qquad e_l = \sum_{n=1}^{S} p_n \cdot (\hat{\pi}_n)^\top h_n. \tag{1.37}$$

Let $w_j = e_l - E_l x_j$. If $\theta_j \geq w_j$, stop and x_j is an optimal solution. Otherwise, return to Step 1 and add the optimality cut

$$\theta_j \geq e_l - E_l x_j. \tag{1.38}$$

1.3.3 Nested Benders Decomposition (NBD)

NBD is an extension of the L-Shaped decomposition, designed to handle multistage stochastic programming (MSP) problems. The fundamental idea behind NBD is to decompose multistage problems into nested two-stage problems and recursively solve them using L-Shaped decomposition over the entire tree structure (Dantzig & Glynn 1990).

Suppose we reformulate (1.1) in a recursive form, where the first-stage problem is

$$\min_{x_1,y_1}\{u_1^\top x_1 + v_1^\top y_1 + \mathbb{E}[V_2(x_1, y_1, \xi_1)] | A_1x_1 + C_1y_1 \geq b_1, x_1, y_1 \geq 0\}, \tag{1.39}$$

the intermediate stages $t = 2, \dots, T-1$ at every node $n \in S$ is

$$V_t(x_{t-1}^n, \xi_t^n) = \min_{x_t^n, y_t^n}\{u_t^{n\top} x_t^n + v_t^{n\top} y_t^n + \mathbb{E}[V_{t+1}^n(x_t^n, y_t^n, \xi_t^n)] | A_t^n x_t^n + C_t^n y_t^n \geq b_t^n - B_t^n x_{t-1}^n, x_t^n, y_t^n \geq 0\}, \tag{1.40}$$

and the final stage T at every node $n \in S$ is

$$V_T^n(x_{T-1}^n, \xi_T^n) = \min_{x_T^n, y_T^n}\{u_T^{n\top} x_T^n + v_{T^n\top} y_T^n | A_T^n x_T^n + C_T^n y_T^n \geq b_T^n - B_T^n x_{T-1}^n, x_T^n, y_T^n \geq 0\}, \tag{1.41}$$

where x, y are state and control variables, respectively, where b are the right-hand side vectors and A, B, C are matrices of the appropriate dimensions, of which b_1, A_1, B_1, C_1 are of known deterministic values, and the random variable ξ is discretized with finite support $\Xi = \{\xi_1, ..., \xi_S\}$.

Following the formulations in BD and L-shaped decomposition, we outline NBD algorithm as follows:

Step 0. Set $j = 0$ (v as iteration counter), $o, f = 0$ (o, f as vectors containing feasibility and optimality cuts).

Step 1. Set $j = j + 1$. Solve the relaxation of (1.40) by removing the cost-to-go functions $V_t^n(x_{t-1}^n, \xi_t^n)$ as

$$\min \quad u_t^{n\top} x_t^n + v_t^{n\top} y_t^n + \theta \tag{1.42}$$

$$\text{s.t.} \quad A_t^n x_t^n + C_t^n y_t^n \geq b_t^n - B_t^n x_{t-1}^n \tag{1.43}$$

$$D_l^n x_t^n \geq d_l^n \qquad \forall l = 1, \ldots, f \tag{1.44}$$

$$\theta_t^n \geq e_l^n - E_l^n x_t^n \qquad \forall l = 1, \ldots, o \tag{1.45}$$

$$x_t^n \geq 0, \quad y_t^n \geq 0, \quad \theta_t^n \in \mathbb{R} \tag{1.46}$$

where constraints (1.44) are feasibility cuts, constraints (1.45) are optimality cuts, and $(x_{tj}^n, y_{tj}^n, \theta_j^n)$ are optimal solution at iteration j. If no optimality cut is present, set $\theta_j^n = -\infty$ and do not consider in solving for x_{tj}^n, y_{tj}^n. Start with stage $t = 1$.

Step 2.1. Check feasibility of optimal solution x_{tj}^n, y_{tj}^n in the second-stage problem. If infeasible, for each discrete realization $\xi_n \in \Xi = \{\xi_1, \ldots, \xi_S\}$, generate a feasibility cut

$$(\sigma_j^n)^\top B_t^n x_{t-1}^n \geq (\sigma_j^n)^\top b_t^n, \tag{1.47}$$

where σ_j^n are the associated dual multipliers at node n and iteration j of the dual problem to the relaxation in Step 1. If all the feasible problems have been solved to optimality, i.e. no additional cut can be added with the incumbent decision x_{tj}^n, y_{tj}^n from the preceding stage) then ascend to parent, otherwise there is then a choice:

1. Ascend step: return to the parent node at stage t–1 and re-solve (as in Step 1).

2. Descend step: move down to children node at stage $t + 1$ of any feasible sub-problems and solve (as in Step 1).

Step 2.2 If there are no infeasbile subproblems, generate optimality cut to parent node. For each discrete realization $\xi_n \in \Xi = \{\xi_1, \ldots, \xi_S\}$, generate an optimality cut by solving

$$\theta_j^n \geq (\hat{\pi}_n)^\top b_t^n - (\hat{\pi}_n)^\top B_t^n x_{t-1}^n, \tag{1.48}$$

where $\hat{\pi}_j^n$ are the dual multipliers associated with the optimal solution at node n and iteration j. If all subproblems have been solved before, then follow Descend step. Otherwise, there are two options:

1. Ascend step: return to the parent node at stage t–1 and re-solve (as in Step 1).

2. Descend step: move down to children node at stage $t+1$ of any feasible sub-problems and solve (as in Step 1).

Step 3. If $\theta_j^n \geq (\hat{\pi_n})^\top b_t^n - (\hat{\pi_n})^\top B_t^n x_{t-1}^n$, stop and we have found an optimal solution.

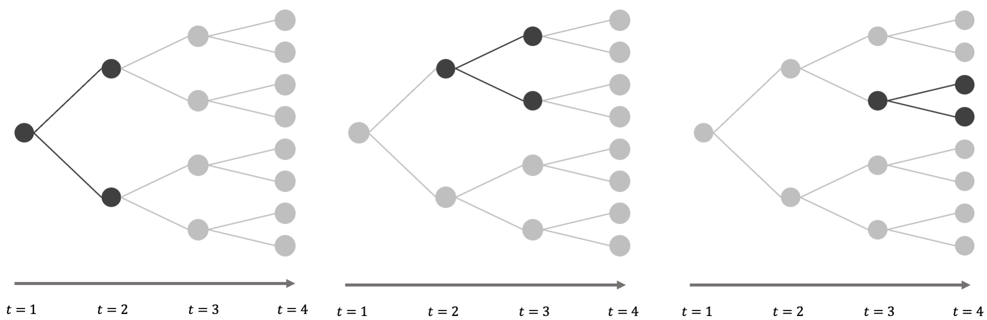


Figure 1.5: MSPs viewed as a nested series of two-stage problems. (a) left, the root node ($t = 0$) and successor nodes ($t = 1$), (b) middle, the successor node ($t = 1$) and its children ($t = 2$) and (c) right, the third-level node ($t = 2$) and its leaf-node children ($t = 3$).

Figure 1.5 shows how NBD deals with MSP as nested series of two-stage problems. Ultimately, a crucial decision in the NBD is determining when to backtrack up the tree and when to move downward. Thompson (1998) presents some *sequencing* heuristics which can be helpful in making this decision. For example, fast-forward approach, aims to traverse through the tree whenever possible. Another one is the fast-back approach which aims to move back up the scenario tree as fast as possible. A combination of both is called the fast-forward-fast-back approach. Another technique to augment teh classical NBD is the Abridged Nested Decomposition (ANBD) algorithm presented in Donohue & Birge (2006). ANBD involves sampling in the forward pass, where not all solutions of the realizations sampled in every stage are explored but only a sample of the solutions. Refer to Murphy (2013) for an in-depth tutorial to NBD.

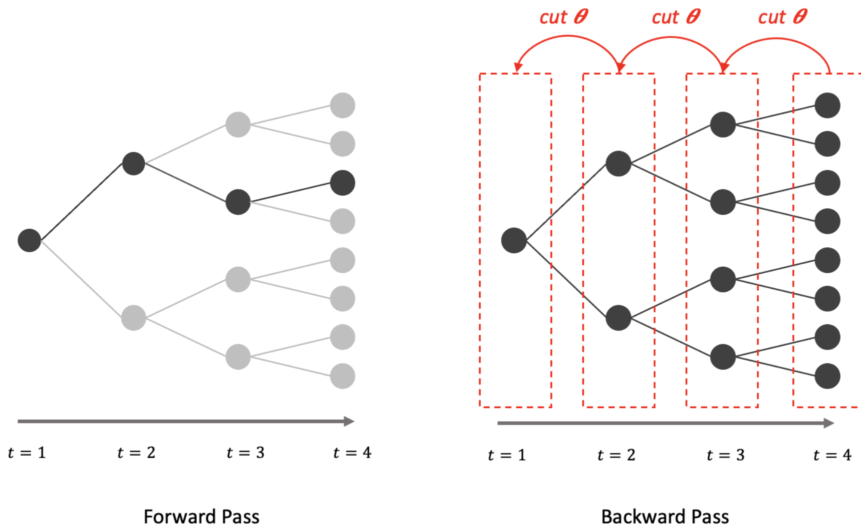


Figure 1.6: The scenario tree is traversed twice in every iteration (a) left, forward pass using a sampled scenario, and (b) right, backward pass where cuts generated.

1.3.4 Stochastic Dual Dynamic Programming (SDDP)

Stochastic Dual Dynamic Programming (SDDP) is a methodology for solving MSPs introduced in Pereira & Pinto (1991) and initially applied to address the long-term Brazilian hydrothermal power scheduling problem. Similar to the Benders and nested Benders decomposition algorithms, SDDP incorporates *Benders cuts* $\theta \in \mathfrak{V}$ – piece-wise linear functions which approximate the expected cost-to-go functions of stochastic dynamic programming. These cuts provide upper and lower bounds for the optimal solution and are derived from the dual solutions of the problem at each stage.

The SDDP algorithm has two key steps - forward simulation and backward pass. Given initial state $\bar{x}_0$, discretization Ω_t and initial approximation of the cost-to-go functions $\theta_t \in \mathfrak{V}_t$, for $t = 2, ..., T$, we begin with forward simulation. In forward simulation, we draw a sample scenario $s \in \Omega$ starting from stage $t = 2$ to final stage T. At each stage t, decisions $\bar{x}_t, \bar{y}_t$ are made based on the realized uncertainties of sample scenario $s \in \Omega$ and the current approximations $\theta \in \mathfrak{V}_t$ of the future costs.

Then we proceed with backward pass. In backward pass, we refine the approximations of the cost-to-go functions. Starting from stage T and working backward to the first stage $t = 1$, we compute the expected cost-to-go based on the decisions made in the forward simulation. For each stage t, we solve each subproblem j to determine the optimal decisions x_t, y_t and the corresponding optimal future cost. Each subproblem j uses the current approximation of the cost-to-go functions $\theta(x_t) \in \mathfrak{V}_{t+1}(x_t)$, so that each subproblem j is essentially a linear program. We obtain the optimal value α_{tj} and dual solutions β_{tj} of the time-coupling constraints which correspond to $z_t = \bar{x}_{t-1}$ to create new cuts of the form $\theta(x) \geq \alpha_t + \beta_t(x - \bar{x}_{t-1})$, and update the approximation of the cost-to-go functions accordingly. Leveraging the convexity of the cost-to-go functions, these cuts can be shared across different scenarios as they are valid underestimators for the entire cost-to-go function in every stage t.

We do this iteratively until the lower bound and the policy value (upper bound) converge or a stopping criteria is met. Figure 1.6 shows graphical scenario tree representation of forward simulation and backward pass. Algorithm 1 shows the classical SDDP algorithm implemented by Pereira & Pinto (1991), as taken from Ding (2020).

One of the significant advantages of the SDDP algorithm is that it avoids the curse-of-dimensionality associated with dynamic programming, as state discretization is not necessary. Through a few iterations, the upper and lower bounds converge, providing an optimal solution (or a near-optimal solution, based on a given tolerance). This convergence makes SDDP an efficient and effective approach for solving large-scale MSPs.

1.3.5 Enhancements of SDDP

The classical SDDP algorithm rests upon several assumptions for its effective implementation. However, numerous refinements and advancements have emerged in recent years, enhancing the original method. For example, the classical SDDP assumes the MSP has a finite planning horizon. Shapiro & Ding (2020) shows how to solve infinite planning horizon problems with what they call the *periodic* SDDP. Another assumption of the classical SDDP is the block-diagonal structure of the constraints of the MSP, i.e. only consecutive stages can be linked by constraints. Shapiro (2011) and Löhndorf & Shapiro (2019) show that this can be overcome by

Algorithm 1: Stochastic Dual Dynamic Programming

initialization: $i = 1$, LB = -∞
Given discretization $\Omega_t = u_{tj}, v_{tj}, A_{tj}, B_{tj}, C_{tj}, b_{tj1\le j\le N_t}$, $t = 2, ..., T$
Given initial value $\bar{x}_0$
Given initial approximation of value functions: $\mathfrak{V}_t^0(.) = \theta : \theta(.) \geq l_t$, $t = 2, ..., T$
while *no stopping criterion is met* **do**
  (Forward Simulation)
  for $t = 1, ..., T$ **do**
    if $t > 1$ **then**
      draw a sample $(u_{tj}, v_{tj}, A_{tj}, B_{tj}, C_{tj}, b_{tj})$ from Ω_t
    $\bar{x}_t, \bar{y}_t =$ argmin $u_t' x_t + v_t' y_t + \mathfrak{V}_{t+1}^i(x_t) : A_t x_t + B_t z_t + C_t y_t \geq b_t, z_t = \bar{x}_{t-1}$
  Backward Pass
  for $t = T, ..., 2$ **do**
    for $j = 1, ..., N_t$ **do**
      Solve $\min\{u_{tj}' x_t + v_{tj}' y_t + \mathfrak{V}_{t+1}^i(x_t) : A_{tj} x_t + B_{tj} z_t + C_{tj} y_t \geq b_{tj}, z_t = \bar{x}_{t-1}\}$
      Get optimal value α_{tj} and a dual solution β_{tj} corresponding to $z_t = \bar{x}_{t-1}$
    $\alpha_t := \frac{1}{N_t} \sum_{j=1}^{N_t} \alpha_{tj}$, $\beta_t := \frac{1}{N_t} \sum_{j=1}^{N_t} \beta_{tj}$
    $\mathfrak{V}_t^i \leftarrow \theta \in \mathfrak{V}_t^{i-1}$, $\theta(x) \geq \alpha_t + \beta_t(x - \bar{x}_{t-1})$
  Policy Value $= \sum_{t=1}^{T} [\gamma^{t-1}(u_t' \bar{x}_t + v_t' \bar{y}_t)]$
  LB = min $u_1' x_1 + v_1' y_1 + \gamma \mathfrak{V}_2^i(x_1) : A_1 x_1 + B_1 z_1 + C_1 y_1 \geq b_1, z_1 = \bar{x}_0$
  $i = i + 1$

an appropriate state-space expansion. Risk-neutrality assumption of MSPs has also been overcome in Guigues & Römisch (2012b), Guigues & Sagastizabal (2013), and Shapiro et al. (2013a). Their contributions extend towards encompassing various risk-averse problems. Another assumption of the classical SDDP is that the realization of the stochastic data process is finite. Guigues (2016) shows convergence analysis over infinite support to overcome such assumption. Finally, stage-wise independence of the probability distribution of the stochastic process is one of the biggest assumptions of the classical SDDP. This assumption creates a unique cost-to-go function per stage, which is efficiently assessed through sampling-based methods, like SAA (Shapiro 2011), and Philpott & Guan (2008) present a proof of its convergence. Nonetheless, Löhndorf

& Shapiro (2019) introduce two approaches to overcome this assumption: (1) the modelling of the random data process as an *auto-regressive time series* by adding state variables to the model and (2) the use of Markov chain discretization of the random data process.

Stochastic Dual Dynamic Integer Programming (SDDiP)

Since the release of SDDP in 1991, it has been extended to many techniques and applications. Its extension to tackle multistage stochastic integer programs (MSIPs), Stochastic Dual Dynamic Integer Programming (SDDiP), stands out as the primary computationally tractable method to obtain both a statistical lower bound and an optimal policy.

As with SDDP, SDDiP involves a reformulation of the MSIP into a dynamic program. Key to this reformulation is the approximation of the cost-to-go function by a convex piece-wise linear function which effectively overcomes the curse-of-dimensionality often associated with traditional dynamic programming methods, making the problem computationally tractable.

Under specific assumptions, SDDiP guarantees convergence to an optimal policy. First, as with SDDP, SDDiP assumes that the stochastic parameters of the MSIP have a discrete number of realizations. This discretization assumption facilitates the handling of uncertainty, allowing for efficient exploration of the feasible solution space over multiple scenarios. Second, all time-coupling state variables, which enter the state space of the dynamic program, should be binary variables. By incorporating binary variables, SDDiP ensures convergence to an optimal solution via Lagrangian relaxation.

In summary, SDDiP is an effective approach to tackle the complexities of the MSIPs by formulating it as a dynamic program and leveraging piece-wise linear approximations of the expected cost-to-go functions. Its convergence to an optimal policy, under the specified assumptions, makes it a valuable tool. With its ability to obtain both lower bounds and optimal solutions, SDDiP proves to be a versatile and efficient technique for decision-making under uncertainty in a wide range of real-world applications.

1.3.6 Contributions and organization

In Chapter 1, we delve into the formal introduction of MSPs, highlighting their significance and applicability in addressing real-world problems. This chapter also offers a panoramic view of solution techniques for MSPs, ranging from tackling its deterministic equivalent to employing the SDDP methodology. Additionally, we trace the history and evolution of SDDP, exploring from its inception in Benders decomposition to its modern advancements.

In Chapter 2, we take a closer look into the multifaceted applications of SDDP as documented in existing literature. An emphasis is laid on showcasing the extent of SDDP's adoption across different domains. We discuss further the predominant utilization of SDDP in the energy realm, with a spotlight on hydro-thermal power production planning. We also outline compelling arguments for the prominence of this specific application. Finally, we move forward with some recommendations of other domains and specific applications where SDDP can be potentially applied.

Chapter 3 introduces two valuable contributions: MSPLib, an open-source library of problems and MSPFormat, a standardized data format designed for benchmarking SDDP. MSPLib aims to facilitate the evaluation of computational performance among different SDDP implementations. It offers a wide array of instances, from real-world problems to synthetic variations with varying complexities. By incorporating MSPFormat into the library, a unified and consistent representation of MSPs is provided, further enhancing their usability and transferability.

Chapter 4 introduces two pivotal contributions to the stochastic programming society: MSPLib and MSPFormat. MSPLib is an open-source library of problems, designed to streamline the assessment of computational efficiency of different SDDP implementations. Its repository ranges from real-world to synthetic problems of varied complexities. Meanwhile, the integration of MSPFormat ensures a uniform and coherent representation of MSPs. This standardization augments the usability and interchangeability of the problems. We close this chapter by presenting numerical results of benchmarking three SDDP implementations commonly used in the literature.

Moving on to Chapter 5, we showcase an MSP application to the optimal location of COVID-19 vaccine facilities under the threat of natural disasters. We introduce a new algorithm, named *shadow price approximation* (SPA), which aims at approximating the shadow price of opening flood-prone vaccine facilities by tuning the parameters of a linear value function approximation which is present in the objective function of base optimization model. We also compare the performance of SPA against stochastic dual dynamic integer programming (SDDiP). The chapter closes with a detailed account of this model's application in two cities of a developing country.

In Chapter 4, we introduce a general model for a novel problem class named the multistage stochastic facility location problem under facility disruption uncertainty (MSFLPD). This new class extends the classical stochastic *capacitated* facility location problem to handle uncertainty arising from facility disruptions. We then compare two solution algorithms, SDDiP and SPA, in a numerical investigation solving famous ORLib data instances.

Finally, we close with a conclusion and discussion of future work in Chapter 6.

Chapter 2

A Survey on the Applications of Stochastic Dual Dynamic Programming and its Variants

In Chapter 1, we are introduced to the world of SDDP - a widely recognized methodology for solving large-scale multistage stochastic linear programming (MSLP) problems. In this chapter, we delve deeper into the multitude of applications of SDDP. The survey aims to contribute to the literature on SDDP within the realm of applications. We systematically identify and analyze the various domains where SDDP has been successfully employed to tackle MSP problems, with a particular focus on real-world problems afflicted by the so-called *curse-of-dimensionality*. Furthermore, we investigate the factors that have facilitated or hindered the adoption of SDDP in specific application areas, shedding light on the limitations and potential barriers to its widespread utilization.

2.1 Introduction

Modelling real-world problems through multi-stage stochastic programming (MSP) presents considerable challenges, particularly when navigating extensive planning horizons and uncertainty scenarios. Such problems frequently run into the infamous *curse-of-dimensionality*, a phenomenon describing the exponential growth in computational requirements as the dimensionality of a certain problem increases.

Enter Stochastic Dual Dynamic Programming (SDDP). Introduced in the seminal work by Pereira & Pinto (1991), SDDP proposes a powerful solution to bypass many of the obstacles associated with MSP problems, particularly the curse-of-dimensionality. Rooted in the principles of dynamic programming and duality, SDDP decomposes an MSP problem, iteratively approximating the expected cost-to-go functions without needing to tackle the entire problem space at once. Consequently, what previously seemed intractable becomes more computationally feasible, allowing practitioners to address large-scale MSP problems in various sectors, from energy planning (Gorenstin et al. 1992) to supply chain optimization (Tong et al. 2020).

The conception of SDDP marked a paradigm shift in the approach to sequential decision-making under uncertainty. Previously, dealing with uncertainty often required, on one hand, simplification and even relaxation of the problem. This potentially leads to suboptimal solutions or ones that failed to mirror the full complexity of a real-world problem. On the other hand, solving the entire scale of MSPs through a tedious exploration and exploitation of the full scenario tree results to expensive computational efforts. However, with SDDP, it became possible to retain much of the intricacies of MSPs while still navigating through the expanse of uncertainties and stages.

Nevertheless, while SDDP is a powerful methodology, it still comes with its challenges and limitations. Affected by factors ranging from computational restraints to domain-specific nuances, SDDP still has a long way to solve even larger problem instances mirroring real-world cases. Füllner & Rebennack (2021) present a more in-depth tutorial-type review of the various improvements and enhancements of SDDP.

To appreciate the relevance of SDDP even further, it is imperative to consider the current challenges faced by different industries worldwide. With increasing globalization and competitiveness, where effects of decisions made today can cascade and amplify over time, industrial applications highlight the invaluable utility of SDDP. For instance, consider the domain of hydro-thermal power system planning, the premier application of SDDP. Governments and private companies across the world harnessing this renewable energy leverage SDDP to aid in their decision-making processes (Maceiral et al. 2018).

Furthermore, as the eminence of SDDP continues to advance, its applications have prospered across a wide array of applications. Nonetheless, a significant proportion of the work in the literature is primarily focused within the field of energy planning. This survey endeavors to explore the breadth and depth of the applications underlining the significance of SDDP and its growing relevance in solving MSP problems.

Through this review, we aim achieve two goals. First, to provide a survey of the practical applications of SDDP in the literature. We explore the literature through the lens of these different applications. Second, we aim to present a compelling discussion on the factors that have facilitated and, possibly, hindered to the adoption of SDDP in specific application areas shedding light on the limitations and potential barriers to its widespread utilization. As we delve deeper into the sphere of SDDP and its applications, it is significant to understand not just where and how SDDP shines, but also where it can falter and require supplemental schemes. This perspective is crucial for both practitioners and researchers aiming to augment the boundaries of solving MSP problems.

The remainder of the paper is organized as follows. Section 2.2 provides an overview of the history, evolution, and technicalities of SDDP. Section 2.3 covers the literature selection process. Section 2.4 constitutes one of the contributions of the paper and discusses the main statistical findings from our literature survey. Section 2.5 discusses various factors of adoption of SDDP in specific application areas. Section 2.6 contains some concluding remarks and provides guidance as to future research directions.

2.2 Background

Before the inception of SDDP thirty years ago, the method drew upon a rich lineage of inspirations and developmental precursors. In this section, we will briefly revisit three pivotal stepping-stone methods that laid the foundation for the evolution of SDDP: (a) Benders decomposition, (b) L-shaped method, and (c) nested Benders decomposition.

2.2.1 Overview of the History and Evolution of SDDP

The origins of SDDP can be traced back from the groundbreaking method developed by Jacques Benders (Benders 1962) - Benders decomposition method. Benders decomposition method is a decomposition method for solving mixed integer programming problems - optimization problems where some variables are constrained to take real values and others only limited to integer values. The basic idea is to decompose the problem into two: a pure linear programming problem [called master problem (MP)] and a pure integer programming [called subproblem (SP)]. Benders decomposition proceeds by solving MP and SP iteratively to arrive to a solution for the overall problem by adding *Benders cuts*, some feasibility or optimality information from the sub-problem to the master problem. The leverage of the algorithm is the fact that two separate decomposed problems are more manageable to solve than the original mixed problem. Notwithstanding the number of iterations to solve each subproblem, this technique is still relatively quicker to arrive at the solution than solving the large mixed integer programs.

Seven years later, Van Slyke & Wets (1969) expanded on this powerful algorithm and introduced the L-Shaped method. L-shaped method is an algorithm to solve stochastic programming problems – optimization problems where some, if not all, of the problem parameters take stochastic or uncertain values. For L-Shaped method, two-stage stochastic problems or stochastic linear problems with recourse are of interest where an initial decision is made succeeded by a second decision which allows remedial (recourse) action after an event have been observed to happen. Similar to Benders' decomposition, L-Shaped method is based on adding additional constraints (or cuts) to the original (in this case, first-stage) problem. These cuts ensure feasibility of the first-stage decision to the second-stage problem and assurance to obtain an optimal solution.

Finally, the nested Benders decomposition (NBD) algorithm from Birge (1985) is an extension to L-Shaped decomposition such that the former proposes an algorithm for multi-stage stochastic programming problems. The simple idea behind nested Benders' decomposition is the notion of decomposing multi-stage problems into nested two-stage problems and recursively, then solving these with L-shaped method over the entire scenario tree structure. With the initial decision of the first-stage (root node), succeeding second-stage sub-problems will be solved. Thereupon, the initial decision will be viewed as a constant and the following stage nodes will become the first-stage (root node) of its own problem. This will continue further until the end of the planning horizon. As with L-shaped method, NBD solve the sub-problems with the current initial decision and use these solutions to create feasibility and optimality cuts. This means that parts of the problem of whichever stage of the tree can be solved as long as initial decision of a preceding node exists. The drawback of NBD is that the scenario tree grows exponentially in the number of stages and it becomes computationally expensive for large instances.

2.2.2 SDDP Methodology and some extensions

SDDP is a straightforward methodology for solving MSP problems by approximating the expected cost-to-go value functions of stochastic dynamic programming using piece-wise linear functions. Similar to Benders decomposition algorithm, SDDP algorithm incorporates *Benders cuts* – which are approximations of the second stage (or the next consecutive stages in the case of a multi-stage problem) – into the first stage (precedent or root) problem. These cuts provide upper and lower bounds for the optimal solution and are derived from the dual solutions of the problem of the succeeding stages.

As mentioned, one of the significant advantages of SDDP is its avoidance of the curse-of-dimensionality associated with dynamic programming. With a few iterations, the upper and lower bounds converge, providing an optimal solution (or a near-optimal solution, based on a given tolerance). This convergence makes SDDP an efficient and effective approach for solving MSPs (Philpott & Guan (2008), Girardeau et al. (2015), and Guigues (2016)). The idea of SDDP stems from NBD but it includes the concept of sampling. In each iteration, not all the scenarios of the stochastic data process scenario tree will be explored but only a sample of those scenarios will be drawn and used. This will significantly reduce the number of linear programs to be solved.

The vanilla SDDP algorithm rests upon several assumptions for its effective implementation. However, numerous refinements and advancements have emerged in recent years, enhancing the original method. As an illustration, an extension to the vanilla SDDP is the incorporation of stage-wise dependent uncertainty of the MSP. The original SDDP limits the problem in consideration to have its random data process stage-wise independent - the case where the uncertainty in different stages do not depend on each other. In this case, the respective value functions of the dynamic programming equations are independent of the data process. Löhndorf & Shapiro (2019) introduce two approaches: (1) the modelling of the random data process as an *auto-regressive time series* by adding state variables to the model and (2) the use of Markov chain discretization of the random data process. Additionally, the pioneering works of Guigues & Römisch (2012b), Guigues & Römisch (2012a), and Shapiro et al. (2013a) stand out as the initial trio of papers that expanded upon the original assumption of risk neutrality in MSP problems for SDDP. Their contributions extend towards encompassing various risk-averse problems.

Furthermore, SDDP has been extended to tackle multistage stochastic integer programs (MSIPs). Stochastic Dual Dynamic Integer Programming (SDDiP) stands out as the primary computationally tractable method to obtain both a statistical lower bound and an optimal policy (Zou et al. 2019). As with SDDP, SDDiP involves a reformulation of the MSIP into a dynamic program. Under specific assumptions, SDDiP guarantees convergence to an optimal policy. Like SDDP, SDDiP assumes that the stochastic parameters in the MSIP follow a discrete distribution. A key ingredient to SDDiP is the binarization of all time-coupling state variables, which ensures convergence.

Finally, the evolution of SDDP remains an ongoing development, marked by continuous refinement and enhancement. Recent efforts are focused on finding of optimal policies quickly and extending its applicability to a wider array of real-world problems.

2.3 Literature Selection and Analysis

We conducted a comprehensive database search for papers that apply, utilize, develop, or enhance SDDP to primarily address optimization problems. The databases

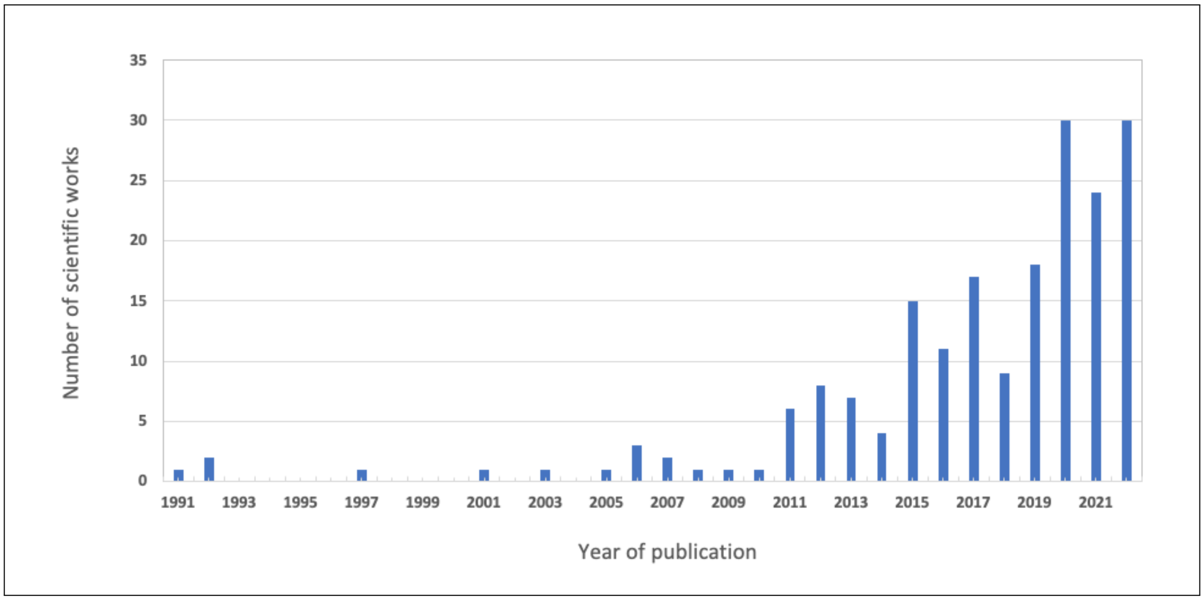


Figure 2.1: Evolution of number of published works over time.

examined included Emerald Insight, Scopus, JSTOR, and Web of Science which included published journal articles until December 2022. We employed the keywords "stochastic dual dynamic programming," "SDDP," "stochastic dual dynamic integer programming," and "SDDiP." Publications not written in English, as well as conference proceedings and book chapters, were excluded from our consideration. Our approach to filtering relevant publications began with an evaluation of the title. If deemed pertinent, we proceeded to assess the abstract, followed by the introduction and conclusion. Only if the paper still appeared relevant after this thorough review did we analyze its entirety. This method led to the identification of 186 significant publications since the release of the seminal paper on the subject in 1991. The collective works surveyed are graphically illustrated in Fig. 2.1. These papers span 61 distinct journals, which are itemized in Table 2.1, covering a spectrum of research domains. We further dissected the significance of these scientific contributions by examining three primary research questions:

1. In which specific optimization problems is SDDP employed, and within which application sectors (e.g., energy, finance, operations management)?

2. What are the key contributions of the research to SDDP (be it theoretical, methodological, or other)?

3. How valuable is SDDP in efficiently resolving the identified problem?

Our survey unveiled several distinct fields of application. Predominantly, a significant portion of the papers concentrates on problems within the energy sector. The Brazilian hydro-thermal power system is a primary focus in this area. This problem manifests in various iterations — different planning horizons, diverse geographical locations, and a range of different random data processes, to name a few. A secondary line of literature navigates challenges in the financial sector, such as portfolio optimization and asset allocation. Operations management is another prominent domain addressed, with a unique set of scientific works exploring problems like inventory, lot sizing, and production. Additionally, certain studies venture into more specialized areas, delving into matters related to disaster management and mining.

2.4 Literature Review and Findings

In this section, we provide an extensive analysis of various applications of SDDP as found in the literature. This examination is structured into distinct subsections that correspond to different fields of application. Notably, the field of energy applications, with its broad scope, is itself subdivided into various domains for a more comprehensive discussion. The distribution of works across these application areas is illustrated in Figure 2.2. While some studies are inherently dedicated to a specific application field, a significant number of papers span multiple domains. For instance, a work might address an energy system application while also incorporating important financial aspects. In such instances, we ensure that these papers are discussed and integrated into all relevant subsections.

2.4.1 Energy

With reference to Figure 2.2, it is evident that the energy sector constitutes a majority of the research works focused on SDDP. This section explores diverse domains within the

Table 2.1: Classification of the papers according to the different journals.

Journal	Number
European Journal of Operational Research	19
IEEE Transactions on Power Systems	14
Mathematical Programming	10
IEEE Transactions on Sustainable Energy	9
Journal of Water Resources Planning and Management - ASCE	9
International Journal of Electric Power and Energy Systems	9
SIAM Journal on Optimization	9
Electric Power Systems Research	7
Water Resources Research	7
Computational Management Science	7
INFORMS Journal on Computing	5
Annals of Operations Research	5
Computational Optimization and Applications	5
Operations Research Letters	5
Optimization and Engineering	4
Energy Policy	4
Operations Research	4
Energies	4
Others	50

energy sector - applications related to hydro-thermal power system, problems associated with short-term economic dispatch and capacity expansion, the integration of various renewable energy sources, and optimization problems arising in energy markets, trading, and investments. Additionally, the integration of different energy storage systems and the intricacies of grid management are also discussed thoroughly.

Hydro-thermal Power Production Problem

The hydro-thermal power production serves as a flagship problem application of the development of SDDP. The central question in this domain revolves around the optimal coordination between hydroelectric and thermal power generation sources to meet demand while minimizing costs and maintaining reservoir storage, considering the unpredictable nature of water inflows into hydroelectric reservoirs, and the fluctuating

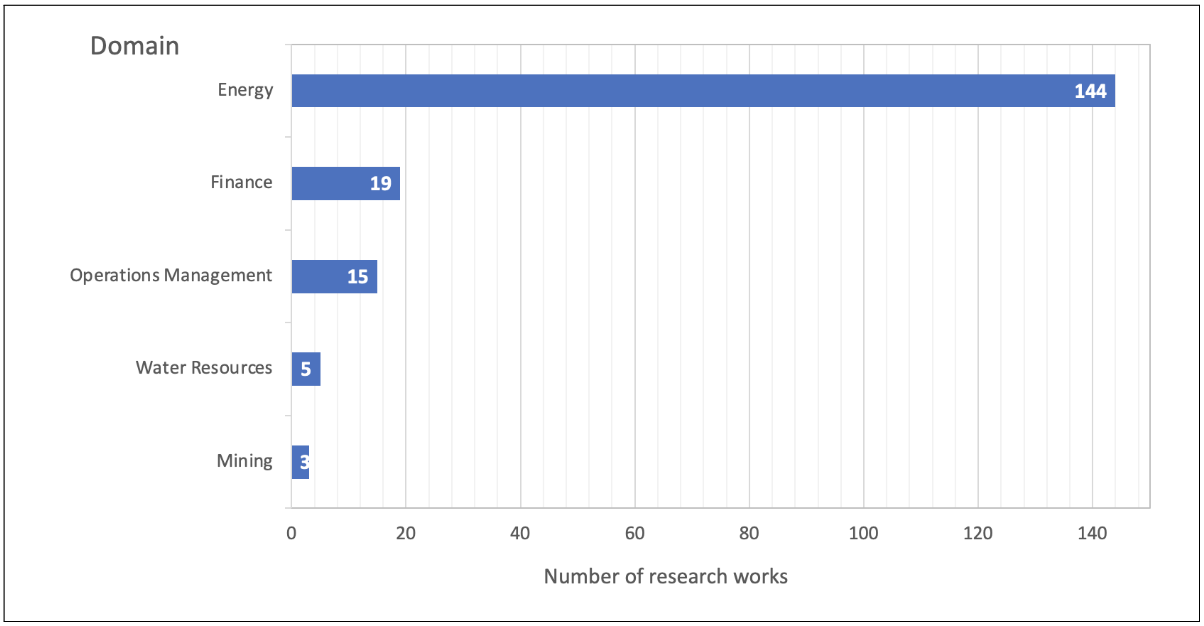


Figure 2.2: Distribution of research works across different application areas.

costs of thermal power generation. Managing these resources is crucial not just for economic efficiency but also for maintaining overall system reliability. Given the complexity arising from the multitude of reservoirs, interconnected systems, and uncertainty of inflows, traditional deterministic methods often fall short in providing practical solutions. This is where SDDP becomes invaluable.

Since the introduction of SDDP over 30 years ago, with the Brazilian interconnected hydro-thermal power system serving as its premier case study, over 100 research publications have been dedicated to addressing the hydro-thermal power production problem using this method. The variations encountered within this domain typically stem from differences in planning horizons, modifications in the uncertainty processes, and diverse geographical contexts. Figure 2.3 presents the distribution according to various geographical locations, while Figure 2.4 illustrates the distribution of research works based on distinct planning horizons. A detailed breakdown, encompassing planning horizons, geographical areas, and uncertainty processes related to the hydro-thermal power production problem, can be found in Tables 2.2 and 2.3 in the Appendix. For an in-depth discussion in this section, we categorize the literature

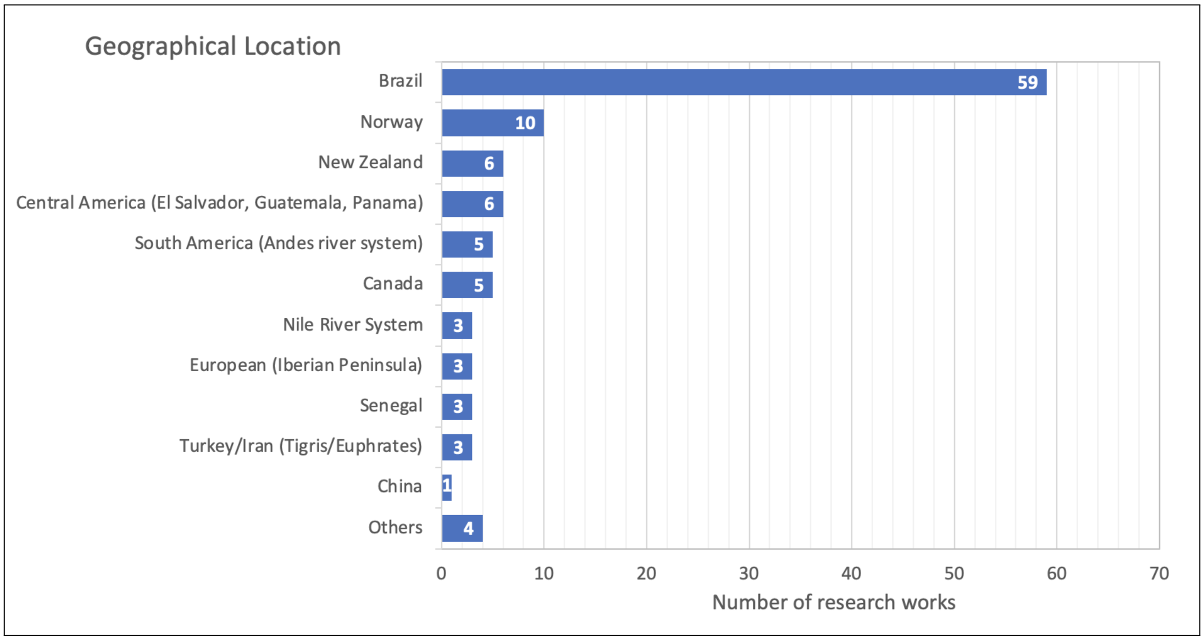


Figure 2.3: Distribution of research works across different geographical locations for hydro-thermal power production problem.

based on temporal scales, which offer more consistent comparisons due to the similar objectives of the problems addressed. The distinct operational planning horizons are as follows: (1) long-term operational planning, covering a span of 3 or more years with monthly decision epochs; (2) medium-term operational planning, typically lasting 1 to 2 years with weekly or monthly decision epochs; and (3) short-term operational planning, extending over hours or days.

Long-term Operational Planning. Long-term operation planning primarily focuses on multi-annual horizons (3 or more years), usually with a monthly decision epoch, ensuring that the system is resilient against seasonal variations and long-term climate changes. The main objective of the solving this problem is to ensure strategic reservoir storage decisions, ensuring that water resources are adequately maintained to meet future demands.

Furthermore, with the seasonal uncertainty of water inflows, long-term planning ensures that reservoirs maintain optimal levels especially in handling drought and surplus events. This helps in preventing potential water shortages in dry periods and

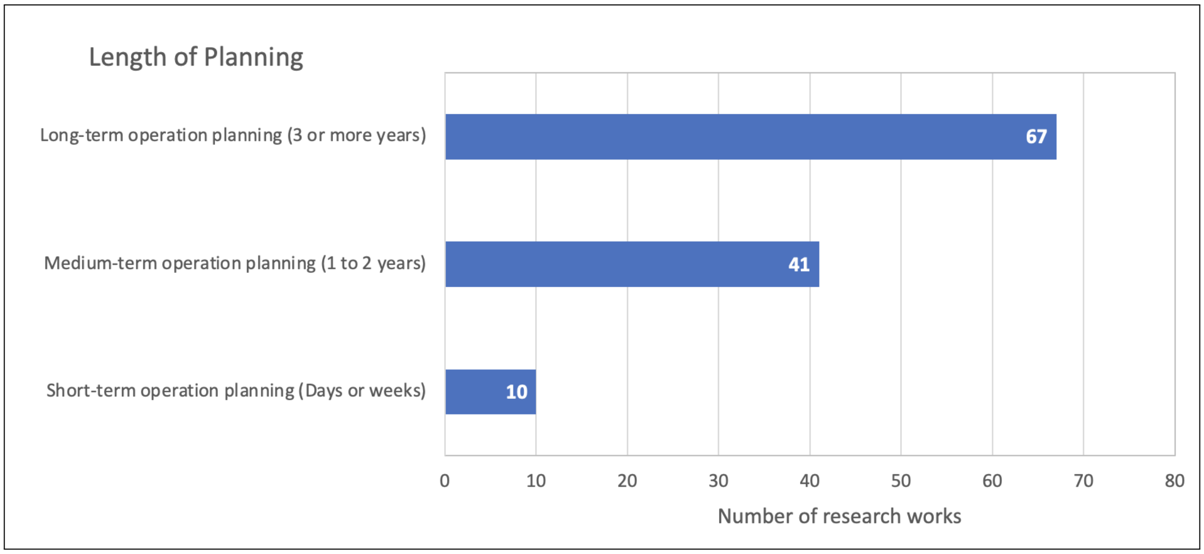


Figure 2.4: Distribution of research works across different planning horizons for hydro-thermal power production problem.

avoids overflow during wet periods. This is the reason that all, if not majority, of research works are devoted to understanding the inflow uncertainty. Sampling from the historical values is one of the easy go-to methods in choosing realizations of inflow uncertainty. Nonetheless, with the understanding that inflow has seasonal patterns, forecasting techniques were also adapted to understand this random process. Various aut-oregressive processes and the inclusion of non-linearities are among the prominent techniques employed in this end.

Long-term planning also involves prediction of future energy demand. Some factors that affect demand include population growth, industrial and local development, and economic trends. Forecasting demand correctly aids in ensuring the optimal operation of the system without frequent and costly re-adjustments. For example, works in Shapiro & Cheng (2021), Raby et al. (2009), and Sauma et al. (2011) include demand uncertainty in their models.

A phenomenon in the long-term operation planning of hydro-thermal power production worth mentioning is the *end-of-horizon* effect. Policies derived from optimization models do not necessarily ensure a sustainable and reliable energy supply beyond the designated planning horizon. The reason being, optimal policies typically exhaust all energy reserves in the reservoirs at the end of the planning horizon. Thus, most works consider doubling the planning horizon (using 10 years instead of 5 years) in order to hedge from this phenomenon.

As mentioned, a more detailed breakdown, encompassing planning horizons, geographical areas, and uncertainty processes related to the hydro-thermal power production problem, can be found in Table in the Appendix.

Medium-term Operational Planning Medium-term operation planning serves as a link between the expansive outlook of long-term operational planning and the immediate requirements of short-term operational planning. Typically encompassing a 1 to 2-year planning horizon, medium-term operation planning operates with a monthly or weekly decision epoch, allowing for a more granular view of the system's operations than long-term planning.

As hydro-thermal power systems cope with the unpredictability of inflows and other uncertainties, medium-term operation planning offers flexibility in accommodating issues that long-term operation planning usually overlooks. One example is the incorporation of planned maintenance schedules to ensure that energy production and distribution assets are kept at optimal levels. Additionally, medium-term operation planning is significant in response to unexpected disturbances in the system like technical malfunctions, sudden demand surges, or other external factors.

Furthermore, works for medium-term operation planning often include the uncertainty of prices and costs. This provides producers, not only to be economically profitable with their operations, but also to perform optimally in energy trading. For example, the works in Kristiansen (2006), Löhndorf et al. (2013), de Matos et al. (2015), Helseth et al. (2016), Rebennack (2016), Hjelmeland et al. (2018), Hjelmeland et al. (2019), and Borges et al. (2022) include uncertainty in prices and costs for the medium-term operation planning of hydro-thermal power system.

Water Resource Management Given the pivotal role of water resources in hydro-thermal power systems, making strategic reservoir storage decisions is crucial - ensuring that water reservoirs are optimally maintained, without compromising the ability to meet future energy demands. Hence, in the operational planning for hydro-thermal power production, the effective and sustainable management of these water resources is important.

Various research works have been directed towards understanding and optimizing water resource management, particularly in the context of major river basins integral to hydro-thermal power systems. Notably, African river basins have been the subject of study, with Rougé & Tilmant (2016) looking at the Zambezi river basin, while Raso et al. (2017) focus on Mananatili river basin. In Europe, the Iberian peninsula, a significant hydro-thermal hub, is the subject of study in Pereira-Cardenal et al. (2016) and Macian-Sorribes et al. (2017). Shifting to the Middle East, Rougé et al. (2018) and Tilmant & Kelman (2007) explore the Tigris-Euphrates river basin considering its pivotal role in the region's energy landscape. Moreover, Treistman et al. (2020) propose innovative synthetic inflow scenario generation techniques tailored to the El Niño–Southern Oscillation, offering insights for the Brazilian hydro-thermal power system.

Short-term Operational Planning - Economic Dispatch and Capacity Expansion The smallest coverage of time horizon is the short-term operation planning which zeroes in on daily or hourly scales. It focuses on fine-tuning generation schedules to respond to daily demand variations. Two domains are included in this area - economic dispatching and capacity expansion planning. The difference between the two lies in the objectives of the operation and planning of any power systems. Economic dispatch focuses on the short-term optimization of existing power generation units to meet the current electricity demand at minimum cost. It aims to determine the optimal output levels of each generator to match demands, while considering different factors such as fuel costs, transmission constraints, and generator constraints. On the other hand, capacity expansion planning deals with determining optimal additions or retirements of power generation units to meet future demand.

The following works have made notable contributions to various aspects of short-term economic dispatch problems: Chabar et al. (2006), Papavasiliou et al. (2017), Stüber & Odersky (2020), Ding et al. (2021), Lan et al. (2022), and Pacaud et al. (2021). Moreover, economic dispatch has been extended to incorporate additional system dynamics. For instance, Lu et al. (2019) integrate energy reserves, while Street et al. (2017) delve into ancillary services. Additionally, Fatouros et al. (2017) explore the operation of distributed energy resources.

Turning to generation expansion problems, researches and works encompass diverse perspectives. da Costa et al. (2020) include reliability constraints using a stochastic risk-averse approach, Ihsan et al. (2021) assess the performance of distributed hybrid renewable power plants in both on-grid and off-grid contexts, and Rebennack (2014) introduce the incorporation of emissions quotas as a critical factor.

Renewable Energy

With the significant attention that production planning for hydro-thermal power, a prominent renewable energy resource, has gained in SDDP research, it is only natural for similar focus to extend to other renewable energy sources. For instance, a couple of works have broadened the scope of hydro-thermal power production problem to incorporate additional renewable energy sources. As initiated in Raby et al. (2009) wind farms have also been integrated into the hydro-thermal power transmission system. Likewise, various other studies such as de Faria & Jaramillo (2017), Alvarez et al. (2018), da Silva Fernandes et al. (2019), Scheben et al. (2020), and Morillo et al. (2020) explore the coupling of hydro-thermal power and wind energy. Moreover, the integration of wind energy production has also been linked with power grid management and storage in Zéphyr & Anderson (2018), and with microgrid systems with storage in Guo et al. (2020). Wozabal & Rameseder (2020) address uncertainty in wind energy production for optimal bidding in the Spanish electricity market.

Other applications integrating renewable energy sources include the economic dispatch problem with different renewable energy resource uncertainties as in Lu et al. (2019) and in Papavasiliou et al. (2017) for storage. Bao et al. (2019) investigate the optimization of renewable bioresources, and cogeneration biomass plants in Leocadio et al. (2020). Finally, Pacaud et al. (2022) and Aaslid et al. (2022b) discuss a complex combination of renewable energies and storage to optimize microgrids.

Energy Markets, Trading, and Investments

In addition to optimizing aspects such as planning, dispatch, and the expansion of energy systems, another critical domain of optimization involves the financial facets of energy – bidding, trading in energy markets, and investments. Energy markets present challenges marked by fluctuating market-clearing prices, dynamic demand patterns, and uncertain renewable energy production. Understanding energy market dynamics provides a framework to optimize bidding strategies capable of robust adaptation to these uncertainties. Trading strategies empower market participants to make real-time decisions that balance profit maximization and risk aversion. Moreover, the use of SDDP provides the ability to make dynamic and well-informed decisions.

The work in Bonnans et al. (2012) initiated this field of application by addressing the optimal management of energy contracts. The model specifically provided bounds on the local and global quantities to be traded, considering uncertainties in prices. This is followed by Bruno et al. (2016) with a development of an MSP model to determine optimal strategies for energy project investments with real options such as postponing, hedging with fixed forward contracts and other sources.

Covering a range of optimal strategic bidding problems, Wozabal & Rameseder (2020) and Shinde et al. (2022) delve into day-ahead and intraday bidding strategies for electricity within a virtual power plant framework, incorporating renewable energy sources. Steeger & Rebennack (2017) address a single-price maker strategic bidding problem, backed by case studies involving El Salvador, Honduras, and Nicaragua. Shifting focus to the domain of natural gas storage valuation, Löhndorf & Wozabal (2021) investigate this problem through an asset-backed trading approach that takes into account agents' risk preferences.

In the context of emissions trading, Zapletal et al. (2022) introduce a multi-stage planning approach involving carbon risk management and the innovative EU emissions management scheme. Their study compares EUA futures (derivatives) and banking. On a broader scale, Scheben et al. (2020) examine the impact of inflow randomness, revealing that a substantial integration of energy storage devices and renewable energies significantly influences electricity price structures within markets. This observation is supported by a case study conducted in Norway.

Energy storage systems

As the energy sector gravitates towards a more decentralized and renewable-centric paradigm, energy storage devices have emerged as viable solutions in energy systems, particularly in their ability to bridge the gap between intermittent energy production and demand uncertainty. These devices, from advanced battery systems to pumped hydroelectric storage, offer the capability to capture energy produced at a certain stage and utilize it at a later stage. Within the context of energy, MSP models have been instrumental to the optimal management and utilization of these storage devices become predominant.

The integration of energy storage devices into management of energy systems poses new challenges and opportunities, and their exploration in the literature is relatively recent. Distributed energy storage modelled with MSPs were first addressed in Gangammanavar & Sen (2016). Building on this, Zéphyr & Anderson (2018) incorporate wind energy uncertainty considerations for power grids.

Furthermore, within the broader spectrum of grid management leveraging the MSP framework, Asamov & Powell (2018) investigate energy storage in large transmission grids, while Bhattacharya et al. (2016) and Aaslid et al. (2022b) focus on optimizing energy storage in microgrids. Further advancements in microgrid systems is showcased by Pacaud et al. (2022) and Hafiz et al. (2019b), with a particular emphasis on integrating solar panels and batteries. Moreover, Guo et al. (2020) introduce a robust, data-driven approach to optimizing microgrids that harness wind energy, and Aaslid et al. (2022a) consider battery degradation.

Other applications of energy storage devices in energy include work in Ding et al. (2021) illustrating the concept of virtual energy storage using distributionally robust SDDP. On the matter of gas storage, Terça & Wozabal (2021), van Ackooij & Warin (2020), and Löhndorf & Wozabal (2021) examine its pricing and valuation. Lastly, Alvarez et al. (2017) discuss hydroelectric storage in conjunction with batteries, and in Guan et al. (2018) concerning flood control applications.

Grid management

Electrical grid management has also received quite an attention for SDDP application. Works in Asamov & Powell (2018) and Yang & Nagarajan (2022) delve deep into large power grid management. Ding et al. (2020) and Lara et al. (2020) extend this with MSIP formulation solved by SDDiP. Additionally, Zéphyr & Anderson (2018) incorporate wind energy and storage elements, while Hafiz et al. (2019a) discuss the enhancement of grid resiliency against natural calamities.

Meanwhile, the realm of microgrid management, Bhattacharya et al. (2016), Ding et al. (2021), Pacaud et al. (2022), Guo et al. (2020), Aaslid et al. (2022b), and Aaslid et al. (2022a) expansively cover its synergy with energy storage devices. Diving deeper into this area, Shi et al. (2019) explore the integration of varied distributed energy resources within microgrids. Furthermore, Carpentier et al. (2020) present an examination of an urban microgrid problem, while Pacaud et al. (2021) emphasize the aspect of economic dispatching within these systems.

2.4.2 Finance

Within the realm of finance, the MSP framework seamlessly aligns with the challenges of optimizing portfolio and asset investments. The seminal work presented in Dantzig & Infanger (1993) offers a comprehensive introduction to modeling portfolio optimization, commonly known in the literature as asset allocation, modelled as an MSP. A crucial feature of multi-stage asset allocation models is the investor's stage-wise capability to recalibrate the portfolio, either by selling or purchasing specific assets, facilitating an ongoing optimization process. The uncertainties of this problem often revolve around the stochastic returns of the chosen investments, e.g., stocks and bonds, and the associated borrowing costs incurred when a specified financial target is elusive. Asset allocation problems, especially when considering large number of assets or over extended time horizons, can exhibit high dimensionality which present a computational challenge for SDDP method.

Furthermore, understanding the behaviour of investors is inherently challenging. Investors naturally gravitate towards risk-aversion to effectively safeguard their investments, which also increases the difficulty of the problem. Homem-de Mello & Pagnoncelli (2016) comprehensively address the formal introduction of risk measures into MSPs, a methodology to which SDDP is easily adapted. Prior to this, three seminal works incorporate risk-aversion into portfolio optimization using SDDP. Specifically, Dupačová & Kozmík (2015) present a risk-averse MSP underpinned by a coherent risk measure. In contrast, both Kozmík (2015) and Kozmík & Morton (2015) apply a risk-averse formulation, but grounded in conditional value-at-risk as the risk metric. Subsequent studies, including Guigues (2017), Valladão et al. (2019), Waga et al. (2022), and Dowson et al. (2022b), built upon these foundational works.

The asset allocation problem has further evolved, embracing diverse contexts. Reus & Prado (2022) delve into large-scale asset allocation, optimizing policies aligned with user-defined (synthetic) indices. Silva et al. (2021) solve a dynamic asset allocation case study underpinned by a data-centric prescriptive analytics framework. Meanwhile, Tsang et al. (2022) put forth a distributionally robust multi-period portfolio model, addressing ambiguity in asset correlations while maintaining fixed metrics for individual asset return mean and variance.

From a technical perspective, asset allocation has also served as a platform to demonstrate advancements in SDDP techniques. For instance, Guigues (2014) present formulas that distribute optimality and feasibility cuts across nodes within the same stage. Various SDDP variants were introduced by Guigues et al. (2020), Guigues (2020), and Dupačová & Kozmík (2017), including regularized, inexact, and preprocessing via scenario tree reduction, respectively. Guigues & Monteiro (2021) add to the discourse with the stochastic dynamic cutting plane (StoDCuP) extension to SDDP. Bandarra & Guigues (2021) outline the nuances of single-cut and multi-cut SDDP, coupled with cut selection for asset allocation. Finally, Guigues (2021) tackle MSPs characterized by an uncertain number of stages.

2.4.3 Operations Management

Outside of energy and finance, operations management also presents an intriguing application area for SDDP. Domains within this sphere encompass inventory management, production and manufacturing, and facility location, among others.

In the realm of inventory management, variations of the classical problem have been explored in references such as Parpas et al. (2015), Guigues (2017), Dowson et al. (2020), Shapiro & Cheng (2021), Ávila et al. (2022), and Guigues et al. (2023). Lot-sizing problems have also received attention. For instance, Ahmed et al. (2022) address a capacitated version, Quezada et al. (2022a) discuss the uncapacitated variant, and Thevenin et al. (2022) investigate a version incorporating component substitution.

Shifting the focus to production and manufacturing, several studies stand out. Nannicini et al. (2021) tackle bike production with uncertain demand, incorporating aspects of backlog and re-balancing. Yıldız & Sütçü (2022) work on optimal pricing strategies for slow-moving items, while Tong et al. (2020) approach the challenges of central kitchen production and distribution. The pastoral dairy farm planning has also been delved into, with Dowson et al. (2019) and Dowson (2020) addressing specific instances. Moreover, Dowson et al. (2022b) provide insights into a particular routing problem.

2.4.4 Other areas and works

While energy, finance, and operations management dominate as the primary sectors for SDDP application, emerging innovative problems are gaining attention as new application areas. Notably, disaster management has begun benefiting from SDDP. For instance, Angün (2015) employs SDDP to address short-term disaster management and Hafiz et al. (2019a) harness SDDP to enhance grid resiliency against natural disasters through distribution service restoration. Raso et al. (2019) also apply SDDP for reservoir management—specifically focusing on drought and flood protection in the upper Seine-Aube river system. Furthermore, in the mining industry, Reus et al. (2019) leverage SDDP to devise optimal policies for more effective management of production incidents.

Stochastic Dual Dynamic Integer Programming (SDDiP)

In the context of SDDiP, several studies stand out for their noteworthy contributions. For instance, both Zou et al. (2019) and Ding et al. (2019) delve into problems surrounding airline revenue management. The classic OR problem of facility location has also seen innovative applications in this domain. Yu & Shen (2022) explore a variant of the capacitated facility location problem, introducing elements of demand uncertainty. Meanwhile, Seranilla & Löhndorf (2023) not only implemented SDDiP for a COVID-19 vaccine facility location instance but also contrast it with a newly proposed methodology.

Additionally, Quezada et al. (2022b) venture into the complexities of remanufacturing under uncertainty. Bakker et al. (2021) solve the problem of optimal timing of investments in mature oil and gas fields in the presence of price uncertainty, like a complex real options problem. For energy planning, Lara et al. (2020) and Hou et al. (2021) present a model on electric power infrastructure planning under uncertainty and Zou et al. (2018) solve a multi-stage stochastic unit commitment problem.

2.5 Discussion

In this section, we delve into the various factors that influence the adoption of SDDP in certain application domains. Our goal is to highlight the challenges and potential obstacles to its more widespread use. To provide a comprehensive perspective, our conclusions draw from both the extensive literature review in Section 2.4 and engaging discussions with prominent researchers in the SDDP and stochastic programming field. Specifically, we have pinpointed four key factors that play a pivotal role in the adoption of SDDP: (1) historical background, (2) state space limitation, (3) stage-wise independence assumption limitation, and (4) data availability.

2.5.1 Historical background

The success of SDDP in addressing its flagship application, the Brazilian hydro-thermal power production problem, in the seminal paper by Pereira & Pinto (1991), played a significant role in its widespread adoption, largely in other energy-related applications. Thus, in the broader academic and research communities, SDDP's evident success on this problem became a compelling narrative. This prompted a surge of literature on the subject. Many researchers were drawn to the challenge of refining the method, expanding its applications, or tailoring it for specific regional or technical nuances. This is highlighted by the literature - from the year of SDDP's inception in 1991 until 2016, all published works in the energy sector that utilized SDDP were centered exclusively on the hydro-thermal power production scheduling problem.

This historical synergy between SDDP and the energy sector has profoundly shaped the trajectory of research and practical applications in this field. SDDP's resilience in handling large-scale, multi-stage stochastic optimization problems, a characteristic that is often inherent in many problems in the energy sector underscores why a bulk of literature on SDDP is anchored in this field of application.

Over time, as the challenges of the energy sector evolved and expanded, the versatility of SDDP kept pace. It became the go-to methodology for many researchers, leading to a wave of academic publications and practical applications. Its historical successes in the energy sector spurred curiosity, experimentation, and eventually, a prolific presence in scholarly discourse - even parallel sessions in conferences are devoted to SDDP in the energy sector.

Currently, when reviewing literature related to optimization, in the context of energy, the echoes of SDDP's historical triumphs are indisputable. Unfortunately, while it testifies to SDDP's efficacy in energy-related applications, it has inadvertently created a perception among some that SDDP is exclusively tailored for energy problems. This misconception limit the exploration and adoption of SDDP in diverse fields, potentially sidelining its benefits in contexts beyond energy.

2.5.2 State space limitation

At its core, SDDP is sophisticated and manages to solve intricate optimization problems, and it has showcased exemplary efficiency. While SDDP undeniably holds merits in solving many problems, its inherent limitations become more pronounced as the scale and complexity of problems increase. A fundamental assumption of SDDP is that consecutive stages should be linked by constraints via *state variables* - variables used to describe the mathematical state of a dynamical system. Naturally, as a problem's dimensionality expands, so does the number of required state variables. Unfortunately, SDDP's efficiency diminishes when tasked with handling an extensive number of state variables and begin to face significant computational and convergence challenges. For instance, the original flagship application on the Brazilian hydro-thermal power production scheduling problem only has 4 state variables per stage corresponding to the stored energy [water level] of each reservoir. This characteristic, in turn, has raised concerns about its performance in vast state spaces, leading to its subdued and limited utilization in other real-world problems. The limitations in handling vast or high-dimensional state spaces can be a considerable deterrent for other problems. High-dimensional problems are increasingly common in numerous fields and a method that struggles with such scale naturally finds limited applicability. Consequently, researchers might be inclined to explore alternative methods that better cater to extensive state space requirements. Moreover, this state space limitation can sometimes lead to suboptimal solutions or expensive computations, which are not always feasible in real-world applications that demand both accuracy and efficiency. As a result, there might be a preference towards methods that offer a more balanced performance across various problem scales. While SDDP undeniably holds benefits and has proven valuable in some problems, the concerns about its performance in extensive state spaces have played a pivotal role in its adoption in other application areas.

2.5.3 Stage-wise independence limitation

Another fundamental assumption of SDDP is that the random data has to be *stage-wise independent* - new information becoming available at the time of decision does not depend on the history of the random process. This assumption, while simplifying the computational complexity in certain cases, becomes conflicting in the context of real-world applications. Many systems and processes, especially those of an economic, financial, or natural nature, inherently possess temporal dependencies, making them Markovian or semi-Markovian structure.

Additionally, the stage-wise independence assumption can limit the accuracy and relevance of SDDP when applied to problems with evident temporal dependencies, potentially resulting in sub-optimal solutions. Thus, there's a risk of overlooking critical dependence, which can lead to significant discrepancy in decisions derived from the model. This limitation can deter researchers from utilizing SDDP in problems where capturing the dynamics of random data over time is crucial.

Nonetheless, to address this issue, the work by Infanger & Morton (1996) introduce interstage dependencies for MSPs and Löhndorf & Shapiro (2019) extended this with two approaches to include dependence of the random process - (1) to model the data process as an auto-regressive process and to add the time series transition equations and (2) another approach based on Markov Chain discretization of the Markovian data process using optimal quantization. Nonetheless, the first approach is restricted only to right-hand side uncertainty and linear cases, and the second approach, although allows modeling any parameter as Markovian data process, has no convergence guarantees. Hence, while SDDP holds significant potential in many optimization problems, the stage-wise independence assumption tempered on its universal applicability, steering its use toward more restricted domains.

2.5.4 Data availability

When a parameter of a on optimization problem is random or uncertain, trying to understand the distribution of this parameter can be a daunting challenge. This complexity is further highlighted especially if data is not readily available. The availability, or lack thereof, of data not only impedes precise modeling but also limits the accuracy and reliability of any derived solution. This phenomenon is universally observed across any problems in the field of decision-making under uncertainty for real-world applications. MSPs naturally falls into this category, where the success of the model often depend on the data for uncertain parameters. Consequently, SDDP methods, which operate under these conditions, might see restricted application in practice due to the inherent challenges posed by data limitations.

2.5.5 Extensions and outlook

Provided the challenges and potential obstacles presented above, we remain optimistic about SDDP's potential to overcome current limitations and address more complex problems. While its major applications have been in the hydrothermal energy sector, finance, and operations management, here are some other domains and specific applications where SDDP can be potentially applied:

Healthcare: The healthcare domain is rife with many complex optimization problems. This includes hospital resource allocation, optimizing treatment plans for chronic illnesses over multiple periods, and surgery scheduling. The uncertainties associated with disease spread, patient outcomes, and resource constraints make SDDP a potential tool for driving productivity and efficiencies.

Disaster Preparedness and Response: Natural disasters, e.g., thunderstorms, earthquakes, and volcanic eruptions, SDDP can be effective in optimizing resource allocation for disaster preparedness and formulating real-time response policies as events unfold.

Environmental Conservation: Optimization problems that involve habitat restoration, wildlife protection, or pollution control over multiple periods, SDDP can help devise policies that account for the uncertainties in ecological settings.

Urban Planning: City infrastructures are complex and interconnected. Whether it is deciding on networks for public transportation, policies for waste management, or land-use planning over different time horizons, SDDP has the potential to handle these problems.

Telecommunications: As communication needs and technologies rapidly evolve, there is a need to optimize network designs, resource allocations, and service over time. SDDP can be a useful tool in making these decisions.

Education: Some common university and educational institution problems are course scheduling, resource allocation (e.g., classrooms or laboratory facilities), and even strategic decisions like opening new programs or campuses.

Marine and Fisheries Management: SDDP can be beneficial from optimizing fishing schedules to route planning for marine vessels to avoid ecologically sensitive areas.

2.6 Concluding Remarks

Since its inception in 1991, SDDP has, for over three decades, demonstrated a phenomenal performance and is still regarded as one of the go-to methods to solve MSPs. This survey presents a review of the practical applications of SDDP in the literature. A total of 186 significant publications, from 1991 to 2022, were examined from an in-depth search in multiple databases. Various advantages and challenges associated with SDDP were examined to pave way for an in-depth exploration of the literature, in the lens of these different applications.

While its applications span to various fields, the energy sector has notably emerged as its predominant user, with approximately 76% of the articles reviewed deploying the SDDP methodology. This is followed by finance, operations management, and water resources domains. Delving deeper, the hydro-thermal power system production surfaced as the predominant domain, constituting roughly 60% of the literature. This is trailed behind by renewable energies, energy market, trading, and investments, energy storage devices, and grid management applications. This analysis is further enriched with an insightful discussion on the factors influencing the adoption of SDDP across distinct application areas. We have identified four factors - historical background, state space limitation, stage-wise independence assumption limitation, and data availability. Finally, given the ongoing research aimed at augmenting, developing, and enhancing SDDP, we remain optimistic about its potential to overcome current limitations and address more complex problems faced by the state-of-the-art.

Appendix 2.A Detailed reference classification

Table 2.2: Reference classification (Hydro-thermal power production) - Brazil and Norway

Location	**Operation**	**Reference**
Brazil	Long-term	Pereira & Pinto (1991), Gorenstin et al. (1992), Mello et al. (1997), da Silva & Finardi (2003), Maceira & Damázio (2006), Homem-de Mello et al. (2011), Zambelli et al. (2011), Guigues & Sagastizábal (2012), de Matos & Finardi (2012), Shapiro et al. (2013a), Guigues & Sagastizabal (2013), Pinto et al. (2013), Philpott et al. (2013), Shapiro et al. (2013b), Guigues (2014), Calili et al. (2014), Maceira et al. (2014), Oliveira et al. (2015), Brandi et al. (2015), de Castro et al. (2015), De Matos et al. (2015), Ferreira et al. (2015), Lohmann et al. (2016), Brigatto et al. (2017), Rego et al. (2017), Soares et al. (2017), De Matos et al. (2017), Street et al. (2017), de Faria & Jaramillo (2017), Brandi et al. (2017), Löhndorf & Shapiro (2019), Van Ackooij et al. (2019), Shapiro & Ding (2020), Machado & Bhagwat (2020), Treistman et al. (2020), Diniz et al. (2020), Liu & Shapiro (2020), Street et al. (2020), Street et al. (2020), Beltrán et al. (2020), Fredo et al. (2021), Machado et al. (2021), Shapiro & Cheng (2021), Resende et al. (2021), Dowson et al. (2022a), Larroyd et al. (2022), Ávila et al. (2022)
	Medium-term	Löhndorf et al. (2013),Maceira et al. (2014), de Matos et al. (2015), Fredo et al. (2019), da Silva Fernandes et al. (2019), Treistman et al. (2020), Machado et al. (2021), Beltrán et al. (2021), Guigues et al. (2021), Siddig & Song (2022),Borges et al. (2022), Dornellas et al. (2022), Pedrini & Finardi (2022)
	Short-term	Beltrán et al. (2021), Borges et al. (2022)
Norway	Long-term	Rotting & Gjelsvik (1992)
	Medium-term	Mo et al. (2001), Kristiansen (2006), Helseth & Braaten (2015), Helseth et al. (2016), Hjelmeland et al. (2018), Hjelmeland et al. (2019), Scheben et al. (2020), Helseth & Mo (2022), Helseth et al. (2022)
	Short-term	- - -

Table 2.3: Reference classification (Hydro-thermal power production) - Other locations

Location	Operation	Reference
New Zealand	Long-term	Duque & Morton (2020)
	Medium-term	Philpott & De Matos (2012), Philpott et al. (2018), Downward et al. (2020), Dowson & Kapelevich (2021)
	Short-term	Pritchard (2015)
Central America	Long-term	Flach et al. (2010), Marques & Tilmant (2013), Rebennack (2014)
	Medium-term	Rebennack et al. (2011), Rebennack (2016), Steeger & Rebennack (2017)
South America	Long-term	Raby et al. (2009), Sauma et al. (2011), Toledo et al. (2015), Morillo et al. (2020), Morillo et al. (2022)
Canada	Long-term	Pina et al. (2017), Mbeutcha et al. (2021)
	Medium-term	Zhang & Ponnambalam (2005), Guan et al. (2018), Côté & Arsenault (2019)
Nile River	Long-term	Goor et al. (2011), Kahsay et al. (2019), Tariku et al. (2021)
Iberian Peninsula	Long-term	Pereira-Cardenal et al. (2016)
	Medium-term	Macian-Sorribes et al. (2017), Leclère et al. (2020)
Senegal	Long-term	Espanmanesh & Tilmant (2022)
	Medium-term	Raso et al. (2017), Raso et al. (2020)
Turkey/Iran	Long-term	Tilmant et al. (2007), Tilmant & Kelman (2007)
	Medium-term	Poorsepahy-Samian et al. (2016)
China	Short-term	Li et al. (2022)
Others	Medium-term	Borges (2022), Cerisola et al. (2012)
	Long-/Short-term	Alvarez et al. (2017), Alvarez et al. (2018)

Chapter 3

MSPLib and MSPFormat: A Library Of Problems and a New Standardized Data Format For Benchmarking Stochastic Dual Dynamic Programming

In this chapter, we introduce two essential contributions: MSPLib, a comprehensive library of multistage stochastic programming problems, and MSPFormat, a new standardized data structure format for multistage stochastic programs. The main objective of MSPLib is to facilitate the evaluation and comparison of computational performance among different implementations of stochastic dual dynamic programming (SDDP). The library encompasses a diverse range of instances, including real-world problems and synthetic variations with varying levels of complexity. By incorporating MSPFormat, we provide a unified and consistent representation of multistage stochastic programs. We also test prevailing implementations of SDDP - including QUASAR, SDDP.jl, and MSPPy.

3.1 Introduction

Problem instances for benchmarking are a necessity for optimization researchers and developers for empirical studies and proof of concepts. Currently, various problem libraries are available for use: the 'netlib' repository for linear programming problems (Browne et al. 1995), the Schittkowski library for non-linear programming (Schittkowski 1986), the 'miplib' library for mixed-integer linear programming (Gleixner et al. 2021), and the more recent CMU-IBM Cyber-Infrastructure (Cyber-Infrastructure 2016) and 'MINLPLib' for mixed-integer non-linear programming problems (Bussieck et al. 2003). Collections of optimal control problems are also readily available. The PROPT toolkit in Matlab (Leek 2016) provides more than 100 test case problems across different applications, complete with their corresponding results and computation times.

For stochastic programming, some efforts have been made by the Stochastic Programming Society (SPS) (SPSociety 2012). Unfortunately, the problem instances included in this collection primarily focus on synthetic problems of either single- or two-stage stochastic (integer) programs. Consequently, multistage stochastic programming (MSP) problems, which deal with sequential problems under uncertainty, have been largely overlooked. However, with the proliferation of MSP algorithms, there's a pressing need for standardized test instances. One such prominent MSP algorithm is the stochastic dual dynamic programming (SDDP) method, which showed promising results in its initial application to hydrothermal scheduling problems and has since been widely employed to address various MSP challenges. Unfortunately, there is a noticeable lack of readily available libraries dedicated specifically to MSP problems. This presents a valuable opportunity.

The goal of this work is three-fold. First, we introduce MSPLib, a benchmark library of MSP problems which encompasses both large-scale, real-world problem instances and smaller, synthetic ones, ranging from simple production problems to the famous Brazilian hydro-thermal power production problem. We aim to provide detailed description of these problems and their corresponding *standard* mathematical formulations. Solutions – including first-stage solutions, and lower and upper bounds gap – obtained by SDDP for these problems will also be provided. MSPLib benchmark library is available online for easy access and utilization by the community. Additionally, we provide ready-to-use scripts to select MSP problems accompanied by some comprehensive notes available at [https://github.com/bonnkleiford/MSPLib-Library].

The second goal is to introduce MSPFormat, our proposed standardized data structure format for MSP problems. Compressed into the JavaScript Object Notation (JSON) file format, MSPFormat separates the mathematical problem formulation and the realizations of the stochastic processes in two distinct JSON files. The recent endeavor to standardize the data structure format for MSP problems was ‘StochOptFormat‘, proposed by the creators of the SDDP.jl solver (Dowson & Garcia 2020). We present the details of the MSPFormat formulation in Section (3.3).

Lastly, we aim to benchmark and to test prevailing implementations of SDDP. These implementations include QUASAR, a multi-stage stochastic programming solver developed by Luxembourg-based consulting company QUANTEGO with Python, MatLab, R, and Java interface (Löhndorf 2021); SDDP.jl, an open-source library in Julia developed by Oscar Dowson and Lea Kapelevich (Dowson & Kapelevich 2021), and MSPPy, an open-source implementation of SDDP from Georgia Institute of Technology written in Python (Ding et al. 2019). We evaluate each solver's performance based on three criteria - time limit criterion, iteration limit criterion, and parallel processing criterion.

3.2 Problem Classification and Variations

The MSPLib comprises both synthetic and real-world problems, all modelled as MSPs. To provide a comprehensive understanding, let us take a closer look into the composition of the MSPLib:

1. **Synthetic Problems:** These are generalized formulations of some classical multi-stage (and two-stage) stochastic optimization problems. What enhances their value is the incorporation of artificial data and parameters, which allow easy adjustments to alter the problem's characteristics.

   Sources: These problems are sourced from textbooks (e.g., Birge & Louveaux (2011), Kall et al. (1994), Wallace & Ziemba (2005), Prékopa (2013), Ross

(2014)), tutorials (e.g., Shapiro & Philpott (2007) and Shapiro (2021)), and current problem instances (e.g., SPSociety (2012)). Initially, they were approached using the SDDP algorithm and/or other MSP algorithms.

Benefits: The advantage of synthetically crafted problems is that they allow for adjustable complexity and difficulty levels. This offers a controlled environment to gauge the efficiency and robustness of various SDDP implementations. A prime example is the classical inventory problem with stochastic demand, which has been enhanced to include lead time and lost sales. This adjustment transforms the problem, making it more complex and challenging to solve, as evidenced by increased number of iterations and wider optimality gap.

2. **Real-world Problems:** These problems are sourced from various articles in the literature where the SDDP algorithm was employed to solve the problem.

   Benefits: It is essential to tackle real-world problems, as the SDDP algorithm is particularly well-suited for large and complex problems. SDDP especially excels when dealing with the *curse-of-dimensionality*.

3. **MSIP Problems:** These problems are sourced from various articles in the literature where the problem is modelled as a multistage stochastic integer program and where the SDDiP algorithm was employed to solve the problem.

   Benefits: The class of MSIPs is also essential to tackle as many real-world problems are modelled as such. The details of the MSIPs included in the MSPLib are found in Appendix 1.1.3.

3.2.1 Problem variations and extensions

Tables 3.1, 3.2, and 3.3 show the problem instance charts. These charts provide in detail the various problems in MSPLib with their unique instance variations and corresponding specifications.

Every synthetic problem extends the original problem (Instance 0) formulation in terms of number of stages (T), problem size (indices I, J, K), and randomness of the problem. Extending planning horizons, increasing the scale of problem sizes, and introducing diverse forms of randomness all contribute to creating more intricate and challenging MSPs. Thus, the different variations and extensions to the original problem formulation have been designed with these considerations. For every problem instance, the problem instance charts also show the vector/matrix and location of the random variable. For example, the randomness can occur in the right-hand side (b / RHS) and the coefficients (A / Con Eff) of the constraints and the objective function (c / Obj Fun). The difficulty level for each problem is based on SDDP's convergence speed, measured in both time (in secs) and number of iterations. *Easy* category problems converge, on average, after 15 to 25 iterations, *medium* ones take 50 to 70 iterations, while *difficult* problems require more than 80 iterations. Discretization of the random variables are also shown, with *easy* category problems featuring the original number of samples (SWI-D) or 100 and 200 scenario realizations per stage. In contrast, *medium* and *difficult* category problems are discretized with 100, 200, and 300 scenario realizations.

On the other hand, the problem instance chart for the real-world problems, shown in Table 3.3, show the specifications of each real-world problem, i.e, planning horizons (timesteps), random variable and its corresponding type (i.e., SWI - stage-wise independent uncertainty, Markovian - Markovian uncertainty, and AR - autoregressive / time series uncertainty), and the level of difficulty. We retain the original data, parameters, model, and randomness / scenarios for the real-world problems from the sourced literature.

The MSPLib adopts a systematic naming convention for instance numbering, which aligns seamlessly with the online repository of MSPLib. Each filename is structured to reflect the problem number, the instance number, and the level of discretization. To illustrate, consider the file named *'(01_2)_100.problem.json'*. In this case:

- *01* denotes the specific problem, in this instance, the simplified hydrothermal scheduling problem.

- *2* represents the instance number.

- *100* indicates that there are 100 discretized scenario realizations.

This consistent naming convention is adhered to throughout the entirety of the library, ensuring clarity and ease of reference for users.

3.2.2 Synthetic Problems

PROBLEM	Problem Number	Instance Number	*Stages* (T)	*Size* $(I/J/K)$	*Randomness*	*Vector / Matrix*	*Location*	*Category*	*Difficulty*	*Discretization*
Simplified Hydrothermal Scheduling Problem *(Shapiro et al, 2013)*	01	0	4	I=1	Inflow	b	RHS	SWI-D	Easy	3
		1	4	I=1	Inflow / Demand	b	RHS	SWI	Easy	100, 200
		2	4	I=1	Inflow / Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		3	10	I=1	Inflow	b	RHS	SWI-D	Easy	3
		4	10	I=1	Inflow / Demand	b	RHS	SWI	Easy	100, 200
		5	10	I=1	Inflow / Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		6	25	I=1	Inflow / Demand	b	RHS	SWI	Easy	100, 200
		7	25	I=1	Inflow / Demand / Price	b / c	RHS	SWI	Easy	100, 200
		8	50	I=1	Inflow / Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		9	100	I=1	Inflow / Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
Airconditioning Production Problem *(Papavasiliou, 2017)*	02	0	3	I=1	Demand	b	RHS	SWI-D	Easy	2
		1	3	I=1	Demand / Price	b / c	RHS / Obj Fun	SWI-D	Easy	2
		2	20	I=1	Demand	b	RHS	SWI	Easy	100, 200
		3	20	I=1	Demand	b	RHS	SWI	Easy	100, 200
		4	20	I=1	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		5	50	I=1	Demand	b	RHS	SWI	Easy	100, 200
		6	50	I=1	Demand	b	RHS	SWI	Easy	100, 200
		7	50	I=1	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		8	100	I=1	Demand	b	RHS	SWI	Easy	100, 200
		9	100	I=1	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
Electricity Production Problem *(Louveaux, 1988)*	03	0	2	I=4 / J=3	Demand	b	RHS	SWI-D	Easy	3
		1	2	I=4 / J=3	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		2	2	I=10 / J=5	Demand	b	RHS	SWI	Easy	100, 200
		3	2	I=10 / J=5	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		4	2	I=20 / J=10	Demand	b	RHS	SWI	Easy	100, 200
		5	2	I=20 / J=10	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		6	2	I=50 / J=30	Demand	b	RHS	SWI	Easy	100, 200
		7	2	I=50 / J=30	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		8	2	I=100 / J=50	Demand	b	RHS	SWI	Easy	100, 200
		9	2	I=100 / J=50	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
Newsvendor Problem *(Arrow et al, 1951)*	04	0	2	I=1	Demand	b	RHS	SWI-D	Easy	10
		1	2	I=2	Demand / Price	b	RHS / Obj Fun	SWI	Easy	100, 200
		2	10	I=1	Demand	b / c	RHS	SWI	Easy	100, 200
		3	10	I=2	Demand / Price	b	RHS / Obj Fun	SWI	Medium	100, 200, 300
		4	20	I=1	Demand	b	RHS	SWI	Medium	100, 200, 300
		5	20	I=2	Demand / Price	b / c	RHS / Obj Fun	SWI	Medium	100, 200, 300
		6	50	I=1	Demand	b	RHS	SWI	Difficult	100, 200, 300
		7	50	I=2	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
		8	100	I=1	Demand	b / c	RHS	SWI	Difficult	100, 200, 300
		9	100	I=2	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
Semiconductor Production Problem *(Ahmed, 2013)*	05	0	4	I=10 / J=10 / K=2	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		1	4	I=10 / J=10 / K=5	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		2	10	I=10 / J=10 / K=2	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		3	10	I=10 / J=10 / K=5	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		4	20	I=10 / J=10 / K=2	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		5	20	I=10 / J=10 / K=20	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		6	50	I=10 / J=10 / K=2	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		7	50	I=10 / J=10 / K=20	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		8	100	I=10 / J=10 / K=2	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		9	100	I=10 / J=10 / K=25	Demand / Price	b / c	RHS / Obj Fun	SWI	Easy	100, 200

Table 3.1: Set of synthetic problems (01-05) with various instance flavors and corresponding specifications

Simplified Hydro-thermal Scheduling Problem

The seminal paper on SDDP primarily tested the algorithm on the Brazilian hydro-thermal interconnected power system. Consequently, it is fitting to commence with this problem. The goal of the Brazilian hydro-thermal interconnected power system is to devise an operational strategy that, for each stage within the planning period, produces electricity generation targets for every plant based on the states of the system's state.

In the simplified version of the problem, given deterministic marginal operating costs for both a thermal and a hydro-power generator, the planner must determine a generation quantity that satisfies known demand for each period within the planning horizon. A reservoir, from which the hydro-power generator sources water, has a maximum capacity. For the purpose of this problem, we assume this reservoir starts at full capacity in the initial planning period. There's also a provision for water to be spilled, bypassing the turbine, to prevent over-topping of the dam.

The primary objective here is to identify an optimal policy that minimizes the expected operation cost over the entire planning horizon. This cost comprises fuel expenses and penalty costs for supply shortfalls. The principal stochastic component of this problem arises from water inflow, which refers to the water entering the reservoir, either from rainfall or river flow. These inflows are unpredictable and introduce the main dilemma in hydro-thermal scheduling: the inclination to utilize water immediately for cost-effective electricity generation versus the peril that future inflows might be insufficient, potentially causing blackouts or necessitating costly thermal generation.

Variations of the simplified hydro-thermal scheduling problem within 'MSPLib' encompass extensions in the planning horizon ($T = 4$, 10, 25, 50, 100), the introduction of more data and parameter uncertainty (ξ_t = inflows, demands, and price), and an increase in the discretization of the problem scenarios ($S = 100$, 200).

Airconditioning Production Problem

The air-conditioner production problem is a synthetic problem derived from the Operations Research lectures of Anthony Papavasiliou (Papavasiliou 2014). The problem presents a scenario where the manager of a manufacturing company aims to devise a production plan for air-conditioners over a given planning period T. The company can function during an 8-hour standard working window or opt for an unlimited overtime schedule, though the latter incurs additional costs. Air-conditioners can be stored inter-monthly at a specific cost, and all demands must be satisfied at every stage t. Variations of the air-conditioner production problem within 'MSPLib' consist of extensions in the planning horizon (T = 3, 20, 50, 100), the introduction of more data and parameter uncertainty (ξ_t = demands and price), and an changes in the discretization of the problem scenarios (S = 100, 200).

Electricity Planning Problem

The electricity planning problem, derived from Louveaux (1988), is a 2-stage synthetic problem. It revolves around determining the optimal capacity investment across various power plants to fulfill the upcoming period's electricity demands. Multiple power plants are taken into account, each capable of operating in distinct modes. The subsequent period's demand for each mode must be satisfied. Constraints are imposed on the budget as well as the minimum total capacity. Variations of the electricity planning problem featured in MSPLib include: A fixed planning horizon (T = 2), an increasing number of power plants (I = 4, 10, 20, 50, 100) and their respective operating modes (J = 3, 5, 10, 30, 50), data and parameter variability (ξ_t = demand and price), and expanded discretization of the problem scenarios (S = 100, 200).

Newsvendor problem

The newsvendor problem, cited from (Arrow et al. 1951), is a classical 2-stage stochastic optimization problem, making it a fascinating study. Consider a scenario where a newsvendor intends to purchase newspapers today and sell them tomorrow. The objective is to determine the optimal quantity of newspapers the newsvendor should procure today to maximize their profit. Tomorrow's newspaper demand must be fulfilled. Variants of the News-vendor problem within 'MSPLib' encompass extensions in the planning horizon (T = 2, 10, 20, 50, 100), introduction of greater data and parameter stochasticity (ξ_t = demand and price) which includes both discrete and continuous distributions, and increasing the discretization of the problem scenarios (S

= 100, 200, 300).

Semiconductor Production Problem

The semiconductor production problem, as outlined in Ahmed (2002), describes a scenario in which a wafer fabrication unit possesses various tool types that can process different wafer types. Each wafer type undergoes a subset of distinct processing steps, and these steps can be executed on one or multiple tool types. Given the time required for these various processing steps, the tasks are to determine the quantity of each tool type to be procured in every period, allocate the processing steps of each wafer type to specific tool types during each period, and ascertain the production volume of each wafer type within each period. Considering the parameters at hand: the cost of tool types in each period, the penalty cost per unit shortage of each wafer type in every period, the per-period capacity (measured in hours) of a single tool of each type, and the per-period demand (quantified in wafer starts) for each wafer type, the overarching goal is to minimize the combined costs of tool purchases and shortage penalties. The semiconductor production problem instance variants in 'MSPLib' include the planning horizon (T = 4, 10, 20, 50, 100), adding more data and parameter stochasticity (ξ_t = demand and price) including discrete and continuous distributions, and increasing the discretization of the problem scenarios (S = 100, 200).

The Farmer's Problem

The farmer's froblem, as presented in Birge & Louveaux (2011), delves into a problem where a farmer specializes in cultivating various crops, such as wheat, corn, and sugar beets, over a designated acreage of land. The central problem involves deciding the allocation of land for each of these crops. Beyond the standard technological, capacity, and quota restrictions, the problem is further compounded by uncertainties stemming from fluctuating crop yields — attributed to weather conditions — and the variable market prices for each crop. The farmer's problem instance variants in 'MSPLib' include the planning horizon (T = 2), adding more data and parameter stochasticity (ξ_t = yields and prices) including discrete and continuous distributions, and increasing the discretization of the problem scenarios (S = 100, 200, 300).

The Farmer's Problem *(Birge and Louveaux, 2011)*	06	0	2	I=4 / J=3	Yield	A	Con Coeff	SWI	Easy	100, 200
		1	2	I=4 / J=3	Yield / Price	A / c	Con Coeff / Obj Fun	SWI	Easy	100, 200
		2	2	I=10 / J=5	Yield	A	Con Coeff	SWI	Medium	100, 200, 300
		3	2	I=10 / J=5	Yield / Price	A / c	Con Coeff / Obj Fun	SWI	Medium	100, 200, 300
		4	2	I=20 / J=10	Yield	A	Con Coeff	SWI	Medium	100, 200, 300
		5	2	I=20 / J=10	Yield / Price	A / c	Con Coeff / Obj Fun	SWI	Medium	100, 200, 300
		6	2	I=50 / J=30	Yield	A	Con Coeff	SWI	Medium	100, 200, 300
		7	2	I=50 / J=30	Yield / Price	A / c	Con Coeff / Obj Fun	SWI	Medium	100, 200, 300
		8	2	I=100 / J=50	Yield	A	Con Coeff	SWI	Medium	100, 200, 300
		9	2	I=100 / J=50	Yield / Price	A / c	Con Coeff / Obj Fun	SWI	Medium	100, 200, 300
Asset Management Problem *(Birge and Louveaux, 2011)*	07	0	4	I=2	Returns	A	Con Coeff	SWI-D	Easy	2
		1	4	I=2	Returns / Cost	A / c	Con Coeff / Obj Fun	SWI-D	Easy	2
		2	4	I=3	Returns	A	Con Coeff	SWI-D	Easy	3
		3	4	I=3	Returns / Cost	A / c	Con Coeff / Obj Fun	SWI-D	Easy	3
		4	4	I=4	Returns	A	Con Coeff	SWI-D	Easy	3
		5	4	I=4	Returns / Cost	A / c	Con Coeff / Obj Fun	SWI-D	Easy	3
		6	4	I=5	Returns	A	Con Coeff	SWI	Medium	100, 200, 300
		7	4	I=5	Returns / Cost	A / c	Con Coeff / Obj Fun	SWI	Medium	100, 200, 300
		8	4	I=6	Returns	A	Con Coeff	SWI	Medium	100, 200, 300
		9	4	I=6	Returns / Cost	A / c	Con Coeff / Obj Fun	SWI	Medium	100, 200, 300
Generation Expansion Problem	08	0	5	I=5	Demand	b	RHS	SWI	Easy	100, 200
		1	5	I=5	Demand / Cost	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		2	10	I=5	Demand	b	RHS	SWI	Easy	100, 200
		3	10	I=10	Demand	b	RHS	SWI	Easy	100, 200
		4	10	I=10	Demand / Cost	b / c	RHS / Obj Fun	SWI	Easy	100, 200
		5	20	I=5	Demand	b	RHS	SWI	Easy	100, 200
		6	20	I=10	Demand	b	RHS	SWI	Easy	100, 200
		7	20	I=10	Demand / Cost	b / c	RHS / Obj Fun	SWI	Medium	100, 200, 300
		8	50	I=5	Demand	b	RHS	SWI	Medium	100, 200, 300
		9	50	I=10	Demand / Cost	b / c	RHS / Obj Fun	SWI	Medium	100, 200, 300
The Flower Seller Problem	09	0	10	I=1	Demand	b	RHS	SWI	Medium	100, 200, 300
		1	10	I=3	Demand	b	RHS	SWI	Medium	100, 200, 300
		2	10	I=5	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
		3	20	I=1	Demand	b	RHS	SWI	Difficult	100, 200, 300
		4	20	I=3	Demand	b	RHS	SWI	Difficult	100, 200, 300
		5	20	I=5	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
		6	50	I=1	Demand	b	RHS	SWI	Difficult	100, 200, 300
		7	50	I=3	Demand	b	RHS	SWI	Difficult	100, 200, 300
		8	50	I=5	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
		9	100	I=5	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
Inventory Problem with Lead Time	10	0	50	LT = 10	Demand	b	RHS	SW	Difficult	100, 200, 300
		1	50	LT = 20	Demand	b	RHS	SWI	Difficult	100, 200, 300
		2	50	LT = 20	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
		3	70	LT = 10	Demand	b	RHS	SWI	Difficult	100, 200, 300
		4	70	LT = 20	Demand	b	RHS	SWI	Difficult	100, 200, 300
		5	70	LT = 20	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
		6	100	LT = 10	Demand	b	RHS	SWI	Difficult	100, 200, 300
		7	100	LT = 20	Demand	b	RHS	SWI	Difficult	100, 200, 300
		8	100	LT = 20	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300
		9	200	LT = 20	Demand / Price	b / c	RHS / Obj Fun	SWI	Difficult	100, 200, 300

Table 3.2: Set of synthetic problems (06-10) with various instance flavors and corresponding specifications

Asset Management Problem

Similar to the farmer's problem, the asset management problem, as introduced in Birge & Louveaux (2011), applies stochastic programming to address financial planning problems. In this context, an investment planner aims to accrue a specific fund over the span of Y years. Starting with an initial investment of b, they have the option to allocate these funds across I different investment vehicles. The primary objective is to ensure the total exceeds the target amount of G by the end of the Y year period.

However, the problem becomes intricate due to uncertainties associated with varying returns on investments, such as stocks and bonds, and the variable percentage costs incurred when borrowing to bridge any shortfall in achieving the G goal. In the 'MSPLib' repository, the asset management problem offers various instance variants. These include a planning horizon $T = 4$. Additionally, there is an enhancement in the complexity of the problem with the addition of more data and parameter uncertainties, represented as (ξ_t = returns and cost). These uncertainties cover both discrete and continuous distributions. Furthermore, to provide a broader spectrum of scenarios, the discretization of problem scenarios has been expanded with values such as ($S = 100$, 200, 300).

Generation Expansion Problem

Generation expansion problem addresses the challenge of making optimal decisions regarding the timing and magnitude of power plant expansions. It aims to strategically invest in electricity generation to cater to anticipated future demands. Over the years, various iterations and extensions of this problem have been formulated to address diverse problems related to investment, capacity, and expansion specifics. At its core, the objective remains to determine the best technology investments that allow for the necessary power plant expansion to satisfy electricity requirements. In 'MSPLib', the generation expansion problem presents multiple instance variants. These include different planning horizons, represented as ($T = 5$, 10, 20, 50). The complexity of the problem is further enhanced by introducing additional data and parameter uncertainties, labeled as(ξ_t = demand and cost). These uncertainties span both discrete and continuous distributions. Moreover, the range of scenario discretization has been broadened, with values like ($S = 100$, 200, 300).

Flower Seller [or Perishability] Problem

The flower seller problem is an adaptation of the news-vendor problem. In this scenario, a flower seller purchases flowers from a supplier for resale. These flowers are procured before sales commence, and any unsold flowers at the day's end are stored and sold the following day as older inventory. However, any roses remaining in stock after this are discarded. The primary challenge for the flower seller is to ascertain the optimal quantity of flowers to purchase each day to maximize profit and minimize waste. The flower seller problem instance variants in 'MSPLib' include the planning horizon (T = 2, 10, 20), adding more data and parameter stochasticity (ξ_t = demands) including discrete and continuous distributions, and increasing the discretization of the problem scenarios (S = 100, 200, 300).

Inventory Planning Problem with Lead Time and Lost Sales

The classical inventory problem revolves around a company's need to fulfill its customer demands by deciding the quantity of products to purchase in each stage, all while minimizing the combined cost of purchasing and inventory holding. This advanced rendition of the problem incorporates a *lead time* for the ordering and receipt of acquired products and lost sales. This addition of lead time heightens the problem's complexity, particularly as the lead time lengthens. The inventory problem with lead time instance variants in 'MSPLib' include the planning horizon (T = 20, 50, 100), adding more data and parameter stochasticity (ξ_t = demands) including discrete and continuous distributions, and increasing the discretization of the problem scenarios (S = 100, 200, 300).

3.2.3 Real-world problems

Brazilian Hydro-Thermal Power Planning Problem

The Brazilian interconnected power system was initially solved in the seminal SDDP paper by Pereira Pereira & Pinto (1991). As described in the Simplified Hydro-thermal Scheduling Problem above, the objective of the Brazilian Hydro-thermal Power System Problem is to determine an operation strategy that, given the states of the system for each stage of the planning period, produces electricity generation targets for each plant. The optimal strategy should minimize the expected cost of the operation, composed of fuel costs and penalty costs for supply failure, throughout the entire planning horizon. In the original version, the behavior of the entire historical data of the random variable (inflow) is disregarded - referred to as the stage-wise independent finite discrete problem.

PROBLEM	Problem Number	Instance Number	Instance Specifications			
			Timesteps	*Randomness*	*Category*	*Difficulty*
Brazilian Hydrothermal Scheduling Problem (Pereira & Pinto, 1988)	**11**	0	120	Inflow	SWI	Difficult
Brazilian Hydrothermal Scheduling Problem (Loehndorf and Shapiro, 2019)	**12**	0	120	Inflow	Markovian	Difficult
		1	120	Inflow	TS	Difficult
Production-Inventory Planning with Intermittent Renewable Energy (Golari et al, 2016)	**13**	0	6	Available Energy	SWI	Medium
European Hydropower Scheduling Problem (Leclere et al, 2018)	**14**	0	12	Inflow Demand Price	SWI	Medium
Multistage Dairy Farm Problem (Dowson et al, 2017)	**15**	0	52	Weather Evapotranspiration Price	SWI	Medium
Coordination in Multi-Market Bidding of Grid Energy Storage (Loehndorf and Wozabal, 2021)	**16**	0	33	Price	AR	Difficult
Gas Storage Valuation Problem (Loehndorf et al, 2013)	**17**	0	24	Price	AR	Difficult
Capacitated Multi-Echelon Lot Sizing Problem with Component Substitution (Thevenin et al, 2020)	**18**	0	10	Demand	SWI	Medium
Integrated Management of Hydro-plant Cascade (Borges P, 2022)	**19**	0	12	Inflow	SWI	Medium
American Put Option Pricing (Yin et al, 2006)	**20**	0	51	Rates	SWI	Medium

Table 3.3: Real-world problems with various instance flavors and corresponding specifications

This is done to fully understand the underlying problem and become familiar with the model.

Variants of Brazilian Hydro-Thermal Power Planning Problem

The original Brazilian interconnected power system planning problem has been extended in Löhndorf & Shapiro (2019) which introduced two different approaches: (1) an autoregressive time series approach, and (2) a Markov Chain discretization of the random data process. In the time series approach, the random variable (inflow) is considered, with a time series model fitted to its historical data. This approach adds additional state variables and reformulates the problem into a stage-wise independent format. On the other hand, the Markov chain discretization approach allows modeling of any parameter as Markovian data process. Thus, this allows usage of a much broader range of stochastic models which may better represent the true process.

Production-Inventory with Intermittent Renewable Energy Problem

The Production-Inventory with Intermittent Renewable Energy Problem, as discussed in Golari et al. (2017), seeks to determine the optimal production quantity, stock level, and renewable energy supply for each period. The objective is to minimize the aggregate production cost, which includes energy expenses. The model's stochastic process focuses on the intermittency of onsite wind and solar generation, as well as grid wind and solar generation, at each power plant.

European Hydro-Thermal Power Planning Problem

This problem closely mirrors the formulation of the Brazilian Hydro-Thermal Power Planning Problem, adapted to its European counterpart as discussed in Leclère et al. (2020). The Électricité de France (EDF), the primary European electricity producer, aims to plan energy production across a multi-period horizon that encompasses a network of production zones at the European scale. Various countries, interconnected through a network, exchange energy with their neighbors. Each country's energy demand must be met locally, but countries can also import energy from their neighbors.

Dairy Farm Production Planning Problem

The dairy farm production planning problem, taken from Dowson et al. (2019), aims to maximize the net profit derived from selling milk produced by the dairy farm after deducting the costs associated with purchasing supplementation, harvesting pasture, and applying irrigation. This model amalgamates three distinct models: a grass growth model, an animal model, and a milk price model. Collectively, they are referred to as POWDer (milk Production Optimizer incorporating Weather Dynamics). After assessing the farm's state - which includes soil moisture, pasture cover, quantity of stored grass, the number of milking cows, and the quantity of milk produced up to that point - as well as the current realization of random variables like potential evapo-transpiration, the quantity of rainfall, and the milk price forecast, the farmer must decide on the amount of irrigation to apply, the volume of pasture to harvest, the number of cows to dry-off, and the quantities of grass (sourced from both pasture and storage) and palm kernel to feed the cows throughout the entire seasonal year (which spans 52 weeks).

Coordination in Multi-Market Bidding of Grid Energy Storage

Coordination in multi-market bidding of grid energy storage delves into the challenges faced by a storage owner trading in a multi-settlement electricity market (Löhndorf & Wozabal 2023). This market encompasses both an auction-based day-ahead segment and a continuous intraday segment. In their stylized model, they demonstrate that a coordinated policy, which reserves capacity explicitly for the intraday market, emerges as optimal. Moreover, the deviation from a sequential policy grows in proportion to the volatility of intraday prices and market liquidity. To address this, they devised a multi-stage stochastic program tailored for day-ahead bidding and hourly intraday trading, complemented by an apt stochastic price model.

Gas Storage Valuation Problem

The storage valuation problem addresses the challenges faced by a price-taking energy trader in an incomplete market (Löhndorf & Wozabal 2021). This trader manages a gas storage contract that grants the right to inject, withdraw, and store gas over a specified time frame. Assuming that the injection and withdrawal limits are unaffected by the storage level, the contract clearly defines the storage capacity limits. Notably, while physical injection or withdrawal results in a marginal cost and an in-kind fuel loss, storing natural gas does not entail any holding costs.

Capacitated Multi-Echelon Lot Sizing Problem with Component Substitution

The integration of component substitution with lot-sizing under demand uncertainty is a crucial topic. Consolidating the demand for components allows for risk pooling, leading to reduced operating costs. The problem explored in Thevenin et al. (2020) holds significance not just within the production realm but also in the domain of distribution planning. The paper introduces a stochastic programming formulation tailored for static and dynamic uncertainty; this means setup decisions remain unchanged, but decisions concerning production and consumption quantities are made dynamically. Addressing the typical scalability challenges inherent to multi-stage stochastic optimization, Thevenin et al. (2020) delves into the potential benefits of stochastic dual dynamic programming (SDDP).

Integrated Management of Hydro-plant Cascade

Borges (2022) evaluates the integrated management of a cascade consisting of 3 hydro-plants over a span of 12 months. Each hydro-plant comprises a reservoir paired with a downward-facing turbine. The interconnected nature of these hydro-plants is facilitated by tubulations, ensuring that water discharged from one hydro-plant for energy production is subsequently stored in the reservoirs of the other plants. Although it is advantageous for profit-driven firms to generate energy during months with peak prices, it is infeasible to focus production solely in the highest-priced month. This is due to potential reservoir overflows or the imposition of environmental restrictions that set minimum and maximum thresholds on daily water discharge volumes.

American Put Option Pricing

An American option is characterized by its early exercise feature, allowing the holder to execute the option before its expiration date. Given this added flexibility, it's logical to anticipate that an American option would be valued higher than its European counterpart, which lacks this feature. The early exercise becomes favorable when the continuation (or time) value dips below the option's intrinsic value. Consequently, at each time interval, the option's value is determined by the greater of these two values: intrinsic or continuation. Yin et al. (2006) considers an example where the spot price stands at 36, the strike price is set at 40, with a volatility of 0.2, an interest rate of 6%, an expiration period of 1 year, and a time step of 0.02.

3.3 MSPFormat: a new data format MSPs

A standardized data structure format for mathematical programs is essential for the exchange of problem test sets used in benchmarking and algorithm testing. The oldest and most widely used format for linear programming (LP) problems is the Mathematical Programming System (MPS) file format from IBM (Orchard-Hays 1984). It has been extended to the 'xMPS' format to cater to non-linear programs (Halldorsson et al. 2000) and to Stochastic Mathematical Programming System (SMPS) file format for stochastic programming (SP) problems (Gassmann 2005). Recently, a new standardized data structure format LPs named 'MathOptInterface' has been proposed (Legat et al. 2022). The same team also introduced the data structure format for SP and MSP under the banner 'StochOptFormat' Dowson & Garcia (2020). Our proposed MSPFormat is our attempt to provide a new standardized data structure format for SP and MSP problems. Like its predecessors, the primary objective of MSPFormat is to ensure easy accessibility and transfer of mathematical models, especially catering to the users of the MSPLib problem library in a simple, clear, and familiar manner. The MSPFormat is serialized and compressed using the JavaScript Object Notation (JSON) format. JSON, an open-source standard for file and data interchange, is frequently used for data exchange across various computer languages. Given its widespread support, parsing JSON files becomes straightforward across different platforms. The creation of MSPFormat was motivated by the following objectives:

- To develop a format capable of describing problems with multiple decision periods, a myriad of state and control variables, in a scalable way;
- To introduce a data file format having an intuitive and familiar architecture, thereby facilitating the verification of both the mathematical model and the stochastic process;
- To establish a standardized data format that distinctly separates the mathematical model from the description of the stochastic process, specifically, discretized samples for MSP problems;
- To craft a comprehensive format that supports various extensions of classic

stochastic programming, such as multi-stage, integrality, and more;

- To maintain consistency across diverse implementations, ensuring smooth data exchange and interoperability;

- To assist researchers and practitioners by eliminating the need for custom data representations, allowing them to focus on advancing SDDP techniques.

In conclusion, it's worth noting that for each problem instance in the MSPLib, we propose two distinct data structures in JSON format: one for the mathematical model and the other for the stochastic data process. Appendix 3.A.1 illustrates a sample MSPFormat data structure for the mathematical model and for the stochastic data process.

3.3.1 Sample Problem

To provide deeper insights into the structure of the MSPFormat, we present Problem 1 - Instance 0 (01_0) from the MSPLib - the simplified hydrothermal power production problem.

For both thermal and hydro-power generators with given deterministic marginal operations costs, we aim to determine the generation quantities, $thermal_{gen}$ and $hydro_{gen}$, and reservoir storage level $volume_t$, to satisfy a known demand D across each period t (with $T = 3$). The hydro-power generator draws from a reservoir with capacity C, which is full at the start. We account for potential water spillages, $hydro_{spill}$, to avoid dam over-topping. The goal is to minimize the expected operational costs, including fuel costs c_f and supply failure penalties c_p. The key stochastic component is water $inflow$, affecting the trade-off between immediate water use for electricity generation and saving for future uncertainties.

The full MSPFormat data structure mathematical model and stochastic data are in Appendix 3.A.1.

3.3.2 Vocabulary

- *State variables*

- A state variable capture the current state of the system. As the system progresses through different stages, this state may change, influencing the decisions made at each stage of the MSP.
- In our sample problem, the state variable is denoted as - *volume*. The transition of this state variable from its value at the beginning of a stage (termed *incoming*) to its value at the end of that stage (termed *outgoing*) is determined by a series of constraints. The *incoming* state variable typically reflects its value from the preceding stage. In our sample problem, it is represented as $volume_{t-1}$. On the other hand, the *outgoing* state variable generally corresponds to its value in the current stage, represented as $volume_t$.

• *Stage or Local variables*

- A stage or local variable refers to a decision variable that is defined and optimized for a specific stage in an MSP problem. This variable denotes the decision made at that particular stage, based on the accumulated information and uncertainties up to that point. The value of the stage variable may be updated at each successive stage as new information emerges, especially in response to stochastic or unpredictable events. Stage variables enable adaptive decision-making, accounting for the evolving nature of uncertainties across multiple stages. This adaptability often results in more robust and optimal solutions in stochastic environments.
- In our sample problem, the stage variables are defined as $hydro\_gen_t$, $thermal\_gen_t$, and $hydro\_spill_t$. These variables are optimized exclusively at the current stage t. Their values are not required for decision-making in subsequent stages.

• *Random variables*

- A random variable represents an uncertain or stochastic parameter that evolves over multiple stages. Typically, random variables are modeled using either discrete or continuous probability distributions, which depict the likelihood of each possible value the variable can assume. In the context of an MSP, random variables may be present in the right-hand side of the constraints or b, serve as parameters of the decision variables or the matrix A, or function as parameters of the objective function represented by the vector c.

- In our sample problem, the random variable is $inflow$, which is found on the right-hand side of the constraints. Following the description above, and depending on the structure of the MSP, random variables can manifest in three distinct areas within the MSPFormat:
  - As the value of the "obj" key in the "variables" object;
  - As the value of the "coefficient" key in the "constraints" object;
  - As a value-pair in the "rhs" of the "constraints" object.

• *Constraints*

- Constraints are mathematical expressions that impose limitations or conditions on the decision variables. Their purpose is to ensure that the decision-making process produces feasible solutions, satisfying specific requirements. Constraints capture various limitations or restrictions that must be observed to obtain a valid solution to the decision problem.
- Within the MSPFormat, constraints are contained in the "constraints" object. Every constraint has its dedicated object, comprising key-value pairs that detail the constraint's "name", "type", and the expressions on both sides of the equation: "lhs" for the left-hand side expressions and "rhs" for right-hand side expressions. In the "lhs", each term specifies the "name" of the variable, the "stage" where the variable appears, and its "coefficient". The "rhs", on the other hand, is structured as key-value pairs, wherein the key indicated the operation (e.g., "ADD", "MUL", "EXP", and "POW"), and its value (be it a random variable or a resource availability). In the context of our sample problem, there are two equality ("EQ") constraints per stage, totaling six. These correspond to (1) the state transition constraint and (2) the demand constraint.

- *State and Successors*
  - Within the MSPFormat, each discretized sample of random variables for every stage t is housed within its unique object. The "state" object contains key-value pairs representing each random variable. On the other hand, the "successors" object contains key-value pairs that detail the successor state of the random variable along with the corresponding transition probability.
  - For our sample problem, every discretized sample of the random variable $inflow$ for every stage t exists as a separate object. The "state" object contains $inflow$ as key and its discretized sample as the associated value. Concurrently, the "successors" object contains key-value pairs detailing the successor state of $inflow$ and the respective transition probabilities.

3.4 SDDP Implementations and Solvers

We benchmark three different SDDP implementations: QUASAR, SDDP.jl, and MSPPy. The selection of these implementations was based on their prominence in the field, ease of use, and accessibility.

QUASAR is a flexible, high-performance modeling system for formulating and solving optimization problems with stochastic parameters. This system was developed by the company Quantego. While QUASAR draws from published scientific research, it also incorporates unique, unpublished features that contribute to scalability and solution quality. The platform provides a documentation with code examples in Java, Python, and MATLAB. QUASAR is available for commercial purchase, but an academic license is also offered Löhndorf (2021).

SDDP.jl is an open-source library designed to solve MPSs using SDDP. This library was co-developed by Oscar Dowson and Lea Kapelevich. Based on JuMP, an algebraic modelling language in Julia, SDDP.jl offers users a high-level interface while maintaining performance levels comparable to low-level languages implementations Dowson & Kapelevich (2021).

MSPPy is a Python-based package that facilitates the construction, resolution, and analysis of MSPs. It was developed by Lingquan Ding during his PhD dissertation titled "Multistage Stochastic Programming" at the Georgia Institute of Technology, under the supervision of Prof. Alexander Shapiro. MSPPy is available under the open source Modified BSD (3-clause) license Ding et al. (2019).

3.5 SDDP Benchmarking Numerical Results

In this section, we present the benchmarking results for the three pre-selected SDDP implementations: QUASAR, SDDP.jl, and MSPPy. Our evaluation focuses on their relative performance across three distinct criteria by solving the problem instances in MSPLib:

- *Iteration Limit*: This criterion evaluates how rapidly the implementation achieves a given number of iterations and determines if the solution has converged by that point.
- *Time Limit*: This stopping criterion assesses the optimality gap after a specified time has elapsed, showing how quickly each implementation approaches an optimal solution.
- *Parallel Processing*: This performance criterion examines the efficiency of each implementation when parallel solving capabilities are utilized, especially when solving the problem instances in MSPLib.

Furthermore, with the pre-classification of the problems in MSPLib based on their relative difficulty, the evaluation criteria differ for each category:

- **Testing criteria for *Easy* Category Problems**:
  - Each solver/implementation is set to its default settings, including parameters such as the choice of solver, cuts selection and aggregation, tolerances, etc.
  - An iteration limit of 25 iterations is imposed per problem.
- **Testing criteria for *Medium* Category Problems**:

- Iteration Limit: evaluated at 100 and 200 iterations.
- Time Limit: assessed at 10 and 60 seconds.
- Parallel Processing: performance evaluated using 1 and 32 processors.

- **Testing criteria for *Difficult* Category Problems**:
  - Iteration Limit: evaluated at 100, 200, and 300 iterations.
  - Time Limit: assessed at 30, 60, and 150 seconds.
  - Parallel Processing: performance evaluated using 1 and 32 processors.

3.5.1 Solver Performances

Easy Category Problems

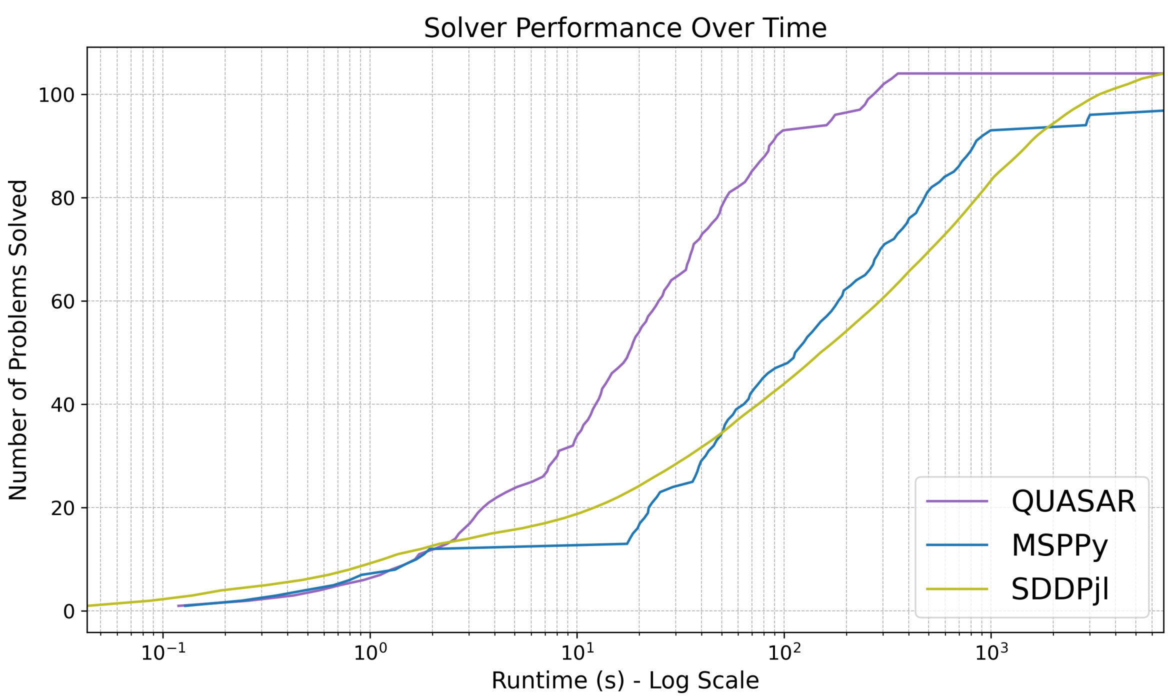


Figure 3.1: Relative performance of the three SDDP solvers for *easy* category problems in MSPLib.

In the MSPLib database, there exists a collection of 105 problem instances that are distinctly categorized as *easy* problems. Specifically, the *easy* category problems include the simplified hydrothermal scheduling problem (01), airconditioning production problem (02), electricity production problem (03), 3 instances of the newsvendor problems (04), semiconductor problem (05), 2 instances of the farmer's problem (06), 6 instances of the asset management problem (07), and 7 instances of the generation expansion problem (08). It is important to note that each scenario discretization correspond to a single problem instance.

A graphical representation of the results can be seen in Figure 3.1. One of the most noteworthy observations from these results is QUASAR's commendable performance. It manages to efficiently tackle and successfully solve all the problem instances falling under the *easy* category, all within an approximate time frame of 350 seconds. To offer a comparative analysis, both MSPPy and SDDP.jl, while being competent solvers, demonstrate a slightly lower performance for these specific problem instances. Within the same 350-second window, they both are able to resolve about 65 instances. Digging a bit deeper into their performance metrics, SDDP.jl finishes all the problem instances, but it requires a more extended period, roughly 6,000 seconds. MSPPy, on the other hand, needs even more time, taking around 12,000 seconds to complete all the *easy* category problem instances.

While all three solvers ultimately achieve the end goal of solving the problems, the disparity in the time taken highlights QUASAR's superior performance for these specific instances. When contrasted with MSPPy and SDDP.jl, QUASAR emerges as a more time-efficient option for solving problem instances classified as *easy* within the MSPLib.

Medium Category Problems

Within the MSPLib, there are 60 problem instances classified as *medium* difficulty. These problems include 3 instances of the newsvendor problem (04), 8 instances of the farmer's problem (06), 4 instances of the asset management problem (07), 3 instances of the generation expansion problem, and 2 instances of the flower seller problem (09), for the synthetic problems. For the real-world problems, problems categorized as *medium*

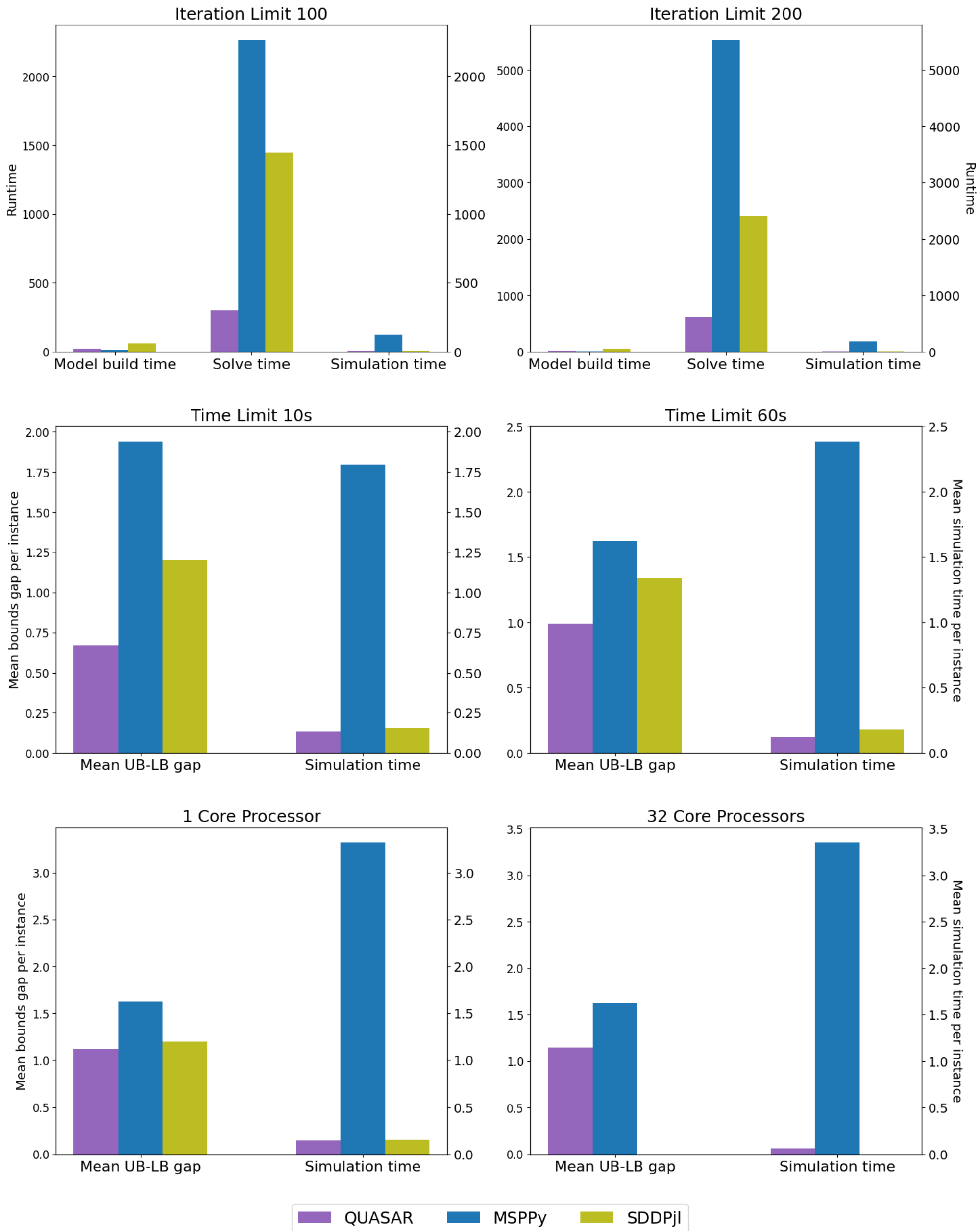


Figure 3.2: Relative performance of the three SDDP solvers for *medium* category problems in MSPLib.

difficulty include production-inventory planning with intermittent renewable energy (13), European hydropower scheduling problem (14), multistage dairy farm problem

(15), capacitated multi-echelon lot sizing problem with component substitution (18), integrated management of hydro-plant cascade (19), and American put option pricing (20).

In Figure 3.2, QUASAR consistently surpasses the other two solvers in all performance metrics. The upper subgraphs showcase the model build time, solve time, and simulation time for each solver, based on iteration limit criteria set at 100 and 200 iterations. Although, the model building time remain largely consistent across all three solvers, stark differences are evident in their solve times. QUASAR emerges as the frontrunner, followed by SDDP.jl, with MSPPy bringing up the rear. The simulation times of QUASAR and SDDP.jl are almost similar while MSPPy performs rather poorly. The central subgraphs highlight the average bound gap and simulation time for each solver, adhering to the time limit criteria of 10 and 60 seconds. Impressively, QUASAR attains a more favorable average bound gap per instance than the others, with SDDP.jl coming in second, followed by MSPPy.

Finally, the bottom subgraphs show the mean bound gap and simulation time for each solver when considering the parallel processing criteria (utilizing either 1 or 32 core processors). For a single-core processor, the distinctions in achieving the least mean bound gap per instance among the three solvers are marginal. However, it is essential to note that, at this juncture, SDDP.jl does not yet support parallel processing for MSPFormat.

Difficult Category Problems

Within the MSPLib, there are 70 problem instances classified as *difficult* level. These problems include 4 instances of the newsvendor problem (04), 8 instances of the farmer's problem (06), 4 instances of the asset management problem (07), 8 instances of the flower seller problem (09), and the inventory problem with lead time and lost sales (10), for the synthetic problems. For the real-world problems, problems categorized as *difficult* level include the original Brazilian hydrothermal scheduling problem with stage-wise independent uncertainty (11) and all its variants (12), coordination in multi-market bidding of grid energy storage (16), and gas storage valuation problem (17). Presently, only QUASAR can address (16) and (17) because of technical

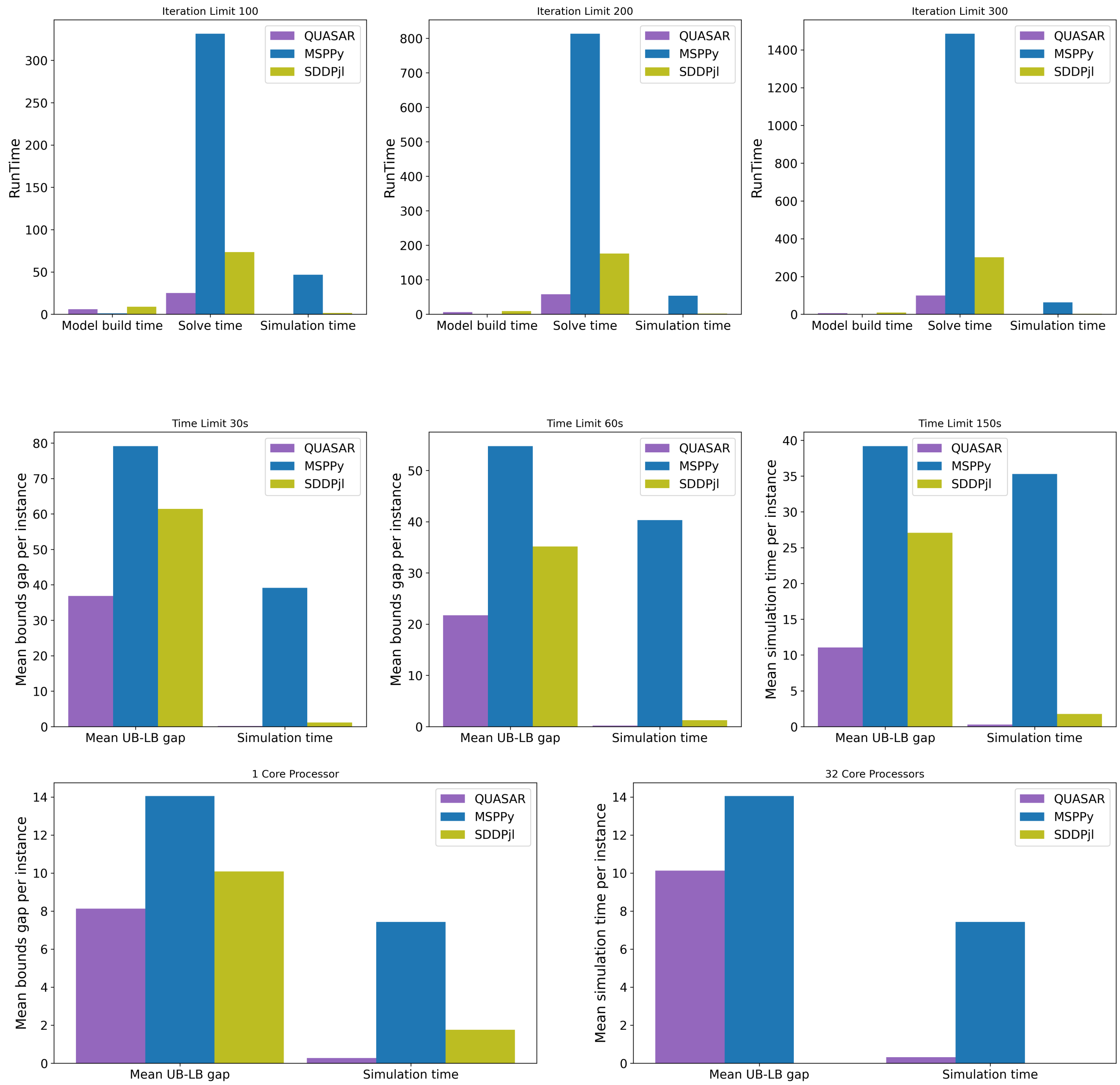


Figure 3.3: Relative performance of the three SDDP solvers for *difficult* category problems in MSPLib.

challenges with the MSPFormat parser in SDDP.jl and the prolonged runtime of MSPPy for the difficult-level problems. Consequently, (16) and (17) are omitted from the benchmarking.

As in Figure 3.3, QUASAR still outperforms the other two solvers across all performance metrics. The uppermost subgraphs detail the model build time, solve time, and simulation time for each solver based on the iteration limit criteria (set at 100, 200, and 300 iterations). While the model build and simulation times are comparable for all three solvers, significant disparities emerge when examining solve times. QUASAR leads in performance, trailed by SDDP.jl and then MSPPy. The central subgraphs present the average bound gap and simulation time for each solver, as dictated by the time limit criteria (10, 60, and 150 seconds). Notably, QUASAR achieves a lower mean bound gap per instance than its counterparts, with SDDP.jl trailing behind and then MSPPy.

Finally, the bottom subgraphs show the mean bound gap and simulation time for each solver when considering the parallel processing criteria (utilizing either 1 or 32 core processors). For a single-core processor, the distinctions in achieving the least mean bound gap per instance among the three solvers are marginal. However, it's essential to note that, at this juncture, SDDP.jl does not yet support parallel processing for MSPFormat.

3.6 Conclusion

In this chapter, we introduce both MSPLib and MSPFormat. MSPLib is a library of MSP problems designed to streamline the evaluation and benchmarking process for various stochastic dual dynamic programming (SDDP) implementations. This library encapsulates a wide array of instances, from real-world problems to synthetic variants, each exhibiting distinct complexity levels. MSPFormat establishes a standardized data format tailored for MSPs. By adopting MSPFormat, we pave the way for a consistent and cohesive representation of MSPs. Additionally, we evaluate leading SDDP implementations, namely QUASAR, SDDP.jl, and MSPPy. QUASAR outperforms the other two solvers across all performance metrics - iteration limit criterion, time limit criterion, and parallel processing criterion.

In this chapter, we delve into a comprehensive introduction of two instrumental tools: MSPLib and MSPFormat. The MSPLib serves as a library housing an extensive collection of MSP problems. Its primary objective is to simplify and make more efficient the process of evaluating and benchmarking various SDDP implementations. This repository comprises a diverse range of instances. From intricate real-world problems that resonate with practical applications to carefully crafted synthetic variants, the library is designed to challenge SDDP implementations with varying degrees of complexity.

On the other hand, the MSPFormat is an initiative that introduces a standardized data format, crafted with MSPs in mind. The adoption of MSPFormat aims to foster a uniform and systematic representation of MSPs, promoting clarity and compatibility.

Further into the chapter, we embark on a rigorous evaluation of some of the leading SDDP implementations in the field: QUASAR, SDDP.jl, and MSPPy. Our results indicate that QUASAR stands out distinctly, surpassing its counterparts in a various of performance metrics. Whether we assess based on the iteration limit criterion, time limit criterion, or even the intricate parallel processing criterion, QUASAR consistently demonstrates superior performance. SDDP.jl trails behind in performance, followed by MSPPy.

Appendix 3.A Sample Problem in MSPFormat

3.A.1 Model and Lattice - Simple Hydro-thermal Power Problem

```
{"version":"MSMLP 1.1",
    "name":"Simple Hydrothermal
    Scheduling Problem",
    "maximize":false,
    "variables":[
        {"name": "volume",
         "stage":0,
         "obj":[0.0],
         "lb":[0.0],
         "ub":[200.0],
         "type":"CONTINUOUS"},
        {"name":"thermal_gen",
         "stage":0,
         "obj":[50.0],
         "lb":[0.0],
         "ub":["inf"],
         "type":"CONTINUOUS"},
        {"name":"hydro_gen",
         "stage":0,
         "obj":[0.0],
         "lb":[0.0],
         "ub":["inf"],
         "type":"CONTINUOUS"},
        {"name":"hydro_spill",
         "stage":0,
         "obj":[0.0],
         "lb":[0.0],
         "ub":["inf"],
         "type":"CONTINUOUS"},
        {"name":"volume",
         "stage":1,
         "obj":[0.0],
         "lb":[0.0],
         "ub":[200.0],
         "type":"CONTINUOUS"},
        {"name":"thermal_gen",
         "stage":1,
         "obj":[100.0],
         "lb":[0.0],
         "ub":["inf"],
         "type":"CONTINUOUS"}
        {"name":"hydro_gen",
         "stage":1,
         "obj":[0.0],
         "lb":[0.0],
         "ub":["inf"],
         "type":"CONTINUOUS"},
        {"name":"hydro_spill",
         "stage":1,
         "obj":[0.0],
         "lb":[0.0],
         "ub":["inf"],
         "type":"CONTINUOUS"},
        {"name":"volume",
         "stage":2,
         "obj":[0.0],
         "lb":[0.0],
         "ub":[200.0],
         "type":"CONTINUOUS"},
        {"name":"thermal_gen",
         "stage":2,
         "obj":[150.0],
         "lb":[0.0],
         "ub":["inf"],
         "type":"CONTINUOUS"},
        {"name":"hydro_gen",
         "stage":2,
         "obj":[0.0],
         "lb":[0.0],
         "ub":["inf"],
         "type":"CONTINUOUS"},
        {"name":"hydro_spill",
         "stage":2,
         "obj":[0.0],
         "lb":[0.0],
         "ub":["inf"],
```

```
            "type":"CONTINUOUS"}],
    "constraints":[
        {"name":"",
         "type":"EQ",
         "lhs":[
             {"name":"volume",
              "stage":0,
               "coefficient":[1.0]},
             {"name":"hydro_gen",
              "stage":0,
              "coefficient":[1.0]},
             {"name":"hydro_spill",
              "stage":0,
              "coefficient":[1.0]}],
        "rhs":["inflow",
               200.0,
               0.0]},
        {"name":"",
         "type":"EQ",
         "lhs":[
             {"name":"hydro_gen",
              "stage":0,
               "coefficient":[1.0]},
             {"name":"thermal_gen",
              "stage":0,
              "coefficient":[1.0]}],
              "rhs":[150.0]},
        {"name":"",
         "type":"EQ",
        "lhs":[
             {"name":"volume",
              "stage":1,
              "coefficient":[1.0]},
             {"name":"volume",
              "stage":0,
              "coefficient":[-1.0]},
             {"name":"hydro_gen",
              "stage":1,
              "coefficient":[1.0]},
             {"name":"hydro_spill",
              "stage":1,
              "coefficient":[1.0]}],
         "rhs":["inflow",
                0.0]},
        {"name":"",
         "type":"EQ",
         "lhs":[
             {"name":"hydro_gen",
              "stage":1,
              "coefficient":[1.0]},
             {"name":"thermal_gen",
              "stage":1,
              "coefficient":
              [1.0]}],
         "rhs":
             [150.0]},
        {"name":"",
         "type":"EQ",
         "lhs":[
             {"name":"volume",
              "stage":2,
              "coefficient":[1.0]},
             {"name":"volume",
              "stage":1,
              "coefficient":[-1.0]},
             {"name":"hydro_gen",
              "stage":2,
              "coefficient":[1.0]},
             {"name":"hydro_spill",
              "stage":2,
              "coefficient":[1.0]}],
         "rhs":["inflow",
                0.0]},
        {"name":"",
         "type":"EQ",
         "lhs":[
             {"name":"hydro_gen",
              "stage":2,
              "coefficient":[1.0]},
             {"name":"thermal_gen",
              "stage":2,
              "coefficient":[1.0]}],
         "rhs":
             [150.0]}]}
```

```
{"0":
    {"stage":0,
    "state":{"inflow":0.0},
    "successors":
        {"1":0.3333333333333333,
         "5":0.3333333333333333,
         "9":0.3333333333333333}},
"4":
    {"stage":0,
    "state":{"inflow":0.0},
    "successors":
        {"1":0.3333333333333333,
         "5":0.3333333333333333,
         "9":0.3333333333333333}},
"8":
    {"stage":0,
    "state":{"inflow":0.0},
    "successors":
        {"1":0.3333333333333333,
         "5":0.3333333333333333,
         "9":0.3333333333333333}},
"1":
    {"stage":1,
    "state":{"inflow":0.0},
    "successors":
        {"2":0.3333333333333333,
         "6":0.3333333333333333,
         "10":0.3333333333333333}},
"5":
    {"stage":1,
    "state":{"inflow":50.0},
    "successors":
        {"2":0.3333333333333333,
         "6":0.3333333333333333,
         "10":0.3333333333333333}},
"9":
    {"stage":1,
    "state":{"inflow":100.0},
    "successors":
        {"2":0.3333333333333333,
         "6":0.3333333333333333,
         "10":0.3333333333333333}},
"2":
    {"stage":2,
    "state":{"inflow":0.0},
    "successors":
        {"3":0.3333333333333333,
         "7":0.3333333333333333,
         "11":0.3333333333333333}},
"6":
    {"stage":2,
    "state":{"inflow":50.0},
    "successors":
        {"3":0.3333333333333333,
         "7":0.3333333333333333,
         "11":0.3333333333333333}},
"10":
    {"stage":2,
    "state":{"inflow":100.0},
    "successors":
        {"3":0.3333333333333333,
         "7":0.3333333333333333,
         "11":0.3333333333333333}},
"3":
    {"stage":3,
    "state":{"inflow":0.0},
    "successors":{}},
"7":
    {"stage":3,
    "state":{"inflow":50.0},
    "successors":{}},
"11":
    {"stage":3,
    "state":
        {"inflow":100.0},
    "successors":{}}}
```

3.A.2 MSP Problem Formulations

Simplified Hydro-thermal Power Production Problem

$$\begin{aligned}
\min\ & \sum_{t=1}^{T} c_t^f * thermal_gen_t \\
s.t.\ & volume_t = volume_{t-1} - hydro_gen_t - spill_t + \xi, && \forall t = 1, \dots, T \\
& hydro_gen_t + thermal_gen_t = D_t, && \forall t = 1, \dots, T \\
& volume_t = C_t, && \forall t = 1, \dots, T \\
& volume_0 = 200, \\
& volume_t, hydro_gen_t, thermal_gen_t, spill_t \geq 0
\end{aligned} \tag{3.1}$$

where, t: indexes the stages ($T = 3$ planning horizon), $volume_t$: reservoir storage level at time t, $thermal\_gen_t$: thermal power generated at time t, $hydro\_gen_t$: hydropower generated at time t, $spill_t$: water spilled at time t, D_t: power demand at time t, C_t: reservoir capacity at time t,

Airconditioning Production Problem

A manager of a factory seeks a production plan for producing air-conditioners over a planning period T. During standard working hours, the factory can produce X units per month at a deterministic cost of p/unit. Unlimited overtime can be scheduled, however the cost increases to I per unit during those hours. In the first month, there is a known deterministic demand of D units. However, in each of months two and three, there is an equally likely demand of either ξ_1 or ξ_2 units. Air-conditioners can be stored between months at a cost of S/unit, and all demands must be met.

$$\begin{aligned}
\min\ & \sum_{t=1}^{T} x_t + I w_t + S y_t \\
s.t.\ & x_t \leq X && \forall t = 1, \dots, T \\
& x_t + w_t + y_{t-1} - y_t = d_t && \forall t = 1, \dots, T \\
& y_0 = 0, d_1 = D_1, d_2, \dots, d_T \leq \xi
\end{aligned} \tag{3.2}$$

Electricity Planning Problem

The following two-stage problem consists of determining the optimal capacity investment in various P types of power plants so as to meet next period demands for electricity. Four power plants are considered and they can operate in C different modes j. The next period demand for each of the three modes are to be met. There is a budget constraint and also a constraint on the minimum total capacity.

$$
\begin{aligned}
& \min \sum_{i=1}^{P} c_i x_i + Q(x) \\
& s.t. \sum_{i=1}^{P} a_{ik} x_i \leq b \qquad \forall k \in R \\
Q(x,\xi) = & \\
& \min \sum_{i=1}^{P} \sum_{j=1}^{C} p_{ij} y_{ij} \\
& s.t. \sum_{j=1}^{C} y_i j \leq x_i \qquad \forall i \in P \\
& \sum_{i=1}^{P} y_{ij} \leq \xi \qquad \forall j \in C
\end{aligned} \tag{3.3}
$$

Newsvendor Problem

Suppose a newsvendor wants to purchase some newspaper today and sell it tomorrow x. The demand for newspaper tomorrow, ξ, is uniformly distributed. The retail price r, production cost p, and recycled value r of one newspaper are all deterministic. The goal is to determine how many newspaper should the newsboy buy b today to maximize profit.

$$
\begin{aligned}
\max \quad & s x_1 + r y_1 - p b_0 \\
s.t. \quad & x_1 + u_1 = \xi \\
& x_1 + y_1 = b_0 \\
& b_t, x_t, y_t, u_t \geq 0
\end{aligned} \tag{3.4}
$$

Semi-conductor Production Problem

Consider a wafer fab consisting of M tool types, that can process N types of wafers. Each product (wafer type) goes through a subset of K processing steps, each of which can be performed on one or more tool types. Let a_{ijk} denote the time (in hrs) required by processing step k (1,...,K) on wafer type j (1,...,N) on tool type i (1,...,M). We set $a_{ijk} = 0$ if step k is not needed for wafer type j, and $a_{ijk} = \infty$ if step k is required for wafer type j but cannot be performed on tool type i. Consider now a planning horizon of T periods. Let us use variables x_{it}, u_{jt}, v_{ijkt}, and w_{jt}, to denote the number of tool type i purchased in period t (1,...,T), the shortage (in units of wafer starts) of wafer type j in period t, the allocation of processing step k of wafer type j to tool type i in period t, and the production of wafer type j in period t, respectively. In addition to a_{ijk}, let us also consider the problem parameters α_{it}, β_{jt}, c_i, and d_{jt} corresponding to the (discounted) cost of tool type i in period t, the penalty cost of unit shortage in wafer type j in period t, the per-period capacity (in hours) of one tool of type i, and the per-period demand (in wafer starts) of wafer type j in period t, respectively. The objective is to minimize total tool purchase costs and shortage penalties.

$$\begin{aligned}
\min \ & \sum_{t=1}^{T}[\sum_{i=1}^{M}\alpha_{it}x_{it} + \sum_{j=1}^{N}\beta_{jt}u_{jt}] & \\
s.t. \ & \sum_{i=1}^{M} v_{ijkt} \geq w_{jt} & \forall i, t \in M, T \\
& \sum_{j=1}^{N}\sum_{k=1}^{K} a_{ijk}v_{ijkt} \leq c_i & \forall j, k, t \in N, K, T \\
& w_{jt} + u_{jt} \geq d_{jt} & \forall j, t \in NT \\
& x_{it}, u_{jt}, v_{ijkt}, w_{jt} \geq 0 & \forall i, j, k, t \in M, N, K, T
\end{aligned} \tag{3.5}$$

The Farmer's Problem (Birge & Louveaux 2011)

Consider a European farmer who specializes in raising wheat, corn, and sugar beets on his 500 acres of land. During the winter, the farmer wants to decide how much land to devote to each crop. The farmer knows that at least 200 tons (T) of wheat and 240 T of corn are needed for cattle feed. These amounts can be raised on the farm or bought from a wholesaler. Any production in excess of the feeding requirement would be sold. Over the last decade, mean selling prices have been \$170 and \$150 per ton of wheat and corn, respectively. The purchase prices are 40% more than this due to the wholesaler's margin and transportation costs. Another profitable crop is sugar beet, which he expects to sell at \$36/T; however, the European Commission imposes a quota on sugar beet production. Any amount in excess of the quota can be sold only at \$10/T. The farmer's quota for next year is 6000 T. Based on past experience, the farmer knows that the mean yield on his land is roughly 2.5 T, 3 T, and 20 T per acre for wheat, corn, and sugar beets, respectively. Furthermore, the farmer has indeed experienced quite different yields for the same crop over different years mainly be- cause of changing weather conditions. Most crops need rain during the few weeks after seeding or planting, then sunshine is welcome for the rest of the growing period. Sunshine should, however, not turn into drought, which causes severe yield reductions. Dry weather is again beneficial during harvest. From all these factors, yields varying 20 to 25% above or below the mean yield are not unusual.

$$
\begin{aligned}
&\min \sum_{i=1}^{I} c^\top x_i + \mathbb{E}_\xi Q(x_i, \xi) \\
&s.t. \sum_{i=1}^{I} x_i \leq 500 \\
Q(x_i, \xi) = &\min 238y_1 - 170w_1 + 210y_2 - 150w_2 - 36w_3 - 10w_4 \\
&s.t.\ \xi_1 x_1 + y_1 - w1 \geq 200, \\
&\xi_2 x_2 + y_2 - w2 \geq 240, \\
&w_3 + w4 \leq \xi_3 x_3, \\
&w_3 \leq 6000, \\
&y, w \geq 0,
\end{aligned}
\tag{3.6}
$$

where x_1 = acres of land devoted to wheat, x_2 = acres of land devoted to corn, x_3 = acres of land devoted to sugar beets, w_1 = tons of wheat sold, y_1 = tons of wheat purchased, w_2 = tons of corn sold, y_2 = tons of corn purchased, w_3 = tons of sugar beets sold at the favorable price, and w_4 = tons of sugar beets sold at the lower price.

Asset Management Problem (Birge & Louveaux 2011)

Suppose we wish to provide for a child's college education Y years from now. We currently have \$$b$ to invest in any of I investments. After Y years, we will have a wealth that we would like to have exceed a tuition goal of \$ G . We suppose that we can change investments every v years, so we have $H = Y/v$ investment periods. For our this problem, we ignore transaction costs and taxes on income although these considerations would be important in reality. We also assume that all figures are in constant dollars.

$$\begin{aligned}
\min\ & \sum_{s} p(s)(qy(s) - rw(s)) & \\
s.t.\ & \sum_{i=1}^{I} x(i,1,s) = b, & \forall s \in S \\
& \sum_{i=1}^{I} \xi(i,t,s)x(i,t-1,s) - \sum_{i=1}^{I} x(i,t,s) = 0, & \forall s \in S, t = 2, ..., H \\
& \sum_{i=1}^{I} \xi(i,H,s)x(i,H,s) - y(s) + w(s) = G, & \\
& \Big(\sum_{s' \in S^t_{J(s,t)}} p(s')x(i,t,s')\Big) - \Big(\sum_{s' \in S^t_{J(s,t)}} p(s')\Big)x(i,t,s) = 0, & \\
& \qquad \forall 1 \le i \le I, \forall 1 \le t \le H, \forall s \in S & \\
& x(i,t,s) \ge 0, y(s) \ge 0, w(s) \ge 0, & \\
& \qquad \forall 1 \le i \le I, \forall 1 \le t \le H, \forall s \in S &
\end{aligned} \tag{3.7}$$

where $J(s,t) = \{s_1, ..., s_{t-1}\}$ such that $s \in S^t_{s_1,...,s_{t-1}}$.

Generation Expansion Problem

Generation expansion aims to optimally choose the timing and levels of investments to meet future demands of a given product. We illustrate the case of power plant expansion for electricity generation: we want to find optimal levels of investment x of generator i in a power plant to meet random future electricity demand D. We want to minimize the cost of investment b, operation cost c, and penalty cost p of unmet demand.

$$
\begin{aligned}
\min \ & \sum_{t=1}^{T} \Big(\sum_{i=1}^{I} b x_{it} \Big) \rho^{t-1} + c g_t + p u_t & \\
s.t. \ & x_{0,t} \geq 0 & \forall i \in I \\
& \sum_{i=1}^{I} x_{it} C_i \geq g_t & \forall t \in T \\
& u_t \geq D_\xi - g_t & \forall t \in T \\
& x_{it} \geq x_{it-1} & \forall i \in I, t \in T \\
& x_{jt} \leq x_{j+1t} & \forall j \in I-1, t \in T \\
& x_{it}, u_t, g_t >= 0 & \forall i \in I, t \in T
\end{aligned}
\tag{3.8}
$$

where x_{it} = investment decision for generator i at stage t, g_t = generation at stage t, u_t = unmet demand at stage t, C_i = capacity of generator i.

Flower Girl Problem

The flower girl problem is a variation of the news-vendor problem where a flower girl buys flowers from a supplier at p^R and sells them for p^S. The flower-girl needs to buy the flowers before she starts selling and whatever is left at the end of the day can be stored to be sold the next day as old stocks. Any roses from the stock are thrown away. The objective is to determine the number of flowers she should buy. ξ_t is the random the demand at time t, I_t is the inventory at time t, R_t is the ordering decision, S_t is the selling decision, In_t is the amount of flowers put into storage in stage t, Out_t is the amount of flowers taken out of storage in stage t, $Disc_t$ is the amount of flowers discarded in stage t, and b is the storage capacity available.

$$
\begin{aligned}
\max\ & p^S S_1 - p^R R_1 + \mathbb{E}[p^S \sum_{t=2}^{T-1} S_t - p^R \sum_{t=2}^{T-1} R_t] \\
s.t.\ & I_{t+1} = cI_t - Out_t + In_t - Disc_t, && \forall t \in [1, ..., T-1] \\
& Out_t + R_{t-1} = S_t + In_t, && \forall t \in [2, ..., T] \\
& S_t \leq \min(I_t, \xi_t), && \forall t \in [2, ..., T] \\
& I_t \leq b, && \forall t \in [2, ..., T] \\
& S_t, I_t, R_t \geq 0, && \forall t \in [2, ..., T]
\end{aligned} \tag{3.9}
$$

Inventory Planning Problem with Lead Time and Lost Sales

The classical inventory problem states that a company needs to satisfy demands of its customer by determining the amount of products to buy in each stage while minimizing the total cost of purchasing and holding inventory. In this more sophisticated version of the classical inventory problem, we introduce lead time for ordering and receiving bought products. The inclusion of lead time adds to the difficulty of the problem especially as lead time increases.

$$
\begin{aligned}
\max\ & p^s sell_t - p^b buy_t - hI_t \\
s.t.\ & I_t = I_{t-1} - sell_t + buy_{t-LT}, && \forall t \in [2, ..., T] \\
& sell_t \leq \xi, && \forall t \in [1, ..., T] \\
& sell_t \leq I_t, && \forall t \in [1, ..., T] \\
& buy_t \leq b, && \forall t \in [1, ..., T] \\
& I_t, sell_t, buy_t \geq 0, && \forall t \in [1, ..., T]
\end{aligned} \tag{3.10}
$$

ξ_t is the random the demand at time t, I_t is the inventory at time t, buy_t is the ordering decision, $sell_t$ is the selling decision, b is the maximum buying capacity, and LT is the lead time of ordering to delivery.

PROBLEM	Problem Number	Instance Number	Instance Specifications			
			Timesteps	*Randomness*	*Category*	*Difficulty*
Airline Revenue Management (Moller et al, 2006)	**21**	1	14	Demand	SWI	Medium
Multistage Stochastic Unit Commitment Problem (Zhou et al, 2019)	**22**	1	24	Demand	SWI	Medium
CCHP Problem (Seranilla et al, 2019)	**23**	1	52	Demand	SWI	Medium
Multistage Facility Location Problem (Kim et al, 2020)	**24**	1	5	Demand Cost Price	SWI	Medium

Figure 3.4: Real-world MSIP problems with various instance flavors and corresponding specifications

3.1.3 Multistage Stochastic Integer Problems

These problems are sourced from various articles in the literature where the problem is modelled as a multistage stochastic integer program and where the SDDiP algorithm was employed to solve the problem. The class of MSIPs is also essential to tackle as many real-world problems are modelled as such. Figure (3.4) show the problem instance charts for MSIPs included in the MSPLib. Presently, although included in the MSPLib, MSIPs are not included in the benchmarking for SDDP implementations.

Multi-stage Stochastic Unit Commitment Problem

The Unit Commitment Problem is one of the pivotal operational problems in power systems. In Zou et al. (2018), a multi-stage, stochastic (MSUC) variation of this problem is presented. MSUC aims to optimize the decision of the commitment schedule of some generation units for a particular planning horizon. Given some deterministic costs and a stochastic demand, the electricity load has to be satisfied as well as some various physical constraints of the generators and the systems, such as generation capacity, minimum up and down time, ramping limits, and the flow limit of transmission lines.

Multistage Facility Location Problem

The facility location problems often involves uncertainties in input parameters such as demands and costs, among others. Kim (2020) presents a problem related to models for facility location problems in the literature that include these characteristics. These uncertainties are often modeled using either stochastic programming or robust optimization, depending on whether probability distributions are known. (stochastic) demand in each period is determined by the outcome of a series of binary events, where the likelihood of outcomes is unknown.

Airline Revenue Management Problem

This problem is taken from Andris Moller, Werner Romisch, and Klaus Weber paper titled "Airline Network Revenue Management by Multistage Stochastic Programming" Möller et al. (2008). An airline company manages flights between three cities. In the middle of the three cities, there is a hub serves as a transition point. The company wants to determine the seat protection level for all itineraries, fare classes to maximize the revenue over a planning horizon. Every stage corresponds to a departure date. Cancellation rate is deterministic. Demand is random.

Combined Cooling, Heating, and Power (CCHP) System Scheduling Problem

Combined Cooling, Heat, and Power (CCHP) system - a system producing electrical, thermal, and cooling energies simultaneously - is seen as a reliable solution for its resource energy efficiency increase and carbon and air pollutant emissions reduction potentials. In order to further recognize these potentials for being more valuable, CCHP must prove to be cost-effective in contrast to traditional systems. Seranilla (2019) models an MSIP to solve for an optimal and cost-efficient operational strategy of a CCHP system. The MSIP aims is to provide the optimal load factor - the utilization rate of the CCHP - which minimizes the total operational cost of producing the electrical, thermal, and cooling demands of the client. The MSIP was designed to linearize the electrical and thermal efficiency functions of a CCHP, which is a quadratic equation, by discretizing its corresponding load factor. A numerical study was conducted in a data center in Italy to evaluate the proposed CCHP system model.

Chapter 4

Optimizing Vaccine Distribution in Developing Countries under Natural Disaster Risk

For many developing countries, COVID-19 vaccination roll-out programs are not only slow but vaccination centers are also exposed to the risk of natural disaster, like flooding, which may slow down vaccination progress even further. Policy-makers in developing countries therefore seek to implement strategies that hedge against distribution risk in order for vaccination campaigns to run smoothly and without delays. We propose a stochastic-dynamic facility location model that allows policy-makers to choose vaccination facilities while accounting for possible facility failure. The model is a multi-stage stochastic variant of the classic facility location problem where disruption risk is modelled as a binary multivariate random process - a problem class that has not yet been studied in the literature.

To solve the problem, we propose a novel approximate dynamic programming algorithm which trains the shadow price of opening a flood-prone facility on historical data, thereby alleviating the need to fit a stochastic model. We trained the model using rainfall data provided by the local government of several major cities in the Philippines which are exposed to multiple flooding events per year. Numerical results demonstrate that the solution approach yields approximately 30-40% lower cost than a baseline approach that does not consider the risk of flooding. Recommendations based on this model were implemented following a collaboration with two large cities in the Philippines which are exposed to multiple flooding events per year.

4.1 Introduction

The coronavirus pandemic of 2019 (COVID-19) demonstrated the vulnerability of public health safety, with approximately 578 million cases and 6.4 million deaths worldwide as of August 2022 (WHO 2022). The development of vaccines against the virus substantially altered the course of the pandemic. As of August 2022, 12.3 billion doses of vaccines have been administered all over the world (WHO 2022). The distribution of these vaccines posed major supply chain challenges for pharmaceutical supply chains and local governments. Since most COVID-19 vaccines are produced and distributed in developed countries, only limited and intermittent vaccine supply reached non-producing, developing countries where vaccination roll-outs are already expected to take longer than in developed countries.

The roll-out of COVID-19 vaccines was further delayed for reasons more than only a restricted supply. Other factors like natural disasters continuously daunt these countries, among them hydro-meteorological events (e.g., flooding, landslides, and storm surges), seismic events (e.g., earthquakes), and volcanic eruptions. Particularly, the absence of adequate infrastructure, catastrophe emergency preparedness, and response systems makes developing countries more vulnerable to natural disasters (Ritchie & Roser 2014) and puts vaccination campaigns in peril. For instance, in early December 2021, Typhoon Rai hit central Philippines with catastrophic thunderstorms missing year-end COVID-19 vaccination target due to delayed schedules and expired vaccines (Philstar 2021). Similarly, in January 2022, deadly floods and landslides caused by torrential rains hit the city of Sao Paulo, Brazil which cancelled the scheduled COVID-19 vaccination campaigns (The Economist 2022). These factors drive policy-makers in developing nations to seek decision support for planning for vaccination campaigns as well as to preparing those campaigns against the risk of natural disaster risk to prevent delays and ultimately save lives.

In this article, we report on models and methods developed as part of *Project FALCON*, which is a collaboration of the authors with the local government of Cagayan de Oro in the Philippines, for planning the roll-out of the COVID-19 vaccination campaign. Cagayan de Oro is a large city on the island of Mindanao that experiences multiple torrential rains and flooding events every year (CDO 2022). Flooding poses a serious threat for successfully orchestrating large-scale vaccination campaigns. The optimal location of vaccination centers and allocation of recipients must therefore account for facility failure due to flooding as well as the resulting cost of relocation and redistribution.

The optimization problem underlying FALCON is a stochastic-dynamic facility location model whose optimal decision policy chooses opening of vaccine facilities while accounting for possible facility failure. The joint risk of facility failure is captured by a multivariate binary random process. As the model entails solution of a high-dimensional stochastic-dynamic program, FALCON uses approximate dynamic programming to optimize the decision policy. Numerical investigations show a reduction of vaccination roll-out operational cost of approximately 30-40% when compared to a greedy approach that ignores the risk of flooding.

4.1.1 Literature Review

On modelling approaches

Choosing vaccination centers is essentially an instance of the well-known facility location problem (FLP). The classic FLP aims at selecting facility locations to meet customer demand, at minimal cost. See Owen & Daskin (1998), Melo et al. (2009), and Snyder (2006) for literature reviews. FLP has also been invaluable in solving optimization problems in healthcare and humanitarian logistics, including vaccine distribution. For example, Bertsimas et al. (2022) discussed the location of COVID-19 mass vaccination facilities in the United States. We refer the reader to Duijzer et al. (2018) for an extensive literature discussion on vaccine supply chains and to Ahmadi-Javid et al. (2017) for surveys on healthcare facility location.

Another important stream of literature is on emergency facility location (EFL) problems, i.e., optimization of facility locations in the face of natural disasters (Li et al. 2011). Furthermore, EFL relates closely to another stream of literature on reliable facility location (RFL) problems. RFL deals with optimal decisions considering failure of a facility due to disruption (e.g., natural disasters, blackouts, fire, etc). A comprehensive review is presented in Snyder et al. (2016). RFL models require knowledge of the occurrence of disruptions which is a difficult undertaking in practice. Thus, the majority of works on RFL models largely assume that the probability distribution of disruptions is known. Nonetheless, the emergence of robust optimization (RO) dealt with the limited information on the uncertainty of disruptions. For example, Lu et al. (2015) introduce a model that allows disruptions to be correlated with an uncertain joint distribution and minimize the expected cost under the worst-case distribution with given marginal disruption probabilities. Lim et al. (2013) also tackled facility location decisions with random disruptions and investigated the impact of misestimating the disruption probability using a stylized continuous location model. Additionally, recent work of Cheng et al. (2021) adopts a two-stage robust optimization method, where facility location decisions are made *here-and-now* (first stage) and *wait-and-see* (second stage) are entail reassigning customers after revealing information on facility availability.

However, most papers in the literature only consider static and two-stage cases. Furthermore, in contrast to the abundant studies on deterministic multi-period setting, there are only a limited number of works that formulate facility location problems as multi-stage stochastic programs, and all of them only consider stochastic demand but not facility disruption. Some multi-stage formulations with stochastic demand are presented in Nickel et al. (2012), Hernandez et al. (2012), and Albareda-Sambola et al. (2013). To the best of our knowledge, there exist no prior work specifically on multistage stochastic facility location with facility disruption that accounts for the risk of facility failure.

On solution methodologies

We model the multistage stochastic facility location problem (MSFLP) under risk of disruption as a multistage stochastic integer program (MSIP). MSIP is a general framework for sequential decision making under uncertainty, when the decision space is high-dimensional, subject to various constraints, and randomness in parameters is modeled by a general stochastic process. The class of MSIP problems are commonly found in various fields such as energy, finance, operations, and logistics (Römisch & Schultz 2001). Nonetheless, due to its inherent complexity, advances in solution methodologies for MSIP problems are limited. Notable methods exploiting extensive MSIP formulation include scenario decomposition (Chen et al. 2002), Lagrangian relaxation (Nowak & Römisch 2000), and progressive hedging (Gade et al. 2016). However, these techniques do not scale well with large problem instances.

Another strand of research considers reformulating MSIPs as stochastic-dynamic programming problems. To be able to approximate the cost-to-go function via cutting planes, a common approach taken by many authors lies in solving a relaxed version of the problem using stochastic dual dynamic programming (SDDP) which is the current *state-of-the-art* to solve linear MSPs. In this approach, integrality is respected during forward passes but relaxed during backward passes of the algorithm, e.g., Löhndorf et al. (2013) and Flach et al. (2010). Cerisola et al. (2012) utilized McCormick envelopes to approximate the bilinear relationship between variables, and Abgottspon et al. (2014) proposed locally valid cuts. An SDDP extension to solve general MSIP problems with binary state variables is the stochastic dual dynamic integer programming (SDDiP) introduced in Zou et al. (2019). Although it can solve MSIP problem exactly, SDDiP limits the stochastic process to be stage-wise independent or, as extended in Löhndorf & Shapiro (2019), to follow a discrete Markov chain.

The algorithm proposed in this work, named shadow price approximation (SPA), proceeds with a more straightforward way by approximating the value function by a linear function. Unlike in dynamic programming, SPA views the parameters of the value function as tunable parameters of the resulting policy. Tuning parameters to improve decision-making in the face of uncertainty is widely used in practice to make deterministic optimization problems more robust against a variety of risks, for example, safety stock and buffers in production planning (Inderfurth 1995), upper and lower limits for water reservoirs to capture rainfall uncertainty (Nápoles-Rivera et al. 2015), and slack time to buffer uncertainty in job completion (Burdett & Kozan 2015).

Although parameterization date back to early 20th century in the statistics community, Ghadimi et al. (2020) formally introduced and proposed *parametric* cost function approximation (CFA) - a method that introduces parameterization of problem parameters either to the objective function or the constraints of the embedded deterministic model of a lookahead policy. In parametric CFA, random parameters are replaced by their expected values. Naturally, these problems suffer from overly optimistic decisions and introduction of tighter constraints provides a hedge against uncertainty. For example, Perkins III & Powell (2017) tackled an energy storage problem to show an improvement over the basic deterministic lookahead model, directly demonstrating the success of applying parametric CFA approach.

While SPA relies on tunable parameters just like CFA, both algorithms work with different model formulations. Parametric CFA chooses parameters of a lookahead model whereas SPA chooses parameters of a linear value function approximation thereby working with a decomposed form of the decision problem. Additionally, SPA allows for bootstrapping random parameters from real, historical data, hence alleviating the need to fit a statistical model before solving the optimization problem. Instead, SPA relies on cross validation to test policy performance, which is a widely known technique in machine learning. This data-driven approach is not suitable for parametric CFA, as CFA requires some form of prediction model at each forward-step to set the parameters of the look-ahead model.

4.1.2 Contributions

The contribution of this work is two-fold: first, the proposed stochastic facility location model as well as the developed solution method are novel to the literature on facility location planning as we further explain below. Second, the work contributes to the practice of vaccine distribution, as Project FALCON provided the scientific background to nudge the central government to accelerate vaccination roll-out. The implementation of Project FALCON led to Cagayan de Oro City hitting the inoculation target and having one of the highest vaccination rates in the Philippines by the end of Year 2021 (DOH-10 2022).

Our work advances the literature in both methodological and managerial directions.

First, the multi-stage variant of the proposed facility location problem has not yet been explored in the literature, specifically the integration of disruption risks due to natural disasters. We close this gap by formulating the problem as a multi-stage stochastic mixed-integer program that captures the risk of natural disaster disruption by a binary random variable that affects the state of each facility across time. We then develop an approximate dynamic programming formulation where the value function of the underlying stochastic-dynamic program is approximated by a function that is linear in the state variables. We show that the optimal upper bound of this approximation only depends on the terms associated with facility state variables. To the best of our knowledge, this is the first time that this problem class is studied.

Furthermore, we propose an algorithm that trains the parameters of the linear value function approximation by minimizing an upper bound on the optimal objective value. Since said parameters behave like shadow prices of the non-anticipatory constraints of the multistage stochastic program, we refer to this algorithm as *shadow price approximation* (SPA). Intuitively, SPA is is simpler than parametric CFA as it does not require formulation of a lookahead model but merely adds a small set of parameters to the objective function of each subproblem. Using observations of the random variables as direct inputs of a deterministic model and the current parameters of the linear value function approximation, going forward in time, each deterministic subproblem is solved. This can be viewed as a *single-stage optimization* in a rolling-horizon approach. With the realized upper bound and the current shadow price parameters, SAP proceeds with a generic update function to produce a new set of parameters of the linear value function approximation until a stopping criterion is met. The optimization model and solution algorithm make up the computational core of the decision support tool that has been developed for Project FALCON.

Although being motivated by COVID-19 vaccination efforts, our model is adequately general to adapt to other settings. Additionally, given the data-driven nature of the solution method, any other form of disruption can be considered, such as blackouts, hurricanes, draughts, etc. The proposed algorithm also has two major advantages over alternative solution approaches for this problem class: (1) it can be easily integrated with gradient-based and stochastic search methods that are widely used in machine learning and global optimization; (2) the algorithm does not require formulation of a stochastic model but merely needs access to independent time series that can also be real data.

The remainder of the paper is organized as follows: Section 2 introduces the multi-stage facility location problem under uncertainty of facility failure. Section 3 presents an introduction to the SPA algorithm which optimally solves the multistage stochastic facility location problem. Section 4 demonstrates the application of the model subsuming the details of integrating rainfall as the random parameter and solution method to large cities in the Philippines. Section 5 summarizes the results and provides an outlook of the research directions.

4.2 Model Formulation

We formulate the facility location model with risk of disruption due to natural disaster as a multi-stage, discrete-time, stochastic-dynamic optimization problem. We decompose the planning horizon T into a sequence of time stages $t \in \{1, ..., T\}$. We define state variable $u_{it} \in \{0,1\}$ such that $u_{it} = 1$ if facility $i \in \{1, \ldots, F\}$ is open at stage t, and $u_{it} = 0$ otherwise. The transition of the state variable from its value at the beginning of the stage to its value the end of the stage is governed by the series of constraints and relationships we detail below. In addition, we observe a realization of the random variable $\xi_{it} \in \{0,1\}$ such that $\xi_{it} = 1$ if the vaccine facility i at stage t fails due to natural disaster and $\xi_{it} = 0$ otherwise.

After the observation of the state variable u_{it} and the random variable ξ_{it} at the beginning of each stage, the values of the decision (or control) variables need to be determined. The control variables are $x_{it} \in \{0,1\}$ (binary variable with value 1 if vaccine facility i should be opened at stage t and 0, otherwise) and z_{ijt} (allocation of vaccinating population of district $j \in \{1, \ldots, B\}$ to vaccine facility i at stage t if vaccine facility i is open). We formulate the corresponding multistage stochastic facility location problem (MSFLP) model under risk of facility disruption due to natural disaster in

Problem (5.5) - (5.10).

$$\min \quad \sum_{t=1}^{T}\sum_{i=1}^{F}(f_{it}x_{it}+\sum_{j=1}^{B}d_{ij}z_{ijt}) \tag{4.1}$$

$$\text{s.t.} \quad u_{it} = u_{it-1}(1-\xi_{it}) + x_{it}, \quad \forall i = 1,\dots,F, t = 1,\dots,T \tag{4.2}$$

$$\sum_{j=1}^{B} z_{ijt} \leq C_i u_{it}, \quad \forall i = 1,\dots,F, t = 1,\dots,T \tag{4.3}$$

$$\sum_{i=1}^{F} z_{ijt} = P_{jt}, \quad \forall j = 1,\dots,B, t = 1,\dots,T \tag{4.4}$$

$$u_{it}, x_{it} \in \{0,1\}, \quad \forall i = 1,\dots,F, t = 1,\dots,T \tag{4.5}$$

$$z_{ijt} \in \mathbb{Z}, \quad \forall i = 1,\dots,F, j = 1,\dots,B, t = 1,\dots,T \tag{4.6}$$

The objective function (4.1) is a straightforward minimization of the total cost function which includes the fixed cost f_{it} of opening vaccine facility i at stage t and the travel cost d_{ij} from district j to vaccine facility i. We need to keep track of the state of vaccine facility i as affected by random variable ξ_{it} across each stage through constraints (4.2). If at stage $t-1$ the vaccine facility i is closed [open] and we decide to open [close] it at stage t, then $x_{it} = 1$ [$x_{it} = 0$], the state at stage t will now become $u_{it} = 1$ [$u_{it} = 0$].

Constraints (4.3) state that the total capacity C_i will be assigned to vaccine facility i only if it is available and should at least be equal to the total number of vaccinating population z_{ijt} in vaccine facility i from district j at stage t. Constraints (4.4) state that the population P_{jt} of district j must be distributed to vaccine facilities i. Constraints (4.5) and (4.6) show the decision variable domains.

Let us formulate a more general version of MSFLP which Problem (5.5) - (5.10) is an instance of. As in Yu et al. (2021), we use bold typeface symbols to denote variable vectors and matrices. Thus, we rewrite MSFLP in the following vector-matrix form,

$$\min_{\mathbf{u}_t,\mathbf{y}_t} \quad \sum_{t\in T}(\boldsymbol{v}_t^\top \mathbf{u}_t + \boldsymbol{w}_t^\top \mathbf{y}_t) \tag{4.7}$$

$$\text{s.t.} \quad \mathbf{u}_t = \mathbf{u}_{t-1}(1-\boldsymbol{\xi}_t) + \mathbf{y}_t, \quad \forall t \in 1,\dots,T \tag{4.8}$$

$$\boldsymbol{A}\mathbf{y}_t \leq \mathbf{u}_t, \quad \forall t \in 1,\dots,T \tag{4.9}$$

$$\boldsymbol{B}\mathbf{y}_t = \boldsymbol{b}, \quad \forall t \in 1,\dots,T \tag{4.10}$$

$$\mathbf{u}_t \in \{0,1\}^F,\ \mathbf{y}_t \in \mathbb{Z}_+^{F\times B}, \quad \forall t \in 1,\dots,T \tag{4.11}$$

where $\mathbf{u}_t$, $\mathbf{y}_t = (\mathbf{x}_t, \mathbf{z}_t)$ are the state and control variables respectively, $\boldsymbol{v}_t^\top$,$\boldsymbol{w}_t^\top$ represent the cost function, and matrices $\boldsymbol{A} \in \mathbb{R}^{F\times B}$, $\boldsymbol{B} \in \mathbb{R}^{F\times B}$, and $\boldsymbol{b} \in \mathbb{R}^B$ correspond to the coefficients of constraints (5.7) and (5.8). This allows a more general setting to accommodate any extension and variant of the MSFLP, e.g., where the objective function depends on $\boldsymbol{u_t}$ in contrast to Problem (5.5) - (5.10).

Recall that random variable $\boldsymbol{\xi}_t \in \{0,1\}$ takes the value 1 if the facility fails at stage t or 0 if it does not. To facilitate the formulation of the dynamic program, we introduce a scenario-path based notation. We take one possible path of realizations from the beginning to the end of the planning horizon, and denote it as a scenario path $\boldsymbol{\omega}$, i.e.,

$$\boldsymbol{\xi}^{\boldsymbol{\omega}} = (\boldsymbol{\xi}_1^{\boldsymbol{\omega}}, \dots, \boldsymbol{\xi}_T^{\boldsymbol{\omega}}). \tag{4.12}$$

Accordingly, we use $\mathbf{u}_t^{\boldsymbol{\omega}}$ and $\mathbf{y}_t^{\boldsymbol{\omega}}$ to denote the state and decision vectors at stage t under scenario path $\boldsymbol{\omega}$. Since the number of realizations is finite, let $\boldsymbol{\Omega}$ be the support set of $\boldsymbol{\omega}$ with each realization $\boldsymbol{\omega} \in \boldsymbol{\Omega}$ having probability $p(\boldsymbol{\omega})$ such that

$$\sum_{\boldsymbol{\omega}\in\boldsymbol{\Omega}} p(\boldsymbol{\omega}) = 1. \tag{4.13}$$

See for example Huang & Ahmed (2009) who make similar assumptions.

4.2.1 Multistage Stochastic Integer Program (MSIP) Formulation

We formulate the multistage stochastic facility location problem (MSFLP) as a multistage stochastic integer program (MSIP), with the form

$$\min_{\mathbf{u_1},\mathbf{y_1}\in\mathcal{X}_1}\Big\{\boldsymbol{v}_1^\top\mathbf{u_1}+\boldsymbol{w}_1^\top\mathbf{y_1}+\mathbb{E}_{\boldsymbol{\xi}_{[2,T]}|\boldsymbol{\xi}_{[1,1]}}\Big[\min_{\mathbf{u_2},\mathbf{y_2}\in\mathcal{X}_2(\mathbf{u_1},\boldsymbol{\xi}_2)}\Big\{\boldsymbol{v}_2^\top\mathbf{u_2}+\boldsymbol{w}_2^\top\mathbf{y_2}+\dots \\ +\mathbb{E}_{\boldsymbol{\xi}_{[T,T]}|\boldsymbol{\xi}_{[1,T-1]}}\Big[\min_{\mathbf{u_T},\mathbf{y_T}\in\mathcal{X}_T(\mathbf{u_{1:T-1}},\boldsymbol{\xi}_T)}\Big\{\boldsymbol{v}_T^\top\mathbf{u_T}+\boldsymbol{w}_T^\top\mathbf{y_T}\Big\}\Big]\dots\Big\}\Big]\Big\}, \tag{4.14}$$

where, for notation simplicity, $\mathbf{u_t}$ and $\mathbf{y_t}$ are the state variables and control (i.e., local or stage) variables respectively, $\boldsymbol{v}_t^\top$ and $\boldsymbol{w}_t^\top$ are the corresponding cost at time $t\in\{1,\dots,T\}$, and $\mathcal{X}_t$ is the feasible set.

In this setting, the stochastic data process is revealed as time goes on. This stochastic data process $(\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_T)$ is modeled where $\boldsymbol{\xi}_1$ is deterministic and $\boldsymbol{\xi}_2,\dots,\boldsymbol{\xi}_T$ will be revealed gradually in time. Thus, the history of the stochastic data process until stage t is given by

$$\boldsymbol{\xi}_{[t]}=(\boldsymbol{\xi}_1\dots\boldsymbol{\xi}_t). \tag{4.15}$$

This formulation allows the decision at each stage to adapt to the realized uncertainty up to that stage. Thus, by solving MSFLP, we get a policy on which facilities should be opened for each possible scenario.

4.2.2 Dynamic Programming Reformulation

We can further formulate (4.14) as a dynamic programming (DP) recursion. The optimal value function at stage t, $V_t(\mathbf{u}_{t-1},\boldsymbol{\xi}_t)$, is the optimal expected objective value given state $(\mathbf{u}_{t-1},\boldsymbol{\xi}_t)$, and assuming that optimal action will be taken at each stage t.

$$V_t(\mathbf{u_{t-1}},\boldsymbol{\xi}_t)=\min_{\mathbf{u}_t,\mathbf{y}_t}\{\boldsymbol{v}_t^\top\mathbf{u}_t+\boldsymbol{w}_t^\top\mathbf{y}_t+\mathcal{V}_{t+1}(\mathbf{u}_t,\boldsymbol{\xi}_t):\boldsymbol{B}_t\mathbf{u}_{t-1}+\boldsymbol{A}_t\mathbf{u}_t+\boldsymbol{C}_t\mathbf{y}_t=\boldsymbol{b}_t\}, \tag{4.16}$$

for $t=1,\dots,T$ where $\mathcal{V}_{t+1}(\mathbf{u}_t,\boldsymbol{\xi}_t)$ is the expected value cost-to-go function,

$$\mathcal{V}_{t+1}(\mathbf{u}_t,\boldsymbol{\xi}_t):=\mathbb{E}[V_{t+1}(\mathbf{u}_t,\boldsymbol{\xi}_{t+1})|\boldsymbol{\xi}_t], \tag{4.17}$$

with $\mathcal{V}_T \equiv 0$.

A common assumption is for $\boldsymbol{\xi}_t$ to be Markovian, i.e., the distribution of $\boldsymbol{\xi}_{t+1}$ only depends on $\boldsymbol{\xi}_t$ rather than the whole history of the data process. Most time series process, such as autoregressive models, can be cast as Markov processes by a sufficient state space expansion.

Finally, let us define the optimal policy as

$$\pi^*(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t) = \text{argmin}_{\mathbf{u}_t, \mathbf{y}_t} \{\boldsymbol{v}_t^\top \mathbf{u}_t + \boldsymbol{w}_t^\top \mathbf{y}_t + \mathcal{V}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t) : \boldsymbol{B}_t \mathbf{u}_{t-1} + \boldsymbol{A}_t \mathbf{u}_t + \boldsymbol{C}_t \mathbf{y}_t = \boldsymbol{b}_t\} \tag{4.18}$$

for $t = 1, \ldots, T$ in set $\boldsymbol{\Pi}$ as the *policy* which specifies the decision to make for all possible states regardless of which state at stage t.

4.2.3 MSIP with Binary State Variables

Each subproblem of MSFLP given realized $\boldsymbol{\xi_t}$ at stage t is a deterministic mixed-integer program. With this, we make the following assumption:

Assumption 4.1. The objective function $\mathbf{z} = \sum_{t \in T} (\boldsymbol{v}_t^\top \mathbf{u}_t + \boldsymbol{w}_t^\top \mathbf{y}_t)$ in each realization $\boldsymbol{\xi}_t$ is an affine function in u_t and y_t, and the constraint set $\mathcal{X}_t$ is a nonempty compact mixed integer polyhedral set.

Mathematically, mixed-integer programs (MIPs) are non-convex. Therefore, given integer local variables, the value functions $V_t(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t)$ and expected cost-to-go functions $\boldsymbol{\mathcal{V}}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t)$ at stage t are both non-convex with respect to the state variables. Although it is impossible to formulate convex polyhedral representations of the non-convex value functions that are tight at the evaluated state variable values, real-valued function of binary variables can be represented exactly by a convex polyhedral function. Zou et al. (2019) also made similar assumption.

Remark 4.1. *The value functions, $V_t(\mathbf{u_{t-1}}, \boldsymbol{\xi}_t)$, are convex in state variable $\mathbf{u}_t \in \{0, 1\}$ if the optimal values of*

$$\min_{\mathbf{u}_t, \mathbf{y}_t} \{\boldsymbol{v}_t^\top \mathbf{u}_t + \boldsymbol{w}_t^\top \mathbf{y}_t + \mathcal{V}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi_t}) : \boldsymbol{B}_t \mathbf{u}_{t-1} + \boldsymbol{A}_t \mathbf{u}_t + \boldsymbol{C}_t \mathbf{y}_t = \boldsymbol{b}_t\} \tag{4.19}$$

are convex in $\mathbf{u}_t$*, which is inherent with the fact that if* $\mathbf{u}_t \in \{0,1\}$ *and* $\mathbf{u}_t$ *is the convex combination of a set of binary vectors, then* $\mathbf{u}_t$ *is a binary vector, for* $t = 1,\ldots,T$*.*

4.2.4 Value Function Approximation

Recall that the value functions $V_t(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t)$ represent the minimal cost assuming optimal action will be taken at stage t. For tractable and small problems with a finite state space, backwards dynamic programming can be used to compute the optimal value function. However, as the number of states grows exponentially with the dimensionality of the state space, many real-world problems become computationally intractable. For this reason, a large body of research is dedicated to methods of approximate dynamic programming (ADP) which aims at using functions of lower complexity to approximate the true value function.

There exist numerous strategies for value function approximation, such as aggregation with lookup tables, basis functions, neural networks, etc. See Powell (2007) for an extensive treatment of the subject.

More sophisticated techniques, often aim at exploiting structure of the underlying optimization problem. For example, *stochastic dual dynamic integer programming* (SDDiP) - the *state-of-the-art* algorithm to solve MSIP problems advanced in Zou et al. (2019) - approximates the convex piece-wise linear expected cost-to-go function by a set of (cutting) hyperplanes obtained from subgradients of the subproblems in stages $t = 1,\ldots,T$ of a set of sampled outcomes. However, the algorithm requires the random process to be either stagewise-independent or to follow a discrete Markov chain (Löhndorf & Shapiro 2019).

In this article, we pursue a much simpler strategy by approximating the value function by a linear function. In what follows, we will demonstrate how to find parameters of a linear value function approximation that not only finds a good solution to the MSFLP but moreover provides practitioners with actionable insights of the risks and benefits of opening a facility under risk of disruption.

4.2.5 Linear Value Function Approximation

We propose to approximate the convex expected value function, $\mathcal{V}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t)$ by a linear function,

$$\mathcal{V}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t) \approx \bar{\mathcal{V}}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t; \boldsymbol{\lambda}_t^u, \boldsymbol{\lambda}_t^\xi, \boldsymbol{\beta}_t) = \boldsymbol{\lambda}_t^{u\top}\mathbf{u}_t + \boldsymbol{\lambda}_t^{\xi\top}\boldsymbol{\xi}_t + \boldsymbol{\beta}_t,\ t = 1, \dots, T-1, \tag{4.20}$$

with $\boldsymbol{\lambda}_t^u$, $\boldsymbol{\lambda}_t^\xi$ as slope vectors and $\boldsymbol{\beta}_t$ as intercept. The vector $\boldsymbol{\lambda}_t^u$ defines the marginal future cost of opening a facility in period t. We will later refer to this as the *shadow price* of opening a facility, as it accounts for the possible future price that needs to be paid later for re-opening facilities in case of failure.

In most cases, calculating the value function is just means to an end. What we are truly interested in is finding an approximation of the optimal policy π^*. The approximate policy that corresponds to the linear value function approximation (4.20) is given by

$$\begin{aligned}\bar{\pi}(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t^\omega; \boldsymbol{\lambda}_t^u, \boldsymbol{\lambda}_t^\xi, \boldsymbol{\beta}_t) = \operatorname{argmin}_{u,y\geq 0}\{ \quad & \boldsymbol{v}_t^\top\mathbf{u}_t + \boldsymbol{w}_t^\top\mathbf{y}_t + \bar{\mathcal{V}}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t^\omega) : \\ & \boldsymbol{B}_t\mathbf{u}_{t-1} + \boldsymbol{A}_t\mathbf{u}_t + \boldsymbol{C}_t\mathbf{y}_t = \boldsymbol{b}_t\},\end{aligned} \tag{4.21}$$

for $t = 1, \dots, T$, where $\omega \in \Omega$ is a scenario path of a stochastic process with its support set Ω.

With this policy, we can now define an upper bound on the optimal objective value of the MSFLP,

$$\bar{\mathrm{z}}(\bar{\pi}) = \mathbb{E}_{\boldsymbol{\omega}\in\boldsymbol{\Omega}_{[1,T]}}\left[\bar{z}(\bar{\pi})\right] \tag{4.22}$$

where (suppressing the dependence of $\bar{\pi}$ on $\boldsymbol{\lambda}_t^u$, $\boldsymbol{\lambda}_t^\xi$, and $\boldsymbol{\beta}_t$)

$$\bar{z}(\bar{\pi}) = \sum_t^T \Big(\boldsymbol{v}_t^\top\mathbf{u}_t(\bar{\pi}(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t^\omega)) + \boldsymbol{w}_t^\top\mathbf{y}_t(\bar{\pi}(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t^\omega))\Big). \tag{4.23}$$

If all we are interested in is finding parameters $(\boldsymbol{\lambda}_t^u, \boldsymbol{\lambda}_t^\xi, \boldsymbol{\beta}_t)$ that provide the lowest upper bound, we can further simplify the policy by removing the terms $\boldsymbol{\lambda}_t^{\xi\top}\xi_t^\omega$ and $\boldsymbol{\beta}_t$ from the right-hand side of (4.20). This is supported by the following proposition.

Proposition 4.1. For a linear value function approximation, $\bar{\mathcal{V}}_t(\cdot;\boldsymbol{\lambda}_t^u,\boldsymbol{\lambda}_t^\xi,\boldsymbol{\beta}_t)$ that is affine in state variables $\boldsymbol{u}_{t-1}$ and random variables $\boldsymbol{\xi}_t$, it holds that the optimal upper bound is independent of parameters $\boldsymbol{\lambda}_t^\xi$ and $\boldsymbol{\beta}_t$.

Proof. By Equations (4.22) and (4.23), we obtain an upper bound $\bar{\mathrm{z}}$ by choosing parameters of approximate policy $\bar{\pi}$, i.e., slope vectors $\boldsymbol{\lambda}_t^u$, $\boldsymbol{\lambda}_t^\xi$, and intercept $\boldsymbol{\beta}_t$. Approximate policy $\bar{\pi}$ is defined by state variables $\mathbf{u_{t-1}}$ and local variables $\mathbf{y_t}$ as shown in Equation (4.21). Now, let us denote

$$\begin{aligned}\bar{\pi}'(\mathbf{u}_{t-1},\boldsymbol{\xi}_t^\omega;\boldsymbol{\lambda}_t^u,\boldsymbol{\lambda}_t^\xi,\boldsymbol{\beta}_t) = \quad & \operatorname{argmin}_{\mathbf{u},\mathbf{y}}\{\boldsymbol{v}_t^\top\mathbf{u}_t+\boldsymbol{w}_t^\top\mathbf{y}_t+\underbrace{\boldsymbol{\lambda}_t^{u\top}\mathbf{u}_t+\boldsymbol{\lambda}_t^{\xi\top}\boldsymbol{\xi}_t^\omega+\boldsymbol{\beta}_t}_{\text{objective function offset}}: \\ & \boldsymbol{B}_t\mathbf{u}_{t-1}+\boldsymbol{A}_t\mathbf{u}_t+\boldsymbol{C}_t\mathbf{y}_t=\boldsymbol{b}_t\}, \qquad \forall t=1,\dots,T.\end{aligned} \tag{4.24}$$

Random variable $\boldsymbol{\xi}_t^\omega$ enters the approximate expected cost-to-go function $\bar{\mathcal{V}}_{t+1}(\cdot)$ as an offset to the objective function. Hence, it does not effect the optimal choice of $\mathbf{u_{t-1}}$ and $\mathbf{y_t}$ in (4.24). Therefore, (4.24) can be simplified to

$$\bar{\pi}'(\mathbf{u}_{t-1},\boldsymbol{\xi}_t^\omega;\boldsymbol{\lambda}_t^u)=\operatorname{argmin}_{\mathbf{u},\mathbf{y}}\{\boldsymbol{v}_t^\top\mathbf{u}_t+\boldsymbol{w}_t^\top\mathbf{y}_t+\boldsymbol{\lambda}_t^{u\top}\mathbf{u}_t : B_t\mathbf{u}_{t-1}+A_t\mathbf{u}_t+C_t\mathbf{y}_t=b_t\}, \tag{4.25}$$

for $t=1,\dots,T$, where the right-hand side is independent of $\boldsymbol{\lambda}_t^{\xi\top}$ and $\boldsymbol{\beta}_t$. Thus, for any realization of random variable $\boldsymbol{\xi}_t^\omega$, policy $\bar{\pi}'$ yields the same values. Accordingly, it holds that

$$\bar{\pi}'(\mathbf{u}_{t-1},\boldsymbol{\xi}_t^\omega;\boldsymbol{\lambda}_t^u)=\bar{\pi}(\mathbf{u}_{t-1},\boldsymbol{\xi}_t^\omega;\boldsymbol{\lambda}_t^u,\boldsymbol{\lambda}_t^\xi,\boldsymbol{\beta}_t),\ \forall t=1,\dots,T. \tag{4.26}$$

Hence, by equivalence of approximate policies $\bar{\pi}$ and $\bar{\pi}'$, the upper bound $\bar{\mathbf{z}}(\bar{\pi})$ remains unchanged, and it follows that upper bound $\bar{\mathrm{z}}(\bar{\pi})\equiv\bar{\mathrm{z}}(\boldsymbol{\lambda}_t^u)$, for $t=1,\dots,T$, is independent of parameters $\boldsymbol{\lambda}_t^\xi$ and $\boldsymbol{\beta}_t$, which concludes the proof. □

Proposition 1 tells us that parameters $\boldsymbol{\lambda}_t^\xi$ and $\boldsymbol{\beta}_t$ do not influence the optimal upper bound, which implies that we can set them to zero, effectively removing them from the objective function. This enables us to simplify the functional form of the approximate expected value function to

$$\bar{\mathcal{V}}_{t+1}(\boldsymbol{u}_t)\equiv\boldsymbol{\lambda}_t^{u\top}\mathbf{u}_t, \tag{4.27}$$

which is independent of the state of the stochastic process $\boldsymbol{\xi}_t$, for $t=1,\dots,T$.

This claim is also supported in Powell (2007) where the notion of 'post-decision state variables' is discussed as a basic strategy to overcome the curse-of-dimensionality. Post-decision state variables are variables that capture the state of the system immediately after a decision has been made but before new information (random event) has arrived. Our linear value function approximation uses the post-decision state which represent the state of the vaccination centers $\boldsymbol{u}_t$ at the end of each period t, whereas the pre-decision state would entail all decision variables of the sub-problem.

Furthermore, the removal of the dependency of the post-decision value on $\boldsymbol{\xi}_t$ reduces the parametric search space, thereby improving algorithmic tractability. Since Proposition 1 proves that the optimal upper bound is dependent only on $\boldsymbol{u_t}$, it implies that we can set the parameters $\boldsymbol{\lambda}_t^{\xi\top}$ and $\boldsymbol{\beta}_t$ to zero, effectively removing them from the objective function thus reducing the search space to the domain of $\boldsymbol{\lambda}_t^u$.

Remark 4.2. *The proposed value function approximation grows linearly in the number of time-coupling control variables and the number of stages, and its parameters have a natural economic interpretation as approximations of the shadow prices (dual values) of the time-coupling balance equations that connect subproblems of successive stages.*

The notion of shadow prices is closely related to the dual solution of linear programming problems which is used by dual dynamic programming algorithms to construct approximations of the cost-to-go function, e.g., Pereira & Pinto (1991), Shapiro (2011), De Matos et al. (2015). For example, in SDDP, an approximation of the expected cost-to-go function is constructed using Benders' cuts. That is, in each stage t, the expected cost-to-go function is replaced by variable $\boldsymbol{\theta}_{t+1}$ and constrained by the set of linear inequalities (which are supporting hyperplanes for $\bar{\mathcal{V}}_{t+1}(\cdot)$ at $\boldsymbol{u}_t$),

$$\boldsymbol{\theta}_{t+1} \geq \bar{\boldsymbol{\delta}}_{t+1,k}^\top \boldsymbol{u}_t + \bar{\boldsymbol{g}}_{t+1,k}, \tag{4.28}$$

for $k = 1, ..., K$, where K is the total number of cuts. The coefficient $\bar{\boldsymbol{\delta}}_{t+1,k} = \mathbb{E}[\boldsymbol{\delta}_{t+1}(\boldsymbol{\xi}_{t+1})]$ corresponds to the dual solution of the time-coupling constraints, also known as the shadow price, which defines the slope of cut k or its gradient $\bar{\boldsymbol{\delta}}_{t+1,k}^\top$, and $\bar{\boldsymbol{\beta}}_{t+1,k}$ is the intercept of cut k. Thus, the approximate cost-to-go function is replaced by $\boldsymbol{\theta}_{t+1}$ in the objective function. Equation (4.28) is included in the set of constraints.

Mapping this into Proposition 1, the expected duals in SDDP correspond to the shadow prices in SPA, i.e.,

$$\bar{\boldsymbol{\delta}}_{t+1,k}^{\top} \equiv \boldsymbol{\lambda}_t^{u\top} \qquad \text{and} \qquad \bar{\boldsymbol{g}}_{t+1,k} \equiv \boldsymbol{\beta}_t \equiv 0. \tag{4.29}$$

Hence, $\boldsymbol{\lambda}_t^u$ can be viewed as both the dual values (shadow prices) as well as the slope vector of the proposed linear value function approximation.

4.3 Solution Method

In this section, we describe a policy approximation algorithm to solve MSFLP. As the algorithm effectively searches for the shadow prices defined by the linear value function approximation, we refer to this method as *shadow price approximation* (SPA).

SPA trains the approximate cost-to-go function, $\bar{\mathcal{V}}_{t+1}(\mathbf{u_{t-1}}, \boldsymbol{\xi}_t)$, $\forall t = 1, \ldots, T$, of the stochastic-dynamic program by directly interacting with a simulation model of the resulting policy.

4.3.1 Dynamic Programming Formulation

We start our exposition of SPA by introducing a reformulation of the MSFLP to a dynamic program using linear value function approximation (4.27). Following the reformulation defined in (1.2), the value function of the dynamic program with a linear value function approximation is given by

$$\begin{aligned} \bar{V}_t(\mathbf{u_{t-1}}, \boldsymbol{\xi}_t; \boldsymbol{\lambda}_t^u) = \min_{u_{it}, x_{it}, z_{ijt}} \quad & \underbrace{\sum_i^F (f_i x_{it} + \sum_j^B z_{ijt} d_{ij})}_{\text{Total Operations Cost}} + \underbrace{\sum_i^F \lambda_{it}^u u_i}_{\text{Shadow Price}} \\ \text{s.t.} \quad & u_{it} = u_{i,t-1}(1 - \xi_{it}) + x_{it}, \quad \forall\, i = 1, \ldots, F, t = 1, \ldots, T, \\ & \text{Eqs.} \quad (5.7) - (5.10), \end{aligned} \tag{4.30}$$

for $t = 1, \ldots, T$, where $\lambda_{iT}^u \equiv 0, \forall i = 1, \ldots, F$.

The resulting parametric policy that uses this reformulation is given by

$$\begin{aligned}\bar{\pi}(\mathbf{u_{t-1}}, \boldsymbol{\xi}_t; \boldsymbol{\lambda}^u_t) = \operatorname{argmin}_{u_{it}, x_{it}, z_{ijt}} \quad & \sum_i^F (f_{it} x_{it} + \sum_j^B z_{ijt} d_{ij}) + \sum_i^F \lambda^u_{it} u_{it} \\ \text{s.t.} \quad & u_{it} = u_{i,t-1}(1 - \xi_{it}) + x_{it}, \quad \forall\ i = 1, \ldots, F, t = 1, \ldots, T, \\ & \text{Eqs.} \qquad (5.7) - (5.10).\end{aligned} \tag{4.31}$$

Given a set of slope vectors, $\boldsymbol{\lambda}^u = (\lambda^u_{i,1}, \ldots, \lambda^u_{i,T-1}), \forall i = 1, \ldots, F$, and a process of facility site failures, $\boldsymbol{\xi}^\omega = (\xi^\omega_{i,1}, \ldots, \xi^\omega_{i,T}), \forall i = 1, \ldots, F$, where $\omega \in \Omega$ is a scenario path of a stochastic process with its support set Ω, we can now use this policy to obtain an estimate of the upper bound $\bar{z}(\boldsymbol{\lambda}^u)$. By sampling N scenarios of possible site failures, $\hat{\boldsymbol{\xi}}^{\boldsymbol{n}} = (\hat{\xi}^n_{i,1}, \ldots, \hat{\xi}^n_{i,T}), \forall i = 1, \ldots, F$, and assuming time-coupling variable $u^n_{i,0} \equiv 0$ where $i = 1, \ldots, F$ and $n = 1, \ldots, N$, we obtain

$$\bar{z}(\boldsymbol{\lambda}^u) = N^{-1} \sum_{n=1}^N \hat{\bar{z}}^n(\boldsymbol{\lambda}^u), \tag{4.32}$$

where

$$\hat{\bar{z}}^n(\boldsymbol{\lambda}^u) = \sum_{t \in T} \left[\boldsymbol{v}_t^\top \boldsymbol{u}_t(\bar{\pi}(\boldsymbol{u}^n_{t-1}, \hat{\boldsymbol{\xi}}^n_t; \boldsymbol{\lambda}^u_t)) + \boldsymbol{w}_t^\top \boldsymbol{y}_t(\bar{\pi}(\boldsymbol{u}^n_{t-1}, \hat{\boldsymbol{\xi}}^n_t; \boldsymbol{\lambda}^u_t)) \right] \tag{4.33}$$

is a realization of the total cost for given slope vectors $\boldsymbol{\lambda}^u$.

4.3.2 Shadow Price Approximation

For the MSFLP, we can view the slope vector $\boldsymbol{\lambda}^u$ as the shadow price of opening facilities. Thus, it accounts for the possible future price that needs to be paid later for re-opening facilities after failure due to natural disaster. Therefore, the objective of SPA is to choose a slope vector that minimizes the upper bound,

$$\boldsymbol{\lambda}^{u*} = \operatorname{argmin}_{\boldsymbol{\lambda}^u} \bar{z}(\boldsymbol{\lambda}^u). \tag{4.34}$$

Unfortunately, the mapping $\boldsymbol{\lambda}^u \mapsto \hat{\bar{z}}$ is noisy and not convex, which entails solution of a non-convex, stochastic optimization problem that is known to be computationally intractable in its general form. We therefore cast SPA as a policy search strategy that can be supported by any method that is suitable for unconstrained (derivative-free) global optimization.

The shadow price approximation algorithm works as follows: first, we initialize shadow prices before the first iteration, i.e., $\lambda_t^{u,0} = 0$, $\forall t = 1, \ldots, T$. Then, at each iteration n, we randomly select a scenario path $(\hat{\boldsymbol{\xi}}_1^n, \ldots, \hat{\boldsymbol{\xi}}_T^n)$ and, going forward in time, we solve Problem (4.31) as a single-stage optimization problem that contains information on the cost-to-go via the shadow prices. The information that we carry from stage $t-1$ to stage t are the state variables u_{it} which then gets affected by $\hat{\xi}_{it}^n$ and the local decisions x_{it}. We do this until we reach the end of the planning horizon T. With the obtained realized upper bound $\hat{\bar{z}}^n$ and parameters $\boldsymbol{\lambda}^{u,n-1}$, the shadow price is updated using a generic update function $U^n(\cdot,\cdot)$, supported by a global optimization method, that returns a new set of trial shadow prices $\boldsymbol{\lambda^{u,n}}$. The notion of a generic update function is introduced in Powell (2007) which allows flexibility on the choice of the global optimization method. In Section 4.4.2, we briefly present a few global optimization methods that were tested to aide SPA with solving the MSFLP.

Finally, at the end of iteration N, the algorithm returns the best slope vector $\boldsymbol{\lambda}^{u,N}$ that minimize $\bar{z}$. This procedure is summarized in Algorithm 2.

An advantage of SPA over conventional stochastic and dynamic programming methods is that generating scenarios can be as simple as drawing a bootstrap sample from historical data. Bootstrapping alleviates the need to develop a stochastic model and effectively turns SPA into a data-driven approximate dynamic programming algorithm. When scenarios are generated by drawing from the empirical distribution of historical data, cross-validation and out-of-sample testing follow naturally. As such techniques are well-known to data scientists and practitioners of machine learning, SPA may be particularly attractive for the large pool of practitioners that are already familiar with such paradigms.

Algorithm 2: Shadow Price Approximation

Initialize $\boldsymbol{\lambda}_t^{u,0} = 0, \ \forall t = 1, \ldots, T$
Set initial state $\boldsymbol{u}_0 \equiv 0$ and $n = 1$
while $n < N$ **do**
  Draw scenario $(\hat{\xi}_1^n, \ldots, \hat{\xi}_T^n)$ from the stochastic process
  for $t \in \{1, \ldots, T\}$ **do**
    $(\boldsymbol{u}_t^n, \boldsymbol{x}_t^n, \boldsymbol{z}_t^n) \leftarrow \text{argmin}\Big\{ \sum_i^F (f_{it} x_{it} + \sum_j^B z_{ijt} d_{ijt}) + \sum_i^F \lambda_{it}^{u,n-1} u_{it} : \text{Eqs (2)-(6)} \Big\}$
  end for
  $\hat{z}^n \leftarrow \sum_{t \in T} \Big[\sum_i^F (f_{it} x_{it}^n + \sum_j^B z_{ijt}^n d_{ijt}) \Big]$
  $\boldsymbol{\lambda}^{u,n} \leftarrow U^n(\hat{z}^n, \boldsymbol{\lambda}^{u,n-1})$
  $n \leftarrow n + 1$
end while
Return $\boldsymbol{\lambda}^{u,N}$

4.4 Case Study

This section presents the construction of the rainfall-to-flood susceptibility mapping to incorporate the variability of rainfall as a random disruption parameter of the overall MSFLP model. This also briefly introduces the geographical and climactic characteristics of Cagayan de Oro City, a large city in the Philippines, which serves as the study case for this work. Section 4.4.2 briefly introduces the three different global optimization methods chosen to aid the SPA algorithm. Finally, we discuss the numerical results in Section 4.4.3 and cross-validation in Section 4.4.4. We programmed our solution approaches as single threaded applications in Python 3.7 with GUROBI 9.1.1 as the solver of the mixed integer linear program. All computations were carried out on a Linux 4.15 server with an Intel Xeon Gold 2.10GHz processor and 256 GB RAM.

4.4.1 Rainfall-to-Flood Susceptibility Mapping

In most developing countries, like the Philippines, one of the most detrimental natural disasters is heavy rainfall resulting in flooding. During the height of the COVID-19 pandemic, in addition to immediate hazards when heavy rainfall and flooding occurs, the probability of vaccination facilities becoming unavailable and inaccessible increases. It is for this reason that we chose flooding, which cause the facility failure, as the natural disaster risk to hedge from for the MSFLP.

In this work, the risk of failure of a vaccine facility i at time t is highly dependent upon its state u_{it} as triggered by the random disruption variable ξ_{it}. Thus, it is beneficial to know flooding event information of vaccine facility i. We constructed rainfall-to-flood susceptibility (RTFS) mapping of the different vaccination facilities F, where we combine real rainfall data with geo-spatial flood risk map data to infer whether a site is accessible or otherwise. The RTFS mapping provides the information for when the vaccination facility i becomes at risk of failure, i.e., closed and inaccessible, due to flooding at time t. Figure 4.1 shows the overall flowchart for the construction of the RTFS mapping.

Case Study: Cagayan de Oro City, Philippines

We performed numerical experiments on the data from a large city in the Philippines, Cagayan de Oro City, which experience heavy rainfall multiple times a year. Cagayan de Oro City is located along the northern central coast of Mindanao Island. As the capital city of the Province of Misamis Oriental, it serves as a regional center and hub for Northern Mindanao region. The city has an estimated population of more than 728,402 as of 2020 census and is the 10th most populated city in the Philippines, divided into eighty (80) districts (locally, *barangays*).

Dividing the city traverses the Cagayan de Oro River, one of the major rivers in Mindanao. This catchment discharges huge amount of water during a heavy downpour, even more so since the city is classified with a tropical monsoon climate. The city's flat slope and swallowing of the channel as it approaches the delta poses flood risks to the residents of the city. In the last twenty years, the city suffered heavy losses due to flooding (Mabao & Cabahug 2014). Thus, appropriate mitigation measure is imperative especially for long-term projects like the COVID-19 vaccination campaigns.

As of July 2021, the local government unit of Cagayan de Oro opens three (3) vaccination facility sites. These three vaccine facilities currently cater to the first few delegations of the vaccination. As the COVID-19 vaccination campaign ramps up, the city's local government plans to open more vaccine facilities to expedite the herd immunity goal of inoculating at least 70% of the total population. The roster of candidate vaccine facilities F were pre-selected to adhere to the standards of Philippines' Department of Health (DOH). The specifics of the RTFS mapping construction take into consideration this Cagayan de Oro City as an example.

RTFS map construction

The Philippine Atmospheric, Geophysical, and Astronomical Services Administration (PAGASA), the national meteorological and hydrological services agency of the Philippines, provided us with twenty-one (21) years of daily rainfall (in mm) data from year 2000 until 2020. The twenty-one years of data induce the different scenario cases S for the rainfall-to-flood susceptibility (RTFS) mapping of the vaccination facilities F. One general assumption of this work is the aggregation of rainfall on a *weekly*

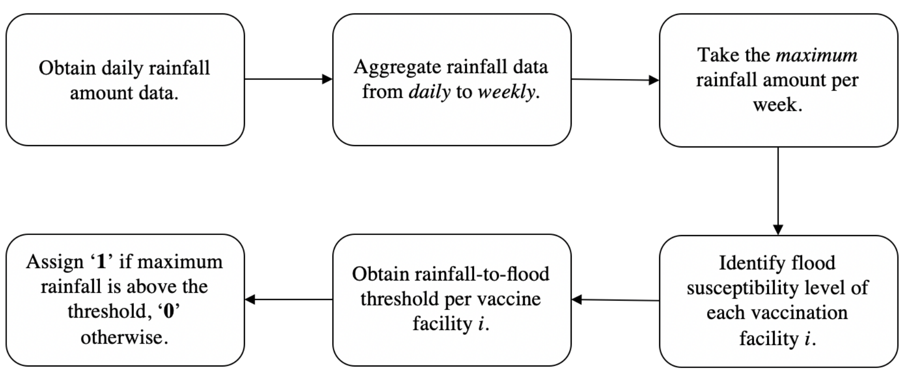


Figure 4.1: Rainfall-to-Flood Susceptibility (RTFS) Mapping Flowchart

basis. Given the twenty-one years of daily rainfall data from PAGASA, we aggregated one week of data following the fifty-two-week calendar year. Thus, the decision epoch t considered for this work is *weekly* over an entire year of planning horizon $T = 52$ decision weeks. It then follows that the decision to open and close facility i dynamically changes on a weekly basis. This assumption is carefully escalated to the local government unit and policy-makers noting that any vaccine facility may be instantaneously opened or closed in a chosen week.

To aggregate the daily precipitations into weekly periods, the *maximum* rainfall amount of week t is taken to represent the aggregated rainfall amount of week t. This implies that if the *maximum* amount of rain on week t surpasses the rainfall-to-flood threshold, the vaccine facility i will be closed and inaccessible for the entire week t. The rainfall-to-flood threshold is identified through a series of circumspect investigation using disaster risk geo-spatial information system (GIS) and the official guidelines of PAGASA on rainfall advisory and classifications.

GeoRisk Philippines led by the Philippine Institute of Volcanology and Seismology (PHIVOLCS) created *HazardHunterPH* - an online GIS tool which generates indicative hazard assessment reports on specified locations in the Philippines (PHIVOLCS & DOST 2019). *HazardHunterPH* maps the entire Philippines and shows different layers of geographical hazards, e.g., seismic (earthquakes), volcanic, and hydro-meteorological (flood, landslides, and storm surges). Figure 4.2 shows a map of Cagayan de Oro City with the overlaid flood hazard map. The different hues of violet represent the flood susceptibility of an area ranging from *Very High Susceptibility* in darker hue to *Low Susceptibility* in lighter hue. This signifies that if a vaccine facility i is located in a *Very High Susceptibility* zone, vaccine facility i is easily flooded more than vaccine facilities located in *Low Susceptibility* zones.

We also mapped the set of candidate vaccine facility sites F, as provided by the local government unit, to distinguish the flood susceptibility of each site location shown in Figure 4.2. For a more thorough investigation, a further examination is conducted with the Engineering Resource Center of Xavier University - Ateneo de Cagayan, a leading university of Cagayan de Oro City, on the flood susceptibility level of the routes leading to the different facilities. The assumption on this further examination insinuates that if *routes* leading to the vaccine facility i has a higher flood susceptibility level, the chosen susceptibility level of facility i will be lifted to the next (higher) susceptibility level. This alludes logically that even though vaccine facility i is not flooded, it will be

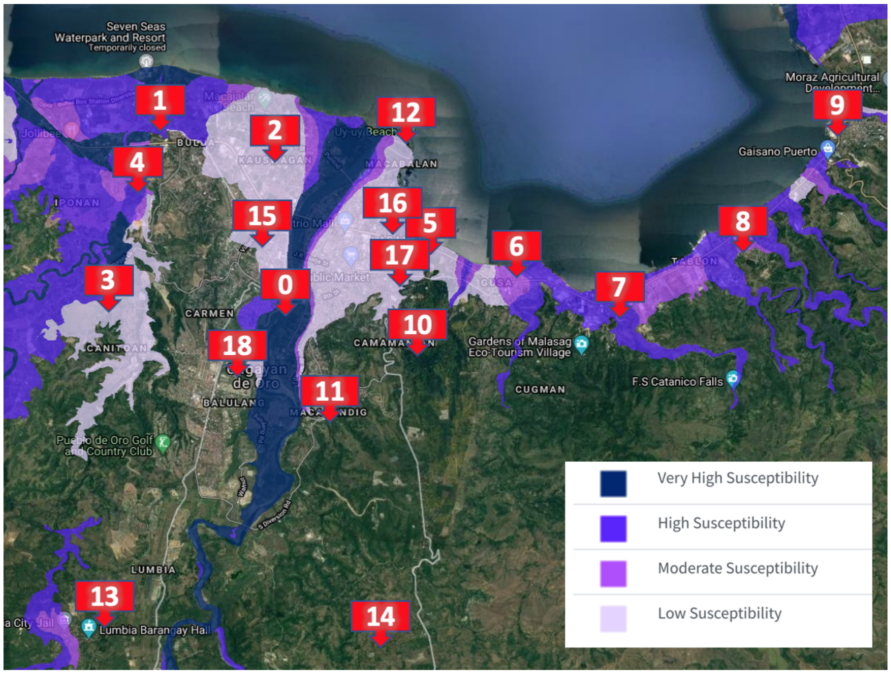


Figure 4.2: Candidate vaccine facility sites of Cagayan de Oro City, Philippines with overlaid flood hazard map

inaccessible due to the routes leading to it being flooded.

After assigning the prominent flood susceptibility level of each vaccine facility site i, the rainfall-to-flood threshold is identified for each site. Nonetheless, the rainfall-to-flood susceptibility (RTFS) mapping leaves room for modeling error. Thus, we have created three conservativity scenario cases that vary the translation of precipitation amounts to whether a facility is flooded or not. These scenario cases are based on the prominent flood susceptibility level and the official rainfall advisories, classification, and measurement of PAGASA (DOST-PAGASA 2019). The cases $\mathbf{S} = \{Low, Medium, High\}$ are instituted from the different ranges of rainfall measurement. Each case $\mathbf{s}$ take the level of conservativity of the ranges of rain measurement. $High$ conservativity case takes the minimum value, the $Medium$ conservativity case takes the median value, and the Low conservativity case takes the maximum value of the rainfall measurement range. The inclusion of the different cases $\mathbf{S}$ parlays more robustness in the decision-making process of the study. This would also allow a more encompassing solution to understand in depth the different plausibilities played by natural disasters.

The different RTFS mapping cases were transformed into zeros and ones to show the value of the random variable ξ_{it} of each vaccine facility i at time t. Recall that if a vaccine facility site i flooded at time t, i.e., $\xi_{it} = 1$, then the facility is disrupted and unavailable, otherwise, $\xi_{it} = 0$. Each of these three different RTFS mapping conservativity cases contain the twenty-one years scenarios with fifty-two weeks, one for each decision epoch t of the entire planning horizon T.

Finally, it should be noted that the steps carried out on the construction of the RTFS mapping for Cagayan de Oro City can be utilized to create RTFS mapping for other cities in other developing countries. Thus, distinct differences on geophysical and geographical characteristics of the city should be assessed thoroughly to provide more conclusive inference on the decision planning. In Section 4.4.4, another large city in the Philippines, General Santos City, served as a validation test data set for the SPA algorithm and the construction of the RTFS mapping.

4.4.2 Global Optimization Methods

In this section, we briefly introduce three different global optimization methods used for the numerical experiments of solving MSFLP using the SPA algorithm. Although, in all fairness, the SPA algorithm may be aided by any global optimization methods, these three methods and algorithms were chosen as they seem to perform well in the literature in comparison to other methods. For the MSFLP, the global optimization method tunes the shadow prices of flood-prone facilities, λ_{it}^{u}, iteratively through evaluations of the objective function.

Stochastic Gradient Descent

Stochastic gradient descent (SGD) is an iterative method to optimize a differentiable or subdifferentiable objective function (Bottou & Bousquet 2011). At large, it is a stochastic approximation of the gradient descent algorithm, as it reinstates the actual gradient, i.e., computed from the total set of data, by an estimate computed from a subset of the data as randomly chosen. SGD aids notably in optimization problems with high dimension in reducing the computational strain which achieves faster iterations as offset for a lower convergence rate.

We use SGD to find the best shadow cost λ_{it}^{u} of each facility i in each period t. As the optimization problem is non-differentiable, we use the post-decision state of each facility, u_{it}, as the subgradient. The learning rate is set by the widely used *adaptive gradient algorithm* (AdaGrad) - a popular enhancement to calculate step-sizes dynamically (Ward et al. 2018). The performance of SGD to solve the MSFLP via SPA is summarized in 4.4.3.

Bayesian Optimization

Bayesian Optimization (BayOpt) is an algorithm that optimizes objective functions which are non-differentiable and expensive to evaluate (Pelikan et al. 1999). It is befitting for optimization problems with continuous domains of around 20 dimensions and is stochasticity-tolerant in function evaluations. BayOpt constructs a surrogate for the objective function and quantifies the stochasticity in that surrogate using a Bayesian machine learning technique, Gaussian process regression, and then uses an acquisition function defined from this surrogate where decision to sample is made.

As a black box optimization algorithm, BayOpt views the MSFLP as a function, with λ_{it}^{u} as function arguments and the resulting total cost as the (random) function value. BayOpt iteratively tunes λ_{it}^{u}, and after a finite number of function evaluations returns λ_{it}^{u} that minimize the function on average. For our numerical experiments, we use the BayOpt implementation provided by Nogueira (2014).

Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

The *covariance matrix adaptation evolution strategy* (CMA-ES) is a powerful stochastic, derivative-free, and population-based search method to solve difficult numerical optimization problems, e.g., non-convex, non-smooth, noisy problems, etc. in continuous domain. It is considered as the *state-of-the-art* in evolutionary computation and adopted in multiple applications and fields. See Hansen (2006) for a detailed review. CMA-ES optimizes a continuous black-box function, i.e., the problem's objective function, $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ by sampling from a non-stationary multivariate normal search distribution $\mathcal{N}(m^{(g)}, \sigma^{(g)^2}, C^{(g)})$, with mean $m^{(g)}$, standard deviation $\sigma^{(g)}$, and correlation matrix $C^{(g)}$. Here, the mean serves as the initial guess and the standard deviation as initial trust region. The algorithm iteratively adjusts the parameters of this distribution based on successful search steps.

Like the former two methods, we use CMA-ES to find the shadow cost of opening a facility. However, for the algorithm to converge quickly, it is imperative to define a good trust region by setting $\sigma^(g)$. In our setting, this is equivalent to setting step-size σ parameter which is multiplied with the shadow cost. Online Appendix B shows the sensitivity of changing the choice of step-size σ parameter on the convergence of the algorithm and the quality of its solution. A good *rule-of-thumb* in choosing initial value step-size σ parameter is also presented in online Appendix B. We use the package in Hansen et al. (2019) for the numerical experiments.

4.4.3 Numerical Results

In this section, we show the results of the numerical experiments of employing SPA with different global optimization methods to solve MSFLP with failure under risk of flooding. We have provided the local government unit with a spreadsheet tool containing vaccine facility information sheet, population information sheet, and distance information sheet which we used for the numerical experiment. Online Appendix A shows these three different spreadsheets. The RTFS mapping for Cagayan de Oro City, discussed in Section 4.4.1, is also adopted.

Initialized with shadow prices $\lambda^{u,0} \equiv 0$ tuned iteratively through the SPA algorithm, Figure 4.3 shows the evolution of the optimal objective function value $\hat{\bar{z}}^n(\boldsymbol{\lambda}^u)$, as computed in Equation (4.33), for the three conservativity cases discussed in Section 4.4.1 using three global optimization methods.

It is notable that both global and derivative-free optimization methods, CMA-ES and BayOpt, perform better than SGD. Both CMA-ES and BayOpt yield approximately 30-40% lower cost than a baseline approach that does not consider the risk of flooding, i.e., where $\lambda^u_{it} = 0, \forall t = 1, \dots, T, i = 1, \dots, F$, across the different conservativity cases.. With the deterministic approach where risk of flooding is neglected, the total cost of the year-long vaccine distribution roll-out will be 47,678,488.81 (in Philippine Peso).

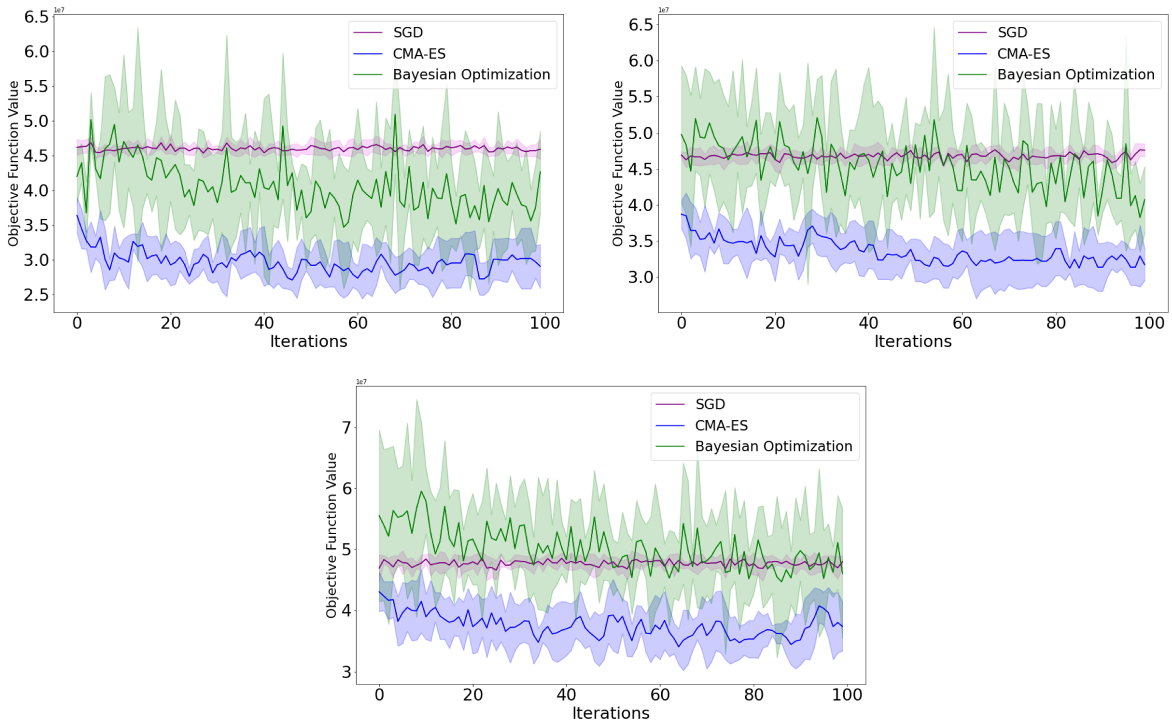


Figure 4.3: Evolution of the objective function value in three different cases - *Low* (top-left), *Medium* (top-right), and *High* (bottom) conservativity case for Cagayan de Oro City using three global optimization methods.

On the other hand, there will be a savings of 15,661,805.95 (in Philippine Peso) when we include the risk of flooding in the optimization model of the *low* conservativity case. Without the inclusion of the risk of failure due to flooding, the chosen facilities will be the same from the start until the end of the planning horizon.

Different parameters for both CMA-ES and BayOpt were investigated - varied number of iterations ($N = 10, 25, 50, 75, 100, 200, 500, 1000$), various population size and different M-values ($M = 1, 1000, 5000, 30000, 50000, 100000, 200000, ..., 100000000$) for CMA-ES, and changing lower- and upper-bounds for BayOpt. We have observed that the algorithm converged around $N = 100$ which prompted us to use $N = 100$ across the different scenario cases and the two different cities. The best parameter configurations for both global optimization methods were tested multiple times to set lower and upper 95% confidence interval as shown by the filled gaps in Figure 4.3 with the darker line as the mean.

Interestingly, the MSFLP solved by SPA with SGD does not constitute any decrease of the objective function value across the three conservativity cases. The objective function value maintains rather smoothly around an initial local optimum solution, even with the use of AdaGrad step-sizes. We conjecture that the poor performance of SGD is due to the non-convexity of the optimization problem, as our state variables, $\boldsymbol{u_t}$, are binary, so that the resulting sub-gradients are misleading. There is an abundant literature on various enhancements and improvements on the stochastic gradient descent that might mitigate this issue, e.g., Khazaei & Powell (2018), Shuai et al. (2019), and Ghadimi & Powell (2022), however, a deeper investigation is beyond scope and left to future work. Nonetheless, this analysis demonstrates that basic SGD is not useful in solving the MSFLP via SPA.

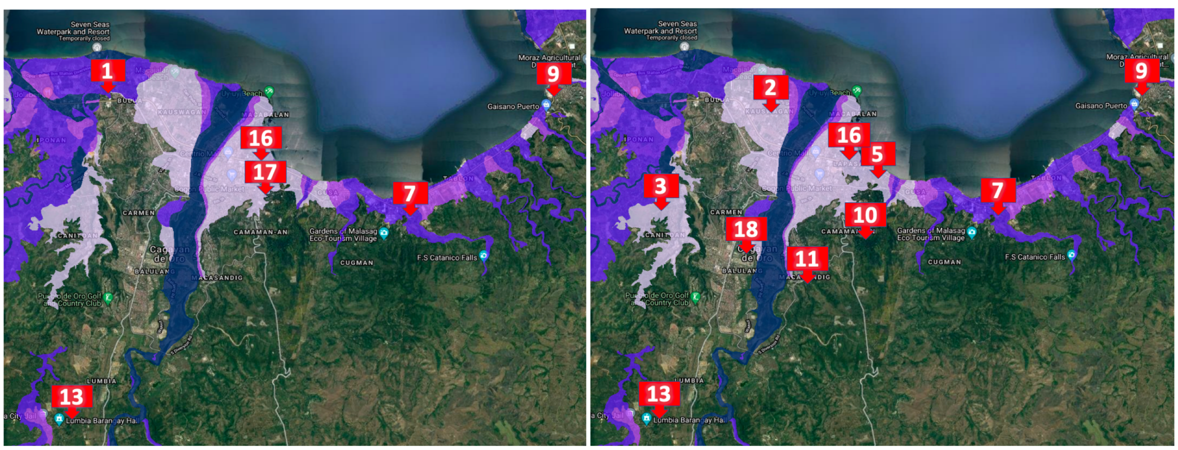
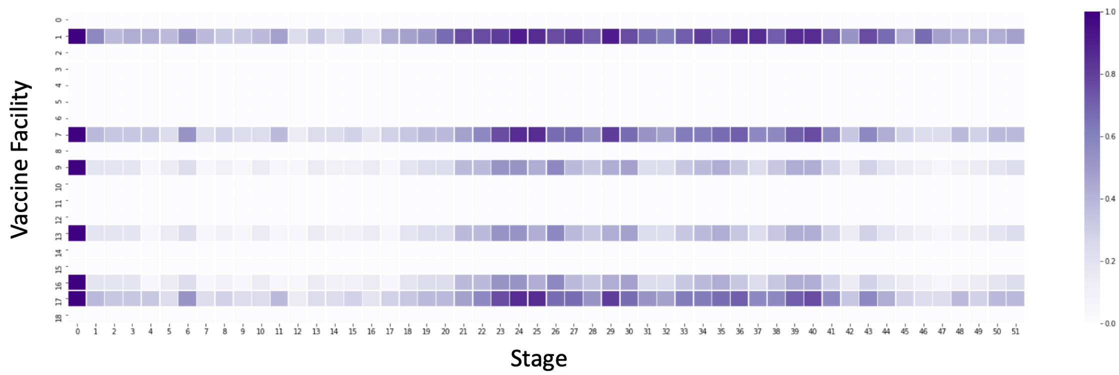
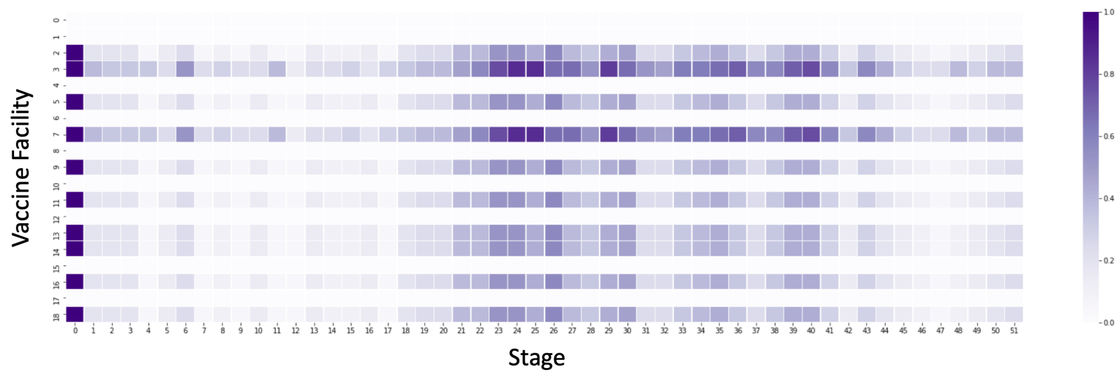


Figure 4.4: Chosen facilities when a) $\boldsymbol{\lambda}^u = 0$ (left) and when b) $\boldsymbol{\lambda}^u = \boldsymbol{\lambda}^{u,N}$ (right) for Cagayan de Oro City.

Furthermore, Figure 4.4 shows the geographical maps of chosen optimal facilities with and without hedging for the risk of disruption due to flooding of *High* conservativity case. As demonstrated, our solution approach lead to a reduction of 30-40% of the overall vaccination roll-out operation cost in contrast to ignoring the risk of natural disaster occurrence. Figure 4.4 shows that opening more facilities reduces the total cost of the vaccination program of the city. The choice of opening facilities, without hedging for the risk of flooding, may incur additional costs in the year-long vaccination campaign since these facilities will possibly be flooded. Thus, they need to be re-opened multiple times across the planning horizon.



[a]



[b]

Figure 4.5: Heatmap of chosen facilities when a) $\boldsymbol{\lambda}^u = 0$ and when b) $\boldsymbol{\lambda}^u = \lambda^{u,N}$ for Cagayan de Oro City.

In Figure 4.4a (left), two out of six facilities are in the very high susceptibility areas and two other facilities are in the moderate susceptibility areas. In contrast to that, Figure 4.4b (right) shows that only one out of ten facilities are in the very high susceptibility area and four are in the moderate to low susceptibility areas, and the rest are in the safe areas.

Additionally, Figure 4.5 shows the heatmap of the average re-opening of each facility. As observed, chosen facilities for policy when $\lambda_{it}^u = 0$, $\forall t = 1, \ldots, T, i = 1, \ldots, F$, Figure 4.5a tend to be re-opened more than the policy when $\lambda_{it}^u = \lambda_{it}^{u,N}$, $\forall t = 1, \ldots, T, i = 1, \ldots, F$, Figure 4.5b. Every time a facility is flooded and re-opened, a corresponding cost is incurred. Thus, if there are more dark hues of violet in the heatmap for a policy, then more facilities are to be re-opened leading higher costs.

	City 1				
Instance	SPA	RunTime (s)	SDDiP	RunTime (s)	Gap
Low	30572531.35	2626.2	34284648.51	2069.7	11.45%
Medium	33016231.32	2422.2	39246742.86	2244.3	17.24%
High	37550449.33	2556.8	41609220.07	2358.3	10.25%
	City 2				
Instance	SPA	RunTime (s)	SDDiP	RunTime (s)	Gap
Low	16381368.74	382.6	21610651.43	606.6	27.52%
Medium	16920797.31	402.0	21978319.44	751.4	26.0%
High	20069987.71	386.4	22892157.47	897.7	13.14%

Table 4.1: Mean optimal objective function value comparison between shadow price approximation (SPA) and stochastic dual dynamic integer programming (SDDiP) algorithms for two cities (City 1 - Cagayan de Oro City and City 2 - General Santos City) under three different susceptibility cases.

Moreover, given the data-driven characteristic of SPA, we used cross-validation to avoid over-fitting. First, we executed 7-fold cross validation where the 21-year RTFS data set was split into seven training sets. Then, we held out one subset at a time and trained the algorithm on the remaining set to obtain $\lambda_{it}^{u,N}$, $\forall t = 1, \ldots, T, i = 1, \ldots, F$. We then evaluated the model on the held-out subset using the obtained $\lambda_{it}^{u,N}$, $\forall t = 1, \ldots, T, i = 1, \ldots, F$. This step was repeated across all the different subset folds.

To assess the solution quality of our approach, we also solved the MSFLP problem using the current gold standard in solving multistage stochastic integer programming problems, stochastic dual dynamic integer programming (SDDiP). We use SDDiP with so-called *strengthened Bender's* cuts to obtain a tightened lower bound (Zou et al. 2019). This not only provides us with an implementable policy but also gives us an optimality gap. We find that SPA is near-optimal and even outperforms the SDDiP policy for all MSFLP instances. Table 4.1 reports the mean optimal objective function value, run time, and gap between SPA and SDDiP algorithm for the two cities, City 1 - Cagayan de Oro City and City 2 - General Santos City (see Section 4.4.4 for further details), under study for all three susceptibility cases.

We refer the reader to Online Appendix C for a brief summary of SDDiP and further numerical results on using it for solving the MSFLP problem.

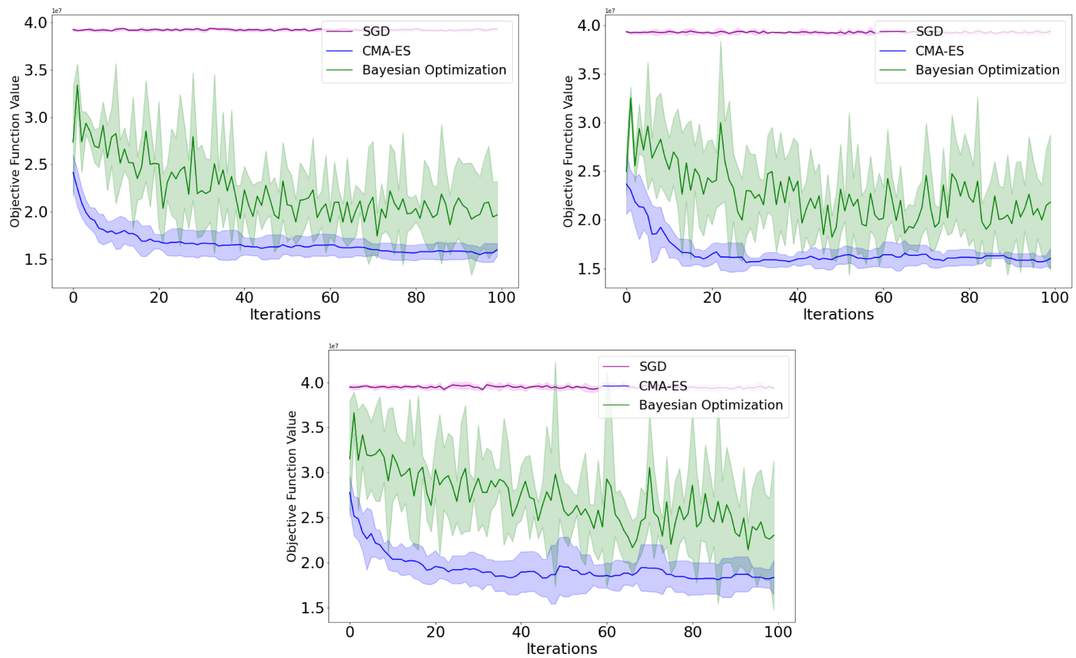


Figure 4.6: Evolution of the objective function value in three different cases - *Low* (top-left), *Medium* (top-right), and *High* (bottom) conservativity case for General Santos City using three global optimization methods.

A more extensive collection of numerical experiments is left to future work. This will involve different modelling choices and algorithmic parameters, e.g., N-iterations, trust region, other gradient and gradient-free optimization methods, as well as more extensive benchmarks against SDDiP, possibly considering recent enhancements, such as the level bundle method and reformulations to obtain tighter cuts.

Finally, we were interested in how well our approach would perform on unseen data. To achieve this, we tested our approach on data for another city in the Philippines, General Santos City.

4.4.4 Test/Validation Data Set: General Santos City (Philippines) Instance

Akin to the first city, MSFLP solved with SPA algorithm yields lower cost, approximately 30-40% lower, across the different cases, i.e., $\lambda_{it}^{u} = 0, \ \forall t = 1, \ldots, T, i = 1, \ldots, F$, for the second city. The evolution of the objective function value is shown in Figure 4.6. Furthermore, Figure 4.7 shows the geographical map of chosen optimal facilities with and without hedging for the risk of failure due to flooding of *High* conservativity case.

Looking at Figure 4.7a (left), three vaccine facilities (Facilities 7, 11, and 25) are chosen when the risk of flooding is not considered, i.e., $\lambda_{it}^{u} = 0, \ \forall t = 1, \ldots, T, i = 1, \ldots, F$. However, when hedging from flooding risks, eight new vaccine facilities (Facilities 1, 3, 5, 8, 14, 18, 20, and 24) are chosen, shown in at Figure 4.7b (right). Although this is counter-intuitive, the eight new vaccine facilities with policy $\lambda_{it}^{u} = \lambda_{it}^{u,N}, \ \forall t = 1, \ldots, T, i = 1, \ldots, F$ are less risky of flooding in contrast to the policy without hedging from natural disaster risk. Thus, lesser cost is incurred following the risk-averse policy.

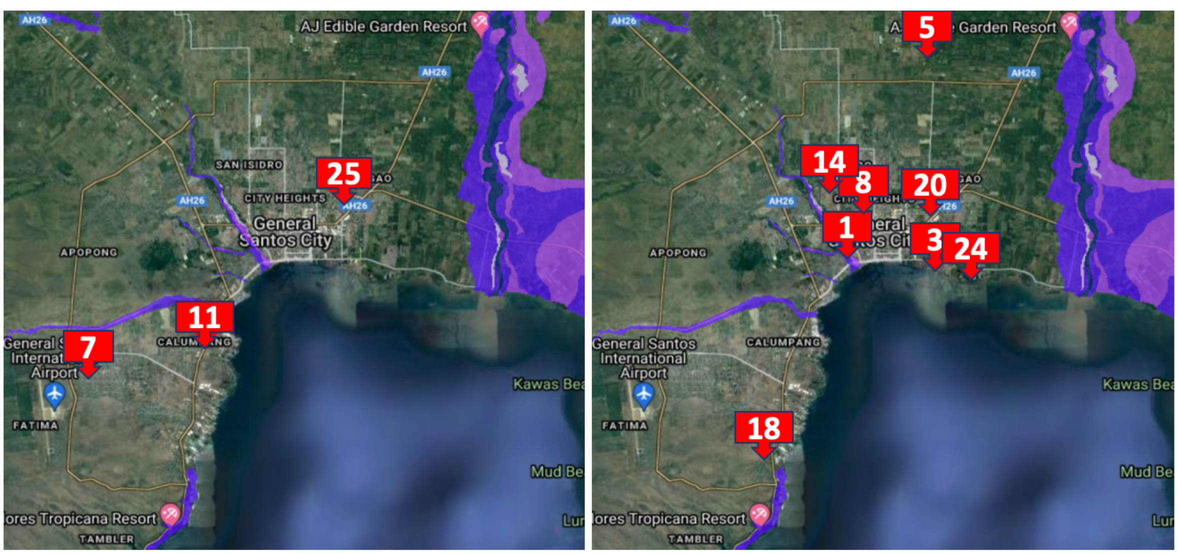


Figure 4.7: Chosen facilities when a) $\boldsymbol{\lambda}^{u} = 0$ (left) and when b) $\boldsymbol{\lambda}^{u} = \lambda^{u,N}$ (right) for General Santos City.

4.5 Conclusion and Outlook

In this chapter, we consider a multistage facility location model for vaccine distribution which integrates the risk of disruption or failure of a vaccine facility due to natural disaster, i.e., in our setting flooding. We solve the multistage stochastic facility location problem (MSFLP) by introducing a new algorithm, named *shadow price approximation* (SPA), which aims at approximating the shadow price of opening flood-prone vaccine facilities. The SPA algorithm proceeds by tuning the parameters of a linear value function approximation which is present in the objective function of base optimization model. For parameter tuning, it resorts existing methods for global optimization. SPA bootstraps sample paths from real, historical data as the random process and uses them as direct inputs of a deterministic model. Thereby, SPA alleviates the need to fit a stochastic model in contrast to most of stochastic optimization methods.

Our approach yields approximately 30-40% lower cost than a baseline approach that does not consider the risk of flooding, across different rainfall-to-flood conservativity cases. An interesting and counter-intuitive insight from the results show that not only are vaccine sites opened in less flood-prone areas, but also more vaccine sites are being opened by the risk-aware solution. This intently reduces, not only the total cost of operation of the vaccination campaign of the city, but also of its duration. Without considering the risk of failure, the chosen facilities would be the same from start to end of the planning horizon, which would lead to facilities re-opening that are prone to failure. To further benchmark the proposed modelling and algorithmic approach, we compare the solution obtained by the *state-of-the-art* methodology to solve multistage stochastic integer program - Stochastic Dual Dynamic Integer Programming (SDDiP). Our analysis shows that our solution is near-optimal for the considered instances and even outperforms the SDDiP policy. The policy produced by SDDiP is clearly suboptimal as the convergence of the algorithm stalls leaving a gap of approximately 20%.

Cagayan de Oro, a large city in the Philippines, which is prone to multiple flooding events per year, provided the data instance for the vaccination campaign costs. Weather and rainfall data were provided by a government agency and an industrial weather company. After having presented the results to the local government of the aforementioned city, our study was used as a basis to request for the independent handling of the inoculation process of the city and be provided with more vaccine supply in order to achieve herd immunity as fast as possible. The request was approved by the Chair of the COVID-19 Vaccine Cluster of the Philippines and by the end of Year 2021, Cagayan de Oro City posted one of the highest vaccination rates in the country and the highest in the region. To test the validity of the model and the solution method, another large city in the Philippines was tested.

Finally, there are several interesting directions to investigate for future research. Future work could look at various vaccine distribution modelling approaches, e.g., a model with endogenous vaccine demand, inclusion of specific vaccine storing requirements, and arrival of vaccine supply. These approaches, although beyond the scope of this work, can be beneficial in proactive planning for future pandemics. Flooding is not the only natural disaster that occur in different nations; tornadoes, storm surges, landslides, earthquakes, and volcanic eruptions daunt all countries and can disrupt long-term vaccination campaigns and other projects. In a future work, we may extend the formulation of the model to include other modelling approaches and other sources of natural disasters to expand the scope further. Various gradient- and gradient-free optimization methods and other algorithmic tuning choices may also be investigated further to explore capabilities of the *shadow price approximation* algorithm.

Appendix 4.A Spreadsheet Tool for Data Collection

Through a spreadsheet tool we have created as a data template, the local governments of the two cities provided us with the necessary cost parameters, e.g., fixed cost f_i to open vaccine facility i, distance cost d_{ij} from vaccine facility i to district j, capacity C_i of each vaccine facility i, and the vaccinating population P_j of each district j. Appendix A shows screenshots of the spreadsheet tool given to the local government for data collection.

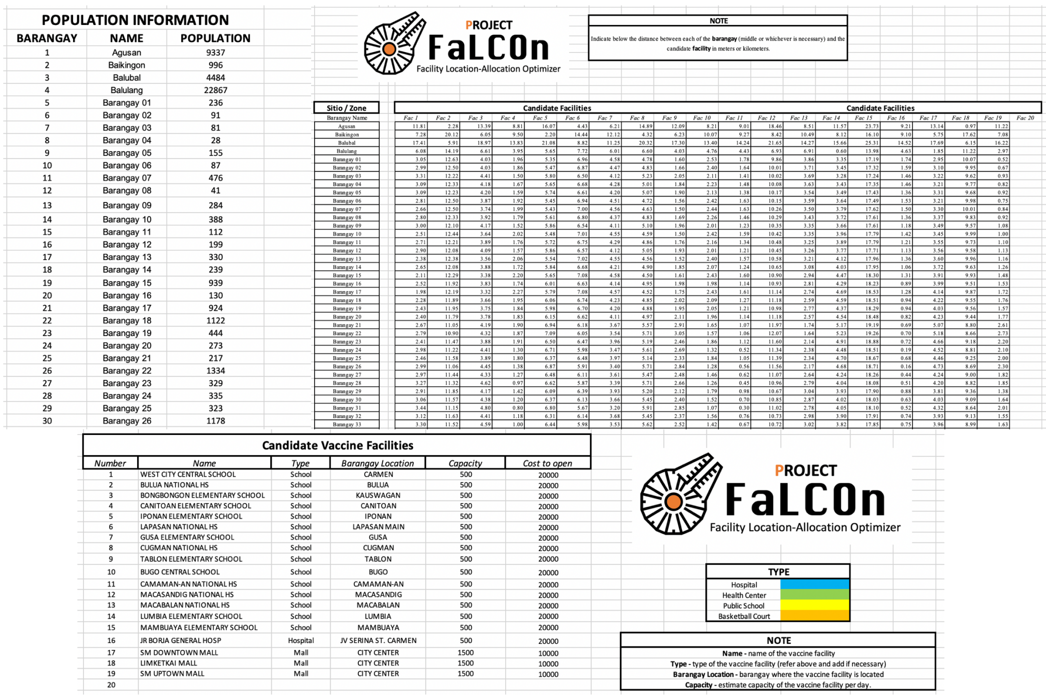


Figure 4.8: Spreadsheet Tool for Data Collection

Appendix 4.B How different values of the step-size σ parameter affects the objective function value

We tested different choices of tunable step-size σ parameter, starting with $\sigma = 1$ and increasing the value incrementally to an upper bound of the shadow price which equals the total expected cost when $\lambda_{it}^{u} = 0$ and more. The following figures show how different values of the step-size σ parameter (x-axis) affects the objective function value (y-axis). As seen, there are optimum values of the step-size σ parameter which coincide to f_{it}, the fixed cost of opening a vaccine facility i at time t, or higher.

A *rule-of-thumb* and starting point where the best step-size σ parameter values which yield the lowest possible cost starts when step-size σ parameter is at least as high as f_i, the cost of opening facility i, or higher. When the step-size σ parameter value is less than the opening cost f_i, there is less, even possibly, no change from when $\lambda_{it}^u = 0$. The step-size σ parameter value, for obvious reasons, affects the tuning of λ_{it} as step-size σ parameter serves as the mean value of cost of failure of facility i due to a natural disaster.

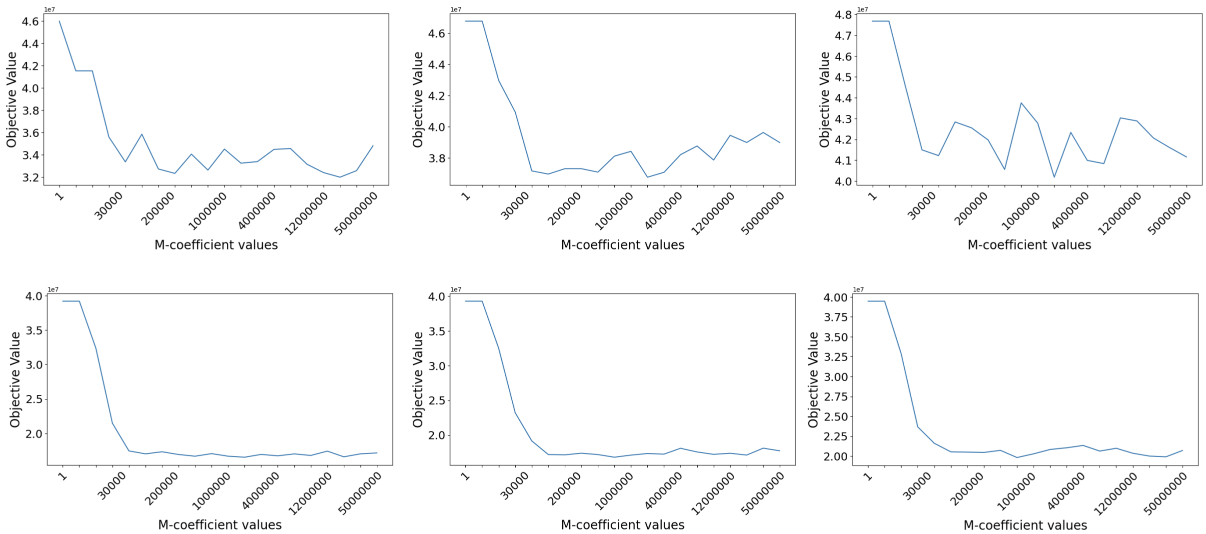


Figure 4.9: Different values of the step-size σ parameter in relation to the objective function value for City 1 (top) and City 2 (bottom) under three different susceptibility cases - *Low* (left), *Medium* (middle), and *High* (right).

Appendix 4.C Stochastic Dual Dynamic Integer Programming (SDDiP)

The only general purpose approach to obtain a lower bound and optimal policy of the MSFLP that is computationally tractable is *stochastic dual dynamic integer programming* (SDDiP). SDDiP requires reformulation of the MSIP as dynamic program which it then solves through approximating the cost-to-go function by a convex piece-wise linear function. SDDiP converges to an optimal policy under the following assumptions: (1) random model parameters follow a discrete distribution and are stagewise independent; and (2) all time-coupling decision variables that enter the state space of the dynamic program are binary variables.

The MSFLP fulfils the requirement of binary states naturally, i.e., u_{it} is 1 if facility i is open at time t and 0, otherwise. However, in order to enforce stagewise independence, we randomly shuffle the discrete distribution of rainfall outcomes at each stage, thereby destroying the serial dependence of rainfall from one period to the next. Although the true problem has serial dependence, this will allow us to compare SPA with SDDiP, the latter providing us with an analytical lower bound. Figure (4.10) show convergence of the upper and lower bound of for the two cities used in the case study under three different conservativity cases.

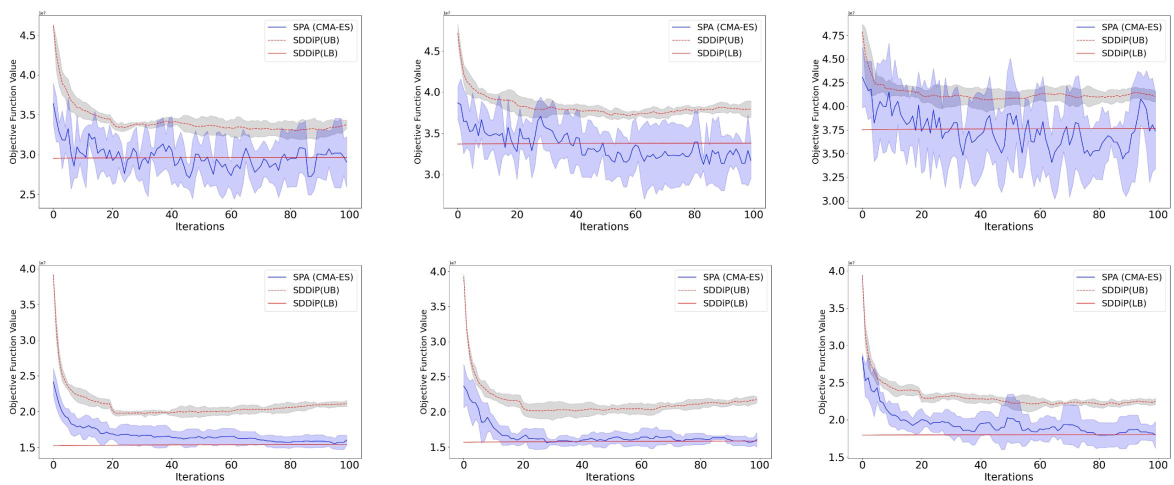


Figure 4.10: Upper- and lower-bound convergence performance of SDDiP algorithm with SPA (CMA-ES) algorithm for the MSFLP problem for City 1 (top) and City 2 (bottom) under three different conservativity cases - *Low* (left), *Medium* (middle), and *High* (right).

As can be seen from Figure 4.10, SPA outperforms SDDiP across all instances. Not only does SPA provide a significantly better policy than SDDiP, we moreover see that SDDiP clearly struggles with closing the gap between the analytical lower bound and the statistical upper bound.

After $N = 100$ iterations and relatively similar running time, even with the use of *strengthened* Bender's cuts - these are valid and finite cuts where a general Benders' cut is tightened by solving a nodal mixed integer program subproblem with its solution equal to a basic optimal LP dual solution (Zou et al. 2019) - the upper and lower bound of SDDiP still exhibit a gap of approximately 20%. This result not only demonstrates that SPA is able to find near-optimal solutions to the multistage stochastic facility location problem but it is also evidence of the immense difficulty of this problem class.

Future work will explore a more extensive collection of numerical experiments involving different modelling choices and various algorithm parameters, e.g., different choice of basis function, step-size σ parameters, other gradient and gradient-free optimization methods, as well as more extensive cross validation. It may also be worthwhile to benchmark the solution against recent enhancements, e.g., new cut families of the SDDiP methodology.

Chapter 5

Multistage Stochastic Facility Location under Facility Disruption Uncertainty

In Chapters 2 and 3, we explored the numerous applications of SDDP and discussed the enhancements and improvements made to the algorithm, including the introduction of SDDiP. Subsequently, in Chapter 4, we focus our discussion on a real-world application of optimizing vaccine distribution in two urban cities in the Philippines during the COVID-19 pandemic under natural disaster risk.

In this chapter, we present a general model for multistage stochastic capacitated facility location to address uncertainty arising from facility disruptions. Facility disruptive incidents, such as power outages, industrial accidents, transportation and infrastructure problems, as well as natural catastrophes, can result in extended periods of facility failures. We propose two solution algorithms for this problem class: (1) stochastic dual dynamic integer programming (SDDiP), the a state-of-the-art algorithm for solving multistage stochastic integer programs, and (2) shadow price approximation (SPA), an algorithm that utilizes trained parameters of the linear value function approximation to minimize an upper bound on the optimal objective value. Through numerical investigations, we demonstrate that SPA consistently outperforms SDDiP, i.e. identifies superior policies, across all tested instances from the literature.

5.1 Introduction

Facility location problem (FLP) is a well-studied problem in operations research that aims to optimally locate facilities in order to provide services to and satisfy demands of customers. There have been many variants of the classic FLP to respond to the diverse needs of different industries and firms; among them are location models in a continuous space (Hunagund et al. 2022), stochastic location models (Correia & Saldanha-da Gama 2019), location of hubs (Alumur et al. 2021), healthcare facilities (Ahmadi-Javid et al. 2017), public sector facilities (Haase et al. 2019), and location for humanitarian logistics (Trivedi & Singh 2018).

In this article, we consider facility location decisions accounting for facility disruptions. Incidents like power outages, industrial accidents, problems with the transportation infrastructure, and natural catastrophes disrupt facilities and may cause facility failures (Cheng et al. 2021). The impact of these events and the probability of their occurrences are difficult to estimate due to lack of high-quality historical data (Cui et al. 2010). Facility location decisions are long-term and difficult to rectify. Thus, it is important to take into account future disruption uncertainties to assure that facility location decisions are sufficiently robust to avert significant costs in the future. Figure 5.1 shows the timeline of a multistage stochastic facility location under facility disruption uncertainty.

Although there exists a stream of literature covering this subject - reliable facility location models - they only cover single- and two-stage models (Snyder et al. 2016). Some disruptions may occur multiple times across the planning horizon and some facilities need to be re-opened when disruptions are overcome. To the best of our knowledge, the *multistage* case of facility location under facility disruptions has only yet been explored in Seranilla & Löhndorf (2023). We present the multistage stochastic facility location problem under facility disruptions as multistage stochastic mixed-integer program (MSIP).

On solution methodology aspect, this problem class has only been recently tackled due to its complexity and difficulty to solve. Stochastic dual dynamic integer programming (SDDiP) is the current state-of-the-art to solve MSIP problems presented in Zou et al. (2019). It requires MSIP to be reformulated as a dynamic program

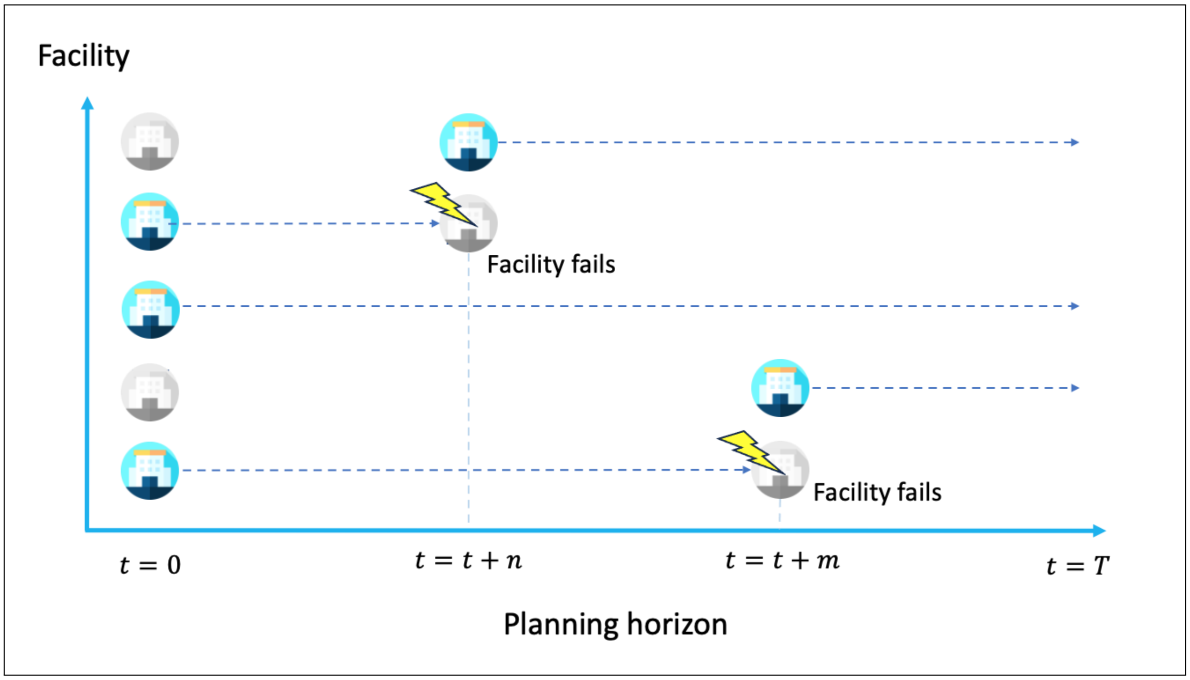


Figure 5.1: Timeline of a multistage stochastic facility location under facility disruption uncertainty.

and can obtain an optimal policy that is computationally tractable. A more novel technique is the shadow price approximation (SPA). SPA trains the parameters of a linear value function approximation by minimizing an upper bound on the optimal objective value. We present numerical results of these two approximate dynamic programming algorithms to solve an instance of the MSFLP.

5.1.1 Multistage Stochastic Integer and Dynamic Programming Formulations

We start by introducing a multistage stochastic mixed-integer problem (MSIP) with the form

$$\begin{aligned} \min_{\mathbf{u_1},\mathbf{y_1}\in\mathcal{X}_1} \Big\{ v_1^\top \mathbf{u_1} + w_1^\top \mathbf{y_1} + \mathbb{E}_{\boldsymbol{\xi}_{[2,T]}|\xi_{[1,1]}} \Big[\min_{\mathbf{u_2},\mathbf{y_2}\in\mathcal{X}_2(\mathbf{u_1},\boldsymbol{\xi}_2)} \Big\{ v_2^\top \mathbf{u_2} + w_2^\top \mathbf{y_2} + \dots \\ + \mathbb{E}_{\boldsymbol{\xi}_{[T,T]}|\xi_{[1,T-1]}} \Big[\min_{\mathbf{u_T},\mathbf{y_T}\in\mathcal{X}_T(\mathbf{u_{1:T-1}},\boldsymbol{\xi}_T)} \Big\{ v_T^\top \mathbf{u_T} + w_T^\top \mathbf{y_T} \Big\} \Big] \dots \Big\} \Big] \Big\}, \end{aligned} \tag{5.1}$$

where $\mathbf{u}_t$ is the state variable and $\mathbf{y}_t$ is a local variable appearing only at stage $t \in T$, $v_t^\top$ and $w_t^\top$ are the corresponding cost at time $t \in 1, \dots, T$, and $\mathcal{X}_t$ is the feasible set. The stochastic data process $(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_T)$ is modeled where $\boldsymbol{\xi}_1$ is deterministic and $\boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_T$ will be revealed gradually in time.

We can now write down the dynamic programming (DP) reformulation of (5.1). The optimal value function at stage t, $V_t(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t)$, is the optimal expected objective value given state $(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t)$, and assuming that optimal action will be taken at each stage t.

$$V_t(\mathbf{u_{t-1}}, \boldsymbol{\xi}_t) = \min_{\mathbf{u}_t, \mathbf{y}_t} \{ v_t^\top \mathbf{u}_t + w_t^\top \mathbf{y}_t + \mathcal{V}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t) : B_t \mathbf{u}_{t-1} + A_t \mathbf{u}_t + C_t \mathbf{y}_t = b_t \}, \quad (5.2)$$

for $t = 1, \dots, T$ where $\mathcal{V}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t)$ is the expected value cost-to-go function,

$$\mathcal{V}_{t+1}(\mathbf{u}_t) := \mathbb{E}[V_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_{t+1}) | \boldsymbol{\xi}_t]. \quad (5.3)$$

We assume $\boldsymbol{\xi}_t$ to be Markovian, i.e. the distribution of $\boldsymbol{\xi}_{t+1}$ only depends on $\boldsymbol{\xi}_t$ rather than the whole history of the data process, with $\mathcal{V}_T \equiv 0$. Finally, let us define the optimal policy as

$$\pi^*(\mathbf{u}_{t-1}, \boldsymbol{\xi}_t) = \operatorname{argmin}_{\mathbf{u}_t, \mathbf{y}_t} \{ v_t^\top \mathbf{u}_t + w_t^\top \mathbf{y}_t + \mathcal{V}_{t+1}(\mathbf{u}_t, \boldsymbol{\xi}_t) : B_t \mathbf{u}_{t-1} + A_t \mathbf{u}_t + C_t \mathbf{y}_t = b_t \} \quad (5.4)$$

for $t = 1, \dots, T$ in set $\mathbf{\Pi}$ as the *policy* which specifies the decision to make for all possible states regardless of which state at stage t.

5.2 Multistage Stochastic FLP under Facility Disruption (MSFLPD) Formulation

Following the formulations above, we propose the multistage stochastic facility location model under facility disruption uncertainty as a multi-stage, discrete-time, stochastic-dynamic optimization problem, as follows

$$\min \quad \sum_{t=1}^{T}(\sum_{i=1}^{F} f_{it}y_{it}+\sum_{i=1}^{F}\sum_{j=1}^{B} d_{ij}x_{ijt}) \tag{5.5}$$

$$\text{s.t.} \quad u_{it} = u_{it-1}(1-\xi_{it}) + y_{it}, \quad \forall i = 1,\dots,F, t = 1,\dots,T \tag{5.6}$$

$$\sum_{j=1}^{B} x_{ijt} \leq C_i u_{it}, \quad \forall i = 1,\dots,F, t = 1,\dots,T \tag{5.7}$$

$$\sum_{i=1}^{F} x_{ijt} = D_{jt}, \quad \forall j = 1,\dots,B, t = 1,\dots,T \tag{5.8}$$

$$u_{it}, y_{it} \in \{0,1\}, \quad \forall i = 1,\dots,F, j = 1,\dots,B, t = 1,\dots,T \tag{5.9}$$

$$x_{ijt} \in \mathbb{N}, \quad \forall i = 1,\dots,F, j = 1,\dots,B, t = 1,\dots,T. \tag{5.10}$$

We define binary variable $y_{it} \in \{0,1\}$ to denote facility opening decision and integer variable x_{ijt} to denote service level as the local variables. We also define binary variable $u_{it} \in \{0,1\}$ as our state variable to track the state of facility $i \in F$ at stage $t \in T$. We denote $u_{it} = 1$ if facility $i \in \{1,...,F\}$ is open at stage $t \in \{1,...,T\}$, and $u_{it} = 0$ otherwise. In addition, we observe a realization of the random variable $\xi_{it} \in \{0,1\}$ such that $\xi_{it} = 1$ if the facility i at stage t is disrupted and $\xi_{it} = 0$ otherwise.

The objective (5.5) minimizes of the total cost which includes the fixed cost of opening facility $i \in \{1,...,F\}$ at stage $t \in \{1,...,T\}$ and the transportation cost d_{ij} from demand point $j \in \{1,...,B\}$ to facility $i \in \{1,...,F\}$. The local variable decisions and transition of the state variable are governed by a series of constraints. Constraints (5.6) keep track of the state of facility $i \in \{1,...,F\}$ as affected by random variable ξ_{it} across each stage. Constraints (5.7) impose assigning of capacity C_i if and only if facility $i \in \{1,...,F\}$ is open. Constraints (5.8) impose that the total demand D_{jt} of customer j must be fully satisfied. Finally, constraints (5.9) and (5.10) show the decision variable domains.

5.3 Solution Methods

We present two solution techniques to solve MSFLPD - stochastic dual dynamic integer programming (SDDiP) and shadow price approximation (SPA).

5.3.1 Stochastic Dual Dynamic Integer Programming (SDDiP)

SDDiP is an extension of the celebrated stochastic dual dynamic programming (SDDP) method to solve MSIP problems (Zou et al. 2019). Unlike SDDP, SDDiP uses Lagrangian relaxation to derive tight cuts of the cost-to-go functions of stochastic mixed-integer problems. SDDiP guarantees to find the exact optimal solution of any MSIP if the time-coupling decision variables that define the state of the dynamic program are binary variables. This bodes well with MSFLPD's binary state variables u_{it} - state of facility $i \in \{1, ..., F\}$ at stage $t \in \{1, ..., T\}$.

Each iteration of SDDiP begins with sampling a subset of scenarios of the stochastic process. Then, SDDiP proceeds by undertaking two key steps - forward simulation and backward pass.

In the forward simulation, the algorithm draws a sample of random realization from the stochastic process, $(\boldsymbol{\xi}_2^\omega, ..., \boldsymbol{\xi}_T^\omega)$, for $\omega \in \Omega$, and then solves the subproblems at each stage t of the dynamic program using the latest cutting plane approximation of the cost-to-go function. Particularly, in every iteration k, the subproblem at stage t is of the form:

$$\begin{aligned} \bar{V}_t^k(\mathbf{u_{t-1}}^k, \boldsymbol{\xi}_t^\omega) := \min_{u_{it}, x_{ijt}, y_{it}} \quad & \sum_{i=1}^{F} (f_i y_{it} + \sum_{j=1}^{B} x_{ijt} d_{ij}) + \bar{\mathcal{V}}_{t+1}^k(\mathbf{u_t}) \\ \text{s.t.} \quad & u_{it} = u_{i,t-1}(1 - \xi_{it}) + y_{it}, \quad \forall\ i = 1, \ldots, F, t = 1, \ldots, T, \\ & \text{Eqs.} \qquad (5.7) - (5.10), \end{aligned} \tag{5.11}$$

for $t = 1, \ldots, T$, where $\boldsymbol{\xi}_t^\omega$ is the ω uncertainty realization at stage t. Each forward simulation, generates a sequence of sample decisions, $((u_{it}^k, x_{ijt}^k, y_{it}^k))_{t=1}^T$ that are made based on the realized uncertainties of the sampled scenario $(\boldsymbol{\xi}_2^\omega, ..., \boldsymbol{\xi}_T^\omega)$.

The function $\bar{\mathcal{V}}_{t+1}^k(\mathbf{u_t})$ is defined as a set of cutting planes (cuts) whose minimum approximates the true expected cost-to-go function $\mathcal{V}_{t+1}^k(\mathbf{u_t})$ from below. The function can be expressed as a linear program which is given by

$$\begin{aligned} \bar{\mathcal{V}}_{t+1}^k(\mathbf{u_t}) := \quad \min \quad & \theta_t \\ \text{s.t.} \quad & \theta_t \geq L_t \\ & \theta_t \geq N_{t+1}^{-1} \sum_{j=1}^{N_{t+1}} (\alpha_{t+1}^{lj} + (\beta_{t+1}^{lj})^\top \mathbf{u_t}) \qquad \forall l \leq k-1. \end{aligned} \tag{5.12}$$

The backward pass begins from the final stage T. Given the solution $\mathbf{u_{T-1}^k}$ from iteration k and an uncertainty realization from $\{\boldsymbol{\xi}_T^j, 1 \leq j \leq N_T\}$, let $P_T^{kj}(\mathbf{u_{t-1}^k}, \boldsymbol{\xi}_t^\omega, \bar{\mathcal{V}}_t^k)$ be a relaxation of the forward problem $\bar{V}_T^k(\mathbf{u_{T-1}^k}, \boldsymbol{\xi}_T^j, \bar{\mathcal{V}}_T^{k+1})$. Solving $P_T^{kj}(\mathbf{u_{t-1}^k}, \boldsymbol{\xi}_t^\omega, \bar{\mathcal{V}}_t^k)$ for each j produces a cut defined by $\theta(\alpha_T^{lj}, \beta_T^{lj})$ which is valid for the value function $V_T(\mathbf{u}_{T-1}, \boldsymbol{\xi}_T^j)$. Then, the cuts $\theta(\alpha_T^{lj}, \beta_T^{lj})$ are aggregated obtaining (5.12) which is valid for expected cost-to-go function $\mathcal{V}_{T-1}^k(\mathbf{u}_{T-1})$. Furthermore, the lower approximation of the expected cost-to-go function is updated from $\bar{\mathcal{V}}_{T-1}^k(\mathbf{u}_t)$ to $\bar{\mathcal{V}}_{T-1}^{k+1}(\mathbf{u}_t)$. Backward pass then proceeds to stage $T-1$. As the first stage computation is completed, and having solved a lower approximation of the original problem, the optimal solution value of the first stage problem $t = 1$ is a valid lower bound of the original problem.

SDDiP introduces a family of valid cuts, called Lagrangian cuts, which are able to obtain strong duality for mixed integer programs. Like SDDP, SDDiP is also sampling-based algorithm and exhibits favorable scalability on solving large-scale problems. The limiting assumption of SDDiP, as with SDDP, is that the random process has to be stagewise independent. Some techniques to incorporate stagewise dependency are presented in Löhndorf & Shapiro (2019).

5.3.2 Shadow Price Approximation (SPA)

SPA is an approximation algorithm proposed in Seranilla & Löhndorf (2023). The main idea of SPA is to train the slope of a linear value function approximation by minimizing an upper bound on the optimal objective value that can be obtained via Monte Carlo simulation. These slopes are similar to the notion of shadow prices of the non-anticipatory constraints that connect successive time periods - hence, the name of the algorithm.

Choosing the slope vector that minimizes an upper bound on the optimal objective value is effectively a nonconvex, stochastic optimization problem that is known to be computationally intractable. Thus, SPA is cast as a policy search strategy that can be supported by any method that is suitable for unconstrained (derivative-free) global optimization.

Instead of developing a stochastic model, SPA only needs access to independent time series, which can also be real data. SPA thereby interacts directly with a simulation model of the resulting policy unlike many other approximate dynamic programming techniques that rely on some form of backwards recursion. SPA can be easily integrated with gradient-based and stochastic search methods which are widely used in machine learning and global optimization.

The SPA algorithm, like SDDiP, undertakes two primary steps - forward simulation and shadow price updating. The algorithm proceeds as follows:

Step 1. Initialize iteration $n = 0$ and shadow prices $\lambda_t^{u,0} = 0$, for every $t = 1, \dots, T$.

Step 2. At each iteration n, we randomly select a scenario path $(\hat{\xi}_1^n, \dots, \hat{\xi}_T^n)$.

[*Forward simulation*]

Step 2.1. Solve the dynamic programming recursion of the form

$$
\begin{aligned}
\bar{V}_t^n(u_{it-1}^n, \xi_{it}^n; \lambda_{it}^{u,n-1}) = \min_{u_{it}^n, x_{ijt}^n, y_{it}^n} \quad & \sum_{i=1}^{F}(f_i y_{it} + \sum_{j=1}^{B} d_{ij} x_{ijt}) + \underbrace{\sum_{i=1}^{F} \lambda_{it}^{u,n-1} u_i}_{\text{Shadow Price}} \\
\text{s.t.} \quad & u_{it} = u_{i,t-1}(1 - \xi_{it}) + y_{it}, \quad \forall\ i = 1, \dots, F, t = 1, \dots, T, \\
& \text{Eqs.} \quad (5.7) - (5.10),
\end{aligned}
\tag{5.13}
$$

as a sequence of mixed-integer problems that encodes information on the cost-to-go only via the slope vector of the linear value function approximation (the shadow prices).

Step 2.2. Obtain realized upper bound $\hat{\bar{z}}^n$ as follows,

$$\hat{\bar{z}}^n(\lambda_{it}^{u,n-1}) = \sum_{t\in T}\left[\sum_{i=1}^{F}(f_{it}y_{it}(\bar{\pi}^n(u_{it-1}^n, \hat{\xi}_{it}^n; \lambda_{it}^{u,n-1})) + \sum_{i=1}^{F}\sum_{j=1}^{B} d_{ij}x_{ijt}(\bar{\pi}^n(u_{it-1}^n, \hat{\xi}_{it}^n; \lambda_{it}^{u,n-1}))\right], \tag{5.14}$$

a realization of the total cost for given slope vectors $\lambda_{it}^{u,n-1}$.

[*Shadow price updating*]

Step 3. With the obtained realized upper bound $\hat{\bar{z}}^n$ and parameters $\lambda_{it}^{u,n-1}$, the shadow price is updated using a generic update function $U^n(\cdot,\cdot)$, supported by a chosen global optimization method, that returns a new set of trial shadow prices,

$$\lambda_{it}^{u,n} \leftarrow U^n(\hat{\bar{z}}^n, \lambda_{it}^{u,n-1}). \tag{5.15}$$

Various choices of global optimization methods include, but are not limited to, stochastic gradient descent Bottou & Bousquet (2011), covariance matrix adaptation - evolutionary strategy (CMA-ES) Hansen et al. (2019), and Bayesian optimization Nogueira (2020).

Step 4. At the end of iteration N, SPA returns a slope vector λ_{it}^{u*} that approximately minimizes $\bar{z}$

$$\lambda_{it}^{u*} \sim \text{argmin}_{\lambda_{it}^{uN}} \hat{\bar{z}}(\lambda_{it}^{uN}). \tag{5.16}$$

Step 5. To obtain an approximate upper bound of the optimal objective value, we simulate the optimal policy $\bar{z}(\lambda_{it}^{u*})$

$$\bar{z}(\lambda_{it}^{u*}) = S^{-1}\sum_{s=1}^{S}\hat{\bar{z}}^n(\lambda_{it}^{u*}). \tag{5.17}$$

Whether or not the chosen global optimization methods is able to find a good linear approximation depends on the initial choice of parameters, the search region, as well as the number of iterations, and is likely to be problem-specific. In the worst case, no improvement over the (initial) greedy policy, with $\lambda_t^{u,0} = 0$, for every $t = 1, \ldots, T$, is possible. In the best case, the method selects the best linear approximation which is optimal if the true cost-to-go function is linear in the region of the state space that can be reached by the optimal policy.

As we will see below, a linear approximation can produce near-optimal results for the chosen problem. In which way this result extends to other problems remains subject of future work.

5.4 Numerical Results

We present the results of employing both SPA and SDDiP to solve Problem (5.5)-(5.10). The case instance chosen, with parameters shown in Table 5.1, is a multi-stage adaptation of the capacitated facility location problem, drawing its test instances from Beasley (1990), a renowned repository of test datasets spanning various facility location and other OR problems. To include a greater depth of robustness in our model's testing and the deployed solution methodologies, different levels of disruptions were synthesized. These disruptions are denoted by $S = \{Low, \; Medium, \; High\}$ levels. Additionally, in a bid to intricately gauge the algorithms' performance, we designated three incremental time limit criteria, represented as $\mathcal{T} = \{300\text{s}, 900\text{s}, 1800\text{s}\}$, to serve as benchmarks.

Table 5.1: Parameters used for the numerical investigation (taken from Beasley (1990).

Parameters	**Value**
T - planning horizons	36, 52, 100
F - set of candidate facilities	16
B - set of customers	50
C_{it} - capacity	50
D_{jt} - demand	taken from Beasley (1990)
f_{it} - fixed cost of opening a facility	taken from Beasley (1990)
d_{ijt} - transportation cost	taken from Beasley (1990)
ξ_{it} - facility failure (uncertainty)	Randomly generated
	[$High$-60%, $Medium$-40%, Low-20%]

The findings from our expansive numerical exploration of MSFLPD via SDDiP and SPA are consolidated in Table 5.2. Distinct columns are reserved for various metrics,

Table 5.2: Numerical investigation of MSFLPD using SDDiP and SPA.

Time Limit Criterion ($\times 10^3$)							
Instance CAP44-A							
Stages (T)	Runtime	S	SDDiP-LB	SDDiP-UB	SPA-UB	GAP-1	GAP-2
36	*300s*	High	138994.0	139445.4	139395.9	0.32%	0.28%
		Medium	133262.4	134092.1	133518.0	0.62%	0.19%
		Low	127614.4	128619.3	127614.8	0.78%	0.00%
	900s	High	138994.0	139170.1	139083.6	0.13%	0.06%
		Medium	133262.3	134043.7	133518.0	0.58%	0.19%
		Low	127614.4	128595.6	127614.8	0.77%	0.00%
	1800s	High	138994.0	139088.0	139020.6	0.07%	0.02%
		Medium	133262.4	133763.6	133518.0	0.38%	0.19%
		Low	127614.4	128421.1	127614.8	0.63%	0.00%
52	*300s*	High	109354.0	110594.0	109857.0	1.13%	0.45%
		Medium	100611.2	102174.4	101551.2	1.54%	0.93%
		Low	92297.0	93850.3	92925.7	1.67%	0.68%
	900s	High	109354.0	110254.3	109857.0	0.82%	0.45%
		Medium	100611.2	102422.7	101482.7	1.78%	0.86%
		Low	92297.0	93978.7	92925.7	1.81%	0.68%
	1800s	High	109354.0	110340.3	109857.0	0.90%	0.46%
		Medium	100611.2	102175.9	101449.5	1.54%	0.83%
		Low	92297.0	93955.6	92925.7	1.78%	0.68%
100	*300s*	High	89966.6	94229.2	93537.2	4.63%	3.89%
		Medium	73972.8	78148.8	77173.9	5.49%	4.24%
		Low	57089.1	63360.7	60764.9	10.41%	6.40%
	900s	High	89966.6	93848.1	93391.8	4.22%	3.76%
		Medium	73972.8	78306.5	76756.4	5.69%	3.69%
		Low	57089.1	62813.4	60764.9	9.55%	6.24%
	1800s	High	89966.6	93678.5	93411.5	4.04%	3.74%
		Medium	73972.8	78308.7	76887.7	5.69%	3.86%
		Low	57089.1	62813.3	60722.8	9.55%	6.17%

including SDDiP lower bounds (SDDiP-LB), SDDiP upper bound (SDDiP-UB), and SPA upper bound (SPA-UB). The percentage gaps delineated represent the %-difference between SDDiP lower bound and SDDiP upper bound (GAP-1), and the %-difference between SDDiP lower bound and SPA upper bound (GAP-2). From our analysis, SPA emerges as a more effective solution, consistently identifying superior policies for

MSFLPD across all case instances. This observation persists even when strengthened *Benders* cuts are employed for SDDiP - these are valid, finite cuts wherein a generic Benders cut is augmented by solving a specific mixed integer program subproblem, where the solution mirrors a basic optimal LP dual solution. Such results underscore the inherent complexity and challenge posed by MSFLPD, especially when considering risk failure.

5.5 Conclusion

In this chapter, we consider a *multi-stage* variant of the facility location problem under facility disruption (MSFLPD). We model the problem as a multi-stage stochastic mixed-integer program, and discuss two methods to solve the problem. The first method is stochastic dual dynamic integer programming (SDDiP), and the second method is the shadow price approximation (SPA) algorithm. While SDDiP is an exact solution approach, it is also difficult to implement, whereas SPA is an approximation method that is relatively easy to apply. We conduct extensive numerical experiments to compare the performance of the two methods in solving the problem. We find that, across all problem instances, SPA emerges as the more effective solution, consistently identifying superior policies for MSFLPD than SDDiP. These results underscore the inherent complexity and challenge posed by MSFLPD, especially when considering facility disruption risk.

Chapter 6

Conclusion

In this dissertation, we delve into the applications of the *state-of-the-art* algorithm, Stochastic Dual Dynamic Programming (SDDP), for solving MSPs. While MSPs offer an effective means to represent many real-world challenges, their intrinsic complexity often makes optimization formidable. Within extant literature, SDDP stands out as a potent methodology adept at addressing MSPs. This thesis highlights the versatility of SDDP in managing sequential decision-making across diverse domains, emphasizing its capability in navigating uncertainty.

In Chapter 2, conduct a comprehensive survey of the diverse applications of SDDP in the literature. This includes an analysis of statistics on the prevalence of SDDP usage in various domains. Moreover, a substantial focus is directed towards the most common application of SDDP in the energy sector, particularly in hydro-thermal power production scheduling. This paper outlines compelling arguments behind the prominence of this specific application, i.e., at least 75% of research work adopting the SDDP methodology exploit this problem.

In Chapter 3, we introduce two valuable contributions: MSPLib, an open-source library of problems and MSPFormat, a standardized data format designed for benchmarking SDDP. MSPLib aims to facilitate the evaluation of computational performance among different SDDP implementations. It offers a wide array of instances, from real-world problems to synthetic variations with varying complexities. By incorporating MSPFormat into the library, a unified and consistent representation of

MSPs is provided, further enhancing their usability and transferability.

In Chapter 4, we introduce a novel problem class called the multistage stochastic facility location problem under facility disruption uncertainty (MSFLPD). This new class extends the classical stochastic *capacitated* facility location problem to handle uncertainty arising from facility disruptions. We then present and compare two solution algorithms tailored for addressing this problem: stochastic dual dynamic integer programming (SDDiP) and shadow price approximation (SPA).

Finally, in Chapter 5, we showcase a specific instance of MSFLPD applied to the optimal location of COVID-19 vaccine facilities under the threat of natural disasters. We introduce a new algorithm, named *shadow price approximation* (SPA), which aims at approximating the shadow price of opening flood-prone vaccine facilities by tuning the parameters of a linear value function approximation which is present in the objective function of base optimization model. SPA yields approximately 30-40% lower cost than a baseline approach that does not consider the risk of flooding, across different rainfall-to-flood conservativity cases. Finally, we provide a detailed account of this model's application in two cities of a developing country.

There are several promising avenues for future research. One notable direction is the expansion of the MSPLib to accommodate problems stemming from the integer enhancement of SDDP, known as Stochastic Dual Dynamic Integer Programming (SDDiP). The realm of Multistage Stochastic Integer Programming (MSIP) presents a class of intricate and intriguing problems. Leveraging our newly proposed solution, the Shadow Price Approximation (SPA), in conjunction with SDDiP can yield substantial benefits when addressing these problems. Incorporating MSIPs into the MSPLib can foster further advancements in the SDDiP and SPA algorithms, enabling the inclusion of more complex real-world problems. Additionally, considering other uncertainties in the MSFLPD, such as demand, capacity, and costs, would present another exciting avenue for future endeavors.

Bibliography

Aaslid, P., Korpås, M., Belsnes, M. M., & Fosso, O. B. (2022a). Stochastic operation of energy constrained microgrids considering battery degradation. *Electric Power Systems Research*, *212*, 108462.

Aaslid, P., Korpås, M., Belsnes, M. M., & Fosso, O. B. (2022b). Stochastic optimization of microgrid operation with renewable generation and energy storages. *IEEE Transactions on Sustainable Energy*, *13*, 1481–1491.

Abgottspon, H., Njálsson, K., Bucher, M. A., & Andersson, G. (2014). Risk-averse medium-term hydro optimization considering provision of spinning reserves. In *2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)* (pp. 1–6). IEEE.

van Ackooij, W., & Warin, X. (2020). On conditional cuts for stochastic dual dynamic programming. *EURO Journal on Computational Optimization*, *8*, 173–199.

Ahmadi-Javid, A., Seyedi, P., & Syam, S. S. (2017). A survey of healthcare facility location. *Computers & Operations Research*, *79*, 223–263.

Ahmed, S. (2002). Semiconductor tool planning via multi-stage stochastic programming. In *Proceedings of the International Conference on Modeling and Analysis in Semiconductor Manufacturing* (p. 157). volume 153.

Ahmed, S., Cabral, F. G., & Freitas Paulo da Costa, B. (2022). Stochastic lipschitz dynamic programming. *Mathematical Programming*, *191*, 755–793.

Albareda-Sambola, M., Alonso-Ayuso, A., Escudero, L. F., Fernández, E., & Pizarro, C. (2013). Fix-and-relax-coordination for a multi-period location–allocation problem under uncertainty. *Computers & operations research*, *40*, 2878–2892.

Alumur, S. A., Campbell, J. F., Contreras, I., Kara, B. Y., Marianov, V., & O'Kelly,

M. E. (2021). Perspectives on modeling hub location problems. *European Journal of Operational Research*, *291*, 1–17.

Alvarez, M., Rönnberg, S. K., Bermudez, J., Zhong, J., & Bollen, M. H. (2017). A generic storage model based on a future cost piecewise-linear approximation. *IEEE Transactions on Smart Grid*, *10*, 878–888.

Alvarez, M., Rönnberg, S. K., Bermúdez, J., Zhong, J., & Bollen, M. H. (2018). Reservoir-type hydropower equivalent model based on a future cost piecewise approximation. *Electric Power Systems Research*, *155*, 184–195.

Angün, E. (2015). Stochastic dual dynamic programming solution of a short-term disaster management problem. *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*, (pp. 225–250).

Arrow, K. J., Harris, T., & Marschak, J. (1951). Optimal inventory policy. *Econometrica: Journal of the Econometric Society*, (pp. 250–272).

Asamov, T., & Powell, W. B. (2018). Regularized decomposition of high-dimensional multistage stochastic programs with markov uncertainty. *SIAM Journal on Optimization*, *28*, 575–595.

Ávila, D., Papavasiliou, A., & Löhndorf, N. (2022). Parallel and distributed computing for stochastic dual dynamic programming. *Computational Management Science*, *19*, 199–226.

Bakker, S. J., Kleiven, A., Fleten, S.-E., & Tomasgard, A. (2021). Mature offshore oil field development: Solving a real options problem using stochastic dual dynamic integer programming. *Computers & Operations Research*, *136*, 105480.

Bally, V., & Pagès, G. (2003). A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli*, *9*, 1003–1049.

Bandarra, M., & Guigues, V. (2021). Single cut and multicut stochastic dual dynamic programming with cut selection for multistage stochastic linear programs: convergence proof and numerical experiments. *Computational Management Science*, *18*, 125–148.

Bao, H., Zhou, Z., Kotsalis, G., Lan, G., & Tong, Z. (2019). Lignin valorization process control under feedstock uncertainty through a dynamic stochastic programming approach. *Reaction Chemistry & Engineering*, *4*, 1740–1747.

Beasley, J. E. (1990). Or-library: distributing test problems by electronic mail. *Journal of the operational research society*, *41*, 1069–1072.

Beltrán, F., Finardi, E. C., Fredo, G. M., & de Oliveira, W. (2020). Improving the performance of the stochastic dual dynamic programming algorithm using chebyshev centers. *Optimization and engineering*, (pp. 1–22).

Beltrán, F., Finardi, E. C., & de Oliveira, W. (2021). Two-stage and multi-stage decompositions for the medium-term hydrothermal scheduling problem: A computational comparison of solution techniques. *International Journal of Electrical Power & Energy Systems*, *127*, 106659.

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numer. Math*, *4*, 238–252.

Bertsimas, D., Digalakis Jr, V., Jacquillat, A., Li, M. L., & Previero, A. (2022). Where to locate covid-19 mass vaccination facilities? *Naval Research Logistics (NRL)*, *69*, 179–200.

Bhattacharya, A., Kharoufeh, J. P., & Zeng, B. (2016). Managing energy storage in microgrids: A multistage stochastic programming approach. *IEEE Transactions on Smart Grid*, *9*, 483–496.

Birge, J. R. (1985). Decomposition and partitioning methods for multistage stochastic linear programs. *Operations research*, *33*, 989–1007.

Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Bonnans, J. F., Cen, Z., & Christel, T. (2012). Energy contracts management by stochastic programming techniques. *Annals of Operations Research*, *200*, 199–222.

Borges, P. (2022). Cut-sharing across trees and efficient sequential sampling for sddp with uncertainty in the rhs. *Computational Optimization and Applications*, (pp. 1–31).

Borges, P., Sagastizábal, C., Solodov, M., Liberti, L., & D'ambrosio, C. (2022). Profit sharing mechanisms in multi-owned cascaded hydrosystems. *Optimization and Engineering*, (pp. 1–39).

Bottou, L., & Bousquet, O. (2011). 13 the tradeoffs of large-scale learning. *Optimization for machine learning*, (p. 351).

Brandi, R. B. S., Marcato, A. L. M., Dias, B. H., Ramos, T. P., & da Silva Junior, I. C. (2017). A convergence criterion for stochastic dual dynamic programming: application to the long-term operation planning problem. *IEEE Transactions on Power Systems*, *33*, 3678–3690.

Brandi, R. B. S., Ramos, T. P., Dias, B. H., Marcato, A. L. M., & da Silva Junior, I. C. (2015). Improving stochastic dynamic programming on hydrothermal systems through an iterative process. *Electric Power Systems Research*, *123*, 147–153.

Brigatto, A., Street, A., & Valladao, D. M. (2017). Assessing the cost of time-inconsistent operation policies in hydrothermal power systems. *IEEE Transactions on Power Systems*, *32*, 4541–4550.

Browne, S., Dongarra, J., Grosse, E., & Rowan, T. (1995). The netlib mathematical software repository. *D-lib Magazine*, *1*.

Bruno, S., Ahmed, S., Shapiro, A., & Street, A. (2016). Risk neutral and risk averse approaches to multistage renewable investment planning under uncertainty. *European Journal of Operational Research*, *250*, 979–989.

Burdett, R. L., & Kozan, E. (2015). Techniques to effectively buffer schedules in the face of uncertainties. *Computers & Industrial Engineering*, *87*, 16–29.

Bussieck, M. R., Drud, A. S., & Meeraus, A. (2003). Minlplib—a collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing*, *15*, 114–119.

Calili, R. F., Souza, R. C., Galli, A., Armstrong, M., & Marcato, A. L. M. (2014). Estimating the cost savings and avoided co2 emissions in brazil by implementing energy efficient policies. *Energy Policy*, *67*, 4–15.

Carpentier, P., Chancelier, J.-P., De Lara, M., & Pacaud, F. (2020). Mixed spatial and temporal decompositions for large-scale multistage stochastic optimization problems. *Journal of Optimization Theory and Applications*, *186*, 985–1005.

de Castro, C. M., Marcato, A. L., Souza, R. C., Junior, I. C. S., Oliveira, F. L. C., & Pulinho, T. (2015). The generation of synthetic inflows via bootstrap to increase the energy efficiency of long-term hydrothermal dispatches. *Electric Power Systems Research*, *124*, 33–46.

CDO, A. (2022). The recent floods in cagayan de oro has precedence that goes way back to the 1970's. URL: https://aboutcagayandeoro.com/recent-floods-cagayan-de-oro/.

Cerisola, S., Latorre, J. M., & Ramos, A. (2012). Stochastic dual dynamic programming applied to nonconvex hydrothermal models. *European Journal of Operational Research*, *218*, 687–697.

Chabar, R., Pereira, M., Granville, S., Barroso, L., & Iliadis, N. (2006). Optimization of fuel contracts management and maintenance scheduling for thermal plants under price uncertainty. In *2006 IEEE PES Power Systems Conference and Exposition* (pp. 923–930). IEEE.

Chen, Z.-L., Li, S., & Tirupati, D. (2002). A scenario-based stochastic programming approach for technology and capacity planning. *Computers & Operations Research*, *29*, 781–806.

Cheng, C., Adulyasak, Y., & Rousseau, L.-M. (2021). Robust facility location under disruptions. *INFORMS Journal on Optimization*, *3*, 298–314.

Correia, I., & Saldanha-da Gama, F. (2019). Facility location under uncertainty. *Location science*, (pp. 185–213).

da Costa, L. C., Thomé, F. S., Garcia, J. D., & Pereira, M. V. (2020). Reliability-constrained power system expansion planning: A stochastic risk-averse optimization approach. *IEEE Transactions on Power Systems*, *36*, 97–106.

Côté, P., & Arsenault, R. (2019). Efficient implementation of sampling stochastic dynamic programming algorithm for multireservoir management in the hydropower sector. *Journal of Water Resources Planning and Management*, *145*, 05019005.

Cui, T., Ouyang, Y., & Shen, Z.-J. M. (2010). Reliable facility location design under the risk of disruptions. *Operations research*, *58*, 998–1011.

Cyber-Infrastructure, C.-I. (2016). Cmu-ibm cyber-infrastructure for minlp collaborative site. URL: http://www.minlp.org/.

Dantzig, G. B., & Glynn, P. W. (1990). Parallel processors for planning under uncertainty. *Annals of Operations research*, *22*, 1–21.

Dantzig, G. B., & Infanger, G. (1993). Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research*, *45*, 59–76.

De Matos, V. L., Morton, D. P., & Finardi, E. C. (2017). Assessing policy quality in a multistage stochastic program for long-term hydrothermal scheduling. *Annals of Operations Research*, *253*, 713–731.

De Matos, V. L., Philpott, A. B., & Finardi, E. C. (2015). Improving the performance of stochastic dual dynamic programming. *Journal of Computational and Applied Mathematics*, *290*, 196–208.

Ding, L. (2020). Multistage stochastic programming. URL: https://repository.gatech.edu/entities/publication/c37bb558-0694-4760-bac2-d960289cb754.

Ding, L., Ahmed, S., & Shapiro, A. (2019). A python package for multi-stage stochastic programming. *Optimization online*, (pp. 1–42).

Ding, T., Qu, M., Huang, C., Wang, Z., Du, P., & Shahidehpour, M. (2020). Multi-period active distribution network planning using multi-stage stochastic programming and nested decomposition by sddip. *IEEE Transactions on Power Systems*, *36*, 2281–2292.

Ding, T., Zhang, X., Lu, R., Qu, M., Shahidehpour, M., He, Y., & Chen, T. (2021). Multi-stage distributionally robust stochastic dual dynamic programming to multi-period economic dispatch with virtual energy storage. *IEEE Transactions on Sustainable Energy*, *13*, 146–158.

Diniz, A. L., Maceira, M. E. P., Vasconcellos, C. L. V., & Penna, D. D. J. (2020). A combined sddp/benders decomposition approach with a risk-averse surface concept for reservoir operation in long term power generation planning. *Annals of Operations Research*, *292*, 649–681.

DOH-10 (2022). Region 10 covid-19 situational report. URL: https://doh.gov.ph/COVID-19-policies.

Donohue, C. J., & Birge, J. R. (2006). The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithmic Operations Research*, *1*, 20–30.

Dornellas, C. R., Leite da Silva, A. M., Costa, J. G., Machado Jr, Z. S., Marcato, A. L., & Mello, J. C. (2022). Toward a new nodal pricing policy of the brazilian transmission system considering multiple hydrological scenarios. *International Transactions on Electrical Energy Systems*, .

DOST-PAGASA (2019). Rainfall advisories, classification, and measurement. URL: https://www.pagasa.dost.gov.ph.

Downward, A., Dowson, O., & Baucke, R. (2020). Stochastic dual dynamic programming with stagewise-dependent objective uncertainty. *Operations Research Letters*, *48*, 33–39.

Dowson, O. (2020). The policy graph decomposition of multistage stochastic programming problems. *Networks*, *76*, 3–23.

Dowson, O., & Garcia, J. (2020). Stochoptformat: a data structure for multistage stochastic programming. URL: https://odow.github.io/StochOptFormat/.

Dowson, O., & Kapelevich, L. (2021). Sddp. jl: a julia package for stochastic dual dynamic programming. *INFORMS Journal on Computing*, *33*, 27–33.

Dowson, O., Morton, D., & Downward, A. (2022a). Bi-objective multistage stochastic linear programming. *Mathematical Programming*, (pp. 1–27).

Dowson, O., Morton, D. P., & Pagnoncelli, B. K. (2020). Partially observable multistage stochastic programming. *Operations Research Letters*, *48*, 505–512.

Dowson, O., Morton, D. P., & Pagnoncelli, B. K. (2022b). Incorporating convex risk measures into multistage stochastic programming algorithms. *Annals of Operations Research*, (pp. 1–25).

Dowson, O., Philpott, A., Mason, A., & Downward, A. (2019). A multi-stage stochastic optimization model of a pastoral dairy farm. *European Journal of Operational Research*, *274*, 1077–1089.

Duijzer, L. E., Van Jaarsveld, W., & Dekker, R. (2018). Literature review: The vaccine supply chain. *European Journal of Operational Research*, *268*, 174–192.

Dupačová, J., & Kozmík, V. (2015). Structure of risk-averse multistage stochastic programs. *OR spectrum*, *37*, 559–582.

Dupačová, J., & Kozmík, V. (2017). Sddp for multistage stochastic programs: preprocessing via scenario reduction. *Computational Management Science*, *14*, 67–80.

Duque, D., & Morton, D. P. (2020). Distributionally robust stochastic dual dynamic programming. *SIAM Journal on Optimization*, *30*, 2841–2865.

Escudero, L. F., Garín, M. A., & Unzueta, A. (2017). Scenario cluster lagrangean decomposition for risk averse in multistage stochastic optimization. *Computers & Operations Research*, *85*, 154–171.

Espanmanesh, V., & Tilmant, A. (2022). Optimizing the management of multireservoir systems under shifting flow regimes. *Water Resources Research*, *58*, e2021WR030582.

de Faria, F. A., & Jaramillo, P. (2017). The future of power generation in brazil: An analysis of alternatives to amazonian hydropower development. *Energy for Sustainable Development*, *41*, 24–35.

Fatouros, P., Konstantelos, I., Papadaskalopoulos, D., & Strbac, G. (2017). Stochastic dual dynamic programming for operation of der aggregators under multi-dimensional uncertainty. *IEEE Transactions on Sustainable Energy*, *10*, 459–469.

Ferreira, P. G. C., Souza, R. C., & Marcato, A. L. M. (2015). The par (p) interconfigurations model used by the brazilian electric sector. *International Journal of Electrical Power & Energy Systems*, *73*, 45–55.

Flach, B. C., Barroso, L., & Pereira, M. (2010). Long-term optimal allocation of hydro generation for a price-maker company in a competitive market: latest developments and a stochastic dual dynamic programming approach. *IET generation, transmission & distribution*, *4*, 299–314.

Fredo, G. L. M., Finardi, E. C., Larroyd, P. V., & Picarelli, L. B. (2021). Inflow aggregation and run-of-the-river inflow energy for reducing dimensionality in the long-term generation scheduling problem. *IEEE Access*, *9*, 98542–98560.

Fredo, G. L. M., Finardi, E. C., & de Matos, V. L. (2019). Assessing solution quality and computational performance in the long-term generation scheduling problem considering different hydro production function approaches. *Renewable energy*, *131*, 45–54.

Füllner, C., & Rebennack, S. (2021). Stochastic dual dynamic programming and its variants. *Optimization Online*, .

Gade, D., Hackebeil, G., Ryan, S. M., Watson, J.-P., Wets, R. J.-B., & Woodruff, D. L. (2016). Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming*, *157*, 47–67.

Gangammanavar, H., & Sen, S. (2016). Two-scale stochastic optimization for controlling distributed storage devices. *IEEE Transactions on Smart Grid*, *9*, 2691–2702.

Gassmann, H. I. (2005). The smps format for stochastic linear programs. In *Applications of stochastic programming* (pp. 9–19). SIAM.

Ghadimi, S., Perkins, R. T., & Powell, W. B. (2020). Reinforcement learning via parametric cost function approximation for multistage stochastic programming. *arXiv preprint arXiv:2001.00831*, .

Ghadimi, S., & Powell, W. B. (2022). Stochastic search for a parametric cost function approximation: Energy storage with rolling forecasts. *arXiv preprint arXiv:2204.07317*, .

Girardeau, P., Leclere, V., & Philpott, A. B. (2015). On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, *40*, 130–145.

Gleixner, A., Hendel, G., Gamrath, G., Achterberg, T., Bastubbe, M., Berthold, T., Christophel, P., Jarck, K., Koch, T., Linderoth, J. et al. (2021). Miplib 2017: data-driven compilation of the 6th mixed-integer programming library. *Mathematical Programming Computation*, (pp. 1–48).

Golari, M., Fan, N., & Jin, T. (2017). Multistage stochastic optimization for production-inventory planning with intermittent renewable energy. *Production and Operations Management*, *26*, 409–425.

Goor, Q., Kelman, R., & Tilmant, A. (2011). Optimal multipurpose-multireservoir operation model with variable productivity of hydropower plants. *Journal of Water Resources Planning and Management*, *137*, 258–267.

Gorenstin, B., Campodonico, N., da Costa, J., & Pereira, M. (1992). Stochastic optimization of a hydro-thermal system including network constraints. *IEEE Transactions on Power Systems*, *7*, 791–797.

Gorski, V. (2017). Scenario trees models vs. lattice models in stochastic optimization.

Guan, Z., Shawwash, Z., & Abdalla, A. (2018). Using sddp to develop water-value functions for a multireservoir system with international treaties. *Journal of Water Resources Planning and Management*, *144*, 05017021.

Guigues, V. (2014). Sddp for some interstage dependent risk-averse problems and application to hydro-thermal planning. *Computational Optimization and Applications*, *57*, 167–203.

Guigues, V. (2016). Convergence analysis of sampling-based decomposition methods for risk-averse multistage stochastic convex programs. *SIAM Journal on Optimization*, *26*, 2468–2494.

Guigues, V. (2017). Dual dynamic programing with cut selection: Convergence proof and numerical experiments. *European Journal of Operational Research*, *258*, 47–57.

Guigues, V. (2020). Inexact cuts in stochastic dual dynamic programming. *SIAM Journal on Optimization*, *30*, 407–438.

Guigues, V. (2021). Multistage stochastic programs with a random number of stages: dynamic programming equations, solution methods, and application to portfolio selection. *Optimization Methods and Software*, *36*, 211–236.

Guigues, V., Juditsky, A., & Nemirovski, A. (2021). Constant depth decision rules for multistage optimization under uncertainty. *European Journal of Operational Research*, *295*, 223–232.

Guigues, V., Lejeune, M. A., & Tekaya, W. (2020). Regularized stochastic dual dynamic programming for convex nonlinear optimization problems. *Optimization and Engineering*, *21*, 1133–1165.

Guigues, V., & Monteiro, R. D. (2021). Stochastic dynamic cutting plane for multistage stochastic convex programs. *Journal of Optimization Theory and Applications*, *189*, 513–559.

Guigues, V., & Römisch, W. (2012a). Sampling-based decomposition methods for multistage stochastic programs based on extended polyhedral risk measures. *SIAM Journal on Optimization*, *22*, 286–312.

Guigues, V., & Römisch, W. (2012b). Sddp for multistage stochastic linear programs based on spectral risk measures. *Operations Research Letters*, *40*, 313–318.

Guigues, V., & Sagastizábal, C. (2012). The value of rolling-horizon policies for risk-averse hydro-thermal planning. *European Journal of Operational Research*, *217*, 129–140.

Guigues, V., & Sagastizabal, C. (2013). Risk-averse feasible policies for large-scale multistage stochastic linear programs. *Mathematical Programming*, *138*, 167–198.

Guigues, V., Shapiro, A., & Cheng, Y. (2023). Duality and sensitivity analysis of multistage linear stochastic programs. *European Journal of Operational Research*, *308*, 752–767.

Guo, Z., Wei, W., Chen, L., Wang, Z., Catalão, J. P., & Mei, S. (2020). Optimal energy management of a residential prosumer: A robust data-driven dynamic programming approach. *IEEE Systems Journal*, *16*, 1548–1557.

Haase, K., Knörr, L., Krohn, R., Müller, S., & Wagner, M. (2019). Facility location in the public sector. *Location Science*, (pp. 745–764).

Hafiz, F., Chen, B., Chen, C., Rodrigo de Queiroz, A., & Husain, I. (2019a). Utilising demand response for distribution service restoration to achieve grid resiliency against natural disasters. *IET Generation, Transmission & Distribution*, *13*, 2942–2950.

Hafiz, F., de Queiroz, A. R., & Husain, I. (2019b). Coordinated control of pev and pv-based storages in residential systems under generation and load uncertainties. *IEEE Transactions on Industry Applications*, *55*, 5524–5532.

Halldorsson, B. V., Thorsteinsson, E. S., & Kristjansson, B. (2000). *A modeling interface to non-linear programming solvers an instance: xMPS, the extended MPS format*. Technical Report Technical report, Carnegie Mellon University and Maximal Software.

Hansen, N. (2006). The cma evolution strategy: a comparing review. *Towards a new evolutionary computation*, (pp. 75–102).

Hansen, N., Akimoto, Y., & Baudis, P. (2019). Cma-es/pycma on github. zenodo, doi: 10.5281/zenodo. 2559634.(feb. 2019).

Helseth, A., & Braaten, H. (2015). Efficient parallelization of the stochastic dual dynamic programming algorithm applied to hydropower scheduling. *Energies*, *8*, 14287–14297.

Helseth, A., Fodstad, M., & Mo, B. (2016). Optimal medium-term hydropower scheduling considering energy and reserve capacity markets. *IEEE Transactions on Sustainable Energy*, *7*, 934–942.

Helseth, A., & Mo, B. (2022). Hydropower aggregation by spatial decomposition—an sddp approach. *IEEE Transactions on Sustainable Energy*, *14*, 381–392.

Helseth, A., Mo, B., Hågenvik, H. O., & Schäffer, L. E. (2022). Hydropower scheduling with state-dependent discharge constraints: An sddp approach. *Journal of Water Resources Planning and Management*, *148*, 04022061.

Hernandez, P., Alonso-Ayuso, A., Bravo, F., Escudero, L. F., Guignard, M., Marianov, V., & Weintraub, A. (2012). A branch-and-cluster coordination scheme for selecting prison facility sites under uncertainty. *Computers & operations research*, *39*, 2232–2241.

Hjelmeland, M. N., Helseth, A., & Korpås, M. (2019). Medium-term hydropower scheduling with variable head under inflow, energy and reserve capacity price uncertainty. *Energies*, *12*, 189.

Hjelmeland, M. N., Zou, J., Helseth, A., & Ahmed, S. (2018). Nonconvex medium-term hydropower scheduling by stochastic dual dynamic integer programming. *IEEE Transactions on Sustainable Energy*, *10*, 481–490.

Hou, S., Fan, Y., & Yi, B.-W. (2021). Long-term renewable electricity planning using a multistage stochastic optimization with nested decomposition. *Computers & Industrial Engineering*, *161*, 107636.

Huang, K., & Ahmed, S. (2009). The value of multistage stochastic programming in capacity planning under uncertainty. *Operations Research*, *57*, 893–904.

Hunagund, I. B., Pillai, V. M., & UN, K. (2022). A survey on discrete space and continuous space facility layout problems. *Journal of Facilities Management*, *20*, 235.

Ihsan, A., Brear, M. J., & Jeppesen, M. (2021). Impact of operating uncertainty on the performance of distributed, hybrid, renewable power plants. *Applied Energy*, *282*, 116256.

Inderfurth, K. (1995). Multistage safety stock planning with item demands correlated across products and through time. *Production and Operations Management*, *4*, 127–144.

Infanger, G., & Morton, D. P. (1996). Cut sharing for multistage stochastic linear programs with interstage dependency. *Mathematical Programming*, *75*, 241–256.

Kahsay, T. N., Arjoon, D., Kuik, O., Brouwer, R., Tilmant, A., & van der Zaag, P. (2019). A hybrid partial and general equilibrium modeling approach to assess the hydro-economic impacts of large dams–the case of the grand ethiopian renaissance dam in the eastern nile river basin. *Environmental Modelling & Software*, *117*, 76–88.

Kall, P., Wallace, S. W., & Kall, P. (1994). *Stochastic programming*. Springer.

Khazaei, J., & Powell, W. B. (2018). Smart-invest: a stochastic, dynamic planning for optimizing investments in wind, solar, and storage in the presence of fossil fuels. the case of the pjm electricity market. *Energy Systems*, *9*, 277–303.

Kim, H. (2020). *Stochastic Capacity and Facility Location Planning with Ambiguous Probabilities*. University of California, Berkeley.

Kleywegt, A. J., Shapiro, A., & Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on optimization*, *12*, 479–502.

Kozmík, V. (2015). On variance reduction of mean-cvar monte carlo estimators. *Computational Management Science*, *12*, 221–242.

Kozmík, V., & Morton, D. P. (2015). Evaluating policies in risk-averse multi-stage stochastic programming. *Mathematical Programming*, *152*, 275–300.

Kristiansen, T. (2006). Hydropower scheduling and financial risk management. *Optimal Control Applications and Methods*, *27*, 1–18.

Lan, Y., Zhai, Q., Liu, X., & Guan, X. (2022). Fast stochastic dual dynamic programming for economic dispatch in distribution systems. *IEEE Transactions on Power Systems*, .

Lara, C. L., Siirola, J. D., & Grossmann, I. E. (2020). Electric power infrastructure planning under uncertainty: stochastic dual dynamic integer programming (sddip) and parallelization scheme. *Optimization and Engineering*, *21*, 1243–1281.

Larroyd, P. V., Pedrini, R., Beltrán, F., Teixeira, G., Finardi, E. C., & Picarelli, L. B. (2022). Dealing with negative inflows in the long-term hydrothermal scheduling problem. *Energies*, *15*, 1115.

Leclère, V., Carpentier, P., Chancelier, J.-P., Lenoir, A., & Pacaud, F. (2020). Exact converging bounds for stochastic dual dynamic programming via fenchel duality. *SIAM Journal on Optimization*, *30*, 1223–1250.

Leek, V. (2016). An optimal control toolbox for matlab based on casadi.

Legat, B., Dowson, O., Garcia, J. D., & Lubin, M. (2022). Mathoptinterface: a data structure for mathematical optimization problems. *INFORMS Journal on Computing*, *34*, 672–689.

Leocadio, C. M., Richa, C. S. G., Fortes, M. Z., Ferreira, V. H., & Dias, B. H. (2020). Economic evaluation of a chp biomass plant using stochastic dual dynamic programming. *Electrical Engineering*, *102*, 2605–2615.

Li, X., Zhao, Z., Zhu, X., & Wyatt, T. (2011). Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research*, *74*, 281–310.

Li, Z., Yang, P., Yang, Y., Lu, G., & Tang, Y. (2022). Solving stochastic hydro unit commitment using benders decomposition and modified stochastic dual dynamic programming. *Frontiers in Energy Research*, *10*, 955875.

Lim, M. K., Bassamboo, A., Chopra, S., & Daskin, M. S. (2013). Facility location decisions with random disruptions and imperfect estimation. *Manufacturing & Service Operations Management*, *15*, 239–249.

Liu, R. P., & Shapiro, A. (2020). Risk neutral reformulation approach to risk averse stochastic programming. *European Journal of Operational Research*, *286*, 21–31.

Lohmann, T., Hering, A. S., & Rebennack, S. (2016). Spatio-temporal hydro forecasting of multireservoir inflows for hydro-thermal scheduling. *European Journal of Operational Research*, *255*, 243–258.

Löhndorf, N., & Shapiro, A. (2019). Modeling time-dependent randomness in stochastic dual dynamic programming. *European Journal of Operational Research*, *273*, 650–661.

Löhndorf, N., & Wozabal, D. (2021). Gas storage valuation in incomplete markets. *European Journal of Operational Research*, *288*, 318–330.

Löhndorf, N., & Wozabal, D. (2023). The value of coordination in multimarket bidding of grid energy storage. *Operations Research*, *71*, 1–22.

Löhndorf, N., Wozabal, D., & Minner, S. (2013). Optimizing trading decisions for hydro storage systems using approximate dual dynamic programming. *Operations Research*, *61*, 810–823.

Louveaux, F. V. (1988). Optimal investments for electricity generateion: A stochastic model and a test problem. *Numerical Techniques for Stochastic Optimization*, .

Lu, M., Ran, L., & Shen, Z.-J. M. (2015). Reliable facility location design under uncertain correlated disruptions. *Manufacturing & Service Operations Management*, *17*, 445–455.

Lu, R., Ding, T., Qin, B., Ma, J., Fang, X., & Dong, Z. (2019). Multi-stage stochastic programming to joint economic dispatch for energy and reserve with uncertain renewable energy. *IEEE Transactions on Sustainable Energy*, *11*, 1140–1151.

Löhndorf, N. (2021). Quasar optimization software 3.1. URL: https://http://www.quantego.com.

Mabao, K., & Cabahug, R. G. (2014). Assessment and analysis of t he floodplain of cagayan de oro river basin. *Mindanao Journal of Science and Technology*, *12*.

Maceira, M., & Damázio, J. (2006). Use of the par (p) model in the stochastic dual dynamic programming optimization scheme used in the operation planning of the brazilian hydropower system. *Probability in the Engineering and Informational Sciences*, *20*, 143–156.

Maceira, M. E. P., Marzano, L., Penna, D. D. J., Diniz, A., & Justino, T. (2014). Application of cvar risk aversion approach in the expansion and operation planning and for setting the spot price in the brazilian hydrothermal interconnected system. In *2014 Power Systems Computation Conference* (pp. 1–7). IEEE.

Maceiral, M., Penna, D., Diniz, A., Pinto, R., Melo, A., Vasconcellos, C., & Cruz, C. (2018). Twenty years of application of stochastic dual dynamic programming in official and agent studies in brazil-main features and improvements on the newave model. In *2018 power systems computation conference (PSCC)* (pp. 1–7). IEEE.

Machado, B. G. F., & Bhagwat, P. C. (2020). The impact of the generation mix on the current regulatory framework for hydropower remuneration in brazil. *Energy policy*, *137*, 111129.

Machado, F. D., Diniz, A. L., Borges, C. L., & Brandão, L. C. (2021). Asynchronous parallel stochastic dual dynamic programming applied to hydrothermal generation planning. *Electric Power Systems Research*, *191*, 106907.

Macian-Sorribes, H., Tilmant, A., & Pulido-Velazquez, M. (2017). Improving operating policies of large-scale surface-groundwater systems through stochastic programming. *Water Resources Research*, *53*, 1407–1423.

Marques, G. F., & Tilmant, A. (2013). The economic value of coordination in large-scale multireservoir systems: The parana river case. *Water Resources Research*, *49*, 7546–7557.

de Matos, V. L., & Finardi, E. C. (2012). A computational study of a stochastic optimization model for long term hydrothermal scheduling. *International Journal of Electrical Power & Energy Systems*, *43*, 1443–1452.

de Matos, V. L., Sierra, M. A., Finardi, E. C., Decker, B. U., & Milanezi, A. A. (2015). Stochastic model for energy commercialisation of small hydro plants in the brazilian energy market. *Computational Management Science*, *12*, 111–127.

Mbeutcha, Y., Gendreau, M., & Emiel, G. (2021). Benefit of parma modeling for long-term hydroelectric scheduling using stochastic dual dynamic programming. *Journal of Water Resources Planning and Management*, *147*, 05021002.

Mello, J., Pereira, M., Granville, S., Lima, M., & Alvarenga, S. (1997). Assessment of transmission cost recovery applying marginal pricing in hydrothermal power systems. *Electric power systems research*, *41*, 67–74.

Homem-de Mello, T., De Matos, V. L., & Finardi, E. C. (2011). Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. *Energy Systems*, *2*, 1–31.

Homem-de Mello, T., & Pagnoncelli, B. K. (2016). Risk aversion in multistage stochastic programming: A modeling and algorithmic perspective. *European Journal of Operational Research*, *249*, 188–199.

Melo, M. T., Nickel, S., & Saldanha-Da-Gama, F. (2009). Facility location and supply chain management–a review. *European journal of operational research*, *196*, 401–412.

Mo, B., Gjelsvik, A., & Grundt, A. (2001). Integrated risk management of hydro power scheduling and contract management. *IEEE Transactions on Power Systems*, *16*, 216–221.

Möller, A., Römisch, W., & Weber, K. (2008). Airline network revenue management by multistage stochastic programming. *Computational Management Science*, *5*, 355–377.

Morillo, J. L., Zéphyr, L., Pérez, J. F., Anderson, C. L., & Cadena, Á. (2020). Risk-averse stochastic dual dynamic programming approach for the operation of a hydro-dominated power system in the presence of wind uncertainty. *International Journal of Electrical Power & Energy Systems*, *115*, 105469.

Morillo, J. L., Zephyr, L., Pérez, J. F., Cadena, A., & Anderson, C. L. (2022). Distribution-free chance-constrained load balance model for the operation planning of hydrothermal power systems coupled with multiple renewable energy sources. *International Journal of Electrical Power & Energy Systems*, *142*, 108319.

Murphy, J. (2013). Benders, nested benders and stochastic programming: An intuitive introduction. *arXiv preprint arXiv:1312.3158*, .

Nannicini, G., Traversi, E., & Calvo, R. W. (2021). A benders squared (b 2) framework for infinite-horizon stochastic linear programs. *Mathematical Programming Computation*, *13*, 645–681.

Nápoles-Rivera, F., Rojas-Torres, M. G., Ponce-Ortega, J. M., Serna-González, M., & El-Halwagi, M. M. (2015). Optimal design of macroscopic water networks under parametric uncertainty. *Journal of Cleaner Production*, *88*, 172–184.

Nemirovski, A., Juditsky, A., Lan, G., & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, *19*, 1574–1609.

Nickel, S., Saldanha-da Gama, F., & Ziegler, H.-P. (2012). A multi-stage stochastic supply network design problem with financial decisions and risk management. *Omega*, *40*, 511–524.

Nogueira, F. (2014). Bayesian Optimization: Open source constrained global optimization tool for Python. URL: https://github.com/fmfn/BayesianOptimization.

Nogueira, F. (2020). Bayesian optimization: Open source constrained global optimization tool for python. 2014. *URL https://github. com/fmfn/BayesianOptimization*, .

Nowak, M. P., & Römisch, W. (2000). Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. *Annals of Operations Research*, *100*, 251–272.

Oliveira, F. L. C., Souza, R. C., & Marcato, A. L. M. (2015). A time series model for building scenarios trees applied to stochastic optimisation. *International Journal of Electrical Power & Energy Systems*, *67*, 315–323.

Orchard-Hays, W. (1984). History of mathematical programming systems. *Annals of the History of Computing*, *6*, 296–312.

Owen, S. H., & Daskin, M. S. (1998). Strategic facility location: A review. *European journal of operational research*, *111*, 423–447.

Pacaud, F., Carpentier, P., Chancelier, J.-P., & De Lara, M. (2022). Optimization of a domestic microgrid equipped with solar panel and battery: Model predictive control and stochastic dual dynamic programming approaches. *Energy Systems*, (pp. 1–25).

Pacaud, F., De Lara, M., Chancelier, J.-P., & Carpentier, P. (2021). Distributed multistage optimization of large-scale microgrids under stochasticity. *IEEE Transactions on Power Systems*, *37*, 204–211.

Papavasiliou, A. (2014). Operations Research. *https://perso.uclouvain.be/anthony.papavasiliou/public_html/SDDP.pdf*, .

Papavasiliou, A., Mou, Y., Cambier, L., & Scieur, D. (2017). Application of stochastic dual dynamic programming to the real-time dispatch of storage under renewable supply uncertainty. *IEEE Transactions on Sustainable Energy*, *9*, 547–558.

Parpas, P., Ustun, B., Webster, M., & Tran, Q. K. (2015). Importance sampling in stochastic programming: A markov chain monte carlo approach. *INFORMS Journal on Computing*, *27*, 358–377.

Pedrini, R., & Finardi, E. (2022). Long-term generation scheduling: A tutorial on the practical aspects of the problem solution. *Journal of Control, Automation and Electrical Systems*, (pp. 1–16).

Pelikan, M., Goldberg, D. E., Cantú-Paz, E. et al. (1999). Boa: The bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99* (pp. 525–532). Citeseer volume 1.

Pereira, M. V., & Pinto, L. M. (1991). Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, *52*, 359–375.

Pereira-Cardenal, S. J., Mo, B., Gjelsvik, A., Riegels, N. D., Arnbjerg-Nielsen, K., & Bauer-Gottwein, P. (2016). Joint optimization of regional water-power systems. *Advances in water resources*, *92*, 200–207.

Perkins III, R. T., & Powell, W. B. (2017). Stochastic optimization with parametric cost function approximations. *arXiv preprint arXiv:1703.04644*, .

Philpott, A., de Matos, V., & Finardi, E. (2013). On solving multistage stochastic programs with coherent risk measures. *Operations Research*, *61*, 957–970.

Philpott, A. B., & De Matos, V. L. (2012). Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *European Journal of operational research*, *218*, 470–483.

Philpott, A. B., & Guan, Z. (2008). On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, *36*, 450–455.

Philpott, A. B., de Matos, V. L., & Kapelevich, L. (2018). Distributionally robust sddp. *Computational Management Science*, *15*, 431–454.

Philstar (2021). Galvez: 'odette' made gov't miss yearend vaccination target. *Philstar Global*, . URL: https://www.philstar.com/headlines/2021/12/31/2151123/galvez-odette-made-govt-miss-yearend-vaccination-target.

PHIVOLCS, & DOST (2019). Hazardhunterph. URL: https://hazardhunter.georisk.gov.ph/.

Pina, J., Tilmant, A., & Côté, P. (2017). Optimizing multireservoir system operating policies using exogenous hydrologic variables. *Water Resources Research*, *53*, 9845–9859.

Pinto, R. J., Borges, C. T., & Maceira, M. E. (2013). An efficient parallel algorithm for large scale hydrothermal system operation planning. *IEEE Transactions on Power Systems*, *28*, 4888–4896.

Poorsepahy-Samian, H., Espanmanesh, V., & Zahraie, B. (2016). Improved inflow modeling in stochastic dual dynamic programming. *Journal of Water Resources Planning and Management*, *142*, 04016065.

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

Prékopa, A. (2013). *Stochastic programming* volume 324. Springer Science & Business Media.

Pritchard, G. (2015). Stochastic inflow modeling for hydropower scheduling problems. *European Journal of Operational Research*, *246*, 496–504.

Quezada, F., Gicquel, C., & Kedad-Sidhoum, S. (2022a). Combining polyhedral approaches and stochastic dual dynamic integer programming for solving the uncapacitated lot-sizing problem under uncertainty. *INFORMS Journal on Computing*, *34*, 1024–1041.

Quezada, F., Gicquel, C., & Kedad-Sidhoum, S. (2022b). A stochastic dual dynamic integer programming based approach for remanufacturing planning under uncertainty. *International Journal of Production Research*, (pp. 1–21).

Raby, M., Ríos, S., Jerardino, S., & Raineri, R. (2009). Integrating large wind farms into transmission planning of hydrothermal system. *Journal of Energy Engineering*, *135*, 89–97.

Raso, L., Bader, J.-C., & Weijs, S. (2020). Reservoir operation optimized for hydropower production reduces conflict with traditional water uses in the senegal river. *Journal of Water Resources Planning and Management*, *146*, 05020003.

Raso, L., Chiavico, M., & Dorchies, D. (2019). Optimal and centralized reservoir management for drought and flood protection on the upper seine–aube river system using stochastic dual dynamic programming. *Journal of Water Resources Planning and Management*, *145*, 05019002.

Raso, L., Malaterre, P.-O., & Bader, J.-C. (2017). An effective streamflow process model for optimal reservoir operation using stochastic dual dynamic programming. *Journal of water resources planning and management*, *143*, 11.

Rebennack, S. (2014). Generation expansion planning under uncertainty with emissions quotas. *Electric Power Systems Research*, *114*, 78–85.

Rebennack, S. (2016). Combining sampling-based and scenario-based nested benders decomposition methods: application to stochastic dual dynamic programming. *Mathematical Programming*, *156*, 343–389.

Rebennack, S., Flach, B., Pereira, M. V., & Pardalos, P. M. (2011). Stochastic hydro-thermal scheduling under co2 emissions constraints. *IEEE Transactions on Power Systems*, *27*, 58–68.

Rego, E. E., de Oliveira Ribeiro, C., do Valle Costa, O. L., & Ho, L. L. (2017). Thermoelectric dispatch: From utopian planning to reality. *Energy Policy*, *106*, 266–277.

Resende, L. d. O., Valladão, D., Bezerra, B. V., & Cyrillo, Y. M. (2021). Assessing the value of natural gas underground storage in the brazilian system via stochastic dual dynamic programming. *Top*, *29*, 106–124.

Reus, L., Pagnoncelli, B., & Armstrong, M. (2019). Better management of production incidents in mining using multistage stochastic optimization. *Resources Policy*, *63*, 101404.

Reus, L., & Prado, R. (2022). Need to meet investment goals? track synthetic indexes with the sddp method. *Computational Economics*, *60*, 47–69.

Ritchie, H., & Roser, M. (2014). Natural disasters. *Our World in Data*, . Https://ourworldindata.org/natural-disasters.

Rockafellar, R. T., & Wets, R. J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, *16*, 119–147.

Römisch, W., & Schultz, R. (2001). Multistage stochastic integer programs: An introduction. In *Online optimization of large scale systems* (pp. 581–600). Springer.

Ross, S. M. (2014). *Introduction to stochastic dynamic programming*. Academic press.

Rotting, T., & Gjelsvik, A. (1992). Stochastic dual dynamic programming for seasonal scheduling in the norwegian power system. *IEEE Transactions on Power Systems*, *7*, 273–279.

Rougé, C., & Tilmant, A. (2016). Using stochastic dual dynamic programming in problems with multiple near-optimal solutions. *Water Resources Research*, *52*, 4151–4163.

Rougé, C., Tilmant, A., Zaitchik, B., Dezfuli, A., & Salman, M. (2018). Identifying key water resource vulnerabilities in data-scarce transboundary river basins. *Water Resources Research*, *54*, 5264–5281.

Sauma, E., Jerardino, S., Barria, C., Marambio, R., Brugman, A., & Mejía, J. (2011). Electric-systems integration in the andes community: Opportunities and threats. *Energy Policy*, *39*, 936–949.

Scheben, H., Klempp, N., & Hufendiek, K. (2020). Impact of long-term water inflow uncertainty on wholesale electricity prices in markets with high shares of renewable energies and storages. *Energies*, *13*, 2347.

Schittkowski, K. (1986). Nlpql: A fortran subroutine solving constrained nonlinear programming problems. *Annals of operations research*, *5*, 485–500.

Seranilla, B. K. (2019). A linear and discrete mathematical model for a combined cooling, heat, and power (cchp) system's optimal operational strategy.

Seranilla, B. K., & Löhndorf, N. (2023). Optimizing vaccine distribution in developing countries under natural disaster risk. *Naval Research Logistics*, .

Shapiro, A. (2011). Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, *209*, 63–72.

Shapiro, A. (2021). Tutorial on risk neutral, distributionally robust and risk averse multistage stochastic programming. *European Journal of Operational Research*, *288*, 1–13.

Shapiro, A., & Cheng, Y. (2021). Central limit theorem and sample complexity of stationary stochastic programs. *Operations Research Letters*, *49*, 676–681.

Shapiro, A., & Ding, L. (2020). Periodical multistage stochastic programs. *SIAM Journal on Optimization*, *30*, 2083–2102.

Shapiro, A., & Philpott, A. (2007). A tutorial on stochastic programming. *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html*, *17*.

Shapiro, A., Tekaya, W., da Costa, J. P., & Soares, M. P. (2013a). Risk neutral and risk averse stochastic dual dynamic programming method. *European journal of operational research*, *224*, 375–391.

Shapiro, A., Tekaya, W., Soares, M. P., & da Costa, J. P. (2013b). Worst-case-expectation approach to optimization under uncertainty. *Operations Research*, *61*, 1435–1449.

Shi, Z., Liang, H., Huang, S., & Dinavahi, V. (2019). Multistage robust energy management for microgrids considering uncertainty. *IET Generation, Transmission & Distribution*, *13*, 1906–1913.

Shinde, P., Kouveliotis-Lysikatos, I., & Amelin, M. (2022). Multistage stochastic programming for vpp trading in continuous intraday electricity markets. *IEEE Transactions on Sustainable Energy*, *13*, 1037–1048.

Shuai, H., Fang, J., Ai, X., Yao, W., Wen, J., & He, H. (2019). On-line energy management of microgrid via parametric cost function approximation. *IEEE Transactions on Power Systems*, *34*, 3300–3302.

Siddig, M., & Song, Y. (2022). Adaptive partition-based sddp algorithms for multistage stochastic linear programming with fixed recourse. *Computational Optimization and Applications*, *81*, 201–250.

da Silva, E. L., & Finardi, E. C. (2003). Parallel processing applied to the planning of hydrothermal systems. *IEEE Transactions on Parallel and Distributed Systems*, *14*, 721–729.

Silva, T., Valladão, D., & Homem-de Mello, T. (2021). A data-driven approach for a class of stochastic dynamic optimization problems. *Computational Optimization and Applications*, *80*, 687–729.

da Silva Fernandes, A., Oliveira, M. T., Marcato, A. L., Oliveira, E. J., Junior, I., & Oliveira, E. (2019). Representation of wind energy scenarios in the mid-term hydrothermal systems operation scheduling. *Journal of Control, Automation and Electrical Systems*, *30*, 413–423.

Snyder, L. V. (2006). Facility location under uncertainty: a review. *IIE transactions*, *38*, 547–564.

Snyder, L. V., Atan, Z., Peng, P., Rong, Y., Schmitt, A. J., & Sinsoysal, B. (2016). Or/ms models for supply chain disruptions: A review. *Iie Transactions*, *48*, 89–109.

Soares, M. P., Street, A., & Valladão, D. M. (2017). On the solution variability reduction of stochastic dual dynamic programming applied to energy planning. *European Journal of Operational Research*, *258*, 743–760.

SPSociety (2012). Stochastic Programming Resources Test Set. URL: https://stoprog.org/resources.

Steeger, G., & Rebennack, S. (2017). Dynamic convexification within nested benders decomposition using lagrangian relaxation: An application to the strategic bidding problem. *European Journal of Operational Research*, *257*, 669–686.

Street, A., Brigatto, A., & Valladão, D. M. (2017). Co-optimization of energy and ancillary services for hydrothermal operation planning under a general security criterion. *IEEE Transactions on Power Systems*, *32*, 4914–4923.

Street, A., Valladão, D., Lawson, A., & Velloso, A. (2020). Assessing the cost of the hazard-decision simplification in multistage stochastic hydrothermal scheduling. *Applied Energy*, *280*, 115939.

Stüber, M., & Odersky, L. (2020). Uncertainty modeling with the open source framework urbs. *Energy Strategy Reviews*, *29*, 100486.

Tariku, T. B., Gan, K. E., Tan, X., Gan, T. Y., Shi, H., & Tilmant, A. (2021). Global warming impact to river basin of blue nile and the optimum operation of its multi-reservoir system for hydropower production and irrigation. *Science of The Total Environment*, *767*, 144863.

Terça, G., & Wozabal, D. (2021). Envelope theorems for multistage linear stochastic optimization. *Operations Research*, *69*, 1608–1629.

The Economist, N. (2022). The world in brief, january 31st 2022. *The Economist*, . URL: https://espresso.economist.com/9352bfb0cb8ca3110212a0aa2499da39.

Thevenin, S., Adulyasak, Y., & Cordeau, J.-F. (2020). Stochastic dual dynamic programming for multi-echelon lot-sizing with component substitution. *Cahier du GERAD*, .

Thevenin, S., Adulyasak, Y., & Cordeau, J.-F. (2022). Stochastic dual dynamic programming for multiechelon lot sizing with component substitution. *INFORMS Journal on Computing*, *34*, 3151–3169.

Thompson, R. T. (1998). Fast sequential and parallel methods for solving multistage stochastic linear programmes.

Tilmant, A., & Kelman, R. (2007). A stochastic approach to analyze trade-offs and risks associated with large-scale water resources systems. *Water resources research*, *43*.

Tilmant, A., Lettany, J., & Kelman, R. (2007). Hydrological risk assessment in the euphrates-tigris river basin: A stochastic dual dynamic programming approach. *Water International*, *32*, 294–309.

Toledo, F., Sauma, E., & Jerardino, S. (2015). Energy cost distortion due to ignoring natural gas network limitations in the scheduling of hydrothermal power systems. *IEEE Transactions on Power Systems*, *31*, 3785–3793.

Tong, X., Yang, L., Luo, X., & Rao, B. (2020). A stochastic dual dynamic programming method for two-stage distributionally robust optimization problems. *Optimization Methods and Software*, *35*, 1002–1021.

Treistman, F., Maceira, M. E. P., Penna, D. D. J., Damázio, J. M., & Rotunno Filho, O. C. (2020). Synthetic scenario generation of monthly streamflows conditioned to the el niño–southern oscillation: application to operation planning of hydrothermal systems. *Stochastic Environmental Research and Risk Assessment*, *34*, 331–353.

Trivedi, A., & Singh, A. (2018). Facility location in humanitarian relief: A review. *International Journal of Emergency Management*, *14*, 213–232.

Tsang, M. Y., Sit, T., & Wong, H. Y. (2022). Robust portfolio optimization with respect to spectral risk measures under correlation uncertainty. *Applied Mathematics & Optimization*, *86*, 8.

Valladão, D., Silva, T., & Poggi, M. (2019). Time-consistent risk-constrained dynamic portfolio optimization with transactional costs and time-dependent returns. *Annals of Operations Research*, *282*, 379–405.

Van Ackooij, W., de Oliveira, W., & Song, Y. (2019). On level regularization with normal solutions in decomposition methods for multistage stochastic programming problems. *Computational Optimization and Applications*, *74*, 1–42.

Van Slyke, R. M., & Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM journal on applied mathematics*, *17*, 638–663.

Waga, M., Valladão, D., Street, A., & Silva, T. (2022). Disentangling shareholder risk aversion from leverage-dependent borrowing cost on corporate policies. *Computational Economics*, (pp. 1–24).

Wallace, S. W., & Ziemba, W. T. (2005). *Applications of stochastic programming*. SIAM.

Ward, R., Wu, X., & Bottou, L. (2018). Adagrad stepsizes: Sharp convergence over nonconvex landscapes, from any initialization. *arXiv preprint arXiv:1806.01811*, *2*.

WHO (2022). Who covid-19 dashboard. URL: https://covid19.who.int/.

Wozabal, D., & Rameseder, G. (2020). Optimal bidding of a virtual power plant on the spanish day-ahead and intraday market for electricity. *European Journal of Operational Research*, *280*, 639–655.

Yang, H., & Nagarajan, H. (2022). Optimal power flow in distribution networks under n–1 disruptions: A multistage stochastic programming approach. *INFORMS Journal on Computing*, *34*, 690–709.

Yıldız, B., & Sütçü, M. (2022). A variant sddp approach for periodic-review approximately optimal pricing of a slow-moving a item in a duopoly under price protection with end-of-life return and retail fixed markdown policy. *Expert Systems with Applications*, (p. 118801).

Yin, G., Wang, J., Zhang, Q., & Liu, Y. (2006). Stochastic optimization algorithms for pricing american put options under regime-switching models. *Journal of optimization theory and applications*, *131*, 37–52.

Yu, X., & Shen, S. (2022). Multistage distributionally robust mixed-integer programming with decision-dependent moment-based ambiguity sets. *Mathematical Programming*, *196*, 1025–1064.

Yu, X., Shen, S., & Ahmed, S. (2021). On the value of multistage stochastic facility location with risk aversion. *arXiv preprint arXiv:2105.11005*, .

Zambelli, M., Soares Filho, S., Toscano, A. E., Santos, E. d., & Silva Filho, D. d. (2011). Newave versus odin: comparison of stochastic and deterministic models for the long term hydropower scheduling of the interconnected brazilian system. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, *22*, 598–609.

Zapletal, F., Šmíd, M., & Kozmík, V. (2022). Multi-stage stochastic optimization of carbon risk management. *Expert Systems with Applications*, *201*, 117021.

Zéphyr, L., & Anderson, C. L. (2018). Stochastic dynamic programming approach to managing power system uncertainty with distributed storage. *Computational Management Science*, *15*, 87–110.

Zhang, J. L., & Ponnambalam, K. (2005). Stochastic control for risk under deregulated electricity market—a case study using a new formulation. *Canadian Journal of Civil Engineering*, *32*, 719–725.

Zou, J., Ahmed, S., & Sun, X. A. (2018). Multistage stochastic unit commitment using stochastic dual dynamic integer programming. *IEEE transactions on Power Systems*, *34*, 1814–1823.

Zou, J., Ahmed, S., & Sun, X. A. (2019). Stochastic dual dynamic integer programming. *Mathematical Programming*, *175*, 461–502.