First Order Methods for Stochastic Optimal Control with an Application to Microgrid Optimization

Johannes Schnebel,
7. Juli 2025

MASTERARBEIT
FSS 2025 · Universität Mannheim,
Lehrstuhl für Mathematische Optimierung,
Prof. Mathias Staudigl, PhD
Zweitgutachter: Prof. Dr. Simon Weißmann

Abstract

Stochastic Optimal Control Problems in discrete time lie at the intersection of optimal control and stochastic programming. Whereas both communities have their own stack of well established theory and algorithms, recently there is a growing effort to use techniques from stochastic programming to solve stochastic optimal control problems. First-order methods are of special interest in this context since they allow to solve large-scale stochastic programs, which are typical for modern applications such as power-grid optimization or reinforcement learning. In this thesis we investigate how to generalize the Dynamic Stochastic Approximation for multistage stochastic programming (DSA), which was introduced by Lan and Zhou (2021). We use different saddle-point solvers as sub-routines and employ DSA to solve stochastic optimal control problems. First we reformulate the stochastic optimal control problem as a nested sequence of saddle-point problems. Then we present the DSA method and a variant with the Proximal Method of Multipliers as saddle-point solver. Finally we apply DSA to dynamically optimize a microgrid under stochastic power demand and solar production.

Eidestattliche Erklärung

Hiermit versichere ich, dass diese Arbeit von mir persönlich verfasst wurde und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internetquellen.

Mannheim, 07.07.2025 ____________________

Johannes Schnebel, Matrikelnummer 1908467

Contents

Abbreviations

Abbreviation	**Full Form**
ALM	Augmented Lagrangian Method
ADMM	Alternating Direction Method of Multipliers
CCP	closed, convex, proper
DGF	Distance Generating Function
DPE	Dynamic Programming Equation
DSA	Dynamic Stochastic Approximation
MSSP	Multistage Stochastic Program
PDHG	Primal Dual Hybrid Gradient
PMM	Proximal Method of Multipliers
SDDP	Stochastic Dual Dynamic Programming
SOC	Stochastic Optimal Control
SP	Stochastic Program
w.p.1	with probability one
w.r.t	with respect to

1 Introduction

1.1 Background

Everyone solves stochastic optimal control (SOC) problems heuristically on a daily basis. We have to make decisions; to do so we form expectations about the future state of the world and consider how today's decisions will influence the scope of future decisions. At least hypothetically, we aim to maximize our overall well-being. Do I study this last chapter of my university course this evening or meet with my friends? There is a chance that the instructor won't ask anything about it in the exam, so my effort would be for nothing. Do I choose my holiday destination now, while the hotel is still inexpensive but at the risk of bad weather - or do I wait until the last minute, when the weather forecast is more accurate.

Formally speaking, the goal of stochastic optimal control is to make sequential optimal decisions over a time horizon with realizations of a random process between decisions. Often previous decisions influence the space of subsequent feasible decisions. Stochastic optimal control is of particular importance in power system operation. Think for instance of a pumped hydro storage plant. Operators want to maximize revenue from buying and selling electricity. To decide which amount of water to pump (buying electricity) or turbine (selling electricity) today, they have to consider what their decision today means for the storage level in the lake tomorrow and therefore for their ability to pump or turbine water tomorrow. Between today and tomorrow there might be a rainfall changing the amount of water in the storage. Besides, tomorrow electricity prices are unknown. The operators have to account for these uncertainties when making today decision to maximize total revenue over today and tomorrow. This is just one example of a typical SOC problem coming up in power system operation.

SOC problems can also be formulated as so-called Multistage Stochastic Programs (MSSP). The difference between stochastic optimal control and multistage stochastic programming is more a question of modelling. While SOC makes a distinction between *state-* and *control-variables*, MSSP does not make this distinction. As we will see later, we can formulate certain SOC Problems as MSSPs.

The goal of this thesis is twofold. At first, we aim to give a unified formulation of SOC-Problems as MSSPs. And second we present a method to formulate a MSSP as a nested sequence of saddle-point problems and solve them with state-of-the-art methods from large scale convex optimization. This reformulation of a MSSP in terms of saddle-point problems was first done by Lan and Zhou (2021) but has not received much attention since then. We see it as a valuable tool among the existing methods to solve MSSPs, since the state-dimension only enters the problem in the dimension of the saddle-point problems and there exists a variety of well performing methods for large-scale saddle-point problems. That means this method does not suffer from the classical *curse of dimensionality*, which denotes the exponential scaling in the state-dimension inherent to many (stochastic) optimal control solution methods. However, our method scales exponentially in the number of stages, which only goes linearly in the effort for standard techniques like *Stochastic Dual Dynamic Programming* (SDDP) (see section 3). Therefore, Lan and Zhou (2021) state that their Dynamic Stochastic Approximation method and therefore also our proposed variants of it can be seen as complementary to Stochastic Dual Dynamic Programming (SDDP), which is nowadays one of the standard techniques to solve MSSPs.

1.2 Contribution of this thesis

This thesis serves as the starting point of a research project where we want to study the potential of first order methods to solve stochastic optimal control problems. Our main goal is to generalize the DSA method of Lan and Zhou (2021). They use the *Primal-Dual Hybrid Gradient* (PDHG)-method of Chambolle and Pock (2011) to solve the upcoming saddle-point problems. However, numerical experiments show that their DSA method converges slowly towards a feasible policy. Since feasibility is crucial in power-system applications, we replace the Primal-Dual Hybrid Gradient subroutine to solve the upcoming saddle-point problems in DSA with the *Proximal Method of Multipliers* (PMM) of Rockafellar (1976). To our knowledge there has not been any further research into this DSA-method since the Lan and Zhou's paper. We implement the DSA with both subroutines in `Python` to get numerical insights of the performance of this new DSA method. Furthermore we present a general reformulation of a SOC as a MSSP. We then reformulate the stochastic optimal control problem of managing a microgrid under uncertain power demand and solar feed-in as a MSSP and compare numerical results both variants of DSA when applied to different instances of this problem.

1.3 State-of-the-art

One of the most widely used techniques to solve multistage stochastic optimization problems is *Stochastic dual Dynamic Programming (SDDP)*, which was first introduced by Pereira and Pinto (Pereira and Pinto, 1991). It was originally invented to solve stochastic linear programs to optimize Brazilian hydrothermal power systems. This technique solves the multistage problem by recursively approximating the later-stage value functions by a convex polyhedral model, starting with the last stage and going backwards in time. In section 3, we present this method and give a short overview of the existing literature. In this thesis we take an alternative approach. Similar to SDDP, our starting point are the dynamic programming equations of the MSSP. And also similar to SDDP, we heavily rely on the fact that we can obtain a subgradient of a value function by computing a dual solution. With that first-order information we can solve the Dynamic Programming Equations (DPEs) with state-of-the-art methods for convex optimization problems. Most of these methods solve a convex optimization problem by solving the saddle-point problem of the associated lagrangian. To obtain a subgradient of the value function, we have to solve the next-stage problem and so on, leading to a solution of the MSSP by solving a nested sequence of saddle-point problems. In recent years so-called first order methods to solve convex optimization problems have gained a lot of attention. They share the advantage of low computational costs at each iteration even for large-scale problems. They converge relatively fast to a near-optimal and near-feasible point, but progress after some iterations slows significantly. These properties make them particularly suited for modern (stochastic) optimization tasks arising in machine-learning and imaging where it is unreasonable to demand accuracies higher than the statistical error of the input-data. Also in stochastic power-system optimization it is not necessary to demand for accuracies higher than the expected error of forecasts for prices, renewable-production and demand. For a comprehensive overview of first-order methods in convex optimization and contemporary applications see Ryu and Yin (2022), Dvurechensky et al. (2021) or Beck (2017). One very popular primal-dual method to solve static deterministic convex optimization problems is the previously mentioned *Primal-Dual-Hybrid-Gradient* (PDHG)-method of Chambolle and Pock (2011), which is used as the saddle-point solver in Lan and Zhou (2021)'s DSA method. As we will later see, PDHG has some limitations in achieving good numerical results in this setting. An alternative to PDHG is the classical *Augmented Lagrangian Method* or *Method of Multipliers* (ALM) and its proximal variant PMM (Rockafellar, 1976). This method adds the squared norm of the constraint violation to the objective, therefore converging faster towards a feasible point. With this method, one has to balance feasibility and optimality. This better feasibility convergence makes Augmented Lagrangian methods well suited for power-system optimization where problems are often large-scale and feasibility is important. A modern extension of the ALM is the alternating direction method of Multipliers which is well suited for optimzation problems where two varables in the objective are coupled via an affine-linear coupling constraint. A contemporary version of ADMM is the Proximal ADMM, see Shefi and Teboulle (2014). Since in the saddle-point problems we want to solve with ADMM, we only have first order information on the next-stage value function in the objective, we need to use a function-linarized ADMM-variant, see Banert et al. (2021). That should not be confused with the *Linearized* ADMM, a variant of ADMM where a quadratic term which always comes up in ADMM-iterations is replaced by a first-order approximation (for an overview see chapters 3 and 8 in Ryu and Yin (2022)).

The use of first-order methods to solve stochastic optimization problems of the form $\min_{x \in \mathcal{X}} \mathbb{E}_\xi \left[F(x, \xi)\right]$ dates back to the invention of the stochastic approximation method by Robbins and Monro (1951). A recent contribution and the underlying of the DSA-method of Lan and Zhou (2021) is the monograph by Nemirovski et al. (2009).

1.4 Organization of this thesis

In the second chapter we will introduce the main mathematical concepts and important results to analyse MSSPs. We introduce the main concepts of convex analysis, set-valued mappings and monotone operator theory. Since analytical properties of value functions of parametric optimization problems are crucial for our methods to solve the DPEs, we will analyse necessary and sufficient conditions on the involved constraints to guarantee convexity and show how to compute subgradients of the value function of very general class of parametric optimization problems.

We then go on to discuss solution methods for MSSPs in chapter three. We give a short overview over SDDP and then show how to reforumalte the MSSPs as a nested sequence of saddle-poin problems. We introduce the DSA-method of Lan and Zhou (2021) and state their main convergence results. In order to use different subroutines for the solution of the saddle-point problems we derive ADMM from an operator splitting method of a monotone inclusion problem and introduce two important variants.

In the next chapter, chapter four, we demonstrate how to transform a SOC problem into a MSSP problem in order to apply first order methods.

In the fith chapter we solve a microgrid optimization problem with DSA. We first formulate the microgrid stochastic optimal control problem as a MSSP and then solve it with PDHG and ADMM as subroutines with different parameter settings and number of iterations.

We conclude this thesis with a critical discussion of our findings and an outlook on further research.

2 Mathematical Background

2.1 Elements of Convex Analysis

We take most of this section from Bauschke and Combettes (2017). Let $\mathcal{H}$ be a Hilbert space.

Definition 2.1 (Convex Function)**.** *Let $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$. Then f is convex, if its epigraph* $\operatorname{epi} f = \{(x, \alpha) \in \mathcal{H} \times \mathbb{R} : f(x) \leq \alpha\}$ *is a convex set in $\mathcal{H} \times \mathbb{R}$.*

We call the set $\operatorname{dom} f = \{x \in \mathcal{H} : f(x) < \infty\}$ the *domain* of f. There are also equivalent properties to convexity as defined above and often convexity is defined by one of them.

Proposition 2.2. *Let $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$. Then convexity of f is euqivalent to one of the following properties*

1. *For $x, y \in \operatorname{dom} f$ and $\lambda \in [0, 1]$*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \tag{1}$$

2. *If f is differentiable,*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle \ \ \forall x, y \in \operatorname{dom} f \tag{2}$$

3. *If f is twice continuosly differentiable*

$$\nabla^2 f(x) \succeq 0 \tag{3}$$

We call a convex function proper, if it never attains the value $-\infty$ and attains a value $f(x) = \alpha < \infty$ somewhere. We call a convex function closed, if its epigraph is a closed subset of $\mathcal{H} \times \mathbb{R}$. Since we will need this three properties of a function f, convexity, closedness and properness often together we will shortly call f a CCP-function. Even though we do not assume any sort of differentiability in convex analysis, we can generalize the concept of a gradient to the setting of convex but not necessarily smooth functions.

Definition 2.3 (Subdifferential)**.** *Let $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ be proper. We call the set-valued operator*

$$\partial f : \mathcal{H} \to \mathcal{P}(\mathcal{H}), \ x \mapsto \{g \in \mathcal{H} : f(y) \geq f(x) + \langle g, y - x\rangle \ \ \forall y \in \mathcal{H}\} \tag{4}$$

the subdifferential of f. We call an element $g \in \partial f(x)$ a subgradient of f at x.

If, however, the function f is given in form of an expectation $f(x) = \mathbb{E}_\xi [F(x, \xi)]$, we usually have to rely on biased estimators of subgradients for f. By introducing the concept of ϵ-subgradients we can "shift" the bias which might be added in the computation to an ϵ and further assume that we can compute unbiased estimators of biased subgradients.

Definition 2.4 (ϵ-Subdifferential)**.** *Let $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ be proper and $\epsilon > 0$. We call the set-valued operator*

$$\partial_\epsilon f : \mathcal{H} \to \mathcal{P}(\mathcal{H}), \ x \mapsto \{g \in \mathcal{H} : f(y) \geq f(x) + \langle g, y - x\rangle - \epsilon \ \forall y \in \mathcal{H}\} \tag{5}$$

the ϵ-subdifferential of f and an element $g \in \partial_\epsilon f(x)$ an ϵ-subgradient of f at x.

Definition 2.5 (Stochastic ϵ-subgradient)**.** *Let $f(x) := \mathbb{E}_\xi[F(x,\xi)]$. We call a random variable $G(\xi)$ a stochastic (ϵ-)subgradient of f at x if for $\epsilon \geq 0$, G is an unbiased estimator of an (ϵ-)subgradient, that is*

$$\mathbb{E}_\xi[G(\xi)] = g \text{ and } g \in \partial_\epsilon f(x) \tag{6}$$

We can further refine the concept of convextiy.

Definition 2.6 (Strong Convexity)**.** *Let $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$. We say f is μ-strongly convex for a $\mu > 0$, if*

$$f(x) - \tfrac{\mu}{2}||x||^2 \tag{7}$$

is convex.

There is also an equivalent definition of strong convexity in terms of the subdifferential

Proposition 2.7. *$f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ is μ-strongly convex, iff*

$$\langle u - v, x - y \rangle \geq \mu||x - y||^2 \text{ for all } x, y \in \operatorname{dom} f \text{ and } u \in \partial f(x),\ v \in \partial f(y) \tag{8}$$

We say that the subdifferential is a μ-strongly coercive operator. We will come back to the operators later in section 2.7. We will now introduce a generalized notion of a distance. This gives us the flexibility to adapt a distance measure to the geometry of our problem.

Definition 2.8 (Distance-generating function, Definition 6.1 in Staudigl (2024))**.** *Let $X \subset \mathcal{H}$ be a closed, bounded and convex subset with a norm $||\cdot||$. Denote X° the relative interior of the set X. A mapping $\psi : X \to \mathbb{R}$ is called a distance generating function (DGF) with modulus $\alpha > 0$ if*

- *ψ is CCP with $X \subset \operatorname{dom}\psi$;*
- *The set $\operatorname{dom}\partial\psi = \{x \in X : \ \partial\psi(x) \neq \emptyset\}$ is nonempty and convex;*
- *ψ restricted to $\operatorname{dom}\partial\psi$ is α-strongly convex w.r.t the norm $||\cdot||$ induced by $\langle\cdot,\cdot\rangle$*

Then we define the Bregman divergence associated with the DGF ψ as

$$D_\psi(u, x) = \psi(u) - \psi(x) - \langle\nabla\psi(x), u - x\rangle \quad \forall(u, x) \in \operatorname{dom}\psi \times \operatorname{dom}\partial h. \tag{9}$$

That is the Bregman divergence measures the difference between the function ψ and its first order Taylor-approximation at x. Note that $D_\psi(u, x) \geq 0$ always since we ψ is convex and $D_\psi(u, x) = 0 \Leftrightarrow u = x$. This justifies the notion of a generalized distance. But in general D_ψ does not satisfy the triangle-inequality. When it is clear from the context, which DGF ψ is used in the definition of $D_\psi(u, x)$, we will just write $D(u, x)$.

Proposition 2.9 (Three-point identity)**.** *Let $D_\psi(x, y)$ be the Bregman-distance associated with the DGF ψ. Then*

$$D_\psi(u, x) - D_\psi(y, x) = \langle\nabla\psi(y) - \nabla\psi(x), u - y\rangle + D_\psi(u, y) \tag{10}$$

In particular in the euclidean case $\psi = ||\cdot||^2$ where the norm $||\cdot||$ is induced by the scalar product $\langle\cdot,\cdot\rangle$, this yields

$$||u - x||^2 - ||y - x||^2 = 2\langle y - x, u - y\rangle + ||u - y||^2. \tag{11}$$

Equality (11) is sometimes also refered to as completion of squares

Proof.

$$
\begin{aligned}
D_\psi(u,x) - D_\psi(u,y) - D_\psi(y,x) &= \psi(u) - \psi(x) - \psi(u) + \psi(y) - \psi(y) + \psi(x) \\
&\quad - \langle \nabla\psi(x), u - x\rangle + \langle \nabla\psi(y), u - y\rangle + \langle \nabla\psi(x), y - x\rangle \\
&= \langle \nabla\psi(x), y - u\rangle + \langle \nabla\psi(y), u - y\rangle \\
&= \langle \nabla\psi(y) - \nabla\psi(y), u - y\rangle
\end{aligned} \tag{12}
$$

□

We now introduce the concept of convex duality. It allows us to formulate convex optimization problem in a more abstract setting and from there gain insights about the original optimization problems. We will only treat this topic as far as we will employ some of the techniques later but unfortunately cannot give a comprehensive overview of this interesting field as this would exceed the scope of this thesis. We refer the interested reader to Bauschke and Combettes (2017), Bot (2009) or Ekeland and Temam (1999). The main object of interest is the so called *convex conjugate* or *Fenchel-conjugate* of a convex function f:

Definition 2.10 (Convex Conjugate). *Let $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ be a convex function. Then the convex conjugate of f is*

$$
\begin{aligned}
&f^* : \mathcal{H} \to \mathbb{R} \cup \{-\infty, \infty\} \\
&f^*(x^*) = \sup_{x \in \mathcal{H}} \{\langle x, x^*\rangle - f(x)\}
\end{aligned} \tag{13}
$$

We can also define the conjugate of the conjugate of f, the biconjugate, as $f^{**} : \mathcal{H} \to \mathbb{R} \cup \{-\infty, \infty\}$. Since here we restict ourselves to the setting of Hilbert spaces, which are self-dual, we will not distinguish between the space $\mathcal{H}$ and its dual. In the more general setting of a vector space $\mathcal{X}$ with dual $\mathcal{X}^*$, we would have $f : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ and $f : \mathcal{X}^* \to \mathbb{R} \cup \{-\infty, \infty\}$. From the definition of f^* we get immediately a famous inequality, the *Fenchel-Young inequalty*: If f is proper,

$$
f^*(x^*) + f(x) \geq \langle x^*, x\rangle \quad \forall x^*, x \in \mathcal{H} \tag{14}
$$

The next theorem is an important result as it states that under some regularity conditions, f and f^{**} coincide

Theorem 2.11 (Fenchel-Moreau). $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$. *Then $f = f^{**}$ if and only if one of the following holds*

1. *f is CCP*
2. *$f \equiv \infty$*
3. *$f \equiv -\infty$*

Proof. In the cases 2. and 3. we see directly $f = f^{**}$ For the proof of 1. we refer to Bauschke and Combettes (2017), Theorem 13.37. □

There is an important connection between the convex conjugate and the subdifferential of a f

Theorem 2.12. *Let $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ be CCP. Then*

$$f^*(x^*) + f(x) = \langle x^*, x\rangle \iff x^* \in \partial f(x) \tag{15}$$

Proof.

$$\begin{aligned} & x^* \in \partial f(x) \\ & \Leftrightarrow 0 \in \partial f(x) - x^* \\ \text{(Fermats Rule)} \quad & \Leftrightarrow x \in \operatorname{argmin}_{z\in\mathcal{H}} \{f(z) - \langle x^*, z\rangle\} \\ & \Leftrightarrow x \in \operatorname{argmax}_{z\in\mathcal{H}} \{\langle x^*, z\rangle - f(z)\} \\ & \Leftrightarrow \langle x^*, x\rangle - f(x) = f^*(x^*), \end{aligned} \tag{16}$$

□

Corollary 2.13. *Now since for f CCP we have $f^{**} = f$, we get*

$$\begin{aligned} x^* \in \partial f(x) & \iff f^{**}(x) = \langle x, x^*\rangle - f^*(x^*) \\ & \iff y^* \in \operatorname*{argmax}_{z^*} \{\langle x, z^*\rangle - f^*(z^*)\} \end{aligned} \tag{17}$$

Proposition 2.14 (Conjugate of Indicator Function)**.** *Let $C \subset \mathcal{E}$ be a convex cone. Then the convex-conjugate of the indicator function $x \mapsto \mathbb{I}_C(x)$ is*

$$x \mapsto \mathbb{I}_{C^*}(-x^*) \tag{18}$$

Proof.

$$\mathbb{I}^*(x^*) = \sup_{x\in\mathcal{E}} \{\langle x, x^*\rangle - \mathbb{I}_C(x)\} = \sup_{x\in C} \langle x, x^*\rangle = \mathbb{I}_{-C^*}(x^*) = \mathbb{I}_{C^*}(-x^*) \tag{19}$$

□

2.2 Set-valued mappings

Let M, B be two sets. A *set-value mapping* or *multifunction* between M and B is a mapping $S : M \to 2^B$. In this and the next subsection 2.3 we state some important properties of set-valued mappings, taken from Rockafellar and Wets (1998) and Shapiro et al. (2021). For a multifunction $S : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$, we define the *domain* of S as

$$\operatorname{dom} S = \{x \in \mathbb{R}^n | S(x) \neq \emptyset\} \tag{20}$$

and the *graph* of S

$$\operatorname{gph} S = \{(x, u) | u \in S(x)\} \subset \mathbb{R}^n \times \mathbb{R}^m. \tag{21}$$

We call a multifunction S *convex-valued* on a convex set $C \subset \mathbb{R}^n$, if $S(x)$ is a convex set in $\mathbb{R}^m$ for all $x \in C$.

Definition 2.15 (Convex Multifunction)**.** *A multifunction $S : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is called convex on a convex set $C \subset \mathbb{R}^n$, if*

$$\operatorname{gph} S \cap (C \times \mathbb{R}^m) \tag{22}$$

is convex or equivalently for $x_1, x_2 \in C,\ \lambda \in (0, 1)$

$$\lambda S(x_1) + (1 - \lambda)S(x_2) \subset S(\lambda x_1 + (1 - \lambda x_2)) \tag{23}$$

or equivalently for $y_1 \in S(x_1),\ y_2 \in S(x_2),\ \lambda \in (0, 1)$,

$$\lambda y_1 + (1 - \lambda)y_2 \in S(\lambda x_1 + (1 - \lambda)x_2). \tag{24}$$

Lemma 2.16. *Let $S_1, S_2 : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ be two convex multifunctions. Then the multifunction $S : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ defined by*

$$S(x) = S_1(x) \cap S_2(x) \tag{25}$$

is convex.

Proof. We need to show $\operatorname{gph}(S_1 \cap S_2) = (\operatorname{gph} S_1) \cap (\operatorname{gph} S_2)$.

$$\begin{aligned}(x, u) \in \operatorname{gph}(S_1 \cap S_2) &\iff u \in S_1(x),\ u \in S_2(x) \\ &\iff (x, u) \in \operatorname{gph} S_1,\ (x, u) \in \operatorname{gph} S_2 \\ &\iff (x, u) \in (\operatorname{gph} S_1) \cap (\operatorname{gph} S_2)\end{aligned} \tag{26}$$

We have that $\operatorname{gph} S_1 \cap (C \times \mathbb{R}^m)$ and $\operatorname{gph} S_2 \cap (C \times \mathbb{R}^m)$ are both convex sets and therefore

$$\operatorname{gph}(S_1 \cap S_2) \cap (C \times \mathbb{R}^m) = (\operatorname{gph} S_1) \cap (\operatorname{gph} S_2) \cap (C \times \mathbb{R}^m) \tag{27}$$

is a convex set. □

In the next proposition we will see that convexity of a multifunction $S(x)$ induces convexity of its indicator function $\mathbb{I}_{S(x)}(y)$.

Proposition 2.17. *If $S : X \rightrightarrows \mathbb{R}^m$ is a convex multifunction, then the indicator function $(x, y) \mapsto \mathbb{I}_{S(x)}(y)$ is jointly convex in (x, y).*

Proof. By definition $\mathbb{I}_{S(x)}(y)$ is convex in (x, y), iff for $x_1, x_2 \in X,\ y_1, y_2 \in Y$ and $\lambda \in (0, 1)$

$$\mathbb{I}_{S(\lambda x_1 + (1-\lambda)x_2)}(\lambda y_1 + (1 - \lambda)y_2) \leq \lambda \mathbb{I}_{S(x_1)}(y_1) + (1 - \lambda)\mathbb{I}_{S(x_2)}(y_2) \tag{28}$$

which holds in any of the following three cases

1. $\lambda y_1 + (1 - \lambda)y_2 \in S(\lambda x_1 + (1 - \lambda)x_2)$ (then LHS is zero)
2. $y_1 \notin S(x_1)$ (then RHS is infinity)
3. $y_2 \notin S(x_2)$ (then RHS is infinity).

Assume now that S is a convex multifunction and $y_1 \in S(x_1),\ y_2 \in S(x_2)$. By the definition of a convex multifunction for $\lambda \in (0, 1)$,

$$\lambda y_1 + (1 - \lambda)y_2 \in S(\lambda x_1 + (1 - \lambda)x_2). \tag{29}$$

and therefore 1. is satisfied. If on the other hand either $y_1 \notin S(x_1)$ or $y_2 \notin S(x_2)$ 2. or 3. is satisfied. That is in any case, if S is a convex multifunction, $\mathbb{I}_{S(x)}(y)$ is convex in (x, y). Assume that S is not convex, that is there are points $x_1, x_2 \in X$ and $y_1 \in S(x_1),\ y_2 \in S(x_2)$ and a $\lambda \in (0, 1)$ such that

$$\lambda y_1 + (1 - \lambda)y_2 \notin S(\lambda x_1 + (1 - \lambda)x_2). \tag{30}$$

Then for these combination of points x_1, x_2, y_1, y_2 and parameter λ, the LHS of (28) is ∞ but the right hand side is zero, therefore $\mathbb{I}_{S(\cdot)}(\cdot)$ is not convex. □

2.3 Measure Theory

In Multistage stochastic optimization we deal with problems of the form

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f(x) + \mathbb{E}_\xi[G(x,\xi)] \\ \text{s.t.} \quad & G(x,\xi) = \min_{y \in \mathcal{Y}(x,\xi)} g(y,x,\xi), \end{aligned} \tag{31}$$

we are interested in the interchangeability of the expectation operator and the minimization operator. Let Ω be an arbitrary set and $\mathcal{A}$ a σ-field on Ω. For a set valued mapping $S : \Omega \rightrightarrows \mathbb{R}^n$, the preimage of a set $A \subset \mathbb{R}^n$ is defined as

$$S^{-1}(A) = \bigcup_{x \in A} S^{-1}(x) = \{\omega \in \Omega \mid S(\omega) \cap A \neq \emptyset\}$$

For the expectation of the value function (36)-(37) to be well-defined we first have to ensure that the value function is measurable. We will use some theory which we take from Rockafellar and Wets (1998) and Shapiro et al. (2021). We begin with the basic notion of measurability of set-valued functions.

Definition 2.18 (Measurability)**.** *A set valued mapping $S : \Omega \rightrightarrows \mathbb{R}^n$ is called measurable if for all open sets $O \subset \mathbb{R}^n$ the preimage is measurable, that is $S^{-1}(O) \in \mathcal{A}$.*

Most interesting features of measurable set valued mappings require closed-valuedness. Therefore we will restrict ourselves to closed valued-mappings, which is not much of a restriction in practical applications, since there most of the time we have closed-valuedness (see Rockafellar and Wets (1998)). Furthermore we have the following nice property which gives us that

Proposition 2.19 (Proposition 14.2 in Rockafellar and Wets (1998))**.** *For $S : \Omega \rightrightarrows \mathbb{R}^n$ we have that*

$$\omega \mapsto S(\omega) \text{ measurable} \implies \omega \mapsto \operatorname{cl} S(\omega) \text{ measurable}$$

For a function $f : \mathbb{R}^n \times \Omega \to \overline{\mathbb{R}}$ we define the epigraph mapping as

$$S_f(\omega) := \operatorname{epi} f(\cdot,\omega) := \{(x,\alpha) \in \mathbb{R}^n \times \mathbb{R} : \ f(x,\omega) \leq \alpha\}$$

Definition 2.20 (normal integrand)**.** *A function $f : \mathbb{R}^n \times \Omega \to \overline{\mathbb{R}}$ is called a normal integrand if its epigraph mapping $S_f(\cdot)$ is a closed valued and measurable multifunction.*

Remark 2.21. *In the context of stochastic programming, normal integrands are also called* random lower semicontinuous *functions.*

There is a slightly smaller class of functions which are normal integrands, so called *Carathéodory functions*, which are the main class of functions, sotchastic optimization deals with.

Definition 2.22 (Carathéodory function)**.** *A single-valued mapping $f : \mathbb{R}^n \times \Omega \to \mathbb{R}^m$ is called a Carathéodory function if the following two conditions hold:*

1. *For fixed $x \in \mathbb{R}^n$, $f(x,\cdot)$ is measurable*

2. For a.e. fixed $\omega \in \Omega$, $f(\cdot,\omega)$ is continuous.

Theorem 2.23 (Theorem 14.37 in Rockafellar and Wets (1998))**.** *Let $f : \Omega \times \mathbb{R}^n \to \overline{\mathbb{R}}$ be a normal integrand. Define*

$$p(\omega) = \inf\ f(\cdot,\omega), \quad P(\omega) = \text{argmin}\ f(\cdot,\omega)$$

Then the function $p : \Omega \to \mathbb{R}$ is measurable and the set valued mapping $P : \Omega \rightrightarrows \mathbb{R}^n$ is closed valued and measurable.
In particular, therefore the set $A = \{\omega \mid \text{argmin}_x f(x,\omega) \neq \emptyset\} \subset \Omega$ is measurable, and it is therefore possible for each $\omega \in A$ to select a minimizing point $x(\omega)$ in such a manner, that the function $\omega \mapsto x(\omega)$ is measurable.

This gives us the measurability of value functions.

Definition 2.24 (Measurable selection)**.** *A measurable selection of a set valued mapping $S : \Omega \rightrightarrows \mathbb{R}^n$ is a measurable function $x : \text{dom}\, S \to \mathbb{R}^n$ such that for all $\omega \in \text{dom}\, S$, $x(\omega) \in S(\omega)$.*

There is an important interconnection between measurability of set valued functions and the existence of measurable selections.

Theorem 2.25 (Measurable Selection Theorem)**.** *Let $S : \Omega \rightrightarrows \mathbb{R}^n$ be a closed-valued mapping. Then, S is measurable, iff* dom *$S \subset \Omega$ is measurable and there exists a countable family $\{x^\nu\}_{\nu\in\mathbb{N}}$ of measurable selections $x^\nu : \text{dom}\, S \to \mathbb{R}^n$, such that for every $\omega \in \Omega$, the set $\{x^\nu(\omega) : \nu \in \mathbb{N}\}$ is dense in $S(\omega)$.*

Proof. First assume that there is a countable family of measurable selections $\{x^\nu\}$ of S. Let $\mathcal{O} \subset \mathbb{R}^n$ be an open set. We need to show that $S^{-1}(\mathcal{O})$ is measurable. We show that

$$S^{-1}(\mathcal{O}) = \bigcup_{\nu\in\mathbb{N}} \{\omega \in \Omega : x^\nu(\omega) \in \mathcal{O}\}\,.$$

Therefore let $\omega \in S^{-1}(\mathcal{O}) = \{\omega \in \Omega :\ S(\omega \cap \mathcal{O} \neq \emptyset)\}$, which implies $S(\omega) \cap \mathcal{O} \neq \emptyset$. Let $a \in S(\omega) \cap \mathcal{O} \subset \mathbb{R}^n$. Then $\exists$ ε-neighborhood $B_\varepsilon(a) \subset \mathcal{O}$ of a in $\mathcal{O}$. But also, since $\{x^\nu(\omega)\}_{\nu\in\mathbb{N}}$ is dense in $S(\omega), \exists \nu_0 \in \mathbb{N}$ s.t. $x^{\nu_0}(\omega) \in B_\varepsilon(a) \subset \mathcal{O}$. Therefore $x^{\nu_0}(\omega) \in \mathcal{O}$ and thus $\omega \in \{\omega \in \Omega : x^{\nu_0}(\omega) \in \mathcal{O}\}$, which gives in total

$$S^{-1}(\mathcal{O}) \subseteq \bigcup_{\nu\in\mathbb{N}} \{\omega \in \Omega : x^\nu(\omega) \in \mathcal{O}\}$$

Now let $\omega \in \bigcup_{\nu\in\mathbb{N}} \{\omega \in \Omega : x^\nu(\omega) \in \mathcal{O}\}$. Then there exists $\nu_0 \in \mathbb{N}$ such that $x^{\nu_0}(\omega) \in \mathcal{O}$. But since $x^{\nu_0}(\omega) \in S(\omega)$, we have that $S(\omega) \cap \mathcal{O} \neq \emptyset$, which gives us together with the above inclusion that $S^1(\mathcal{O})$ is the countable union of measurable sets and therefore measurable.
For the reverse implication that every measurable set-valued mapping S has a measurable selection, see Theorem 14.5 in Rockafellar and Wets (1998) □

Lemma 2.26 (Corollary 14.6 in Rockafellar and Wets (1998))**.** *Every closed valued measurable mapping $S : \Omega \rightrightarrows \mathbb{R}^n$ has a measurable selection.*

2.4 Interchangeability principles

We need one last definition about function spaces:

Definition 2.27 (Decomposable space). *A linear space $\mathcal{X}$ of measurable functions $x : \Omega \to \mathbb{R}^n$ is decomposable w.r.t a measure μ on $\mathcal{A}$ if for every function $x_0 \in \mathcal{X}$, every set $A \in \mathcal{A}$ with $\mu(A) < \infty$ and bounded, measurable function $x_1 : A \to \mathbb{R}^n$, the function $x : \Omega \to \mathbb{R}^n$ defined as*

$$x(\omega) := x_0(\omega)\mathbf{1}_{\Omega\setminus A}(\omega) + x_1(\omega)\mathbf{1}_A(\omega)$$

is also an element of $\mathcal{X}$.

Probably the most important examples of a decomposable space are the Lebesque spaces $L^p(\Omega, \mathcal{A}, \mu)$ for $p \in [1, \infty]$. To be decomposable, a linear space $\mathcal{X}$ must in particular contain every bounded measurable function that vanishes outside some set of finite measure. (Rockafellar and Wets (1998)). We now clarified all important notions to state the interchangeability principle.

Theorem 2.28 (Theorem 9.108 in Shapiro et al. (2021)). *Let $\mathcal{X}$ be a μ-decomposable space of measurable functions $\chi : \Omega \to \mathbb{R}^n$ for a σ-finite measure μ on $\mathcal{A}$. Let $f : \mathbb{R}^n \times \Omega \to \overline{\mathbb{R}}$ be a normal integrand. Then, as long as $\mathbb{E}\left[f(\chi(\omega), \omega)\right] < \infty$ for some $\chi \in \mathcal{X}$,*

$$\mathbb{E}\left[\inf_{x\in\mathbb{R}^n} f(x, \omega)\right] = \inf_{\chi\in\mathcal{X}} \mathbb{E}\left[f(\chi(\omega), \omega)\right] \tag{32}$$

Moreover, if the common value is not $-\infty$, one has for $\overline{\chi} \in \mathcal{X}$ that

$$\overline{\chi} \in \underset{\chi\in\mathcal{X}}{\operatorname{argmin}}\ \mathbb{E}\left[f(\chi(\omega), \omega)\right] \iff \overline{\chi}(\omega) \in \underset{x\in\mathbb{R}^n}{\operatorname{argmin}}\ f(x, \omega) \text{ for a.e. } \omega \in \Omega \text{ and } \overline{\chi} \in \mathcal{X}$$

2.5 Multistage stochastic optimization models

The general T-stage stochastic optimization problem can be formulated as

$$\operatorname{Min}_{x_1\in\mathcal{X}_1} f_1(x_1) + \mathbb{E}_{|\xi_1}\left[\inf_{x_2\in\mathcal{X}_2(x_1,\xi_2)}\left\{f_2(x_2,\xi_2) + \cdots + \mathbb{E}_{|\xi_{[T-1]}}\left[\inf_{x_T\in\mathcal{X}_T(x_{T-1},\xi_T)} f_T(x_T,\xi_T)\right]\right\}\right], \tag{33}$$

where $f_t : \mathbb{R}^{n_t} \times \mathbb{R}^{d_t} \to \bar{\mathbb{R}}$, $t = 1, ..., T \in \mathbb{N}$ are the objective functions, $\xi(\omega) = (\xi_1(\omega), ..., \xi_T(\omega))$ is a random process on the probability space $(\Omega, \mathcal{A}, \mathbb{P})$ and ξ_1 is deterministic[1] and $\mathcal{X}_t : \mathbb{R}^{n_{t-1}} \times \mathbb{R}^{d_t} \rightrightarrows \mathbb{R}^{n_t}$, $t + 2, ..., T$ is a measurable closed-valued multifunction. The first-stage feasible set is a deterministic set $\mathcal{X}_1$. Even this problem can be furher generalized. We could for example replace the expectation operator with a general risk measure $\mathcal{R}$ (see e.g. Shapiro et al. (2021), ch. 6) or allow for an infinite time horizon. In this thesis we want to deal with multistage stochastic optimization problems which satisfy the following assumptions

1. The time horizon T is finite and deterministic.
2. The probability distribution of the random process $(\xi)_{t\in[\![T]\!]}$ is known.

[1] we write $\xi_{[t]} := (\xi_1, ..., \xi_t)$ for the whole history of the process up to time t

3. The randomness is exogenous. That is the random variables $(\xi_t)_{t \in [\![T]\!]}$ are independent from the decision variables $(x_t)_{t \in [\![T]\!]}$.

Remark 2.29. *Note that at this point we have neither made some assumptions on the dependence of the random variables ξ_t and $\xi_{[t-1]}$ nor on the convexity of the stage-cost-functions f_t. Both are crucial properties which we will discuss later and make some assumptions there.*

Definition 2.30 (Policy). *A* policy *(also called* decision rule*) for problem (33) is a sequence $\{x_1, \mathbf{x}_2(\xi_{[2]}), ..., \mathbf{x}_T(\xi_{[T]})\}$ of measuarble mappings $\mathbf{x}_t : \mathbb{R}^{d_1} \times ... \times \mathbb{R}^{d_t} \to \mathbb{R}^{n_t}$.*

Since we have to make a decision x_t *after* we observed the realization of the randomness ξ_t, we only can state a solution $x_t = \mathbf{x}_t$ as a function of $\xi_{[t]}$ and therefore we deal with an infinite dimensional optimization problem where we choose $\mathbf{x}_t$ from an appropriate function-space.

Remark 2.31. *If our probability space is discrete, i.e. on stage t there are only $K < \infty$ possible outcomes of $\xi_t = \xi_t(\omega_k)$, $k = 1, ..., K$, we can index our solution as $\mathbf{x}_t(\xi_t(\omega_k)) := x_t^k$ and the optimization is performed over vectors $\mathbf{x}_t \in \mathbb{R}^{n_t K}$ in the finite dimensional space $\mathbb{R}^{n_t K}$.*

Definition 2.32 (Feasible policy). *A policy $\mathbf{x}$ is called feasible, if $x_1 \in \mathcal{X}_1$ and for $t = 2, ..., T$*

$$\mathbf{x}_t(\xi_{[t]}) \in \mathcal{X}(\mathbf{x}_{t-1}(\xi_{[t-1]}), \xi_t) \quad \text{w.p.1.} \tag{34}$$

With the tower property of the expected-value operator $\mathbb{E}\left[\mathbb{E}_{|Y}[X]\right] = \mathbb{E}[X]$, we can rewrite (33) as

$$\begin{aligned} \min_{x_1, \mathbf{x_2}, ..., \mathbf{x}_T} \quad & \mathbb{E}\left[f_1(x_1) + f_2(\mathbf{x}(\xi_{[2]}), \xi_2) + \cdots + f_T(\mathbf{x}_T(\xi_{[T]}), \xi_T)\right] \\ s.t. \quad & x_1 \in \mathcal{X}_1,\ \mathbf{x}_t(\xi_{[t]}) \in \mathcal{X}_t(\mathbf{x}_{t-1}(\xi_{[t-1]}), \xi_t),\ t = 2, ...T \end{aligned} \tag{35}$$

The interchangeability principle Theorem 2.28 allows us to reformulate problem (33) in terms of so called *value-functions*. This leads to the dynamic-programming formulation of our multistage stochastic optimization problem. Therefore for the last stage T define the value-function

$$V_T(x_{T-1}, \xi_T) = \min_{x_T \in \mathcal{X}(x_{T-1}, \xi_T)} f_T(x_T, \xi_T) \tag{36}$$

and recursively for stages $t = 2, ..., T-1$

$$\begin{aligned} V_t(x_{t-1}, \xi_{[t]}) &= \min_{x_t \in \mathcal{X}(x_{t-1}, \xi_t)} \left\{ f_t(x_t, \xi_t) + \mathbb{E}_{|\xi_{[t]}}\left[V_{t+1}(x_t, \xi_{[t+1]})\right] \right\} \\ &=: \min_{x_t \in \mathcal{X}(x_{t-1}, \xi_t)} \left\{ f_t(x_t, \xi_t) + v_{t+1}(x_t, \xi_{[t]}) \right\}, \end{aligned} \tag{37}$$

where we definded the expected value-functions $v_{t+1}(x_t, \xi_{[t]})$. In particular at the first stage, where ξ_1 is deterministic we solve the deterministic finite-dimensional problem

$$\min_{x_1 \in \mathcal{X}_1} f_1(x_1) + v_2(x_1). \tag{38}$$

Now if we could evaluate the value function $v_2(x_1)$ at every x_1 we could simply solve (38) as a standard optimization problem. Unfortunately we rarely have an analytical expression for v. Nevertheless, under certain regularity conditions, we can do some calculus on value functions. We will see some important properties of value functions in the next chapter.

2.6 Analytical Properties of Value Functions

In optimization we are mostly interested in convexity of a function and how to calculate (sub-gradients). We strongly follow Chapters 2 and 9 of Shapiro et al. (2021). Let $\psi : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R} \cup \{\infty\}$ be convex and extended-real-valued. Define the value function

$$V(s) := \inf_{x \in \mathbb{R}^n} \ \psi(x, s) \tag{39}$$

Note that if $\operatorname{dom} \psi(\cdot, s) = \emptyset$ at $s \in \mathbb{R}^p$, we have $V(s) = \infty$. The proof of the next theorem is taken from Fiacco and Kyparisis (1986).

Theorem 2.33 (Convexity of the optimal-value function)**.** *Let first $\psi : X \times S \to \overline{\mathbb{R}}$ be a jointly convex function on the convex sets $X \subset \mathbb{R}^n$, $S \times \mathbb{R}^p$. The optimal-value function $V : S \to \mathbb{R} \cup \{-\infty, \infty\}$, $V(s) := \inf_{s \in S} \psi(x, s)$ is convex.*

Proof. For $s_1, s_2 \in S$ and $\lambda \in (0, 1)$, we have

$$\begin{aligned} V(\lambda s_1 + (1-\lambda)s_2) &= \inf_{x \in X} \ \psi(\lambda s_1 + (1-\lambda)s_2, x) \\ &= \inf_{x_1, x_2 \in X} \psi(\lambda s_1 + (1-\lambda)s_2, \lambda x_1 + (1-\lambda)x_2) \\ &\leq \inf_{x_1, x_2 \in X} \ \{\lambda \psi(s_1, x_1) + (1-\lambda)\psi(s_2, x_2)\} \\ &= \inf_{x_1 \in X} \lambda \psi(s_1, x_1) + \inf_{x_2 \in X} (1-\lambda)\psi(s_2, x_2) \\ &= \lambda V(s_1) + (1-\lambda)V(s_2) \end{aligned} \tag{40}$$

□

Consider now the two-stage problem with first stage

$$\min_{x \in \mathcal{X}} f(x) + \mathbb{E}_\xi \left[V(x, \xi)\right] \tag{41}$$

where $V(x, \xi)$ is the value-function of the second stage problem

$$V(x, \xi) = \min_{y \in \mathcal{G}(x,y,\xi)} g(x, y, \xi). \tag{42}$$

We now want to study conditions under which the extended-real-valued function $\bar{g}_\xi(x, y) = \bar{g}(x, y, \xi) = g(x, y) + \mathbb{I}_{\mathcal{G}(x,\xi)}(y)$ for convex g remains convex in x. We know from Proposition 2.17 that if $x \mapsto \mathcal{G}_\xi(x)$ is a convex multifunction (Definition 2.15). It remains to investigate, which multifunctions are convex. In multistage stochastic optimization, often the stages are connected via some recourse function of the form

$$\mathcal{G}_\xi(x) := \{y \in Y : \ T(x, \xi) + W(y, \xi) \in -C\}, \tag{43}$$

where the set $Y \subset \mathbb{R}^m$ is closed and convex, $T = (t_1, ..., t_l) : \mathbb{R}^n \times \xi \to \mathbb{R}^l$, $W = (w_1, ..., w_l) : \mathbb{R}^m \times \xi \to \mathbb{R}^l$ and C is a closed convex cone. In the next proposition we will see, that a cone $C \subset \mathbb{R}^m$ defines a partial order on $\mathbb{R}^m$.

Proposition 2.34 (Partial Order induced by a Cone)**.** *Let $C \subset \mathbb{R}^m$ be a closed convex cone. Then by defining*

$$a \preceq_C b \iff b - a \in C \tag{44}$$

we have a partial order "$\preceq_C$" on $\mathbb{R}^m$, that is for $x, y, z \in \mathbb{R}^m$

$$
\begin{aligned}
\textit{(Antisymmetry) } x \preceq_C y,\ y \preceq_C x &\Longleftrightarrow\ x = y \\
\textit{(Reflexivity) } x &\preceq_C x \\
\textit{(Transitivity) } x \preceq_C y,\ y \preceq_C z &\ \Longrightarrow x \preceq_C z.
\end{aligned}
\tag{45}
$$

Furhtermore we have for $\alpha \in \mathbb{R}$ the scaling

$$
\begin{aligned}
\textit{If } \alpha > 0,\ x \preceq_C y &\ \Longrightarrow \alpha x \preceq_C \alpha y \\
\textit{and if } \alpha < 0,\ x \preceq_C y &\ \Longrightarrow\ \alpha x \succeq_C \alpha y
\end{aligned}
\tag{46}
$$

Proof. We skip the proof here. □

That means we can write $T(x) + W(y) \in -C$ equivalently as $T(x) + W(y) \preceq_C 0$

Example 2.35. *1. If $C := \mathbb{R}^l_+$, the expression $T_\xi(x) + W_\xi(y) \preceq_C 0$ means $t_i(x, \xi) + w_i(y, \xi) \leq 0,\ i = 1, ..., l$.*

2. If $C := \{0\}$, the recourse-constraints read $t_i(x, \xi) + w_i(y, \xi) = 0,\ i = 1, ..., l$

Therefore in this notation we can incapsulate convex optiimization problems with equality and inequality constraints.

Definition 2.36 (Convexity w.r.t. Cone)**.** *Let C be a closed convex cone. We say a mapping $G : \mathbb{R}^n \to \mathbb{R}^l$ is* convex with respect to C, *if the multifunction $M(x) := G(x)+C$ is convex or equivalently for $x_1, x_2 \in R^n,\ \lambda \in (0, 1)$*

$$
G(\lambda x_1 + (1 - \lambda)x_2) \preceq_C \lambda G(x_1) + (1 - \lambda)G(x_2) \tag{47}
$$

Now we are able to deduce convexity of $\mathcal{G}_\xi(x)$ from cone-convexity of the multifunctions T_ξ and W_ξ:

Theorem 2.37. *If $C \subset \mathbb{R}^l$ is a closed convex cone and the multifunctions $T_\xi : \mathbb{R}^n \to \mathbb{R}^l$ and $W_\xi : \mathbb{R}^m \to \mathbb{R}^l$ are convex w.r.t C, then the multifunction $\mathcal{G}_\xi : \mathbb{R}^n \rightrightarrows R^m$ defined by*

$$
\mathcal{G}_\xi(x) := \{y \in Y :\ T(x, \xi) + W(y, \xi) \in -C\}, \tag{48}
$$

is convex.

Proof. Fix $\xi \in \xi$ and pairs $x_1 \in \mathbb{R}^n, y_1 \in \mathbb{R}^m$ and $x_2 \in \mathbb{R}^n, y_2 \in \mathbb{R}^m$ such that $y_1 \in \mathcal{G}_\xi(x_1),\ y_2 \in \mathcal{G}_\xi(x_2)$. Then we have

$$
T_\xi(x_1) + W_\xi(y_1) \preceq_C 0,\ T_\xi(x_2) + W_\xi(y_2) \preceq_C 0 \tag{49}
$$

Since we assumed T_ξ and W_ξ to be convex w.r.t C, we have for a $\lambda \in (0, 1)$,

$$
T_\xi(\lambda x_1 + (1 - \lambda)x_2) \preceq_C \lambda T_\xi(x_1) + (1 - \lambda)T(x_2). \tag{50}
$$

and the same with T_ξ, x_2 respective W_ξ, y_1, y_2. Since by definition the cone C is convex iff $C + C \subseteq C$, we also have

$$
\begin{aligned}
T_\xi(\lambda x_1 + (1 - \lambda)x_2) +& W(\lambda y_1 + (1 - \lambda)y_2) \\
&\preceq_C \lambda(T_\xi(x_1) + W_\xi(y_1)) + (1 - \lambda)(W_\xi(y_1) + W_\xi(y_2))
\end{aligned}
\tag{51}
$$

and by the transitivity (45) and scaling (46) of $\preceq_C$ we arrive at

$$T_\xi(\lambda x_1 + (1-\lambda)x_2) + W_\xi(\lambda y_1 + (1-\lambda)y_2) \preceq_C 0 \tag{52}$$

which induces

$$\lambda y_1 + (1-\lambda)y_2 \in \mathcal{G}_\xi(\lambda x_1 + (1-\lambda)x_2), \tag{53}$$

which verifies that $\mathcal{G}_\xi$ is a convex multifunction. □

Example 2.38. *1. If $C := \mathbb{R}^l_+$ and $t_i(\cdot,\xi),\ w_i(\cdot,\xi),\ i = 1,...,l$ are convex functions then T_ξ and W_ξ are convex w.r.t. C and therefore $\mathcal{G}_\xi$ is a convex multifunction*

2. *If $C := \{0\}$ and $t_i(\cdot,\xi),\ w_i(\cdot,\xi),\ i = 1,...,l$ are linear functions, then T_ξ and W_ξ are convex w.r.t. C and therefore $\mathcal{G}_\xi$ is a convex multifunction*

3. *If we encode with*

$$\mathcal{G}_{1,\xi}(x) := \left\{y \in Y :\ T_{1,\xi}(x,) + W_{1,\xi}(y) \in \mathbb{R}^l_-\right\}, \tag{54}$$

inequality recourse constraints and with

$$\mathcal{G}_{2,\xi}(x) := \left\{y \in Y :\ T_{2,\xi}(x,) + W_{2,\xi}(y) \in \mathbb{R}^l_-\right\}, \tag{55}$$

equality recourse constraints, we have a convex constraints-multifunction $\mathcal{G}_\xi(x) = \mathcal{G}_{1,\xi}(x) \cap \mathcal{G}_{2,\xi}(x)$

Lemma 2.39. *The conjugate of V is*

$$V^*(s^*) = \psi^*(0, s^*) \tag{56}$$

Proof.

$$\psi^*(x^*, s^*) := \sup_{(x,s)\in\mathbb{R}^n\times\mathbb{R}^p} \{\langle x^*, x\rangle + \langle s^*, s\rangle - \psi(x,s)\} \tag{57}$$

and therefore

$$\begin{aligned} V^*(s^*) &= \sup_{s\in\mathbb{R}^p} \{\langle s^*, s\rangle - V(s)\} \\ &= \sup_{s\in\mathbb{R}^p} \left\{\langle s^*, s\rangle - \inf_{x\in\mathbb{R}^n} \psi(x,s)\right\} \\ &= \sup_{(x,s)\in\mathbb{R}^n\times\mathbb{R}^p} \{\langle s^*, s\rangle - \psi(x,s)\} = \psi^*(0,s^*) \end{aligned} \tag{58}$$

□

Then by definition

$$V^{**}(s) = \sup_{s^*\in\mathbb{R}^p} \{\langle s^*, s\rangle - \psi^*(0,s^*)\}. \tag{59}$$

We call

$$\sup_{s^*\in\mathbb{R}^p} \{\langle s^*, s\rangle - \psi^*(0,s^*)\} \tag{60}$$

the dual problem to (39). We know from Theorem 2.33 that V is convex. Furthermore we know from Corollary 2.13 that

$$\partial V(s) = \underset{s^*\in\mathbb{R}^p}{\operatorname{argmax}} \{\langle s^*, s\rangle - \psi^*(0,s^*)\} \tag{61}$$

We now want to calculate subgradients of V at $\bar{s}$ for different instances of this problem.

Proposition 2.40. *Consider the following problem:*

$$\begin{aligned} V(s) = \min_{x\in X} & f(x) \\ s.t. \quad & G(x) \preceq_C s \end{aligned} \tag{62}$$

where $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R} \cup \{\infty\}$ is a convex extended-real-valued function, $C \subset \mathbb{R}^m$ is a convex cone and $G : \mathbb{R}^m \to \mathbb{R}^n$ is convex w.r.t the cone C (see Definition 2.36). Then

$$\partial V(s) = \underset{y\in C^*}{\operatorname{argmax}}\ L(x,y) \tag{63}$$

where L is the Lagrangian to (62) and C^* is the dual cone. In light of (61) that means an subgradient of V at s is a dual solution of (39).

Proof. By setting $\bar{f} = f + \mathbb{I}_X$ and $F = \mathbb{I}_C$, we can write $V(s) = \min_{x\in\mathbb{R}^n} \bar{f}(x) + F(s - G(x)) = \min_{x\in\mathbb{R}^n} \psi(x,s)$ and ψ is convex under the above assumptions. First we calculate the conjugate of ψ

$$\begin{aligned} \psi^*(x^*, s^*) &= \sup_{(x,s)\in\mathbb{R}^n\times\mathbb{R}^p} \left\{\langle x^*, x\rangle + \langle s^*, s\rangle - \bar{f}(x) - F(s - G(x))\right\} \\ &= \sup_{x\in\mathbb{R}^n} \left\{\langle x^*, x\rangle - \bar{f}(x) + \langle s^*, G(x)\rangle + \sup_{s\in\mathbb{R}^p} \{\langle s^*, (s - G(x))\rangle - F(s - G(x))\}\right\}. \end{aligned} \tag{64}$$

Now by a change of variables $r = s - G(x)$, which is possible since we perform the supremumg over s in all $\mathbb{R}^p$ we have

$$\sup_{s\in\mathbb{R}^p} \{\langle s^*, (s - G(x))\rangle + F(s - G(x))\} = \sup_{r\in\mathbb{R}^p} \{\langle s^*, r\rangle - F(r)\} = \mathbb{I}_{C^*}(s^*) \tag{65}$$

and hence

$$\psi^*(x^*, s^*) = \sup_{x\in X} \{\langle x^*, x\rangle - f(x) + \langle s^*, G(x)\rangle\} + \mathbb{I}_{C^*}(s^*). \tag{66}$$

Plugging this into the definition of the dual (60) with $x^* = 0$ we have

$$V^{**}(s) = \sup_{s^*\in\mathbb{R}^p} \left\{\langle s^*, s\rangle + \inf_{x\in X} \{f(x) - \langle s^*, G(x)\rangle + \mathbb{I}_{C^*}(s^*)\}\right\} \tag{67}$$

By defining $L(x, s^*) = f(x) + \langle s^*, s - G(x)\rangle$ we get the classical Lagrangian dual problem

$$\sup_{y\in C^*} L(x,y). \tag{68}$$

□

Now we want to generalize the dependence of (62) on s.

Corollary 2.41 (Proposition 2.22 in Shapiro et al. (2021))**.** *Assume the problem reads*

$$\begin{aligned} V(s) = \min_{x\in X} & f(x) \\ s.t. \quad & G(x) \preceq_C -T(s), \end{aligned} \tag{69}$$

where $T : \mathbb{R}^p \to \mathbb{R}^m$ is also convex w.r.t the cone C. Further assume that T is differentiable at s and strong duality for (69) holds. Suppose that

$$0 \in \{T(s) + \nabla T(x)\mathbb{R}^m - \operatorname{dom} V\}. \tag{70}$$

Then

$$\partial V(s) = \nabla T(s)^T \mathcal{D}(s) \tag{71}$$

where $\mathcal{D}(s)$ is the set of Dual solutions to (69).

2.7 Monotone Operator theory

Let us now come back to set-valued mappings. In this section we will employ them in a different context to solve convex optimization problems. In the earlier section we used them to write parametric constraints as values of set-valued mappings. Now we will show that we can transform an optimization problem into the problem of finding an x such that for a set valued mapping T, $0 \in T(x)$ or $0 \in \operatorname{Zer} T$. This problem is called an *inclusion problem* and we call the mapping T in this context a *(set-valued) operator*. Today it is the state-of-the-art to formulate and analyse convex optimization problems within the framework of operator inclusion problems. The bridge between convex optimization and operator inclusions is fermats principle, which is a generalization of the well known first-order condition for extremas in smooth optimization.

Theorem 2.42 (Fermat's principle). *Let $\mathcal{H}$ be a Hilbert space, $f : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ be convex and proper. Then*

$$x \in \operatorname{argmin} f \Longleftrightarrow 0 \in \partial f(x) \tag{72}$$

Again as in the definition 2.3, the subdifferential $\partial f(x)$ of a convex function f at x is a set-valued operator $\partial f : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$. We know want to give a brief overview of operator analysis in order to later formulate the *Alternating direction methods of multipliers* as an instance of an operator-splitting. By shifting convex optimization problems into this more abstract world of operators, the asymptotic convergence proofs of methods such as basic gradient descent become two-liners. A very readable resource on the connection of convex optimization and operator theory is Ryu and Yin (2022). For the more advanced reader, Bauschke and Combettes (2017) present this topic in the more abstract setting of an infinite dimensional Hilbert space. In the remainder of this section we shortly introduce the most important notions and properties of operator analysis. We hereby follow closly Ryu and Yin (2022) and also skip most of the proofs. Again for a detailed introduction we refer to the literature. We start with some basic properties of an operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$. We write $Tx := T(x)$.

Definition 2.43. *We call an operator L-Lipschitz for an $L > 0$, if*

$$||u - v|| \leq L||x - y|| \text{ for all } (x, u), (y, v) \in T \tag{73}$$

Note that if T is Lipschitz, it is single valued, since otherwise we could set $x = y$ on the right-hand side but can choose $u \neq v$ on the left-hand side. We will also write $||Tx - Ty||$ instead of $||u - v||$ $\forall (x, u), (y, v) \in T$ although technically there we take the norm of a set. For the remainder of this section, we will deal with a specific class of operators, so called *monotone* operators.

Definition 2.44. *An operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is called monotone, if*

$$\langle Tx - Ty, x - y\rangle \geq 0 \;\forall x, y \in \mathbb{R}^n. \tag{74}$$

Single valued monotone operators are exactly the non-decreasing functions. A further important property of monotone operators is maximality.

Definition 2.45. *An operator T is maximal monotone, if there is no other monotone operator S, such that $\operatorname{gph} T \subset \operatorname{gph} S$.*

We can further refine the notion of monotonicity by introducting strong monotonicity.

Definition 2.46 (strongly monotone). *An operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is μ-strongly monotone, or μ-coercive, if $\mu > 0$ and*

$$\langle Tx - Ty, x - y \rangle \geq \mu ||x - y||^2. \tag{75}$$

Definition 2.47 (cocoercive). *A single valued operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is β-cocoercive, if T^{-1} is β-coercive. This is equivalent to*

$$\langle Tx - Ty, x - y \rangle \geq \beta ||Tx - Ty||^2 \ \forall x, y \in \mathbb{R}^n \tag{76}$$

By the Cauchy-Schwarz-inequalty we have for a β-cocoercive operator

$$\beta^{-1} ||x - y|| \geq ||Tx - Ty|| \ \forall x, y \in \mathbb{R}^n. \tag{77}$$

There is a direct connection between convexity properties of a fuction f and monotonicity properties of its subdifferential ∂f:

Lemma 2.48. *Assume f is CCP. Then*

1. *f is μ-strongly convex, iff ∂f is μ-strongly monotone*
2. *f is L-smooth, iff ∂f is $(1/L)$-cocoercive*

Definition 2.49 (Nonexpansive, Contraction). *We say an operator T is nonexpansive, if $||Tx - Ty|| \leq ||x - y|| \ \forall x, y \in \operatorname{dom} T$. This is equivalent to saying T is L-Lipschtiz with $L = 1$. If T is L-Lipschitz with $L < 1$, we say T is a contraction*

Proposition 2.50 (Operations preservinge contraction, nonexpansiveness). *For two nonexpansive operators T, S the composition TS is nonexpansive. If at least one of T and S is a contraction, TS is a contraction. Furthermore the convex combination $\theta T + (1 - \theta) S$ with $\theta \in [0, 1]$ of two nonexpansive operators T, S is nonexpansive. And if T is contractive and $\theta > 0$, $\theta T + (1 - \theta) S$ is contractive.*

Definition 2.51 (Averaged operator). *Let S be nonexpansive. The operator $T := (1 - \theta) I + \theta S$, $\theta \in (0, 1)$ is called θ-averaged*

We will later see that we often can find solutions of a monotone inclusion problem for an operator $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ by finding a fixed point of another operator $S : \mathbb{R}^n \to \mathbb{R}^n$, that is a point $x \in \mathbb{R}^n$ satisfying $x = Sx$. This is done by a so called fixed-point iteration or *Picard*-iteration, that is we produce a sequence $\{x^k\}_{k \in \mathbb{N}}$ with $x^{k+1} = Tx^k$. If S is a contraction on $\mathbb{R}^n$, then the classic Banach fixed-point theorem states that S has at least one fixed point and a fixed point iteration converges towards this fixed point (see for example Bauschke and Combettes (2017), Theorem 1.50). Most operators of interest do not posses the nice property of being a contraction. But luckily also fixed-point iterations with averaged operators converge towards a fixed-point, if one exists. Such an iteration is called *Krasnosel'skii-Mann*-iteration.

Theorem 2.52 (Theorem 1 in Ryu and Yin (2022)). *Assume $T : \mathbb{R}^n \to \mathbb{R}^n$ is θ-averaged with $\theta \in (0, 1)$ and T has at least one fixed-point. Then for any starting point $x^0 \in \mathbb{R}^n$, the Picard-iteration $x^{k+1} = Tx^k$ converges to one fixed point, that is*

$$x^k \longrightarrow \bar{x} \tag{78}$$

for some $\bar{x} \in \operatorname{Fix} T$. The quantities $\operatorname{dist}(x^k, \operatorname{Fix} T)$, $||x^{k+1} - x^k||$, *and* $||x^k - \bar{x}||$ *for any* $\bar{x} \in \operatorname{Fix} T$ *are monotonically nonincreasing with k. Finally,*

$$\operatorname{dist}(x^k, \operatorname{Fix} T) \longrightarrow 0 \tag{79}$$

and

$$||x^{k+1} - x^k||^2 \leq \frac{\theta}{(k+1)(1-\theta)} \operatorname{dist}(x^0, \operatorname{Fix} T)^2 \tag{80}$$

Proof. For the proof, see Ryu and Yin (2022), Theorem 1. □

This theorem becomes even more usefull in light of the next statement which shows us, that we can "average" a nonexpansive but not necessarily averaged operator T to find its fixed-points.

Lemma 2.53. *Let T be a nonexpansive operator and choose $\theta \in (0, 1)$. Define the averaged operator $\bar{T} := (1 - \theta)I + \theta T$. Then*

$$\operatorname{Fix} T = \operatorname{Fix} \bar{T} \tag{81}$$

We will now come to some more concrete examples of monotone operators. Let A be an operator. The *resolvent* of A is the operator

$$J_A := (I + A)^{-1}. \tag{82}$$

From the resolvent we can define the *reflected resolvent* or *Cayley-operator* of A:

$$R_A := 2J_A - I. \tag{83}$$

If A is maximal monotone, J_A is (1/2)-averaged with $\operatorname{dom} J_A = \mathbb{R}^n$ and R_A is nonexpansive single valued with $\operatorname{dom} R_A = \mathbb{R}^n$. The next theorem connects inclusion problems with fixed-point problems.

Theorem 2.54. *Let A be maximal monotone. Then*

$$0 \in Ax \iff x \in x + Ax \iff J_A x = x. \tag{84}$$

Another important operator in convex analysis is the *proximal operator* of a CCP-function f. For $\alpha > 0$ define

$$\operatorname{Prox}_{\alpha f}(z) := \operatorname*{argmin}_{x \in X} \{\alpha f(x) + \tfrac{1}{2}||x - z||^2\} \tag{85}$$

The proximal operator is a generalization of the projection operator. Consider a convex set $C \subset \mathbb{R}^n$. Then $\Pi_C(z) = \operatorname*{argmin}_{x \in C} \{1/2||x - z||^2\} = \operatorname{Prox}_{\mathbb{I}_C}(z)$. For many functions f, the evaluation of the proximal operator $\operatorname{Prox}_{\alpha f}$ is quite cheap, in some cases it even can be done analytically. A list of proximal operators can be find for example online at Chierchia et al. (2020). We have for f CCP and $\alpha > 0$

$$J_{\alpha \partial f} = \operatorname{Prox}_{\alpha f} \tag{86}$$

which allows us to evaluate the resolvent and therefore also the reflected resolvent. We also get a formula for the prox-operator of a conjugate function.

$$\begin{aligned} v = \operatorname{Prox}_{\alpha f^*(A^T \cdot)}(z) \quad \iff \quad & v = z - \alpha A x \\ & x \in \operatorname*{argmin}_{x} \{f(x) - \langle z, Ax \rangle + \tfrac{\alpha}{2}||Ax||^2\} \end{aligned} \tag{87}$$

For the reflected resolvent we have the identity

$$R_{\alpha A}(I + \alpha A) = I - \alpha A. \tag{88}$$

2.8 Douglas-Rachford-Splitting

We now want to present one famous operator inclusion problem and its connection to convex optimization, the so called *Douglas-Rachford-splitting*. The aim is to solve

$$\operatorname*{find}_{x\in\mathbb{R}^n} 0 \in (A+B)x \tag{89}$$

with maximal monotone operators A and B. We show how to reforumalte this as a fixed-point problem. Therefore observe, that for any $\alpha > 0$, we have

$$\begin{aligned} 0 \in (A+B)x &\Leftrightarrow 0 \in (I+\alpha A)x - (I-\alpha B)x \\ &\Leftrightarrow 0 \in (I+\alpha A)x - R_{\alpha B}(I+\alpha B)x \\ &\Leftrightarrow 0 \in (I+\alpha A)x - R_{\alpha B}z,\ z \in (I+\alpha B)x \\ &\Leftrightarrow R_{\alpha B}z \in (I+\alpha A)J_{\alpha B}z,\ x = J_{\alpha B}z \\ &\Leftrightarrow J_{\alpha A}R_{\alpha B}z = J_{\alpha B}z,\ x = J_{\alpha B}z \\ &\Leftrightarrow R_{\alpha A}R_{\alpha B}z = z,\ x = J_{\alpha B}z \end{aligned} \tag{90}$$

This splitting is called the *Peaceman-Rachford splitting*. The operator $R_{\alpha A}R_{\alpha B}$ is only nonexpansive, so the fixed-point iteration

$$z^{k+1} = R_{\alpha A}R_{\alpha B}(z^k) \tag{91}$$

is not guaranteed to converge. Therefore we average with $1/2$. We remain with

$$0 \in (A+B)x \iff \left(\tfrac{1}{2}I + \tfrac{1}{2}R_{\alpha A}R_{\alpha B}\right)(z) = z, \quad x = J_{\alpha B}(z). \tag{92}$$

This splitting is now called *Douglas-Rachford splitting* (DRS). The fixed-point-iteration with DRS reads

$$\begin{aligned} x^{k+1/2} &= J_{\alpha A}(z^k) \\ x^{k+1} &= J_{\alpha A}(2x^{k+1/2} - z^k) \\ z^{k+1} &= z^k + x^{k+1} - x^{k+1/2}. \end{aligned} \tag{93}$$

By Theorem 2.52 it converges for any $\alpha > 0$ towards a zero of $A+B$ if $\operatorname{Zer}(A+B) \neq \emptyset$.

3 Multistage Stochastic Optimization Methods

In this section we investigate methods to solve multistage stochastic optimization problems of the form

$$\min_{\substack{A_1x_1-b_1\in K_1\\ x_1\in X_1}} f_1(x_1,c_1)+\mathbb{E}\left[\min_{\substack{A_2x_2-B_2x_1-b_2\in K_2\\ x_2\in X_2}} f_2(x_2,c_2)+\cdots+\mathbb{E}\left[\min_{\substack{A_Tx_T-B_Tx_{T-1}-b_T\in K_T\\ x_T\in X_T}} f_T(x_T,c_T)\right]\right] \tag{94}$$

(Shapiro et al., 2021) with decisions $\{x_1,\mathbf{x}_2(\xi_{[2]}),...,\mathbf{x}_T(\xi_{[T]})\}$ and uncertainty $\xi_t(\omega)=(c_t,A_t,B_t,b_t)$ with $c_t\in\mathbb{R}^{n_{c,t}}$, $A_t\in\mathbb{R}^{l_t\times n_t}$, $B_t\in\mathbb{R}^{l_t\times n_{t-1}}$ and $b_t\in\mathbb{R}^{l_t}$ and convex cones $K_t,\ t=1,...,T$. The first stage data c_1,A_1,b_1 is deterministic. Note that here we have replaced the general recourse constraint $x_t\in\mathcal{X}(x_{t-1},\xi)$ with the linear-affine recourse $A_tx_t-B_tx_{t-1}-b_t\in K_t$. First we will give an introduction into stochastic dual dynamic programming, which is widely used to for MSSPs. Then we will present another viewpoint of multistage stochastic optimization methods, which is from the perspective of recursive saddle-point problems, which makes MSSPs amenable to first-order methods from static convex optimization.

3.1 Stochastic Dual Dynamic Programming

In this section we want to give a short overview *Stochastic Dual Dynamic Programming.* Since SDDP is not the type of method investigated in this thesis but is often used as a benchmark to test new methods against, we see it important to present it here but will refer to the classical literature for details. We will give a short overview of some important work at the end of this section. For an excellent overview of SDDP and all its properties and extensions, see Füllner and Rebennack (2023). If not stated otherwise, we rely in this section on the presentation of SDDP therein and in Shapiro et al. (2021). For SDDP to work properly we make additional assumptions to the ones on (94)

1. The objective functions $f_t(\cdot,\cdot)$ are random l.s.c. and $f_t(\cdot,c_t)$ are convex for a.e. ξ_t .
2. The random variables ξ_t have discrete support Ξ_t with $|\Xi_t|=N_t$. Denote $\mathcal{N}=\Pi_{t=2}^T N_t$
3. The random variables ξ_t are stagewise independent, that is ξ_t is independent of ξ_s whenever $s\neq t$.
4. The feasible sets $X_t\subset\mathbb{R}^{n_t},\ t=1,...,T$ are closed and convex.
5. The cones are trivial $K_t=\{0\},\ t=1,...,T$
6. The problem has relatively complete recourse.

We therefore have a convex multistage stochastic program with linear dynamics. Assumpions 2. and 3. are very restricitve since in real world applications most of the time we face continous unvertainty with at least markovian stagewise dependency. In case of a continously supported random process, we discretize the support and draw samples from this discretization. This approximate model of our true optimziation model is the so called *Sample-Average-Approximation (SAA)* and is not only the setup for SDDP but a general technique in Stochastic Programming (cf. Shapiro et al. (2021), ch. 5). For the moment we assume that we have stagewise independent random variables and the SAA-model of a multistage stochastic program. As in (36)-(36) we can write (94) as a sequence of dynamic programming equations with the t-stage value function

$$\begin{aligned} V_t(x_{t-1}, \xi_t) &= \min_{x_t \in X_t} \{ f_t(x_t, \xi_t) + v_{t+1}(x_t) : \quad A_t x_t - B_t x_{t-1} = b_t \}, \quad t = 1, ..., T, \\ v_t(x_{t-1}) &= \mathbb{E}_{\xi_t} [V_t(x_{t-1}, \xi_t)], \quad t = 1, ..., T, \\ v_{T+1} &\equiv 0. \end{aligned} \tag{95}$$

We know from Theorem 2.33 that under the above assumptions the cost-to-go-function $V_t(\cdot, \xi_t)$ is convex and therefore also $v_t(\cdot) = \mathbb{E}[V_t(\cdot, \xi_t)]$ is convex. The main idea of the SDDP algorithm is to approximate the convex $v_t(\cdot)$ with the maximum of a collection of affine cutting planes $l^i(x) = \alpha + \beta^\intercal x,\ i = 1, ..., N$. This idea was first introduced by Kelley (1960) and is therefore called *Kelley's cutting plane method.*
We denote such a polyhedral outer approximation of $v_t(\cdot)$ as $\underline{\mathcal{V}}_t(\cdot)$, and given an approximation $\underline{\mathcal{V}}_t(\cdot)$, we denote

$$\underline{V}_t(x_{t-1}, \xi_t) := \min_{x_t \in X_t} \left\{ f_t(x_t, c_t) + \underline{\mathcal{V}}_{t+1}(x_t) : \ A_t x_t - B_t x_{t-1} = b_t \right\} \tag{96}$$

and correspondingly

$$\underline{v}_t(x_{t-1}) = \mathbb{E}_{\xi_t} [\underline{V}_t(x_{t-1}, \xi_t)].$$

We will now state in detail the procedure of one iteration i of SDDP. In each iteration i of SDDP. To focus on the main principles of the algorithm, we omit the index i of the current iteration here. We first pick a set of feasible points (candidate points) $\bar{x}_1, ..., \bar{x}_T$. We call this the forward pass. Then at each stage, beginning with the last one, we calculate a cutting plane approximation of the value function at the candidate point. We refer to this as the *backward pass*. First assume we have a set of candidate points $\bar{x}_1, ..., \bar{x}_T$ and present one backward pass.

We start with the problem at last stage $t = T$. For a given realization ξ_T^j of ξ_T and a candidate point $\bar{x}_{T-1}$ we have

$$V_T(\bar{x}_{T-1}, \xi_T^j) = \min_{x \in X_T} \left\{ f_T(x, c_T^j),\ A_T^j x - B_T^j \bar{x}_{T-1} = b_T^j \right\}, \tag{97}$$

which is a convex optimization problem.. Assume that total duality holds for (97). We know from section 2.6 that we obtain a subgradient of (97) by solving its dual problem. Then by Corollary 2.41 with the dual solution π_T^j we get a subgradient $g_T^j = B_T^{j\intercal} \pi_T^j$ of $V_T(\cdot, \xi_T^j)$ at $\bar{x}_{T-1}$. We calculate the expectation of the subgradients $g_T := \mathbb{E}\left[g_T^j\right]$ which gives us a new cut for v_T:

$$v_T(x) \geq l_T(x) := v_T(\bar{x}_{T-1}) + \langle g_T, x - \bar{x}_{T-1} \rangle \tag{98}$$

Given a set of candidate points $\bar{x}_{T-1}^k,\ k = 1, ..., K$, and cutting planes $l_T^k(x) := v_T(\bar{x}_{T-1}^k) + \langle g_T^k, x - \bar{x}_{T-1}^k \rangle$ we calculate a polyhedral approximation as

$$\underline{\mathcal{V}}_T(x) := \max_{1 \leq k \leq K} \left\{ l_T^k(x) \right\}. \tag{99}$$

If we already have a current polyhedral approximation $\underline{\mathcal{V}}_T^{i-1}$ from an earlier backward-pass $i-1$ of SDDP, and compute a new approximation $\underline{\mathcal{V}}_T$, we can update the current approximation by taking the maximum of both approximations:

$$\underline{\mathcal{V}}_T^i(x) := \max \left\{ \underline{\mathcal{V}}_T^{i-1}(x), \underline{\mathcal{V}}_T(x) \right\}. \tag{100}$$

Now for stage $T-1$ we want to obtain an approximation for $\underline{v}_{T-1}$. Therefore again we need to compute for each realization ξ^i_{T-1} of ξ_{T-1} a subgradient g^j_{T-1} of $\underline{V}_{T-1}(\cdot, \xi^j_{T-1})$. Remember that

$$\underline{V}_t(x_{T-1}, \xi^j_t) := \min_{x \in X_{T-1}} \left\{ f_{T-1}(x, c^j_{T-1}) + \underline{\mathcal{V}}_T(x) : \; B^j_{T-1} x_{T-2} + A^j_{T-1} x = b^j_{T-1} \right\} \tag{101}$$

and $\underline{\mathcal{V}}_T(x)$ is convex, therefore we still have a convex optimization problem. By computing the dual solution π^j_{T-1} of (101) we obtain a subgradient $g^j_{T-1} = -B^j_{T-1}\pi^j_{T-1}$ of $\underline{V}_{T-1}(\cdot, \xi^j_{T-1})$ at $\bar{x}_{T-2}$. We can now calculate the expectation of the subgradients $g_{T-1} = \mathbb{E}_{\xi_{T-1}}\left[g^j_{T-1}\right]$ and obtain a new cut for the value function at stage $T-1$:

$$\underline{v}_{T-1}(x) \geq l_{T-1}(x) := \underline{v}_{T-1}(\bar{x}_{T-2}) + \langle g_{T-1}, x - \bar{x}_{T-2} \rangle. \tag{102}$$

We proceed this way for $t = T-2, ..., 1$ until we are left with the deterministic first stage problem

$$\min_{x \in X_1} \left\{ f_1(x) + \underline{\mathcal{V}}_2(x) : \; A_1 x = b_1 \right\} \tag{103}$$

This completes one backward pass. In the forward pass, we sample a set of scenarios $(\xi_1, \xi^k_2, ..., \xi^k_T)$[2], $k = 1, .., K$ from the scenario tree for a $K \ll \mathcal{N}$. Then by solving (96) with the current approximations $\underline{\mathcal{V}}_t$ of the expected cost-to-go function, for each scenario k, we obtain a sequence of candidate points $(\bar{x}^k_1, \bar{x}^k_2, ..., \bar{x}^k_T)$. To start the algorithm, we need a first guess for the value function approximates. We can either skip the forward pass in the first iteration, starting the backward pass with randomly selected feasible candidate points. Or if we have lower bounds θ_t for the expected cost-to-go functions, we can initialize the cut approximations $\underline{\mathcal{V}}_t \equiv \theta_t$.

The SDDP now consists of a forward- and a backward-pass in each iteration i. That is in each i, we sample a set of scenarios $(\xi^i_1, \xi^{ik}_2, ..., \xi^{ik}_T), k = 1, ..., K_i$. Then we run the forward pass to obtain candidate points $(\bar{x}^{ik}_1, \bar{x}^{ik}_2, ..., \bar{x}^{ik}_T)$ and update the cut approximations at these trail points in the backward pass. We run the algorithm $i = 1, ..., N$ times until we meet a stopping criterion. For a detailed discussion of convergence and stopping of SDDP see Füllner and Rebennack (2023).

Since its invention, there grew a large literature about SDDP and its properties, variants and extensions. It would go beyond the scope of this thesis to give a comprehensive overview about this research. Convergence for linear problems with stagewise independent randomness was first formally proved by Chen and Powell (1999). One of the main assumptions for SDDP is that the random process is finite, which is not the case in most real-world applications. Shapiro (2011) shows convergence and statistical properties of SDDP if the true continous random process is discretized. Also Löhndorf and Shapiro (2019) theoretically analyse how to deal with problems where the randomness is not stagewise independent anymore but instead is markovian. Both in the original paper by Pereira and Pinto (1991) and in most of the literature since then, SDDP is stated for linear stochastic multistage problems with a linear objective function and affine-linear dynamics. In this case the $t+1$-stage value-functions are polyhedral and convex and therefore the t-stage can be formulated as an equivalent linear problem, so the whole problem remains linear. Girardeau et al. (2015) give a convergence proof for general convex problems.

[2]Note again, that ξ_1 is deterministic

3.2 Saddle-point formulation

The dynamic programming equations of the MSSP (94) read as (95) which is an instance of a more generic convex optimization problem.

$$\begin{aligned} V(u,\xi) = \min_{x\in X} \quad & f(x,c) + v(x,\xi) \\ s.t. \quad & Ax - Bu - b \in K \end{aligned} \tag{104}$$

with convex objective f, v, closed convex feasible set X and closed convex cone K. $v(\cdot,\xi)$ is a realization of the next-stage expected value function, conditional on ξ. Here the previous stage x_t and the realization of the uncertainty process up to time t, $\xi_{[t]}$ go into this problem as parameters u and ξ respectively. The Lagrangian of (104) is

$$L(x,y,u,\xi) = f(x,\xi) + v(x,\xi) + \langle Ax - Bu - b, y\rangle . \tag{105}$$

We assume that total duality holds for all parameters u, ξ, that is for every u, ξ there is a pair $\bar{x}, \bar{y}$, such that

$$L(\bar{x},y,u,\xi) \leq L(\bar{x},\bar{y},u,\xi) \leq L(x,\bar{y},u,\xi), \tag{106}$$

which is the case if slaters constraint qualification holds. That is for any u, ξ there is a $\tilde{x} \in \operatorname{int} X$ such that $A\tilde{x} - Bu \in \operatorname{int} K$. We call a pair $(\bar{x},\bar{y})$ which satisfies (106) a *saddle-point* of the Lagrangian $L(\cdot,\cdot,u,\xi)$ at (u,ξ). We call

$$\max_{y\in K^*} \min_{x\in X} L(x,y,u,\xi) \tag{107}$$

the dual problem to (104) and

$$\min_{x\in X} \max_{y\in K^*} L(x,y,u,\xi) \tag{108}$$

the primal problem. We say total duality holds if

$$V(u,\xi) = \min_{x\in X} \max_{y\in K^*} L(x,y,u,\xi) = \max_{y\in K^*} \min_{x\in X} L(x,y,u,\xi) = L(\bar{x},\bar{y},u,\xi) \tag{109}$$

and call $\bar{x}$ a primal solution and $\bar{y}$ a dual solution. For many popular saddle-point solvers for an objective function of the form $f(x) + g(x) + f(Kx)$ there exists a variant which only uses a linear approximation of $g(x)$, therefore they only need an oracle for the gradient or subgradient. This is possible since they only use first-order information, i.e. (sub-)gradients. In this case we need to ensure, that the optimization problems in the subroutines remain solvable by adding a proximal term $||x^k - x||^2$. This is very valuable for us, since in most cases we cannot evaluate the value functions $v(x,\xi)$ but we can compute gradients $\nabla v(x,\xi)$ respective subgradients $v'(\xi) \in \partial_x v(x,\xi)$ as in section 2.6. There we have that by computing a dual solution $\bar{y}(u,\xi)$ to (104), we have that $B^T\bar{y}(u,\xi)$ is an ϵ-subgradient of $V(u,\xi)$ at u,ξ. We will now present an approach how to solve the Multistage stochastic program by solving a nested sequence of saddle-point problems. This approach generalizes the Dynamic-Stochastic Approximation algorithm of Lan and Zhou (2021). Assume we have an iterative method to solve the saddle point problem (109):

$$x^{k+1}, y^{k+1} = \mathrm{SPS}(x^{[k]}, y^{[k]}, v', u, \xi, X, K^*, \mathrm{param}) \tag{110}$$

where $x^{[k]}, y^{[k]}$ denotes the history of previous iterates, v' is a stochastic ϵ-subgradient of v at $x^{[k]}, \xi$ and param is a set of algorithmic parameters like step-sizes etc. Note that the uncertainty ξ_t in the stage-t Problem in (94) determines the parameters, therefore $\xi = (c, A, B, b)$. We then have the algorithm (algorithm 3 in Lan and Zhou (2021))

Algorithm 1 The general DSA algorithm for three-stage problems

Require: Initial points (z_1^0, z_2^0, z_3^0), parameters $\xi^1 = (A_1, b_1, c_1)$.
for $i = 1, 2, \dots, N_1$ **do**
  Generate a random realization of $\xi_2^i = (A_2^i, B_2^i, b_2^i, c_2^i)$.
  for $j = 1, 2, \dots, N_2$ **do**
    Generate a random realization of $\xi_3^j = \left(A_3^j, B_3^j, b_3^j, c_3^j\right)$ conditional on ξ_2^i.
    for $k = 1, 2, \dots, N_3$ **do**
      $(x_3^k, y_3^k, \tilde{y}_3^k) = \text{SPS}\,(x_3^{[k-1]}, y_3^{[k-1]}, 0, x_{j-1}^2, \xi_3^j, X^3, K_*^3, \theta_3^k, \tau_3^k, \eta_3^k)$.
    end for
    $\left(\bar{x}_3^j, \bar{y}_3^j\right) = \sum_{k=1}^{N_3} w_3^k\,(x_3^k, y_3^k) \,/\, \sum_{k=1}^{N_3} w_3^k$.
    $\left(x_2^j, y_2^j, \tilde{y}_2^j\right) = \text{SPS}\,(x_2^{[j-1]}, y_2^{[j-1]}, (B_3^j)^T \bar{y}_3^j, x_{i-1}^1, \xi_2^i, X^2, K_*^2, \theta_2^j, \tau_2^j, \eta_2^j)$.
  end for
  $(\bar{x}_2^i, \bar{y}_2^i) = \sum_{j=1}^{N_2} w_2^j \left(x_2^j, y_2^j\right) / \sum_{j=1}^{N_2} w_2^j$.
  $(x_1^i, y_1^i, \tilde{y}_1^i) = \text{SPS}\,(x_1^{[i-1]}, y_1^{[i-1]}, (B_2^i)^T \bar{y}_2^i, 0, \xi^1, X^1, K_*^1, \theta_1^i, \tau_1^i, \eta_1^i)$.
end for
return $(\bar{x}^1, \bar{y}^1) = \sum_{i=1}^{N_1} w_1^i\,(x_1^i, y_1^i) \,/\, \sum_{i=1}^{N_1} w_1^i$.

where w's are the weights of the ergodic averages. Most of the time their are set to 1, but we will leave them undetermined here for full generality.

3.3 Dynamic Stochastic Approximation

Lan and Zhou (2021) use the PDHG method from Chambolle and Pock (2011) as a subroutine to solve the upcoming saddle-point problems. Their paper is to the best of our knowledge the first publication formualting MSSPs as a nested sequence of saddle-point problems. Here we present the main proof with some preliminary assumptions to ease notation and to focus on the main ideas of the proof. In particular, we set the weights w_k of the averaged solution to 1 and the parameters η_k, τ_k are fixed to η respective τ within each stage. Furthermore the extrapolation parameter in the PDHG-subroutine is set to 1 as in Chambolle and Pock (2011). Then the PDHG iteration to solve the saddle-point problem (109) at each stage reads

$$\begin{aligned}
\tilde{y}_{k-1} &= 2y_{k-1} - y_{k-2}, \\
x_k &= \underset{x \in X}{\operatorname{argmin}}\ \langle b + Bu - Ax, \tilde{y}_{k-1} \rangle + f(x) + \langle x, v' \rangle + \tau D(x, x_{k-1}) \\
y_k &= \underset{y \in K_*}{\operatorname{argmin}}\ \langle -b - Bu + Ax_k, y \rangle + \frac{\eta}{2} ||y - y_{k-1}||^2
\end{aligned} \tag{111}$$

where $D(\cdot, \cdot)$ is the Bregman distance w.r.t an α-strongly convex distant generating function ψ. Define the diameter of the set X as

$$\Omega^2 := \max_{x,y \in X} D(y, x) \tag{112}$$

Define the primal-dual gap function

$$\begin{aligned}
Q(\bar{z}, z) := &\langle y, b + Bu - A\bar{x} \rangle + f(\bar{x}) + v(\bar{x}) \\
&- \langle \bar{y}, b + Bu - Ax \rangle - f(x) - v(x)
\end{aligned} \tag{113}$$

The first result gives a bound for the gap-function at each iteration.

Lemma 3.1. *We have for any $1 \leq k \leq N$ and $(x, y) \in X \times K_*$*

$$\begin{aligned}&Q(z_k, z) + \langle A(x_k - x), y_k - y_{k-1}\rangle - \langle A(x_{k-1} - x), y_{k-1} - y_{k-2}\rangle - \langle A(x_k - x_{k-1}), y_{k-1} - y_{k-2}\rangle \\ &\leq \tau\left[D(x, x_{k-1}) - D(x, x_k) - \tfrac{\alpha}{2}||x_k - x_{k-1}||^2\right] + \tfrac{\eta}{2}\left[||y - y_{k-1}||^2 - ||y_k - y||^2 - ||y_k - y_{k-1}||^2\right] \\ &\quad + \langle \Delta_{k-1}, x_{k-1} - x\rangle + (M + ||G_{k-1}||_*)||x_k - x_{k-1}|| + \bar{\epsilon}\end{aligned} \tag{114}$$

We want to use the Lipschitz continuity of v to get an estimate for $v(x_k) - v(x)$ in the gap-function: We assume that for a stochastic $\bar{\epsilon}$-subgradient G_k of v at x_k there exists $M > 0$ such that

$$\mathbb{E}\left[||G_k||_*^2\right] \leq M^2 \quad \forall k \geq 1 \tag{115}$$

We define the bias of the subgradient-estimator G_k as

$$\Delta_k := g(x_k) - G_k. \tag{116}$$

Then v is Lipschitz continuous with Lipschitz-constant M and we have

$$\begin{aligned}v(x_k) &\leq v(x_{k-1}) + M||x_k - x_{k-1}|| \\ &\leq v(x) + \langle g(x_{k-1}), x_{k-1} - x\rangle + M||x_k - x_{k-1}|| + \bar{\epsilon}\end{aligned} \tag{117}$$

by the definition of an $\bar{\epsilon}$-subgradient. Furthermore we have

$$\begin{aligned}\langle g(x_{k-1}), x_{k-1} - x\rangle &= \langle G_{k-1}, x_{k-1} - x\rangle + \langle \Delta_{k-1}, x_{k-1} - x\rangle \\ &= \langle G_{k-1}, x_k - x\rangle + \langle G_{k-1}, x_{k-1} - x_k\rangle + \langle \Delta_{k-1}, x_{k-1} - x\rangle \\ &\leq \langle G_{k-1}, x_k - x\rangle + \|G_{k-1}\|_* \|x_k - x_{k-1}\| + \langle \Delta_{k-1}, x_{k-1} - x\rangle .\end{aligned} \tag{118}$$

Which gives us an upper bound for $v(x_k) - v(x)$:

$$v(x_k) - v(x) \leq \langle G_{k-1}, x_k - x\rangle + \langle \Delta_{k-1}, x_{k-1} - x\rangle + (M + \|G_{k-1}\|)\|x_k - x_{k-1}\| + \bar{\epsilon} \tag{119}$$

From the first order conditions of the minimization steps in the algorithm we get the following estimates (Lemma 1 in Lan et al. (2011))

$$\begin{aligned}\langle -A(x_k - x), \tilde{y}_{k-1}\rangle &+ f(x_k) - f(x) + \langle G_{k-1}, x_k - x\rangle \\ &\leq \tau\left[D(x, x_{k-1}) - D(x, x_k) - D(x_k, x_{k-1})\right], \ \forall x \in X, \\ \langle -b - Bu + Ax_k, y_k - y\rangle &\leq \tfrac{\eta}{2}\left[||y - y_{k-1}||^2 - ||y_k - y||^2 - ||y_k - y_{k-1}||^2\right], \ \forall y \in K_*\end{aligned} \tag{120}$$

Summing both terms and plugging in 119 we get

$$\begin{aligned}&\langle -A(x_k - x), \tilde{y}_{k-1}\rangle + \langle -b - Bu + Ax_k, y_k - y\rangle + f(x_k) - f(x) + v(x_k) - v(x) \\ &\leq \tau\left[D(x, x_{k-1}) - D(x, x_k) - D(x_k, x_{k-1})\right] + \tfrac{\eta}{2}\left[||y - y_{k-1}||^2 - ||y_k - y||^2 - ||y_k - y_{k-1}||^2\right] \\ &\quad + \langle \Delta_{k-1}, x_{k-1} - x\rangle + (M + ||G_{k-1}||_*)||x_k - x_{k-1}||^2 + \bar{\epsilon}\end{aligned} \tag{121}$$

Now we deal with the first two summands in the first line

$$\begin{aligned}&\langle -A(x_k - x), \tilde{y}_{k-1}\rangle + \langle -b - Bu + Ax_k, y_k - y\rangle \\ &= \langle -A(x_k - x), \tilde{y}_{k-1}\rangle + \langle -b - Bu + A(x_k + x - x), y_k\rangle - \langle -b - Bu + Ax_k, y\rangle \\ &= \langle -A(x_k - x), \tilde{y}_{k-1} - y_k\rangle + \langle -b - Bu + Ax, y_k\rangle - \langle -b - Bu + Ax_k, y\rangle\end{aligned} \tag{122}$$

We have

$$\begin{aligned}\langle -A(x_k - x), \tilde{y}_{k-1} - y_k\rangle &= \langle -A(x_k - x), (y_{k-1} - y_{k-2}) + y_{k-1} - y_k\rangle \\ &= \langle -A(x_k - x + x_{k-1} - x_{k-1}), (y_{k-1} - y_{k-2})\rangle \\ &\quad - \langle -A(x_k - x), y_k - y_{k-1}\rangle \\ &= \langle -A(x_k - x_{k-1}), y_{k-1} - y_{k-2}\rangle \\ &\quad + \langle -A(x_{k-1} - x), y_{k-1} - y_{k-2}\rangle \\ &\quad - \langle -A(x_k - x), y_k - y_{k-1}\rangle\,,\end{aligned} \tag{123}$$

Then (121) becomes

$$\begin{aligned}&\langle -b - Bu + Ax, y_k\rangle - \langle -b - Bu + Ax_k, y\rangle + f(x_k) - f(x) + v(x_k) - v(x) \\ &+ \langle -A(x_k - x_{k-1}), y_{k-1} - y_{k-2}\rangle + \langle -A(x_{k-1} - x), y_{k-1} - y_{k-2}\rangle - \langle -A(x_k - x), y_k - y_{k-1}\rangle \\ &\le \tau\left[D_X(x, x_{k-1}) - D_X(x, x_k) - \tfrac{\alpha}{2}||x_k - x_{k-1}||^2\right] + \tfrac{\eta}{2}\left[||y - y_{k-1}||^2 - ||y_k - y||^2 - ||y_k - y_{k-1}||^2\right] \\ &\quad + \langle \Delta_{k-1}, x_{k-1} - x\rangle + (M + ||G_{k-1}||_*)||x_k - x_{k-1}||^2 + \bar{\epsilon},\end{aligned} \tag{124}$$

which yields the estimate.

Theorem 3.2 (Theorem 3 in Lan and Zhou (2021)). *If the parameters τ and η in (111) satisfy*

$$\alpha\tau\eta \ge 2||A||^2 \tag{125}$$

we have for any $z = (x, y) \in X \times K$

$$\begin{aligned}Q(\bar{z}_N, z) \le \frac{1}{N}\Bigg(&\tau D(x, x_0) + \frac{\eta}{2}\|y - y_0\|^2 - \frac{\eta}{2}\|y - y_N\|^2 \\ &+ \sum_{k=1}^{N}\left[\tfrac{(M + \|G_{k-1}\|_*)^2}{\alpha\tau} + \langle \Delta_{k-1}, x_{k-1} - x\rangle\right]\Bigg) + \bar{\epsilon}.\end{aligned} \tag{126}$$

Proof. We sum both sides of (124) over $k = 1, ..., N$ with $= 1$ and $y_{-1} = y_0$ and obtain

$$\begin{aligned}&\sum_{k=1}^{N}\left[\langle A(x_k - x), y_k - y_{k-1}\rangle - \langle A(x_{k-1} - x), y_{k-1} - y_{k-2}\rangle - \langle A(x_k - x_{k-1}), y_{k-1} - y_{k-2}\rangle\right] \\ &= \langle A(x_N - x), y_N - y_{N-1}\rangle - \sum_{k=1}^{N}\langle A(x_k - x_{k-1}), y_{k-1} - y_{k-2}\rangle\,.\end{aligned} \tag{127}$$

By the α-strong convexity of D we have $D(x_k, x_{k-1}) \ge \alpha/2||x_k - x_{k-1}||^2$

$$\begin{aligned}&\sum_{k=1}^{N}\left[\tau\left[D(x, x_{k-1}) - D(x, x_k) - D(x_k, x_{k-1})\right] + \tfrac{\eta}{2}\left[||y - y_{k-1}||^2 - ||y_k - y||^2 - ||y_k - y_{k-1}||^2\right]\right] \\ &\le \tau\left(D(x, x_0) - D(x, x_N)\right) + \tfrac{\eta}{2}\left(||y - y_0||^2 - ||y - y_N||^2\right) \\ &\quad - \sum_{k=1}^{N}\left[\tfrac{\alpha\tau}{2}||x_k - x_{k-1}||^2 + \tfrac{\eta}{2}||y_k - y_{k-1}||^2\right]\end{aligned} \tag{128}$$

wich yields together

$$\sum_{k=1}^{N} Q(z_k, z) \leq -\langle A(x_N - x), y_N - y_{N-1}\rangle + \sum_{k=1}^{N} \langle A(x_k - x_{k-1}), y_{k-1} - y_{k-2}\rangle$$
$$+ \tau D(x, x_0) - \tau D(x, x_N) + \tfrac{\eta}{2}||y - y_0||^2 - \tfrac{\eta}{2}||y - y_N||^2$$
$$- \sum_{k=1}^{N} \left[\tfrac{\alpha\tau}{4}||x_k - x_{k-1}||^2 + \tfrac{1}{2\eta}||y_{k-1} - y_{k-2}||^2\right] - \tfrac{\eta}{2}||y_N - y_{N-1}||^2$$
$$+ \sum_{k=1}^{N} \left[-\tfrac{\alpha\tau}{4}||x_k - x_{k-1}||^2 + \langle \Delta_{k-1}, x_{k-1} - x\rangle + (M + ||G_{k-1}||_*)||x_k - x_{k-1}||^2 + \bar{\epsilon}\right]. \quad (129)$$

We have by the Cauchy-Schwarz inequality, the strong convexity of D and the parameter conditions

$$\begin{aligned} &- \langle A(x_N - x), y_N - y_{N-1}\rangle - \tau D(x, x_N) - \tfrac{\eta}{2}||y_N - y_{N-1}||^2 \\ &\leq ||A||||x_N - x||||y_N - y_{N-1}|| - \tfrac{\alpha\tau}{2}||x - x_N||^2 - \tfrac{\eta}{2}||y_N - y_{N-1}||^2 \quad \leq 0 \end{aligned} \quad (130)$$

and similarly

$$-\sum_{k=1}^{N} \left[\tfrac{\alpha\tau}{4}||x_k - x_{k-1}||^2 + \tfrac{\eta}{2}||y_{k-1} - y_{k-2}||^2 + \langle A(x_k - x_{k-1}), y_{k-1} - y_{k-2}\rangle\right] \leq 0 \quad (131)$$

which simplifies (129) to

$$\sum_{k=1}^{N} Q(z_k, z) \leq \tau D(x, x_0) + \tfrac{\eta}{2}||y - y_0||^2 - \tfrac{\eta}{2}||y - y_N||^2$$
$$+ \sum_{k=1}^{N} \left[-\tfrac{\alpha\tau}{4}||x_k - x_{k-1}||^2 + \langle \Delta_{k-1}, x_{k-1} - x\rangle + (M + ||G_{k-1}||_*)||x_k - x_{k-1}||^2 + \bar{\epsilon}\right] \quad (132)$$

We have with the Young inequality $-at/2 + b \leq b^2/(2at)$

$$\sum_{k=1}^{N} \left[-\tfrac{\alpha\tau}{4}||x_k - x_{k-1}||^2 + (M + ||G_{k-1}||_*)||x_k - x_{k-1}||\right] \leq \sum_{k=1}^{N} \tfrac{(M+||G_{k-1}||_*)^2}{\alpha\tau}. \quad (133)$$

□

Finally we arrive at (132)

$$\sum_{k=1}^{N} Q(z_k, z) \leq \tau D(x, x_0) + \tfrac{\eta}{2}\|y - y_0\|^2 - \tfrac{\eta}{2}\|y - y_N\|^2$$
$$+ \sum_{k=1}^{N} \left[\tfrac{(M+\|G_{k-1}\|_*)^2}{\alpha\tau} + \langle \Delta_{k-1}, x_{k-1} - x\rangle + \bar{\epsilon}\right]. \quad (134)$$

Dividing by N and using the convexity of the gap function Q, we obtain the desired bound for the gap-function $Q(\bar{z}_N, z)$.

To measure optimality, we will define the following gap-functions:

$$\begin{aligned} \mathrm{gap}_*(\bar{z}) &:= \max\{Q(\bar{z}; x, y_*) : \ x \in X\} \\ \mathrm{gap}_\delta(\bar{z}) &:= \max\{Q(\bar{z}, x, y) + \langle \delta, y \rangle : \ (x, y) \in X \times K_*\} \end{aligned} \tag{135}$$

In the proof of the next theorem, we need an essential lemma to calculate $\mathbb{E}\left[\langle \Delta_{k-1}, x_{k-1} - x \rangle\right]$. This step was introduced by Nemirovski et al. (2009) (Lemma 2.1):

Lemma 3.3. *For a distance generating function ψ with modulus α define the bregman divergence (see Definition 2.1)*

$$\begin{aligned} &D : X \times X^\circ \to \mathbb{R}_+ \\ &D(u, x) := \psi(u) - [\psi(x) + \langle \nabla\psi(x), u - x \rangle] \end{aligned} \tag{136}$$

Define the respective proximal map

$$P_x^\tau(y) := \underset{z \in X}{\operatorname{argmin}} \ \left\{ \langle y, z - x \rangle + \tau D(z, x) \right\}. \tag{137}$$

Then for any $u \in X$, $x \in X^\circ$ and y we have the following inequality

$$\tau D(u, P_x^\tau(y)) \leq \tau D(u, x) + \langle y, u - x \rangle + \frac{||y||_*^2}{2\alpha\tau} \tag{138}$$

Proof. For the proof see the Appendix. □

In the euclidean case this gives us

$$\langle y, x - u \rangle \leq \tfrac{\tau}{2}||u - x||_2^2 - \tfrac{\tau}{2}||u - P_x^\tau(y)||_2^2 + \tfrac{||y||_*^2}{2\tau} \tag{139}$$

Now we want to use this lemma to generate an artificial sequence which we will later use in the proof of Theorem 3.5.

Lemma 3.4. *Define $\nu_0 = x_0$ and for $k \geq 1$*

$$\nu_k := P_{\nu_{k-1}}^\tau(\Delta_{k-1}) = \underset{z \in X}{\operatorname{argmin}} \ \left\{ \langle \Delta_{k-1}, z \rangle + \tfrac{\tau}{2}||\nu_{k-1} - z||_2^2 \right\}. \tag{140}$$

Then for any $x \in X$

$$\sum_{k=1}^{N} \langle \Delta_{k-1}, \nu_{k-1} - x \rangle \leq \tfrac{\tau}{2}||x - x_0||_2^2 + \textstyle\sum_{k=1}^{N} \frac{||\Delta_{k-1}||_*^2}{2\tau} \tag{141}$$

Proof. By using Lemma 3.3 with $y = \Delta_{k-1}, x = \nu_{k-1}$ and $u = x$, and recalling that we are in the euclidean setting, therefore $\alpha = 1$, we have

$$\langle \Delta_{k-1}, \nu_{k-1} - x \rangle \leq \tfrac{\tau}{2}||x - \nu_{k-1}||_2^2 - \tfrac{\tau}{2}||x - \nu_k||_2^2 + \tfrac{||\Delta_{k-1}||_*^2}{2\tau}. \tag{142}$$

Summing from $k = 1$ to $k = N$ and omitting nonpositive terms on the right hand side, we arrive at (141). □

Theorem 3.5. *[Theorem 5 in Lan and Zhou (2021)] Set $\delta := 1/N(\eta(y_0 - y_N))$. If the parameters τ, η satisfy assumption (125) then we have the following estimations*

1.
$$\mathbb{E}\left[\text{gap}_*(\bar{z}_N)\right] \leq \frac{1}{N}\left(2\Omega_X^2\tau + \tfrac{\eta}{2}||y_* - y_0||_*^2\right) + \left[\tfrac{6M^2}{\tau}\right] + \bar{\epsilon}. \tag{143}$$

2.
$$\mathbb{E}\left[\text{gap}_\delta(\bar{z}_N)\right] \leq \frac{1}{N}\left(2\Omega_X^2\tau + \tfrac{\eta}{2}||y_0||_*^2\right) + \left[\tfrac{6M^2}{\tau}\right] + \bar{\epsilon} \tag{144}$$

3.
$$\mathbb{E}\left[||\delta||\right] \leq \frac{\eta}{N}\left(2||y_* - y_0||_* + 2\sqrt{\tfrac{\tau}{\eta}}\Omega_X + \sqrt{\tfrac{2}{\eta}N\left[\tfrac{6M^2}{\tau} + \bar{\epsilon}\right]}\right) \tag{145}$$

4.
$$\mathbb{E}\left[||y_* - \bar{y}_N||^2\right] \leq ||y_* - y_0||^2 + \frac{4\Omega_X^2\tau}{\eta} + \frac{N+1}{\eta}\left(\frac{6M^2}{\tau} + \bar{\epsilon}\right) \tag{146}$$

Proof. Plugging in y_* for y in (126), maximizing w.r.t. $x \in X$ and taking expectations, we have

$$\mathbb{E}\left[\text{gap}_*(\bar{z}_N), y_*\right] = \frac{1}{N}\Bigg(\Omega_X^2\tau + \frac{\eta}{2}||y_* - y_0||_*^2 - \frac{\eta}{2}||y_* - y_N||_*^2 \\ + \mathbb{E}\left[\max_{x\in X}\left\{\sum_{k=1}^{N}\left[\langle\Delta_{k-1}, x_{k-1} - x\rangle + \tfrac{(M+\|G_{k-1}\|_*)^2}{\tau} + \bar{\epsilon}\right]\right\}\right]\Bigg). \tag{147}$$

where we have used that for two functions $f(x), g(x)$, $\max_x \{f(x) + g(x)\} \leq \max_x f(x) + \max_x g(x)$. If we define $\nu_k,\ k = 1, ..., N$ as in Lemma 3.4,

$$\max_{x\in X}\left\{\sum_{k=1}^{N}\langle\Delta_{k-1}, x_{k-1} - x\rangle\right\} = \sum_{k=1}^{N}\langle\Delta_{k-1}, x_{k-1} - \nu_{k-1}\rangle + \max_{x\in X}\left\{\sum_{k=1}^{N}\langle\Delta_{k-1}, \nu_{k-1} - x\rangle\right\} \\ \leq \Omega_X^2\tau + \sum_{k=1}^{N}\left[\langle\Delta_{k-1}, x_{k-1} - \nu_{k-1}\rangle + \tfrac{||\Delta_{k-1}||_*^2}{2\tau}\right] \tag{148}$$

Now by definition $\mathbb{E}_{|\xi_{[t-1]}}\left[\Delta_k\right] = 0$. Thus we have

$$\begin{aligned}\mathbb{E}\left[\langle\Delta_{k-1}, x_{k-1} - \nu_{k-1}\rangle\right] &= \mathbb{E}\left[\mathbb{E}_{|\xi_{[k-1]}}\left[\langle\Delta_{k-1}, x_{k-1} - \nu_{k-1}\rangle\right]\right] \\ &= \mathbb{E}\left[\left\langle\mathbb{E}_{|\xi_{[k-1]}}\left[\Delta_{k-1}\right], x_{k-1} - \nu_{k-1}\right\rangle\right] \\ &= 0\end{aligned} \tag{149}$$

Furthermore, since we assumed $\mathbb{E}\left[\|G_{k-1}\|_*^2\right] \leq M^2$ and $\|g\left(x_{k-1}\right)\|_* \leq M$ we obtain with the triangle inequality and the fact $(a+b)^2 \leq 2a^2 + 2b^2$

$$\mathbb{E}\left[\|\Delta_{k-1}\|_*^2\right] = \mathbb{E}\left[\|G_{k-1} - g\left(x_{k-1}\right)\|_*^2\right] \leq \mathbb{E}\left[\left(\|G_{k-1}\|_* + \|g\left(x_{k-1}\right)\|_*\right)^2\right] \leq 4M^2 \tag{150}$$

Therefore,

$$\mathbb{E}\left[\max_{x\in X}\left\{\sum_{k=1}^{N}\langle\Delta_{k-1}, x_{k-1} - x\rangle\right\}\right] \leq \Omega_X^2\tau + \sum_{k=1}^{N}\frac{2M^2}{\tau} \tag{151}$$

Also

$$\mathbb{E}\left[(M + \|G_{k-1}\|_*)^2\right] \leq \mathbb{E}\left[2M^2 + 2\|G_{k-1}\|_*^2\right] \leq 4M^2 \tag{152}$$

which together yields the estimate 1.. To proof the estimate 2., we add $\langle \delta, y\rangle$ on both sides of (126). We have

$$\begin{aligned} Q(\bar{z}_N, z) + \langle \delta, y\rangle \leq &\frac{1}{N}\left(\tau D(x, x_0) + \eta\left[\tfrac{1}{2}\|y_0 - y\|_*^2 - \tfrac{1}{2}\|y_N - y\|_*^2 + \langle y_0 - y_N, y\rangle\right]\right. \\ &\left. + \sum_{k=1}^{N}\left[\langle \Delta_{k-1}, x_{k-1} - x\rangle + \tfrac{(M+\|G_{k-1}\|_*)^2}{\tau} + \bar{\epsilon}\right]\right) \\ \leq &\frac{1}{N}\left(\tau D(x, x_0) + \tfrac{\eta}{2}||y_0||^2 + \textstyle\sum_{k=1}^{N}\left[\langle \Delta_{k-1}, x_{k-1} - x\rangle + \tfrac{(M+\|G_{k-1}\|_*)^2}{\tau} + \bar{\epsilon}\right]\right) \end{aligned} \tag{153}$$

We take the maximum w.r.t. $(x, y) \in X \times K_*$ and take the expectation to obtain

$$\begin{aligned} \mathbb{E}[\mathrm{gap}_\delta(\bar{z}_N)] \leq \frac{1}{N}\Bigg(&\Omega_X^2\tau + \frac{\eta}{2}\|y_0\|_*^2 \\ &+ \mathbb{E}\left[\max_{x\in X}\left\{\sum_{k=1}^{N}\left[\langle \Delta_{k-1}, x_{k-1} - x\rangle + \tfrac{(M+\|G_{k-1}\|_*)^2}{\tau} + \bar{\epsilon}\right]\right\}\right]\Bigg) \end{aligned} \tag{154}$$

With the same estimate as in the proof of 1. for the expectation-term on the RHS, we get (144). To show 3., we first fix $x = x_*$ in (147) and use the fact that $Q(\bar{x}_N, x_*, y_*) \geq 0$ to obtain

$$\frac{\eta}{2}||y_* - y_N||^2 \leq \Omega_X^2\tau + \tfrac{\eta}{2}||y_* - y_0||^2 + \textstyle\sum_{k=1}^{N}\left[\langle \Delta_{k-1}, x_{k-1} - x\rangle + \tfrac{(M+\|G_{k-1}\|_*)^2}{\tau} + \bar{\epsilon}\right]. \tag{155}$$

Again, we take expectations and use the same technique for the expectation of the last summand on the RHS to get

$$\tfrac{\eta}{2}\mathbb{E}\left[||y_* - y_N||^2\right] \leq 2\Omega_X^2\tau + \tfrac{\eta}{2}||y_* - y_0||^2 + \textstyle\sum_{k=1}^{N}\left[\tfrac{6M^2}{\tau} + \bar{\epsilon}\right] \tag{156}$$

which implies

$$\mathbb{E}\left[||y_* - y_N||\right] \leq 2\sqrt{\tfrac{\tau}{\eta}}\Omega_X + ||y_* - y_0|| + \sqrt{\tfrac{2}{\eta}N\left[\tfrac{6M^2}{\tau} + \bar{\epsilon}\right]}. \tag{157}$$

This equality together with the fact that by definition

$$||\delta|| \leq \frac{1}{N}\left(\eta(||y_* - y_0||_* + ||y_* - y_N||)\right), \tag{158}$$

yields (145). To lastly proof 4., we first observe, that (156) holds for any $y_k,\ k = 1, ..., N$ and hence that

$$\tfrac{\eta}{2}\mathbb{E}\left[||y_* - y_k||^2\right] \leq 2\Omega_X^2\tau + \tfrac{\eta}{2}||y_* - y_0||^2 + \textstyle\sum_{i=1}^{k}\left[\tfrac{6M^2}{\tau} + \bar{\epsilon}\right]. \tag{159}$$

$$\mathbb{E}\left[||y_* - y_k||^2\right] \leq \tfrac{4\Omega_X^2\tau}{\eta} + ||y_* - y_0||^2 + \tfrac{2}{\eta}k\left[\tfrac{6M^2}{\tau} + \bar{\epsilon}\right]. \tag{160}$$

We use this, the convexity of $||\cdot||^2$ and the definition of $\bar{y}_N$

$$\mathbb{E}\left[||y_* - \bar{y}_N||^2\right] \leq \tfrac{4\Omega_X^2\tau}{\eta} + ||y_* - y_0||^2 + \tfrac{2}{\eta}\left[\tfrac{6M^2}{\tau} + \bar{\epsilon}\right]\tfrac{(N+1)}{2}. \tag{161}$$

□

Now we will cite two Corollaries from Lan and Zhou (2021) where bounds for the quantities from Theorem 3.5 are obtained for certain parameter settings.

Lemma 3.6 (Corollary 6 in Lan and Zhou (2021))**.** *Let M be the bound for the subgradients of the subsequent stage. If we set*

$$\tau_k = \tau = \max\left\{\frac{M\sqrt{3N}}{\Omega}, \sqrt{2}\|A\|_2\right\}, \quad \eta_k = \eta = \sqrt{2}\|A\|_2 \quad \forall\ 1 \le k \le N, \tag{162}$$

$$\mathbb{E}\left[\text{gap}_*\left(\bar{z}_N\right)\right] \le \frac{\sqrt{2}\|A\|_2\left(2\Omega^2+\|y_*-y_0\|_*^2\right)}{N} + \frac{4\sqrt{3}M\Omega}{\sqrt{N}} + \bar{\epsilon}, \tag{163}$$

$$\mathbb{E}\left[\text{gap}_\delta\left(\bar{z}_N\right)\right] \le \frac{\sqrt{2}\|A\|_2\left(2\Omega^2+\|y_0\|_*^2\right)}{N} + \frac{4\sqrt{3}M\Omega}{\sqrt{N}} + \bar{\epsilon}, \tag{164}$$

$$\mathbb{E}[\|\delta\|_*] \le \frac{2\sqrt{2}\|A\|_2\|y_*-y_0\|_*+4\Omega\|A\|_2}{N} + \frac{2M\left(\sqrt{6}\|A\|_2+\sqrt{3}\right)}{\sqrt{N}} + \sqrt{\frac{3\|A\|_2\bar{\epsilon}}{N}}, \tag{165}$$

$$\mathbb{E}\left[\|y_* - \bar{y}_N\|_*^2\right] \le \|y_* - y_0\|_*^2 + 4\Omega^2 + \frac{2\sqrt{6N}M\Omega}{\|A\|_2} + \frac{3(N+1)M^2}{\|A\|_2^2} + \frac{(N+1)\bar{\epsilon}}{2}. \tag{166}$$

Proof. We only need to check that the parameter set in (162) satisfy (125). Then this follows directly from Theorem 3.5. □

For these parameters, we can make $\mathbb{E}\left[||\delta||_*\right]$ arbitrarily small by increasing N in (165), but cannot bound $\mathbb{E}_|\left[||y_* - \bar{y}_N\right]|_*^2$ in (166). For that we need to use a slighlty different set of parameters.

Lemma 3.7 (Corollary 7 in Lan and Zhou (2021))**.** *If we set*

$$\tau_k = \tau = \max\left\{\frac{M\sqrt{3N}}{\Omega}, \frac{\sqrt{2}\|A\|_2}{\sqrt{N}}\right\} \text{ and } \eta_k = \eta = \sqrt{2N}\|A\|_2, \forall 1 \le k \le N \tag{167}$$

then

$$\mathbb{E}\left[\text{gap}_*\left(\bar{z}_N\right)\right] \le \frac{2\sqrt{2}\|A\|_2\Omega^2}{N\sqrt{N}} + \frac{||A||_2\,||y_*-y_0||_*^2+4\sqrt{3}M\Omega}{\sqrt{N}} + \bar{\epsilon}, \tag{168}$$

$$\mathbb{E}\left[\text{gap}_\delta\left(\bar{z}_N\right)\right] \le \frac{2\sqrt{2}\|A\|\Omega^2}{N\sqrt{N}} + \frac{||A||_2\,||y_0||_*^2+4\sqrt{3}M\Omega}{\sqrt{N}} + \bar{\epsilon}, \tag{169}$$

$$\mathbb{E}[\|\delta\|_*] \le \frac{2\sqrt{2}\|A\|_2\,\|y_*-y_0\|_*+4\sqrt{M\Omega\|A\|_2}}{\sqrt{N}} + \frac{2M\sqrt{6}\|A\|M}{\sqrt{N}} + \sqrt{\frac{3\|A\|\bar{\epsilon}}{\sqrt{N}}}, \tag{170}$$

$$\mathbb{E}\left[\|y_* - \bar{y}_N\|_*^2\right] \le \|y_* - y_0\|_*^2 + \frac{2\Omega^2}{N} + \frac{2\sqrt{6}M\Omega}{\|A\|_2} + \frac{\sqrt{N}\bar{\epsilon}}{\sqrt{2}||A||_2} \tag{171}$$

We have estimates for optimality gaps and dual variable norms for the iterates of one saddle-point problem. Now we need to connect these results at each stage in order to get asymptotic convergence rates for the solutions of the Multistage problem. We have the following important result, guaranteeing the ϵ-optimality and δ-feasibility of a point $\bar{x}$ for the generic saddle point problem (109)

Lemma 3.8 (Lemma 8 in Lan and Zhou (2021))**.** *Let x_* be an optimal solution to (109). If there exists a random vector $\delta(\xi) \in \mathbb{R}^m$ and $\bar{z} \in Z$, such that*

$$\mathbb{E}\left[\text{gap}_\delta(\bar{z})\right] \le \epsilon_0, \tag{172}$$

then

$$\begin{aligned} &\mathbb{E}\left[f(\bar{x}, c) + v(\bar{x}) - f(x_*, c) - v(x_*)\right] \le \epsilon_0, \\ &A\bar{x} - Bu - b - \delta \in K, \ \ a.s.. \end{aligned} \tag{173}$$

Proof. At first we see that

$$f(\bar{x}, c) + v(\bar{x}) - f(x_*, c) - v(x_*) \tag{174}$$

□

Now if we choose N large enough such that the right-hand side of (164) is smaller than an ϵ, we know from Lemma 3.8 that $\bar{z}_N$ is an (ϵ, δ)-optimal solution of (104) and by (170) with an appropriate choice of N we also can bound δ in expectation by ϵ. In the folowing we summarize Lemmas 10 and 11 and Theorem 12 from Lan and Zhou (2021), which use the previously established bounds for a one-stage saddle-point problem to give bounds for the solution of the saddle-point formulation of the MSSP.

Theorem 3.9. *Let ϵ be our desired accuracy. If we run Algorithm 1 with the parameters at stages $t = 1$ and $t = 3$ set to (162) and for the inner loop $t = 2$ set to (167) and the last stage iteration number N_3 set to*

$$N_3 := \frac{3\sqrt{2}||A||_2\left[2\Omega^2 + ||y^j_{*,3} - y^0_3||^2\right]}{\epsilon}, \tag{175}$$

$$\tag{176}$$

then $B^T\bar{y}^j_3$ is a stochastic $\epsilon/3$-subgradient of the expected cost-to-go function v_3 at x^{j-1}_2. Moreover there exists a constant M_3 such that $|v_3(x) - v_3(y)| \leq M3||x - y||_2\ \forall x, y \in X^2$. and

$$\mathbb{E}\left[||B^T\bar{y}^j_3||^2_*\right] \leq M^2_3 \tag{177}$$

And if we set for the second stage

$$N_2 := \left(\frac{12\sqrt{2}\|A\|_2\Omega}{\epsilon}\right)^{\frac{2}{3}} + \left[\frac{6\left(||A||_2\ ||y^i_{*,2} - y^0_2||^2 + 4\sqrt{3}M_3\Omega\right)}{\epsilon}\right]^2, \tag{178}$$

$$\tag{179}$$

then $B^T\bar{y}^i_2$ is a stochastic $2\epsilon/3$-subgradient of the expected cost-to-go function v_2 at x^1_{i-1}. Moreover there exists a constant M_2 such that $|v_2(x) - v_2(y)| \leq M_2||x - y||_2\ ,\forall x, y \in X^2$ and

$$\mathbb{E}\left[||B^T\bar{y}^i_2||^2_*\right] \leq M^2_2. \tag{180}$$

And if we set for the first stage

$$N_1 := \max\left\{\frac{6\sqrt{2}||A||_2\left[2\Omega^2 + \left\|y^0_1\right\|^2_*\right]}{\sqrt{\alpha}\epsilon} + \left(\frac{24\sqrt{3}M_2\Omega}{\epsilon}\right)^2,\right. \tag{181}$$

$$\left.\frac{6\|A\|_2\left(\sqrt{2}\left\|y_{*,1} - y^0_1\right\| + 2\Omega + 3\right)}{\epsilon} + \left(\frac{6\sqrt{3}M_2\left(\sqrt{2}\|A\| + 1\right)}{\epsilon}\right)^2\right\}, \tag{182}$$

then there exist vectors δ_3, δ_2 *and* δ_1 *s.t.*

$$\mathbb{E}\left[f_3\left(\bar{x}_3\right) - V_3\left(\bar{x}_3, \xi_3\right)\right] \leq \epsilon/3, \tag{183}$$

$$A\bar{x}_3 - B\bar{x}_2 - b_3 - \delta_3 = 0 \ a.s. \tag{184}$$

$$\mathbb{E}\left[\|\delta_3\|_*\right] \leq \epsilon/3, \tag{185}$$

$$\mathbb{E}\left[f_2\left(\bar{x}_2\right) + v_3\left(\bar{x}_2\right) - V_2\left(\bar{x}_1, \xi_2\right)\right] \leq 2\epsilon/3, \tag{186}$$

$$A\bar{x}_2 - B\bar{x}_1 - b_2 - \delta_2 = 0 \ a.s. \tag{187}$$

$$\mathbb{E}\left[\|\delta_2\|_*\right] \leq 2\epsilon/3 \tag{188}$$

and

$$\mathbb{E}\left[f_1\left(\bar{x}_1\right) + v^2\left(\bar{x}_1\right) - \left(f_1\left(x_*\right) + v^2\left(x_*\right)\right)\right] \leq \epsilon, \tag{189}$$

$$A\bar{x}_1 - b - \delta_1 = 0 \ a.s. \tag{190}$$

$$\mathbb{E}[\|\delta_1\|_*] \leq \epsilon \tag{191}$$

Proof. For the proof see Lemmas 10, 11 and Theorem 12 in Lan and Zhou (2021). □

3.4 Alternating Direction Methods of Multipliers

We now want to present an alternative method to solve the saddle-point problems which might give us better control over the feasibility of an approximate solution by penalizing constraint violations with a quadratic term in the objective. We again assume that the cones K_t in (94) are trivial, that is we only have equality terms in the coupling constraint. Consider the composite optimization problem of the form

$$\inf_{x \in X} h(x) + g(Ax), \tag{192}$$

with closed convex set X, CCPs functions $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, $g : \mathbb{R}^m \to \bar{\mathbb{R}}$ and matrix $A \in \mathbb{R}^{m \times n}$. This is equivalent to the linear constraint convex problem

$$\begin{aligned} \inf_{x \in X,\, z \in \mathbb{R}^m} \; & h(x) + g(z), \\ s.t. \quad & Ax = z. \end{aligned} \tag{193}$$

Our generic optimization problem (104) is an instance of this problem.

$$\begin{aligned} \min_{x \in X} \quad & f(x, c) + v(x, \xi) + \mathbb{I}_{\{z:\, z = Bu + b\}}(z) \\ s.t. \quad & Ax = z, \end{aligned} \tag{194}$$

parametrized through ξ and u. We want to cast this problem in the framework of monotone inclusions, and solve the monotone inclusion problem by applying Douglas-Rachford-Splitting as introduced in sections 2.7 - 2.8. The resulting iteration is the famous *Alternating Direction Method of Multipliers* (ADMM). In our case, g is only an indicator function of a point set, which reduces ADMM to the classic *Augmented Lagrangian Method* (ALM).We will here treat ALM as a special case of ADMM and will not refer to the classic ALM literature but to the more contemporary ADMM literature. We will further introduce a more contemporary variant involving a proximal term, which was introduce by Shefi and Teboulle (2014). Finally since we only have first-order information about the value function-part v in (104), we present a variant of ADMM which only uses a linear model to v in the iteration. This was studied by Banert et al. (2021).

Remark 3.10. *Here we consider problems with linear variable coupling $Ax = z$, but this can be extended to the more general case of affine linear coupling $Ax + By = c$, see e.g Ryu and Yin (2022), ch.3.1..*

For the presentation of ADMM we follow Ryu and Yin (2022). The Lagrangian of (193) is

$$L(x, z, y) = h(x) + g(z) + \langle y, Ax - z \rangle \,. \tag{195}$$

and therefore the corresponding dual problem to (193) reads

$$\sup_{y \in \mathbb{R}^n} \; - f^*(-A^T u) - g^*(y) \tag{196}$$

As before we assuem that total duality (109) holds. We further introduce the augmented Lagrangian associated with problem (193), which is the Lagranian plus a quadratic penalty term, penalizing the constraint violation with a factor $\rho/2 > 0$.

$$L_\rho(x, z, y) = h(x) + g(z) + \langle y, Ax - z \rangle + \tfrac{\rho}{2} ||Ax - z||_2^2. \tag{197}$$

We apply DRS to the dual problem (196). We write $\tilde{f}(y) = f^*(-A^T u)$. The goal is to find $\bar{y} \in \operatorname{dom} g^*$ which solves

$$0 \in -A\,\partial f^*(-A^T\bar{y}) + \partial g^*(\bar{y}) = \partial\tilde{f}(\bar{y}) + \partial g^*(\bar{y}). \tag{198}$$

The Douglas-Rachford-splitting seeks to equivalently find a fixed point of the operator $\frac{1}{2}I + \frac{1}{2}R_{\rho\partial\tilde{f}}R_{\rho\partial g^*}$ with $\rho > 0$. The fixed-point iteration then reads (see section 2.8)

$$\begin{aligned}
\mu^{k+1/2} &= J_{\rho\partial g^*}(\psi^k) \\
\mu^{k+1} &= J_{\rho\partial\tilde{f}}(2\mu^{k+1/2} - \psi^k) \\
\psi^{k+1} &= \psi^k + \mu^{k+1} - \mu^{k+1/2}.
\end{aligned} \tag{199}$$

With the identities (86) and (87) we can write this as

$$\begin{aligned}
\tilde{z}^{k+1} &\in \underset{z}{\operatorname{argmin}}\ \left\{g(z) - \langle\psi^k, z\rangle + \tfrac{\rho}{2}||z||_2^2\right\} \\
\mu^{k+1/2} &= \psi^k - \rho\tilde{z}^{k+1} \\
\tilde{x}^{k+1} &\in \underset{x}{\operatorname{argmin}}\ \left\{h(x) + \langle\psi^k - 2\rho\tilde{z}^{k+1}, Ax\rangle + \tfrac{\rho}{2}||Ax||_2^2\right\} \\
\mu^{k+1} &= \psi^k + \rho A\tilde{x}^{k+1} - 2\rho\tilde{y}^{k+1} \\
\psi^{k+1} &= \psi^k + \rho A\tilde{x}^{k+1} - \tilde{y}^{k+1}.
\end{aligned} \tag{200}$$

We can remove $\mu^{k+1/2}$ and μ^{k+1}, as they no longer have any explicit dependence and reorder to obtain

$$\begin{aligned}
\tilde{z}^{k+1} &\in \underset{z}{\operatorname{argmin}}\left\{g(z) - \langle\psi^k, z\rangle + \frac{\rho}{2}\|Bz\|_2^2\right\} \\
\tilde{x}^{k+1} &\in \underset{x}{\operatorname{argmin}}\ \left\{h(x) + \langle\psi^k - \rho\tilde{y}^{k+1}, Ax\rangle + \tfrac{\rho}{2}||Ax - \tilde{z}^{k+1}||_2^2\right\} \\
\psi^{k+1} &= \psi^k + \rho(A\tilde{x}^{k+1} - \tilde{y}^{k+1}).
\end{aligned} \tag{201}$$

By substituting $y^k = \psi^k - \rho A\tilde{x}^k$ we get

$$\begin{aligned}
\tilde{z}^{k+1} &\in \underset{z}{\operatorname{argmin}}\left\{g(z) - \langle y^k, z\rangle + \frac{\rho}{2}\left\|A\tilde{x}^k - z\right\|_2^2\right\} \\
\tilde{x}^{k+1} &\in \underset{x}{\operatorname{argmin}}\left\{h(x) + \langle y^{k+1}, Ax\rangle + \frac{\rho}{2}\left\|Ax - \tilde{z}^{k+1}\right\|_2^2\right\} \\
y^{k+1} &= y^k + \rho\left(A\tilde{x}^k - \tilde{z}^{k+1}\right).
\end{aligned} \tag{202}$$

We ultimately need to swap the order of the y^{k+1} and the $\tilde{x}^{k+1}$ update to get the correct dependency structure and substitute $x^{k+1} = \tilde{x}^k$ and $z^k = \tilde{y}^k$ and obtain the celebrated ADMM:

$$\begin{aligned}
x^{k+1} &\in \underset{x}{\operatorname{argmin}}\ L_\rho(x, z^k, y^k) \\
z^{k+1} &\in \underset{x}{\operatorname{argmin}}\ L_\rho(x^{k+1}, z, y^k) \\
y^{k+1} &= y^k + \rho(Ax^{k+1} - z^{k+1}).
\end{aligned} \tag{203}$$

In our case, g is only an indicator function of a point set therefore the z-update in ADMM vanishes and $z^k = Bu + b$ and ADMM reduces to the *Augmented-Lagrangian Method* (ALM). The iteration then reads

$$\begin{aligned}
x^{k+1} &\in \underset{x}{\operatorname{argmin}}\ \left\{h(x) + \langle y^k, Ax\rangle + \frac{\rho}{2}||Ax - Bu - b||_2^2\right\} \\
y^{k+1} &= y^k + \rho(Ax^{k+1} - Bu - b).
\end{aligned} \tag{204}$$

This reduction also applies to the following variants of ADMM but to give an overview of the ADMM method in all generality we will always present the full iteration with z-update.

3.5 Proximal ADMM

Let M_1, M_2 be two positive semi-definite matrices and define the induced seminorm $||u||_M^2 = \langle u, Mu \rangle$. Then the Proximal-ADMM with initial points (x_0, x_0, z_0) reads

$$\begin{aligned} x^{k+1} &\in \operatorname*{argmin}_x \left\{ h(x) + \langle y^k, Ax \rangle + \frac{\rho}{2}||Ax - z^k||_2^2 + \tfrac{1}{2}||x - x^k||_{M_1}^2 \right\} \\ z^{k+1} &\in \operatorname*{argmin}_z \left\{ g(z) - \langle y^k, z \rangle + \frac{\rho}{2}||Ax^{k+1} - z||_2^2 + \tfrac{1}{2}||z - z^k||_{M_1}^2 \right\} \\ y^{k+1} &= y^k + \rho(Ax^{k+1} - z^{k+1}). \end{aligned} \tag{205}$$

We state the main convergence results of the Prox-ADMM from Shefi and Teboulle (2014).

Theorem 3.11 (Theorem 5.1 in Shefi and Teboulle (2014))**.** *Let (x^*, z^*, y^*) be a saddle point of the Lagrangian (195) associated with (193) and let $\{x^k, z^k, y^k\}$ be a sequence generated by (205) with $M_1, M_2 \succeq 0$ and $\rho > 0$. Define further the ergodic average $\bar{x}_N = \sum_{k=1}^N x^k$ and in the same manner $\bar{z}_N, \bar{y}_N$. Then*

$$L(\bar{x}_N, \bar{z}_N, y) - L(x, z, \bar{y}_N) \leq \frac{1}{2N}\left(||x - x_0||_{M_1}^2 + ||z - z_0||_{M_2}^2 + \tfrac{1}{\rho}||y - y_0||_2^2 + \rho||Ax - z^0||_2^2 \right) \tag{206}$$

In particular we obtain

$$h(\bar{x}_N) + g(\bar{z}_N) + r||A\bar{x}_N - \bar{z}_N||_2 - h(x^*) - g(z^*) \leq \frac{c}{N} \tag{207}$$

where $c := \max_{||y|| \leq r} \left\{ ||x^* - x_0||_{M_1}^2 + ||z^* - z_0||_{M_2}^2 + \tfrac{1}{\rho}||y - y_0||_2^2 + \rho||Ax^* - z^0||_2^2 \right\}$

3.6 Function-Linearized Proximal ADMM

The Function-Linearized Proximal ADMM or FLiP-ADMM, only uses a linear model for a one part of the composite functions h or g. In the most general way, we have $h = h_1 + h_2$ and $g = g_1 + g_2$ with differentiable parts h_2 and g_2. In our model with $h = f + v$ we will use a linear model for v, using subgradient information. For the general case with also only a linear model to a part of g, we refer to Ryu and Yin (2022), ch. 8.1.. Although in the literature it is assuemd that h_2 is smooth and we have gradient information, we only have subgradient information of v. The iterates of FliP-ADMM applied to our problem (194) read

$$\begin{aligned} x^{k+1} &\in \operatorname*{argmin}_x \left\{ h(x) + \langle y^k, Ax \rangle + \langle v', x \rangle + \frac{\rho}{2}||Ax - z^k||_2^2 + \tfrac{1}{2}||x - x^k||_{M_1}^2 \right\} \\ z^{k+1} &\in \operatorname*{argmin}_z \left\{ g(z) - \langle y^k, z \rangle + \frac{\rho}{2}||Ax^{k+1} - z||_2^2 + \tfrac{1}{2}||z - z^k||_{M_1}^2 \right\} \\ y^{k+1} &= y^k + \rho(Ax^{k+1} - z^{k+1}). \end{aligned} \tag{208}$$

Note that here we assumed for FLiP-ADMM that v is differentiable. But we have only access to stochastic ϵ-subgradients of v. But since convex functions are almost everywhere differentiable (Theorem 25.5 in Rockafellar (1997)), this is not much of a drawback for the algorithmic performance. For the case where v is L-Lipschitz smooth, we obtain the same convergence rate for the primal-dual gap as in 3.11.

Theorem 3.12 (Theorem 9 in Banert et al. (2021)). *Assume $M_1 - L\,\mathrm{Id}$ and M_2 are symmetric, positive semidefinite. Then for the ergodic sequences $\bar{x}_N, \bar{z}_N, \bar{y}_N$ as defined Theorem 3.11 we have*

$$L(\bar{x}_N, \bar{z}_N, y) - L(x, z, \bar{y}_N) \leq \frac{1}{2N}\left(||x - x_0||^2_{M_1} + ||z - z_0||^2_{M_2} + \tfrac{1}{\rho}||y - y_0||^2_2 + \rho||Ax - z^0||^2_2\right) \tag{209}$$

We will use a slightly modified version of FLiP-ADMM, where we scale the dual step-size ρ by an additional factor $\varphi = 1/\rho$ such that it always is 1, that means the dual step reads

$$y^{k+1} = y^k + Ax^{k+1} - z^{k+1}. \tag{210}$$

From the literature, we know that to ensure convergence φ should be in the inerval $(0, \frac{1=\sqrt{5}}{2})$ (Ryu & Yin, 2022), where the upper bound interestingly is the golden ratio.

4 Stochastic programming formulation of stochastic optimal control problems

Before we formulate a SOC-Problem, we have to make a crucial assumption about the sequence in which the events occur in the modelled system. We can find us in one of two possible types of stochastic control-systems regarding the order of control and uncertainty: The *hazard-decision*-setting, where control u_t for time-interval $[t, t+1)$ is chosen *after* the uncertainty ξ_t during $[t, t+1)$ is revealed. These models are also sometimes referred to as *wait-and-see*-models. Or the *decision-hazard*-setting , where control u_t is chosen *before* the uncertainty ξ_t is revealed. This is sometimes referred to as *here-and-now*-setting. When modelling a SOC-Problem, we have to cautiously incorporate this control-uncertainty-order into the model. We present for both cases the SOC-model and then transform it into an equivalent SP-Model. Let us first deal with the hazard-decision-setting.

4.1 Hazard-Decision-Model

In the hazard-decision-setting, the evolution of the system over time is

$$x_1 \to \xi_1 \to u_1 \to x_2 \to \xi_2 \to u_2 \to \cdots \tag{211}$$

with x_1 and ξ_1 are known and deterministic. Suppose we are given a discrete-time stochastic optimal control problem of the form:

$$\begin{aligned}
\min_{\pi} \quad & \mathbb{E}\left[\sum_{t=1}^{T} L_t(x_t, u_t, \xi_t) + L_{T+1}(x_{T+1})\right] \\
\text{s.t.} \quad & x_1, \xi_1 \text{ given,} \\
& \text{and } \forall\, t = 1, ..., T, \\
x_{t+1} =& F_t(x_t, u_t, \xi_t), \text{ a.s.,} \\
G_t(x_t, u_t, \xi_t) =& 0, \text{ a.s.,} \\
u_t \in& \mathcal{U}_t^{ad}, \\
x_t \in& \mathcal{X}_t^{ad}, \ x_{T+1} \in \mathcal{X}_t^{ad}, \\
\sigma(u_t) \subset& \ \sigma(x_t, \xi_t)
\end{aligned} \tag{212}$$

where minimization is performed over policies $\pi = [\pi_1, ..., \pi_{T+1}]$ with $\pi_t = (x_t, u_t),\ t = 1, ..., T,\ \pi_{T+1} = x_{T+1},\ x_t \in \mathbb{R}^{n_t},\ u_t \in \mathbb{R}^{m_t}$. We incorporate the admissibility constraints $u_t \in \mathcal{U}_t^{\text{ad}},\ x_t \in \mathcal{X}_t^{\text{ad}}$ into the objective by adding the respective indicator functions. The specific form of the measurability-constraints $\sigma(u_t) \subset\ \sigma(x_t, \xi_t)$ stems from the hazard-decision-setting. Control u_t is a measurable function only of the information revealed up to time t, which in this case is $x_1, ..., x_t$ and $\xi_1, ..., \xi_t$. To formulate this as a stochastic programming problem, we define for $t = 1, ..., T+1$

$$\begin{aligned}
y_t &:= \begin{pmatrix} x_t \\ u_t \end{pmatrix} \\
A_t &:= [\mathrm{Id}_{n_t}, 0_{n_t \times m_t}] \in \mathbb{R}^{n_t \times (n_t + m_t)} \\
\mathcal{Y}_t^{ad} &:= \mathcal{X}_t^{ad} \times \mathcal{U}_t^{ad}.
\end{aligned} \tag{213}$$

Note that the component-projection matrix A_t only depends on the time index t through its dimension. Now we can write problem (224) as

$$\begin{aligned}
\min_{y} \quad & \mathbb{E}\left[\sum_{t=1}^{T} L_t(y_t,\xi_t) + L_{T+1}(A_{T+1}y_{T+1})\right] \\
\text{s.t.} \quad & x_1, \xi_1 \text{ given}, \\
& A_1 y_1 = F_0(x_1,\xi_1), \\
& \text{and } \forall\, t = 1, ..., T, \\
A_{t+1}y_{t+1} &= F_t(y_t,\xi_t), \ \text{a.s.}, \\
G_t(y_t,\xi_t) &= 0, \ \text{a.s.}, \\
\sigma(y_t) &\subset \sigma(y_{t-1},\xi_t),
\end{aligned} \tag{214}$$

where we again have incorporated the admissible sets $\mathcal{Y}_t^{ad}$ into the objective by indicator functions. Here on the first stage we have part of the decision y_1 given, x_1, ξ_1 and only have to choose the control-part u_1 of y_1. Therefore the specific form of the constraint $A_1 y_2 = F_0(x_1,\xi_1)$.

We also can formulate this in terms of dynamic-programming equations. For $t = 2, ..., T$ define the value-functions

$$\begin{aligned}
V_t(y_{t-1},\xi_{[t]}) = \min_{y_t} \quad & L_t(y_t,\xi_t) + v_{t+1}(y_t,\xi_{[t]}) \\
\text{s.t.} \quad & F_{t-1}(y_{t-1},\xi_{t-1}) - A_t y_t = 0, \\
& G_t(y_t,\xi_t) = 0
\end{aligned} \tag{215}$$

and the expected value-functions

$$v_t(y_{t-1},\xi_{[t-1]}) = \mathbb{E}_{\xi_{[t-1]}}\left[V_t(y_{t-1},\xi_{[t]})\right] \tag{216}$$

with the terminal value function

$$V_{T+1}(y_T,\xi_{[T+1]}) = L_{T+1}(A_{T+1}F_T(y_T,\xi_t)). \tag{217}$$

At the first stage, we solve

$$\begin{aligned}
\min_{y_1} \quad & L_1(y_1,\xi_1) + v_2(y_1,\xi_1) \\
\text{s.t.} \quad & A_1 y_1 = F_0(x_1,\xi_1), \\
& G_1(y_1,\xi_1) = 0
\end{aligned} \tag{218}$$

since L_{T+1} is only a function of the n_{T+1} x_{T+1}-components of y_{T+1} and therefore u_{T+1} is free. Since the final cost is a deterministic function of y_T, we can define a value for ξ_{T+1} arbitrarily since we need it only for notational completeness. By taking the realizations $\xi_{[t]}$ parametrically into the t-stage DPE and taking the expectation over all future $\xi_{t+1}, ..., \xi_T$, we satisfy the non-anticipativiy-constraint with this formulation. We can view (218) as an instance of the more generic optimization problem

$$\begin{aligned}
V(a,\xi) = \min_{y} \quad & f(y,\xi) + v(y,\xi) \\
\text{s.t.} \quad & F(a) - Ay = 0, \\
& G(y,\xi) = 0
\end{aligned} \tag{219}$$

parametrized through a and ξ. From here on we can procede as in Lan and Zhou (2021). The respective lagrangian of (219) is

$$\mathcal{L}(y, \lambda_1, \lambda_2, a, \xi) = f(y, \xi) + v(y, \xi) + \lambda_1^T(F(a) - Ay) + \lambda_2^T G(y, \xi) + \mathbb{I}_{\mathcal{Y}^{ad}}(y). \tag{220}$$

The primal problem is

$$\min_{y \in \mathcal{Y}^{ad}} \max_{\lambda_1, \lambda_2} \mathcal{L}(y, \lambda_1, \lambda_2, a, \xi) \quad (P) \tag{221}$$

and the dual problem is

$$\max_{\lambda_1, \lambda_2} \min_{y \in \mathcal{Y}^{ad}} \mathcal{L}(y, \lambda_1, \lambda_2, a, \xi) \quad (D) \tag{222}$$

If we assume convexity of the functions $L_t(\cdot, \xi),\ v(\cdot, \xi),\ g(\cdot, \xi)$ for all ξ and convexity of the set $\mathcal{Y}^{ad}$ and Slaters Constraint Qualifiaction, we have strong duality between (P) and (D) and we can solve instead of (219) the saddle point problem (P).

4.2 Decision-Hazard-Model

In the hazard-decision-setting, the evolution of the system over time is

$$x_1 \to u_1 \to \xi_1 \to x_2 \to u_2 \to \xi_2 \to \cdots \tag{223}$$

with x_1 known and deterministic. Here the SOC model is the same as in (224) except of the measurability constraint:

$$\begin{aligned}
\min_{\pi} \quad & \mathbb{E}\left[\sum_{t=1}^{T} L_t(x_t, u_t, \xi_t) + L_{T+1}(x_{T+1})\right] \\
\text{s.t.} \quad & x_1, \xi_1 \text{ given}, \\
& \text{and } \forall\, t = 1, ..., T, \\
x_{t+1} =& F_t(x_t, u_t, \xi_t), \\
G_t(x_t, u_t, \xi_t) =& 0, \\
u_t \in& \mathcal{U}_t^{ad}, \\
x_t \in& \mathcal{X}_t^{ad}, x_{T+1} \in \mathcal{X}_{T+1}^{ad}, \\
\sigma(u_t) \subset& \sigma(x_t).
\end{aligned} \tag{224}$$

Here, all information available at time t is contained in x_t, hence u_t is a measurable function of x_t, which is also called a state-feedback-solution. Here we can not transform the SOC Problem into a SP in the same straightforward manner as the hazard-decision-model, since we do not know the all parameters of the optimization model during the period for which the decision has to be made. One possible way to model this is as a chance-constraint SP with probabilisitc constraints. Therefore we must set permissible tolerances $\epsilon_{1,t}, \epsilon_{2,t}$, which are the maximal chances we are willing to take that a constraint is violated.

$$
\begin{aligned}
\min_{y} \quad & \mathbb{E}\left[\sum_{t=1}^{T} L_t(y_t,\xi_t) + L_{T+1}(A_{T+1}y_{T+1})\right] \\
\text{s.t.} \quad & x_1 \text{ given}, \\
& A_1 y_1 = x_1, \\
& \text{and } \forall\, t = 1,...,T, \\
& A_{t+1}y_{t+1} = F_t(y_t,\xi_t), \\
& \mathbb{P}\left[G_t(y_t,\xi_t) = 0\right] \geq 1-\epsilon_{1,t}, \\
& \mathbb{P}\left[y_t \in \mathcal{X}_t^{ad} \times \mathcal{U}_t^{ad}\right] \geq 1-\epsilon_{2,t}, \\
& \sigma(y_{t+1}) \subset \sigma(y_t)
\end{aligned}
\tag{225}
$$

where again $L_{T+1}(A_{T+1}y_{T+1}) = \tilde{L}_{T+1}(A_{T+1}y_{T+1}) + \mathbb{I}_{\mathcal{Y}_{T+1}^{ad}}(A_{T+1}y_{T+1})$. Again we can formulate this in terms of dynamic-programming equations. For $t = 2,...,T$ define the value-functions

$$
\begin{aligned}
V_t(y_{t-1},\xi_{[t-1]}) = \min_{y_t} \quad & \mathbb{E}_{|\xi_{[t-1]}}\left[L_t(y_t,\xi_t) + V_{t+1}(y_t,\xi_{[t]})\right] \\
\text{s.t.} \quad & F_{t-1}(y_{t-1},\xi_{t-1}) - A_t y_t = 0, \\
& \mathbb{P}\left[G_t(y_t,\xi_t) = 0\right] \geq 1-\epsilon_{1,t}, \\
& \mathbb{P}\left[y_t \in \mathcal{X}_t^{ad} \times \mathcal{U}_t^{ad}\right] \geq 1-\epsilon_{2,t}
\end{aligned}
\tag{226}
$$

and at the first stage we solve

$$
\begin{aligned}
\min_{y_1} \quad & \mathbb{E}\left[L_1(y_1,\xi_1) + V_2(y_1,\xi_{[1]})\right] \\
\text{s.t.} \quad & A_1 y_1 = x_1, \\
& \mathbb{P}\left[G_1(y_1,\xi_1) = 0\right] \geq 1-\epsilon_{1,1}, \\
& \mathbb{P}\left[y_1 \in \mathcal{X}_1^{ad} \times \mathcal{U}_1^{ad}\right] \geq 1-\epsilon_{2,1}
\end{aligned}
\tag{227}
$$

5 Microgrid Optimization

5.1 Modelling

Here we introduce the microgrid SOC problem from Pacaud et al. (2021). We then transform into a MSSP as shown in the section before. We deal with a microgrid consisting of a set of nodes $\mathcal{N}$ nodes and a set of edges $\mathcal{E}$. The time horizon is $T = 3$ (in hours). Denote $N = |\mathcal{N}|$ and $E = |\mathcal{E}|$. At each node, there is a prosumer, an entity which can consume and produce power. Some prosumers have a solar panel, some have a battery in their home. All prosumers can exchange energy through the microgrid within the line limits of the edges. Additionally every prosumer is connected to the wholesale market and can buy and sell energy from/to the external grid. The residual load in each period, which is solar-production minus consumption, is unknown in advance and therefore introduce makes the problem stochastic. A central energy managment system (EMS) controls all flows and control variables for every prosumer on the microgrid. In each period, the goal of the EMS is to minimize the cost of electricity purchased from the grid and the cost arising when energy flows through the edges of the network, possibly due to resistive losses or a fixed toll for the edges. We take whole-sale electricity prices as given.

5.1.1 Variables and Parameters

All variables are stated in MW respective MWh.

State Variables
s_t^n: Storage state of battery . We impose the capacity limit $0 \leq s_t^n \leq \overline{S}^n$. Define the admissible state set $\mathcal{S}^{ad} := \left[0, \overline{S}^n\right]$.

Controls
$u_t^n = u_t^{ne,n}, u_t^{bd,n}, (u_t^{bc,n})$: Amount of energy charged into resp. discharged from battery and energy exchanged with external grid during time interval $[t, t+1)$. We impose the power limits $0 \leq u_t^{bc,n}, u_t^{ne,n} \leq \overline{U}^{b,n}$, $-\overline{U}^{b,n} \leq u_t^{bd,n} \leq 0$. Note that we write the model such that both flows from and to the battery appear as non-negative variables. The set of admissible controls is $\mathcal{U}^{ad} := \left[-\overline{U}^{ne,n}, \overline{U}^{ne,n}\right] \times \left[0, \overline{U}^{b,n}\right] \times \left[0, \overline{U}^{b,n}\right]$

Uncertainty
$\xi_t^n = d_t^n$: Electricity demand minus solar production during $[t, t+1)$. A major advantage of the Stochastic Approximation method is that it can deal with stagewise depenency of the uncertainty process, therefore we will allow it here. If we want to refer to the history of the stochastic process up to time t we write $\xi_{[t]}^n := (\xi_1^n, ..., \xi_t^n)$

Flows
At each node n we denote the power flowing into this node as f_t^n and at each edge e q_t^e. We have the node-edge incidence matrix $C \in \{-1, 0, 1\}^{N \times E}$. The edge-flows are non-negative $0 \leq q_t^e \leq \overline{Q}_t^e$ and for the node-flows we have $\underline{F}_t^n \leq f_t^n \leq \overline{F}_t^n$

For alle parameters and variables $u_t^n, u_t^{bc,n}, \xi_{t+1}^n$ etc., if we write them with an upper index n (or e), we mean the respective quantity at node n (edge e). If we leave out such an index, we mean the vector of the quantities at all nodes (edges). For example $u_t^{bc} = (u_t^{bc,1}, ..., u_t^{bc,N})^\top \in \mathbb{R}^N, \; q_t = (q_t^1, ...q_t^E) \in \mathbb{R}^E$

We impose that all nodes are identical, that is we have for all controls and states n and e the same admissible sets.

5.1.2 Dynamics

Battery dynamics

The state of the battery at time $t+1$ is

$$s_{t+1}^n = \alpha_b s_t^n + \Delta T \left(\gamma_c u_t^{bc,n} - \frac{1}{\gamma_d} u_t^{bd,n} \right) \tag{228}$$

with α_b being the roundtrip efficiency and γ_d, γ_c are the charge resp. discharge efficiency.

5.1.3 Balances

Local load Balance

$$f_t^n = u_t^{ne,n} - d_t^{el,n} - u_t^{bc,n} + u_t^{bd,n}$$

Network flow balance
We make a simplified physical model and model the network dynamics with Kirchhoff's current law. That is we have the network-flow balance equation

$$Cq_t + f_t = 0 \tag{229}$$

5.1.4 Objective function

The objective is to minimize the cost of electricity purchased from the grid and the cost arising when energy flows through the edges of the network, possibly due to resisitve losses or a fixed toll for the edges.

Cost of electricity
The cost of electricity purchased on the wholesale market during the time interval $[t, t+1)$

$$L_t^n \left(s_t^n, u_t^n, \xi_t^n \right) = p_t^{el} u_t^{ne,n}. \tag{230}$$

edge cost
Transportation cost for energy flow q_t^e on edge e is modelled by a quadratic function

$$l_t^e(q_t^e) = c_2^e (q_t^e)^2 + c_1^e q_t^e + c_0^e \tag{231}$$

with coeffictions $c_2^e > 0, c_1^e, c_0^e$. Note that this cost function is strongly convex.

Battery degradation

To incorporate degradation of the battery with every charge-discharge-cycle into our model, we penalize charging and discharging with a small extra cost. We therefore add the term

$$p_t^{bat} u_t^{bc,n} - p_t^{bat} u_t^{bd,n}, \tag{232}$$

with $p_t^{bat} > 0$. Note that $u_t^{bd,n} \geq 0$ by assumption.

5.1.5 SOC-Model

The SOC problem of the energy managment system then reads

$$\begin{aligned}
\min_{x,u,f,q} \quad & \mathbb{E}\left[\sum_{t=1}^{T}\left(\sum_{n\in\mathcal{N}} L_t^n\left(s_t^n, u_t^n, \xi_t^n\right) + \sum_{a\in\mathcal{E}} l_t^e(q_t^e)\right)\right] && \text{(233a)}\\
\text{s.t.} \quad & x_1 = \tilde{x}_1, \quad \xi_1 = \tilde{\xi}_1 \text{ fix }, && \text{(233b)}\\
& f_t^n = u_t^{ne,n} - d_t^n - u_t^{bc,n} + u_t^{bd,n} \qquad , && \text{(233c)}\\
& s_{t+1}^n = \alpha_b s_t^n + \Delta T\left(\gamma_c u_t^{bc,n} - \frac{1}{\gamma_d} u_t^{bd,n}\right) t = 1, ..., T-1, && \text{(233d)}\\
& Cq_t + f_t = 1 \qquad , && \text{(233e)}\\
& s_t^n \in \mathcal{S}^{ad}, \; u_t^n \in \mathcal{U}^{ad}, \; f_t^n \in \mathcal{F}_t^{ad}, \; q_t^e \in \mathcal{Q}^{ad}, && \text{(233f)}\\
& \sigma(u_t) \subset \sigma(\xi_1, \ldots, \xi_t) && \text{(233g)}
\end{aligned}$$

5.1.6 SP-Formulation

We now want to translate the microgrid model into a MSSP. We have that at every node n, edge e and timestep t, the following must hold:

1. $u_t^{ne,n} - u_t^{bc,n} + u_t^{bd,n} - f_t^n = d_t^n$ (Local load balance (233c))
2. $s_{t+1}^n = \alpha_b s_t^n + \Delta T(\gamma_c u_t^{bc,n} - \frac{1}{\gamma_d} u_t^{bd,n})$ (Battery Dynamics (233d))
3. $Cq_t + f_t = 0$ (Network Flow Balance (233e))
4. $s_t^n \in \mathcal{S}^{ad}, \; u_t^n \in \mathcal{U}^{ad}, \; f_t^n \in \mathcal{F}^{ad}, \; q_t^e \in \mathcal{Q}^{ad}$ (Admissibility of Controls (233f))

We first combine the states, controls and flows to a decision variable:

$$\begin{aligned} x_t := (&u_t^{ne,1}, u_t^{bc,1}, u_t^{bd,1}, s_t^1, f_t^1, u_t^{ne,2}, u_t^{bc,2}, u_t^{bd,2}, s_t^2, f_t^2, ..., \\ &u_t^{ne,n}, u_t^{bc,n}, u_t^{bd,n}, s_t^n, f_t^n, q_t)^\intercal \in \mathbb{R}^{5N+E} \end{aligned} \tag{234}$$

Since our final matrices will have a block-structure let us first define the block matrices. Since the balances and the dynamics do not vary over time, the matrices will not depend on the time index t. Define the matrices

$$A = \begin{array}{c} \begin{matrix} u_t^{ne} & u_t^{bc} & u_t^{bd} & s_t & f_t \end{matrix} \\ \begin{pmatrix} 1 & -1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \in \mathbb{R}^{3\times 5},$$

$$B = \begin{array}{c} \begin{array}{ccccc} u_{t-1}^{ne} & u_{t-1}^{bc} & u_{t-1}^{bd} & s_{t-1} & f_{t-1} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \Delta T\gamma_c & -\Delta T/\gamma_d & \alpha_b & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \in \mathbb{R}^{3\times 5}$$

Furthermore denote $\mathcal{C}^{(n)}$, $n = 1, ..., N$ the n-th row of the node-edge incidence matrix $\mathcal{C}$. Define the matrix

$$C^n = \begin{array}{c} q_t \\ \begin{pmatrix} 0 \\ 0 \\ \mathcal{C}^{(n)} \end{pmatrix} \end{array} \in \mathbb{R}^{3\times E}.$$

The first row of A and B corresponds to the local load balance (233c), the second row to the battery dynamics (233d) and the last row together with the last row of C^n to the network flow balance (233e).
Then we define the block-matrices

$$\mathbf{A} = \begin{bmatrix} A & 0 & \dots & & C^1 \\ 0 & A & & & \vdots \\ \vdots & & \ddots & & \\ 0 & \dots & & A & C^N \end{bmatrix} \in \mathbb{R}^{3N\times(5N+E)}$$

and

$$\mathbf{B} = \begin{bmatrix} B & 0 & \dots & & 0 \\ 0 & B & & & \vdots \\ \vdots & & \ddots & & \\ 0 & \dots & & B & 0 \end{bmatrix}. \in \mathbb{R}^{3N\times(5N+E)}$$

Define the right-hand side vector

$$b_t = (d_t^1, 0, 0, d_t^2, 0, 0, \dots, d_t^n, 0, 0)^\intercal \in \mathbb{R}^{3N}.$$

Further define the set of admissible solutions

$$X_t^{ad} = \left[\mathcal{U}^{ad} \times \mathcal{S}^{ad} \times \mathcal{F}^{ad}\right]^N \times (\mathcal{Q}^{ad})^E$$

So in this model the vector b_t introduces randomnes to the problem via the uncertain demand for electricity and hot water. It is therefore a random variable. Rewrite the objective function as

$$\mathcal{J}_t(x_t) = \sum_{n\in\mathcal{N}} p_t^{el^n} u_t^{ne,n} + \sum_{e\in\mathcal{E}} l_t^e(q_t^e) + \sum_{n\in\mathcal{N}} p_t^{bat}(u_t^{bd,c} - u_t^{bd,n})$$

The DPEs of the MSSP then read

$$\begin{aligned} \min_{x_1} \quad & \mathcal{J}_1(x_1) + v_2(x_1, \xi_1) && (235a) \\ \text{s.t.} \quad & \mathbf{A}x_1 = b_t, && (235b) \\ & x_1 \in \mathcal{X}^{ad} && (235c) \end{aligned}$$

and for $t = 2, ..., T$

$$
\begin{aligned}
v_t(x_{t-1}, \xi_{[t-1]}) = \min_{x_t} \quad & \mathbb{E}\left[\mathcal{J}_t(x_t) + v_{t+1}(x_t, \xi_{[t]}) \middle| \xi_{[t-1]}\right] && \text{(236a)} \\
\text{s.t.} \quad & \mathbf{A}x_t - \mathbf{B}x_{t-1} = b_t, && \text{(236b)} \\
& x_t \in \mathcal{X}^{ad} && \text{(236c)}
\end{aligned}
$$

5.2 Numerical Results

In this section we study numerical behaviour of DSA when applied to the stochastic optimal control of the microgrid model introduced above. We run the DSA with FLiP-ADMM and PDHG as saddle-point solvers with different algorithmic parameters for different sizes of the grid. Our goal is to study the convergence and feasibility behaviour of the presented DSA-Algorithm. We call the DSA with PDHG as saddle-point solver PDHG-DSA and DSA with FLiP-ADMM PMM-DSA, since in our setting we do not have a z-update in the FLiP-ADMM, therefore it reduces to a function linearized version of Proximal Method of Multipliers. To solve the upcoming quadratic optimization problems in the primal step of PDHG and FLiP-ADMM we employ Gurobi v12.0.2 (Gurobi Optimization, LLC, 2025). We assume that in the microgrid the power-demand minus solar feed-in at each node $d_t^n \sim \mathcal{N}(1, 1/2),\ n \in \mathcal{N},\ t = 0, ..., 2$[3] is normally distributed with $\mu = 1$ and $\sigma^2 = 1/2$. Therefore randomness enters our model only in the RHS b_t of the coupling constraint . Note that it is also possible to optimize a model with random data A, B and cost-function-parameter c. Here we have stage-wise independent randomness but we emphasize that it is a major advantage of our solution approach that it can handle also stage-wise dependent randomness. There is a comprehensive literature on scenario tree modelling in general (see e.g. Heitsch and Römisch (2009) Dupačová et al. (2000), Pranevicius and Sutiene (2007)) and on modelling uncertainties in microgrids in particular (see e.g. Zakaria et al. (2020), Kumar and Saravanan (2017)). Also since we do not possess reliable data for the wholesale-price and the transmission losses, which determine the linear respective the quadratic term of the objective function, we manually set the wholesale prices and the quadratic coefficients for the transmission losses. Besides the stochastic case we also solved the problem for the deterministic case to understand convergence behaviour. Therefore we set $\mu = 1, \sigma = 0$. The objective function and recourse matrices are defined as in section 5.1, that is we have the objective function

$$\mathcal{J}(x) = \sum_{t=0}^{2} \left[x_t^T C x_t + p_t^T x_T \right] \tag{237}$$

We set the factor for the quadratic transmission price to $c_2^e = 1/2$ and the following electricity prices for each stage: $p_0 = 1,\ p_1 = 5,\ p_2 = 10$. Furthermore we set the battery degradation cost $p_t^{bat} = 1,\ t = 0, 1, 2$. To get a better sense for the inter-stage dynamics we set the battery efficiency parameters $\alpha_b, \gamma_c, \gamma_d = 1$. We also tested smaller values for the transmission price c_2^e, but even for a price of zero, no substantial flows over the network arcs are computed. This is probably since we model all nodes to have the same physical properties and constraints. The terminal reward for the battery is $R = 10$ EUR/MWh, that means we assume electricity price will be -10 EUR/MWh and therefore penalizes a full battery with 10 EUR/MWh for each MWh which is in the battery in the end and therefore encourage to use energy from the battery now.

We model three different microgrids with the following dimensions.

[3]In order to keep notation consistent with the Python code, in this section we enumerate the stages with $t = 0, 1, 2$

Model	N	E	dim A	$\|\|A\|\|_2$
mg_3	3	3	9×18	2.49
mg_10	10	8	30×58	2.59
mg_100	100	80	300×580	2.96

Table 1: Network models and optimization model size.

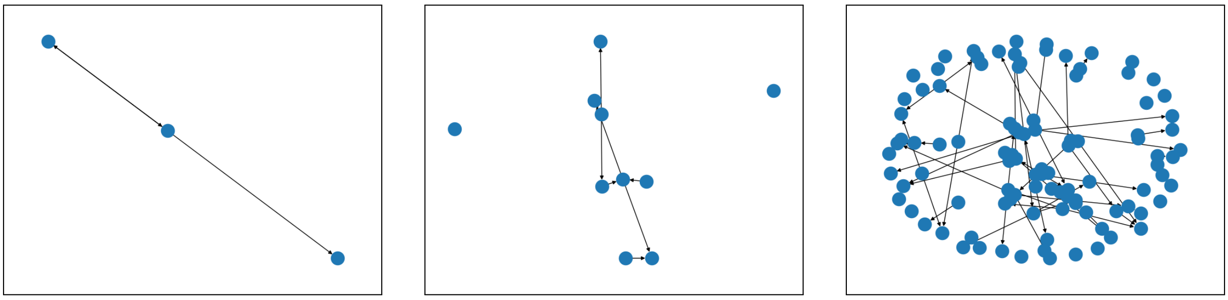

Figure 1: The three directed networks

We use the euclidean distance $\frac{1}{2}||\cdot - \cdot||_2^2$ as the Bregman distance in the primal step of PMM respective PDHG. Furthermore we impose the following bounds on the decision-variables at each node, respective edge and each timestep, which define the admissible sets X_t^{ad}, $t = 0, 1, 2$.

$[MW]$	u^{ne}	u^{bc}	u^{bd}	s	f	q
LB	-10	0	0	0	-1	0
UB	10	2	2	2	1	1

Table 2: Power-limits for nodes and edges

This means we set our model such that we can in every time step fully charge/discharge the battery. Since the parameter setting in Theorem 3.9 are merely theoretical guarantees, and N_t becomes that large that it lead to impractical run times, we conducted a numerical study with different parameter settings.

Benchmark-Solution

We know from Theorem 3.9 that an approximate solution from a run of DSA satisfies the constraints in expectation up to a error δ. This means we can benchmark our model by solving a deterministic linearly constraint quadratic problem, where the demand in the RHS-vector b is exactly the mean $\mu = 1$. We compare only the mg_3 iterates to this benchmark since for larger problems, we would loose sight because of the large number of variables. The Benchmark solution reads

t/u^n	$u_t^{ne,n}$	$u_t^{bd,n}$	$u_t^{bc,n}$	s_t^n	f_t^n
			$n=1$		
$t=0$	3	2	0	0	0
$t=1$	1	0	0	2	0
$t=2$	-1	0	2	2	0
			$n=2$		
$t=0$	3	2	0	0	0.0001
$t=1$	1	0	0	2	0.0001
$t=2$	-1	0	2	2	0.0001
			$n=3$		
$t=0$	3	2	0	0	-0.0001
$t=1$	1	0	0	2	-0.0001
$t=2$	-1	0	2	2	-0.0001
	q_t^1	q_t^2	q_t^2		
$t=0$	0.0001	0.0001	0.0001		
$t=1$	0.0001	0.0001	0.0001		
$t=2$	0.0001	0.0001	0.0001		

Table 3: Benchmark-solution of mg_3 for all n, e

The exact objective values are

	Optimal Value
mg_3	6
mg_10	0
mg_100	1400

Table 4: Benchmark Objective values

This solution is intuitive since it is economically optimal to buy electricity at the first stage when the price is at 1 EUR/MWh and fully charge the battery, then only buy electrcity to satisfy demand at the second stage where the price is 5 EUR/MWh and leave the battery full to discharge it to satisfy demand and sell 1 MWh back to the market, when prices are 10 EUR/MWh in the last stage. Also we assume that a full battery will cost us 10 EUR/MWh, therefore we wont keep any energy in the battery. The arc cost for transferring energy x is quadratic and is $1/2\,x^2$. Therefore no exchange takes place in our network, since transmission costs are too high.

DSA-solutions

We solve 7 models, mg_3, mg_10 and mg_100, each with and without stochastics with PMM-DSA and mg_3 with PDHG-DSA. We choose three different iteration numbers $Niter_{\{a,b,c\}} = \{[50, 50, 50], [50, 100, 50], [100, 200, 100]\}$. We choose N_1 in $Niter_b$ and $Niter_c$ larger than the respective N_0 and N_2 since (178) suggest to put more iterations on the middle stage. This is also intuitive, since the second stage constraint violation depends on x_0 and x_1 and the third stage constraint violation depends on x_1 and x_2, therefore x_1 goes twice into the overall model. We run the mg_100-model only with the first two iteration counts, since the last one would take unreasonably long. In our study we especially focus on feasibility. The box constraints of Table 2 are always satisfied, since we enforce them explicitly in the computation of the iterates with Gurobi. Whereas the linear coupling constraints only go into the lagrangian with a mupltiplier term or in case of PMM as a solver additionally with a quadratic penalty term. For PMM we compare the two parameter settings

	τ	ρ	φ
$param_0$	$t=0:1/10$ $t=1:1/10$ $t=2:1/10$	$t=0:10$ $t=1:10$ $t=2:10$	$t=0:0.1$ $t=1:0.1$ $t=2:0.1$
$param_1$	$t=0:1/10$ $t=1:1/10$ $t=2:1/10$	$t=0:0.1$ $t=1:10$ $t=2:10$	$t=0:10$ $t=1:0.1$ $t=2:0.1$

Table 5: Parameter settings for PMM

and for PDHG we choose the parameter settings[4]

	$1/\tau$	$1/\eta$
$param_0$	$t=0:1/10$ $t=1:1/10$ $t=2:1/10$	$t=0:1$ $t=1:1$ $t=2:1$
$param_1$	$t=0:1/10$ $t=1:1/10$ $t=2:1/10$	$t=0:1$ $t=1:10$ $t=2:10$

Table 6: Parameter settings for PDHG

Note that in the parameter setting $param_1$ at $t=0$ we have $\varphi = 10$ which is larger than $\frac{\sqrt{5}+1}{2} \approx 1.6$. But we still get interesting results. Even though Lemmas 3.6, 3.7 and 3.9 suggest larger values for τ and η in PDHG-DSA, we got better results for the smaller parameter settings as in Table 6. For PDHG-DSA the parameter setting $param_0$ and the first stage of setting $param_1$ satisfy the parameter bound $\tau\eta > 2||A||_2$ (see Table 1 for the respective norm of A).

We report for all models the sum of deviations from the linear coupling terms. Economically this means we would have bought/sold too much energy, resulting in an imbalance in the Balancing group. This would lead to an activation of positive resp. negative balancing energy, resulting in costs. Since by German federal law StromNZV §4 it is prohibited to trade against balancing energy, possibly in expectation of a more favorable balancing energy price than the market price, we have to enforce this constraints and can not put them, multiplied by the expected balancing energy price, into the objective.

We run the computations on a 2024 MacBook Pro with M4 Chip. The computation times are in the order of

[4]Note that in order to maintain consistency with Lan and Zhou (2021), the stepsizes for PDHG are stated in the inverse form compared to the step sizes of ADMM.

	N_a	N_b	N_c
mg_3	90s	180s	1400s
mg_10	110s	220s	1720s
mg_100	630s	1240s	9792s

Table 7: Computation times in seconds

We see in Table 7 that the time-complexity grows moderately with the problem dimension, especially the step from $\dim A = 9 \times 18$ in mg_3 to $\dim A = 30 \times 58$ in mg_10 result only in a small relative increase in computation times. This is due to the good scaling of the underlying Gurobi-solver with respect to the dimension of the problem.

When comparing results for the mg_3-model solved with PDHG-DSA and with PMM-DSA, we see from Tables 9 and 17 that especially for the later stages, PMM-DSA gives lower relative errors for the constraints then PDHG-DSA. We also see the bad oscillation occurring at the solution process of PDHG-DSA in Figure 2. In comparison the trajectories of PMM-DSA for the mg_3-model in Figure 3 are more monotone. These findings align with our intuition since PMM penalizes quadratic constraint-violation and also the quadratic term in the objective leads to smoother trajectories since the iteration-sub-problems are strongly convex with larger modulus ρ.

Interestingly, in our setting adding stochasticity to the problem does not make results much worse, in fact for some parameter-settings results for the stochastic model are by a tiny margin better. Compare for example the sum of constraint violations of mg_10 and mg_10_stoch in Tables 12 and15. A possible explanation is that due to the stochasticity, the solver gets not stuck at an oscillation equilibrium since stochastic perturbations lets iterates deviate from oscillation equilibrias. This can be seen in the plots of the constraint violation at stages 1 and 2 for mg_10 (Figure 4) and mg_10_stoch (Figure 7).

The obtained objective values in Tables 23 - 28 are quite off the benchmark objective values in Table 4. This is especially the case, if relative constraint violations are high. Compare for example mg_10 and mg_10_stoch. Here, again stochasticity seems to lead to better convergence results. A next step towards better convergence behaviour would be to use adaptive step-size policies.

When using DSA for the operation of a microgrid, we are confronted with the problem that DSA is only pracically solvable up to 3-5 time steps due to its exponential complexity in the number of stages. But a typical microgrid optimization would need to optimize over 96 time steps, corresponding to the 24 quarter-hour products traded in the Continous Intraday-market. Nevertheless, a possible deployment of DSA would be as an online optimizer. Then we would add a value function to the last stage, incorporating all future gains/losses of the day as a function of the state. Such a function could be externally estimated. We additionally would require it to be convex. In this way we could run DSA in an online manner to get optimal trading decisions for the next 3-5 time steps and update at each step the distribution of the uncertainty and the value function for the remaining trading products.

6 Outlook

We presented a framework to solve stochastic optimal control problems with first order methods for large scale saddle-point problems. We first formulated a SOC as a MSSP and then generalized the approach of Lan and Zhou (2021) by incorporating different first-order methods to solve the upcoming saddle-point problems in our specific problem of controlling an electric microgrid under uncertainty. In particular, we thereby overcame the slow feasibility convergence of the PDHG-method used by Lan & Zhou. In energy-system applications feasibility of an approximate solution is crucial to sustain the stability of the network. We finally did a numerical study of the performance of the different methods for the solution of the microgrid problem.

This thesis is the first part of a research project in which we want to thoroughly study and analyze the potential of first order methods to solve stochastic optimal control problems. Here we focused on the conceptual framework and collected numerical intuition. The next step will be to analyze the mathematical properties of the methods presented here. That is we want to obtain convergence rates and error bounds of the methods introduced here. Besides the Proxmial Method of Multipliers presented here, we also want to calculate convergence rates for variants of PDHG like fully intertial PDHG as presented in Chambolle and Pock (2016) and other refined versions of existing saddle-point problem solving algorithms. These methods all stem from an operator splitting and are therefore closely related, but have slightly different convergence properties. For a good overview of these relations see chapter 3 of Ryu and Yin (2022). It will be part of our investigation to see if there are variants of operator splittings which are better suited to our MSSP application than others. Or if it even might be possible to state the problem in a modular way and choose the solver for the upcoming saddle-point problems for each problem instance separately. Since the choice of the algorithmic parameters for most of these methods is still a research topic even for the static case, we will have to specifically pay attention on how to choose these parameters in the dynamic programming context. Especially there are variants of these algorithms with adaptive step size policies, see e.g. Chambolle et al. (2024). It might be also useful in our stochastic context to apply an adaptive step-size policy. All these operator splitting methods also have different properties in balancing feasibility and optimallity. Depending on the specific problem we want to solve, we need to take this into account.

We fixed the number of stages to $T = 3$. This was done for two reasons. On the one hand, the $T = 3$-case reveals the full problem structure and we believe it will be straightforward to generalize our findings to the $T > 3$ case. On the other hand, since the number of overall solutions of a saddle-point problem grows exponentially in the number of stages, even for the $T = 3$ case the computation for realistic problem sizes takes minutes to hours. Since electric power is traded within 15-min time-slices, computation times which well exceed 10 minutes would be unacceptable.

Our approach solves a MSSP, which means if we want to solve a SOC, we first have to transform it into a MSSP as presented in section 4. Therefore as part of our future research we will also investigate how to apply first order methods to the DPEs which come directly from the SOC. The DPEs for a SOC problem read

$$\begin{aligned}
\min_{u,x} \quad & f_t(x_t, u_t, \xi_t) + v_{t+1}(x_{t+1}, \xi_{[t]}) \\
s.t. \quad & x_{t+1} = A_t(x_t, u_t, \xi_t) \\
& G_t(x_t, u_t, \xi_t) = 0 \\
& x_t \in X_t, \ u_t \in U_t. \\
& \sigma(u_t) \subset \sigma(x_t, \xi_t)
\end{aligned} \tag{238}$$

with a linear dynamic A_t. This is exactly the form of variable coupling ADMM can deal with. So when investigating how to solve SOC Problems directly with first order methods, ADMM will be our first method under consideration.

Another research direction in this area builds on the *Dual Approximate Dynamic Programming* (DADP) method (Barty et al. (2010), Pacaud et al. (2021), Leclere (2014)). This method solves a distributed optimization problem by dualizing the coupling constraint and then optimizing the network flows. Their iterative scheme is based on Dual Ascent. It might be interesting to see the effect of replacing the dual ascent subroutine with a splitting method. Also, here convergence bounds and choice of parameters are the open questions one needs to address. Finally, it might be interesting to compare the DSA first order methods with the standard DADP and also the modified DADP with the Dual Ascent replaced.

All these approaches rely on the fact that we are in the hazard-decision setting, that is the random variable ξ_t which acts during time period $[t, t+1)$ is revealed before making the decision for time period $[t, t+1)$. This allows us to take the random variable as a parameter into the dynamic programming equation at each stage. It is not clear if and how a saddle-point problem based solution method can be deployed for decision-hazard problems as presented in section (4.2). It would be another research area to build up theory how to formulate DPEs in this context.

For the modelling of the microgrid underlying our optimization we choose a very simplified approach. We model the grid coupling by a simple linear equation representing Kirchhoff's current law. For deployment of this control algorithm by an energy grid operator, it would be necessary to make a more physically accurate model. This could make the solution of the problem harder since AC-power-flow equations are non-linear and even non-convex.

With this thesis we lay out the main points of a first-order approach to stochastic optimal control. We analyzed all the components and got promising first numerical results. Now it will be our task to plug the modules together in order to come up with a convergence analysis of the proposed first-order methods for stochastic dynamic programming.

7 Appendix

7.1 Proofs

Proof of Lemma 3.3

Proof. Let $x \in X^\circ$ and define $v := P_x(y)$. Since v is of the form $\operatorname{argmin}_{z\in X}\{w(z) + \langle p, z\rangle\}$, $v \in X^\circ$ and thus ψ is differentiable at v. As $\nabla_v \tau D(v,x) = \tau(\nabla\psi(v) - \nabla\psi(x))$, we get from the optimality conditions of (137)

$$(\tau(\nabla\psi(v) - \nabla\psi(x)) + y)^\intercal (v-u) \leq 0 \quad \forall u \in X. \tag{239}$$

For $u \in X$ we have therefore

$$\begin{aligned}\tau(D(u,v) - D(u,x)) &= \tau[\psi(u) - \nabla\psi(v)^\intercal(u-v) - \psi(v)] - \tau\left[\psi(u) - \nabla\psi(x)^\intercal(u-x) - \psi(x)\right] \\ &= (\tau(\nabla\psi(v) - \nabla\psi(x)) + y)^\intercal(v-u) + y^\intercal(u-v) \\ &\quad - \tau\left[\psi(v) - \nabla\psi(x)^\intercal(v-x) - \psi(x)\right] \\ &\leq y^\intercal(u-v) - \tau D(v,x).\end{aligned} \tag{240}$$

With the Young-inequality we have

$$y^\intercal(x-v) \leq \frac{||y||_*^2}{2\tau\alpha} + \frac{\tau\alpha}{2}||x-v||_2^2. \tag{241}$$

and $\tau D(v,x) \geq \frac{\tau\alpha}{2}||x-v||_2^2$ since $D(\cdot,x)$ is α-strongly convex. Together we get

$$\begin{aligned}\tau D(u,v) - \tau D(u,x) &\leq y^\intercal(u-v) - \tau D(v,x) \\ &= y^\intercal(u-x) + y^\intercal(x-v) - \tau D(v,x) \\ &\leq y^\intercal(u-x) + \frac{||y||_*^2}{2\tau\alpha}\end{aligned} \tag{242}$$

□

7.2 Tables

PDHG

	$Niter_0$	$Niter_1$	$Niter_2$
$param_0$	$t = 0 : 0.0646$ $t = 1 : 2.8662$ $t = 2 : 4.6329$	$t = 0 : 0.0606$ $t = 1 : 4.1294$ $t = 2 : 3.8350$	$t = 0 : -0.0394$ $t = 1 : 4.3170$ $t = 2 : 4.2629$
$param_1$	$t = 0 : 0.0606$ $t = 1 : 3.2100$ $t = 2 : 5.7000$	$t = 0 : 0.0606$ $t = 1 : 4.4574$ $t = 2 : 4.3476$	$t = 0 : -0.0394$ $t = 1 : 4.4781$ $t = 2 : 4.4244$

Table 8: Sum of contraint violation mg_3

	$Niter_0$	$Niter_1$	$Niter_2$
$param_0$	$t = 0 : 0.0192$ $t = 1 : 1.0831$ $t = 2 : 1.2081$	$t = 0 : 0.0186$ $t = 1 : 1.4869$ $t = 2 : 1.0347$	$t = 0 : 0.0135$ $t = 1 : 1.4867$ $t = 2 : 1.1024$
$param_1$	$t = 0 : 0.0186$ $t = 1 : 1.0068$ $t = 2 : 1.3742$	$t = 0 : 0.0186$ $t = 1 : 1.4520$ $t = 2 : 1.1235$	$t = 0 : 0.0135$ $t = 1 : 1.4755$ $t = 2 : 1.1205$

Table 9: Relative constraint violation mg_3

	$Niter_0$	$Niter_1$	$Niter_2$
$param_0$	-4.1204	-5.8083	-2.7459
$param_1$	8.7786	8.2536	7.0571

Table 10: Objective Values mg_3

PMM

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	$t = 0 : -0.1191$ $t = 1 : 0.6214$ $t = 2 : 2.2287$	$t = 0 : -0.1203$ $t = 1 : 0.4687$ $t = 2 : 1.9037$	$t = 0 : -0.0600$ $t = 1 : 0.1329$ $t = 2 : 3.5118$
$param_1$	$t = 0 : -0.0002$ $t = 1 : 1.2524$ $t = 2 : 2.2268$	$t = 0 : -0.0052$ $t = 1 : 1.5591$ $t = 2 : 2.7715$	$t = 0 : -0.0834$ $t = 1 : 0.7272$ $t = 2 : 3.4184$

Table 11: Sum of contraint violation mg_3

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	$t=0:-2.3831$ $t=1:1.3842$ $t=2:14.2326$	$t=0:-2.3046$ $t=1:-2.8069$ $t=2:11.7336$	$t=0:-1.2085$ $t=1:8.9193$ $t=2:11.2040$
$param_1$	$t=0:-2.2358$ $t=1:11.7563$ $t=2:14.5335$	$t=0:-2.3268$ $t=1:-6.3316$ $t=2:11.8544$	$t=0:-1.2635$ $t=1:8.9274$ $t=2:11.2040$

Table 12: Sum of contraint violation mg_10

	$Niter_a$	$Niter_b$
$param_0$	$t=0:-12.2005$ $t=1:27.6426$ $t=2:116.0926$	$t=0:-12.1556$ $t=1:20.4376$ $t=2:100.6798$
$param_1$	$t=0:-10.8383$ $t=1:33.0749$ $t=2:136.9764$	$t=0:-10.8057$ $t=1:36.7089$ $t=2:114.3195$

Table 13: Sum of constraint violations mg_100

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	$t=0:-0.1190$ $t=1:-0.5558$ $t=2:3.4517$	$t=0:-0.1201$ $t=1:-0.6061$ $t=2:1.8739$	$t=0:-0.0600$ $t=1:0.5436$ $t=2:3.9772$
$param_1$	$t=0:-0.0005$ $t=1:0.2085$ $t=2:3.4665$	$t=0:-0.0086$ $t=1:0.5039$ $t=2:2.3847$	$t=0:-0.0733$ $t=1:1.2959$ $t=2:3.6829$

Table 14: Sum of constraint deviation mg_3_stoch

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	$t=0:0.0021$ $t=1:-0.5144$ $t=2:9.4155$	$t=0:-0.0019$ $t=1:-0.3791$ $t=2:9.7570$	$t=0:0.0014$ $t=1:2.2243$ $t=2:8.0698$
$param_1$	$t=0:0.2898$ $t=1:2.0289$ $t=2:9.4192$	$t=0:0.2985$ $t=1:3.1328$ $t=2:9.2632$	$t=0:-0.0344$ $t=1:2.9976$ $t=2:9.0316$

Table 15: Sum of constraint violations mg_10_stoch

	$Niter_a$	$Niter_b$
$param_0$	$t=0:-11.9908$ $t=1:15.1187$ $t=2:133.8596$	$t=0:-11.9891$ $t=1:18.4519$ $t=2:134.1373$
$param_1$	$t=0:-8.7828$ $t=1:34.3462$ $t=2:123.5259$	$t=0:-9.2516$ $t=1:6.3745$ $t=2:129.4561$

Table 16: Sum of constraint violations mg_100_stoch

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	$t = 0 : 0.0281$ $t = 1 : 0.4315$ $t = 2 : 0.8365$	$t = 0 : 0.0284$ $t = 1 : 0.2663$ $t = 2 : 0.8248$	$t = 0 : 0.0141$ $t = 1 : 0.1002$ $t = 2 : 0.9784$
$param_1$	$t = 0 : 0.0174$ $t = 1 : 0.6344$ $t = 2 : 0.8364$	$t = 0 : 0.0226$ $t = 1 : 0.6237$ $t = 2 : 0.8857$	$t = 0 : 0.0197$ $t = 1 : 0.2974$ $t = 2 : 0.9683$

Table 17: Relative constraint violation mg_3

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	$t = 0 : 0.168509$ $t = 1 : 0.139834$ $t = 2 : 1.083824$	$t = 0 : 0.162968$ $t = 1 : 0.280757$ $t = 2 : 1.003162$	$t = 0 : 0.085453$ $t = 1 : 0.891932$ $t = 2 : 0.999955$
$param_1$	$t = 0 : 0.168044$ $t = 1 : 1.176021$ $t = 2 : 1.097041$	$t = 0 : 0.167355$ $t = 1 : 0.633140$ $t = 2 : 1.005783$	$t = 0 : 0.089390$ $t = 1 : 0.892741$ $t = 2 : 0.999955$

Table 18: Relative constraint violation mg_10

	$Niter_a$	$Niter_b$
$param_0$	$t = 0 : 0.0863$ $t = 1 : 0.4049$ $t = 2 : 0.9774$	$t = 0 : 0.0860$ $t = 1 : 0.2678$ $t = 2 : 0.9385$
$param_1$	$t = 0 : 0.0769$ $t = 1 : 0.4595$ $t = 2 : 1.0642$	$t = 0 : 0.0766$ $t = 1 : 0.4292$ $t = 2 : 0.9717$

Table 19: Relative constraint violation mg_100

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	$t = 0 : 0.02807$ $t = 1 : 0.61438$ $t = 2 : 1.08625$	$t = 0 : 0.02832$ $t = 1 : 0.50353$ $t = 2 : 0.77950$	$t = 0 : 0.01414$ $t = 1 : 0.26840$ $t = 2 : 1.12861$
$param_1$	$t = 0 : 0.01745$ $t = 1 : 0.79248$ $t = 2 : 1.08886$	$t = 0 : 0.02410$ $t = 1 : 0.75979$ $t = 2 : 0.81748$	$t = 0 : 0.01729$ $t = 1 : 0.42668$ $t = 2 : 1.10027$

Table 20: Relative constraint violation mg_3_stoch

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	$t = 0 : 0.00088$ $t = 1 : 0.52154$ $t = 2 : 0.99710$	$t = 0 : 0.00098$ $t = 1 : 0.38670$ $t = 2 : 1.04098$	$t = 0 : 0.00024$ $t = 1 : 0.33735$ $t = 2 : 0.91978$
$param_1$	$t = 0 : 0.02992$ $t = 1 : 0.70838$ $t = 2 : 0.99715$	$t = 0 : 0.03387$ $t = 1 : 0.63171$ $t = 2 : 1.03172$	$t = 0 : 0.00291$ $t = 1 : 0.38935$ $t = 2 : 0.95281$

Table 21: Relative constraint violation mg_10_stoch

	$Niter_a$	$Niter_b$
$param_0$	$t = 0 : 0.08479$ $t = 1 : 0.66714$ $t = 2 : 1.19864$	$t = 0 : 0.08478$ $t = 1 : 0.55397$ $t = 2 : 1.21834$
$param_1$	$t = 0 : 0.06670$ $t = 1 : 0.80853$ $t = 2 : 1.14742$	$t = 0 : 0.07334$ $t = 1 : 0.56994$ $t = 2 : 1.22529$

Table 22: Relative constraint violation mg_100_stoch

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	-6.7139	-4.2921	-4.7685
$param_1$	-6.9795	-13.7602	-5.1939

Table 23: Objective Value mg_3

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	106.68	127.11	148.37
$param_1$	126.41	138.81	149.01

Table 24: Objective Value mg_10

	$Niter_a$	$Niter_b$
$param_0$	779.27	869.78
$param_1$	789.98	859.34

Table 25: Objective values mg_100

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	-14.1563	-16.4302	1.9276
$param_1$	-14.8944	-22.9641	1.6001

Table 26: Objective values mg_3_stoch

	$Niter_a$	$Niter_b$	$Niter_c$
$param_0$	-34.8858	-19.0519	5.6173
$param_1$	-35.7420	-21.9483	7.6591

Table 27: Objective values mg_10_stoch

	$Niter_a$	$Niter_b$
$param_0$	163.02	336.76
$param_1$	64.84	607.47

Table 28: Objective values mg_100_stoch

7.3 Figures

mg_3 - PDHG

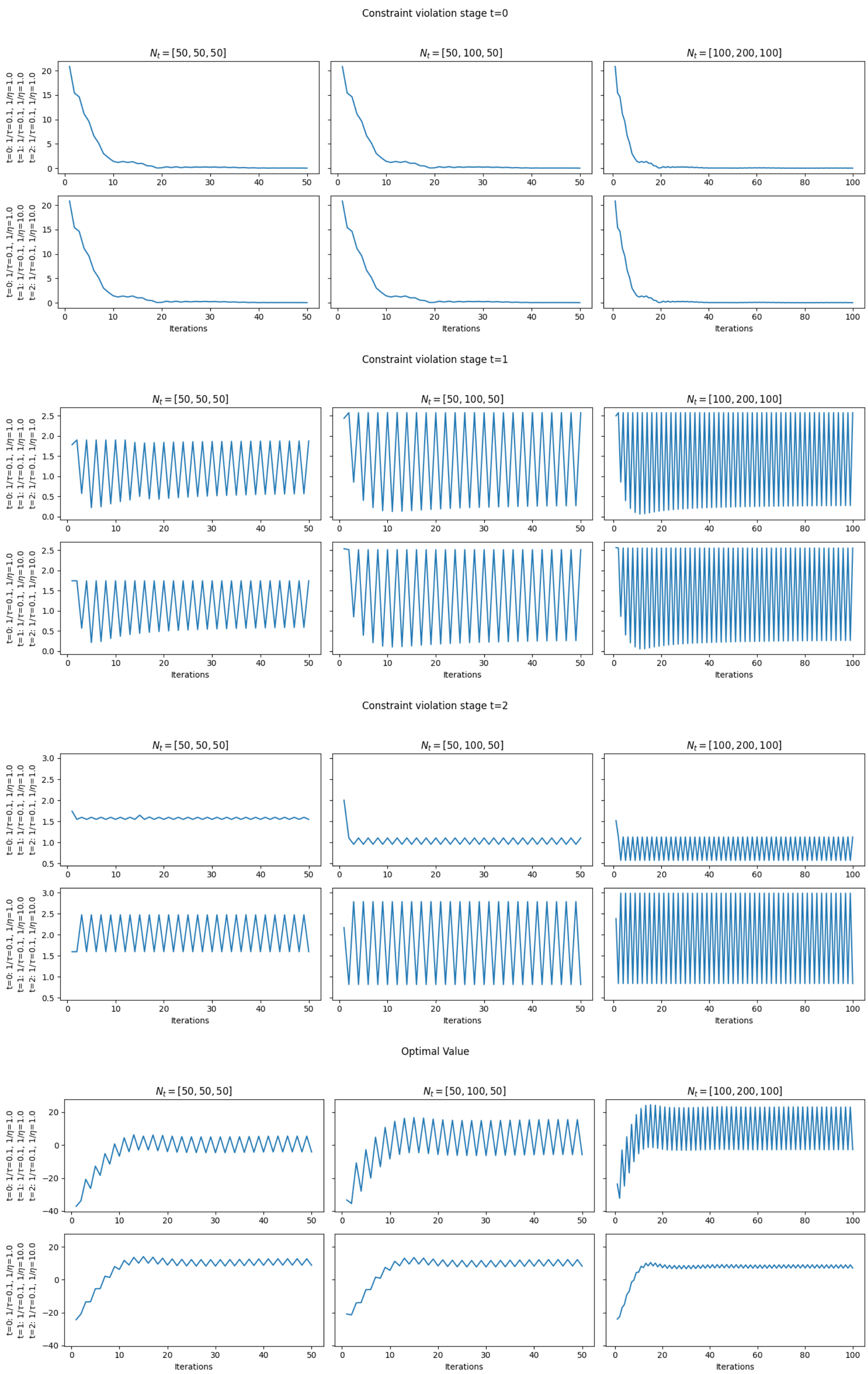

Figure 2: Convergence plots of l_2-norm of constraint violation and objective value of mg_3 solved with PDHG-DSA

mg_3 - PMM

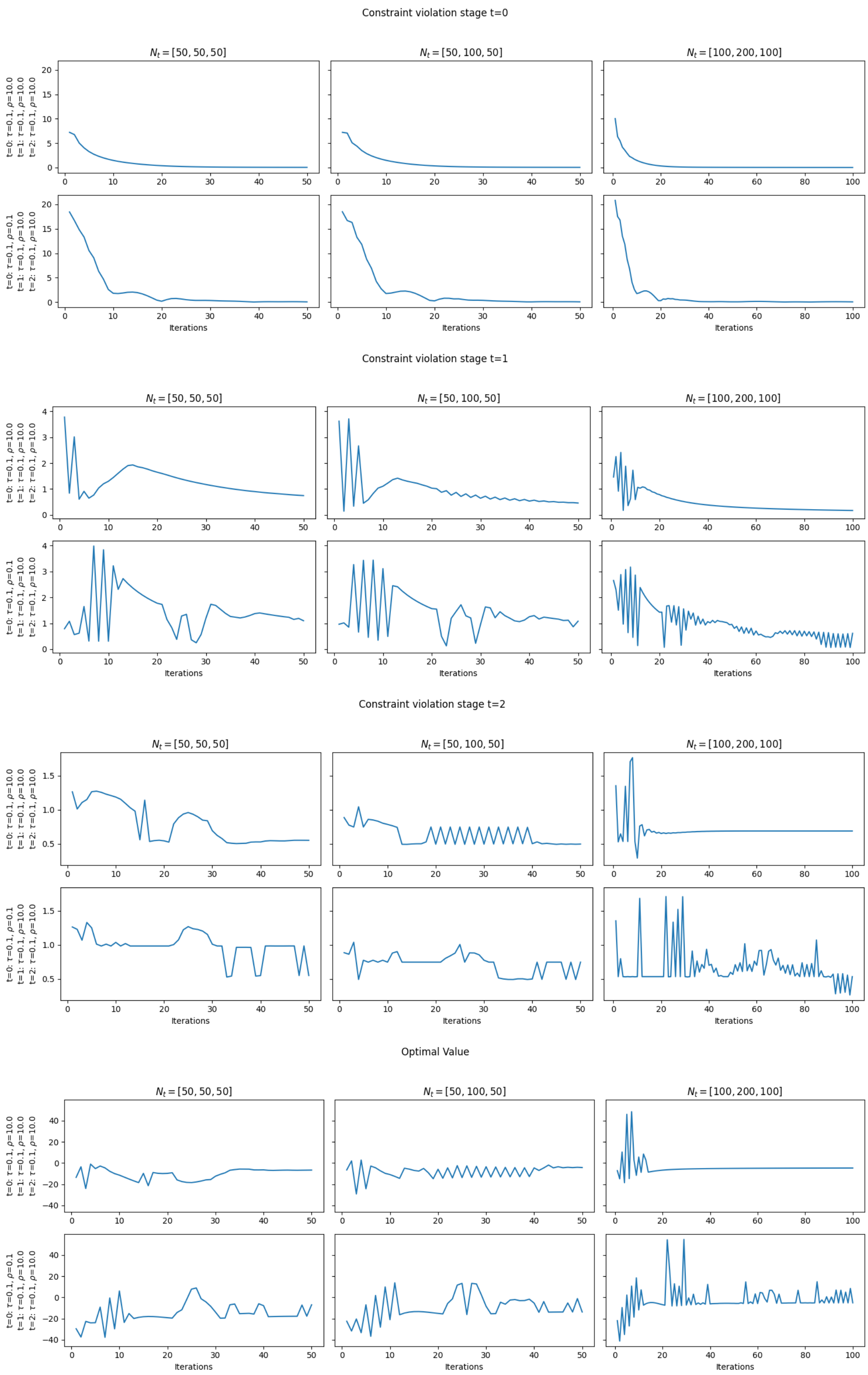

Figure 3: Convergence plots of l_2-norm of constraint violation and objective value of mg_3 solved with PMM-DSA

mg_10

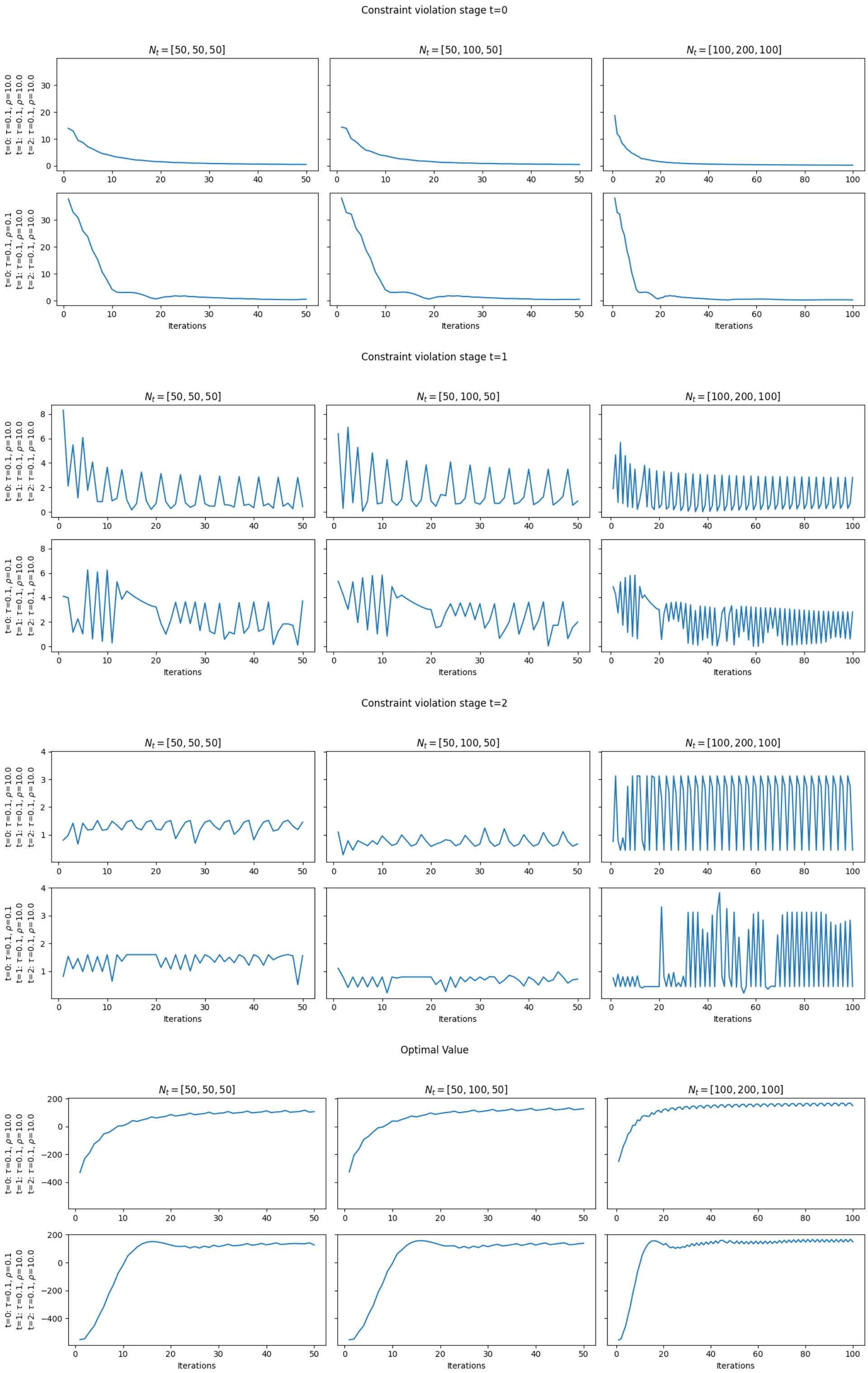

Figure 4: Convergence plots of l_2-norm of constraint violation and objective value of mg_10 solved with PMM-DSA

mg_100

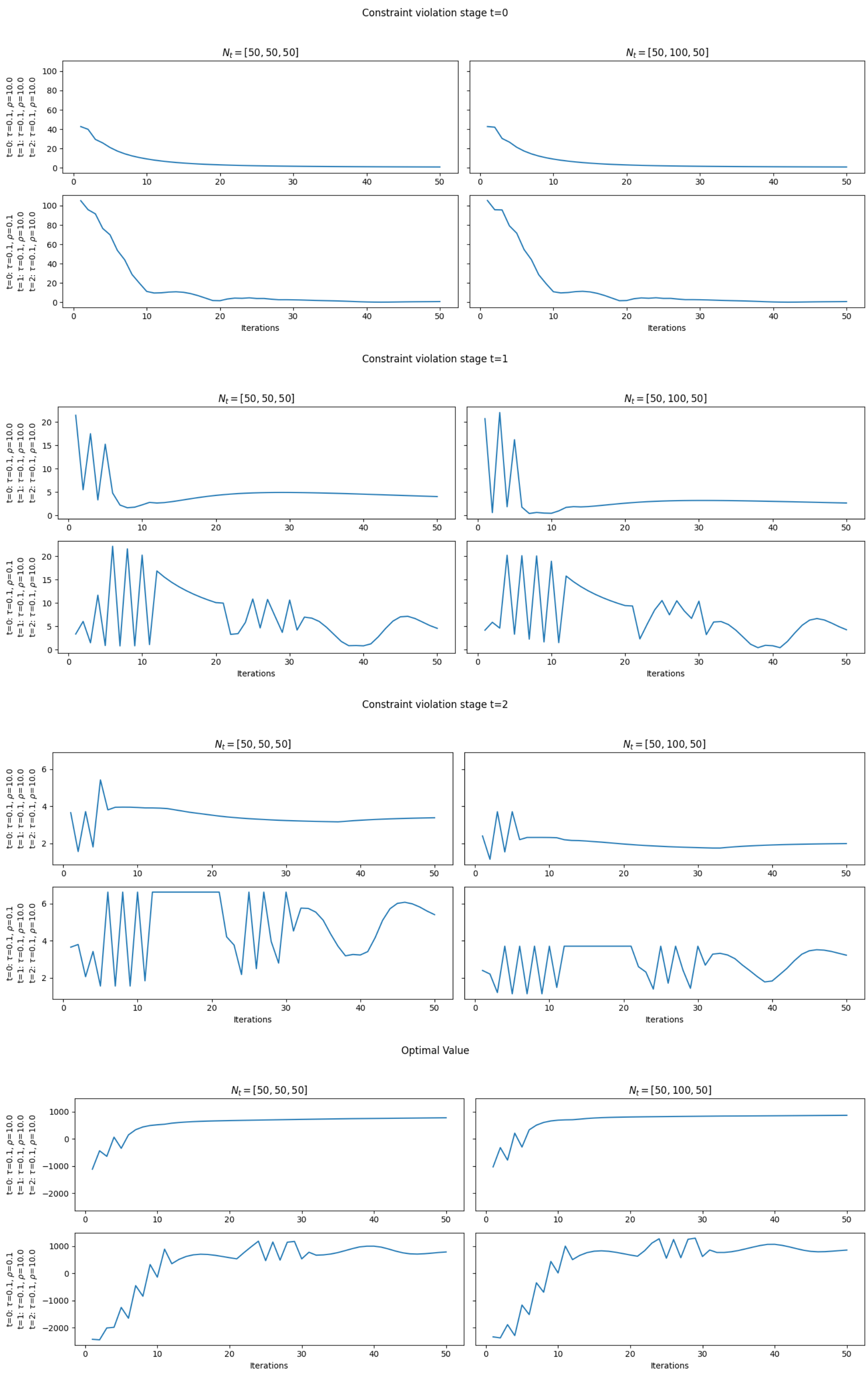

Figure 5: Convergence plots of l_2-norm of constraint violation and objective value of mg_100 solved with PMM-DSA

mg3_stoch

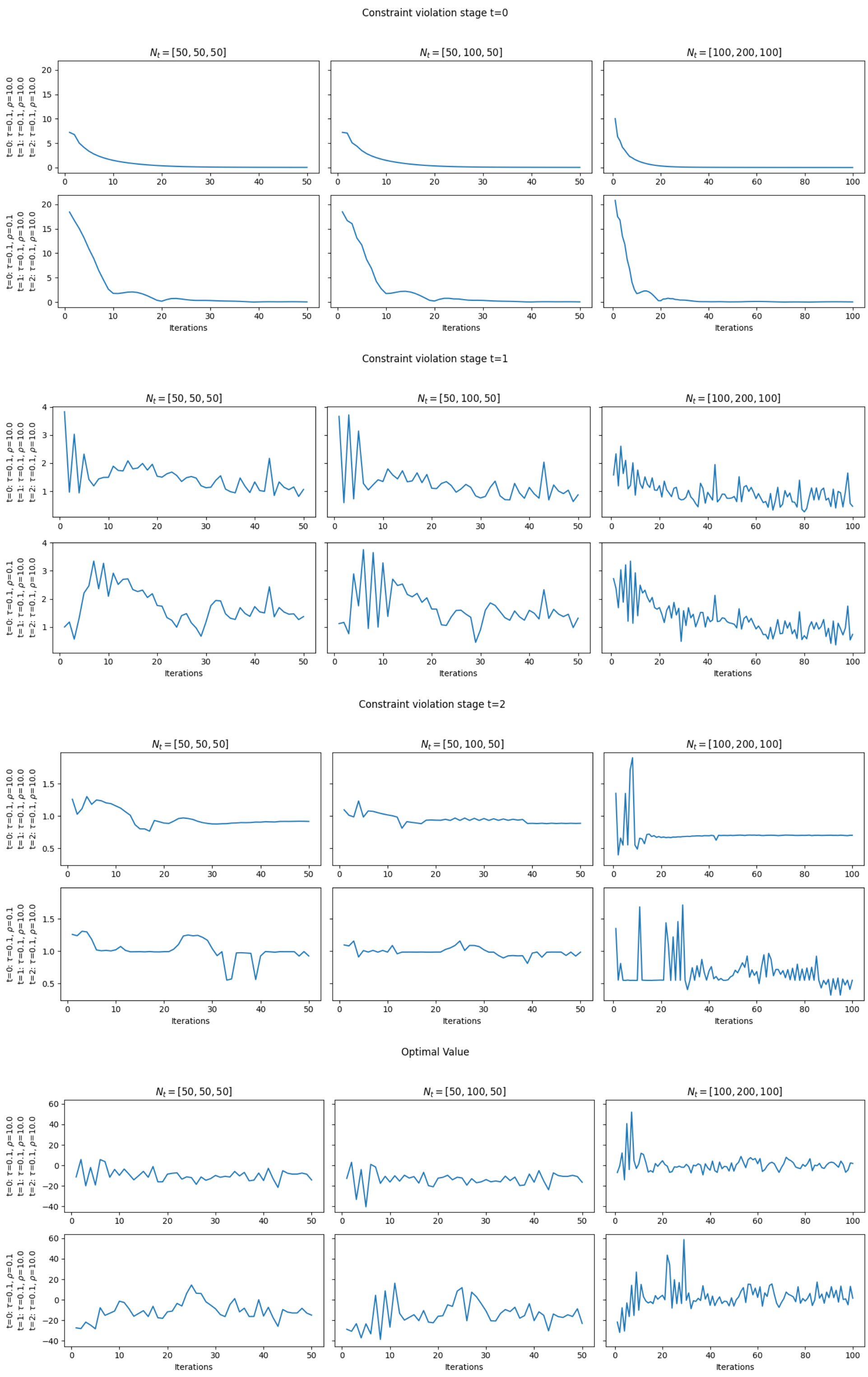

Figure 6: Convergence plots of l_2-norm of constraint violation and objective value of mg3_stoch solved with PMM-DSA

mg_10_stoch

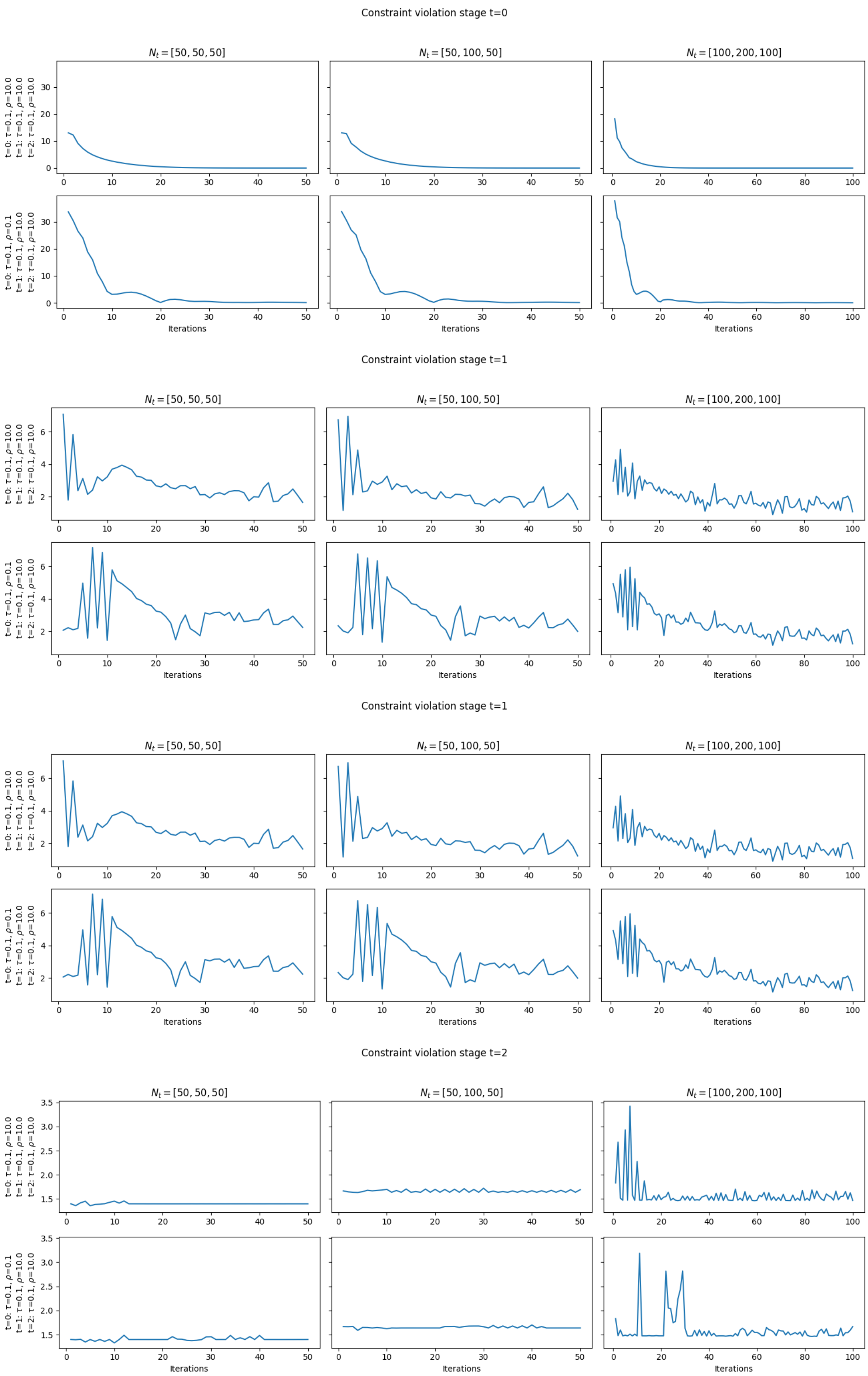

Figure 7: Convergence plots of l_2-norm of constraint violation and objective value of mg_10_stoch solved with PMM-DSA

mg_100_stoch

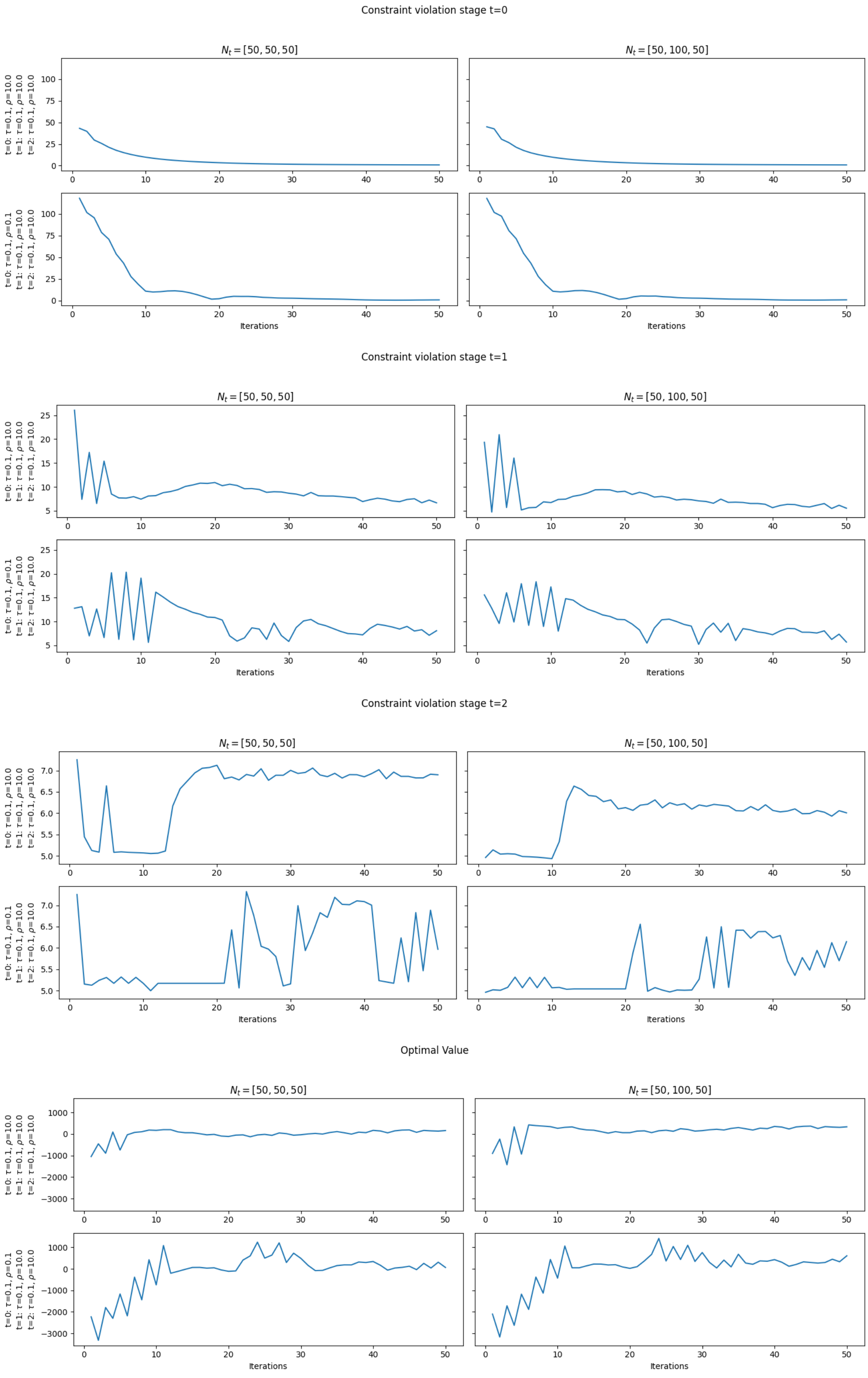

Figure 8: Convergence plots of l_2-norm of constraint violation and objective value of mg_100_stoch solved with PMM-DSA

List of Figures

List of Tables

References

Banert, S., Boţ, R. I., & Csetnek, E. R. (2021). Fixing and extending some recent results on the admm algorithm. *Numerical Algorithms*, *86*, 1303–1325.

Barty, K., Carpentier, P., & Girardeau, P. (2010). Decomposition of large-scale stochastic optimal control problems. *RAIRO-Operations Research*, *44*(3), 167–183.

Bauschke, H., & Combettes, P. (2017). *Convex analysis and monotone operator theory in hilbert spaces, second edition*. Springer New York.

Beck, A. (2017). *First-order methods in optimization*. SIAM.

Bot, R. I. (2009). *Conjugate duality in convex optimization* (Vol. 637). Springer Science & Business Media.

Chambolle, A., Delplancke, C., Ehrhardt, M. J., Schönlieb, C.-B., & Tang, J. (2024). Stochastic primal–dual hybrid gradient algorithm with adaptive step sizes. *Journal of Mathematical Imaging and Vision*, *66*(3), 294–313.

Chambolle, A., & Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, *40*, 120–145.

Chambolle, A., & Pock, T. (2016). On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, *159*(1), 253–287.

Chen, Z.-L., & Powell, W. B. (1999). Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse. *Journal of Optimization Theory and Applications*, *102*, 497–524.

Chierchia, G., Chouzenoux, E., Combettes, P. L., & Pesquet, J.-C. (2020). The proximity operator repository. *User's guide http://proximity-operator. net/download/guide. pdf. Accessed*, *6*.

Dupačová, J., Consigli, G., & Wallace, S. W. (2000). Scenarios for multistage stochastic programs. *Annals of operations research*, *100*, 25–53.

Dvurechensky, P., Shtern, S., & Staudigl, M. (2021). First-order methods for convex optimization. *EURO Journal on Computational Optimization*, *9*, 100015.

Ekeland, I., & Temam, R. (1999). *Convex analysis and variational problems*. SIAM.

Fiacco, A. V., & Kyparisis, J. (1986). Convexity and concavity properties of the optimal value function in parametric nonlinear programming. *Journal of optimization theory and applications*, *48*(1), 95–126.

Füllner, C., & Rebennack, S. (2023). Stochastic dual dynamic programming and its variants—a review. *Preprint, available at https://optimization-online.org/wp-content/uploads/2021/01/SDDP-Review.pdf*.

Girardeau, P., Leclere, V., & Philpott, A. B. (2015). On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, *40*(1), 130–145.

Gurobi Optimization, LLC. (2025). Gurobi Optimizer Reference Manual. https://www.gurobi.com

Heitsch, H., & Römisch, W. (2009). Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, *118*, 371–406.

Kelley, J. E., Jr. (1960). The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, *8*(4), 703–712.

Kumar, K. P., & Saravanan, B. (2017). Recent techniques to model uncertainties in power generation from renewable energy sources and loads in microgrids–a review. *Renewable and Sustainable Energy Reviews*, *71*, 348–358.

Lan, G., Lu, Z., & Monteiro, R. D. (2011). Primal-dual first-order methods with iteration-complexity for cone programming. *Mathematical Programming*, *126*(1), 1–29.

Lan, G., & Zhou, Z. (2021). Dynamic stochastic approximation for multi-stage stochastic optimization. *Mathematical Programming*, *187*(1), 487–532.

Leclere, V. (2014). *Contributions to decomposition methods in stochastic optimization* [Doctoral dissertation, Université Paris-Est].

Löhndorf, N., & Shapiro, A. (2019). Modeling time-dependent randomness in stochastic dual dynamic programming. *European Journal of Operational Research*, *273*(2), 650–661.

Nemirovski, A., Juditsky, A., Lan, G., & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, *19*(4), 1574–1609.

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*(1), 204–211.

Pereira, M. V., & Pinto, L. M. (1991). Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, *52*, 359–375.

Pranevicius, H., & Sutiene, K. (2007). Scenario tree generation by clustering the simulated data paths. *Proc 21st Eur Conf on Model and Simul (ECMS)*.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Rockafellar, R. T. (1976). Augmented lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, *1*(2), 97–116.

Rockafellar, R. T. (1997). *Convex analysis* (Vol. 28). Princeton university press.

Rockafellar, R. T., & Wets, R. J.-B. (1998). *Variational analysis* (Vol. 317). Springer Science & Business Media.

Ryu, E. K., & Yin, W. (2022). *Large-scale convex optimization: Algorithms & analyses via monotone operators*. Cambridge University Press.

Shapiro, A. (2011). Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, *209*(1), 63–72.

Shapiro, A., Dentcheva, D., & Ruszczynski, A. (2021). *Lectures on stochastic programming: Modeling and theory*. SIAM.

Shefi, R., & Teboulle, M. (2014). Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM Journal on Optimization*, *24*(1), 269–297. https://doi.org/10.1137/130910774

Staudigl, M. (2024). *Convex optimization, methods and algorithms* [Lecture Notes Summerterm 2024, University of Mannheim].

Zakaria, A., Ismail, F. B., Lipu, M. H., & Hannan, M. A. (2020). Uncertainty models for stochastic optimization in renewable energy applications. *Renewable Energy*, *145*, 1543–1571.

Declaration of the used AI tools

In the writing of this thesis I only used AI tools as stated below. I checked all output from the tools.

Tool	Purpose	Where
ChatGPT	Checking English grammar and spelling	Throughout
ChatGPT	Python code generation	All Python files
ChatGPT	Table generation	Sec. 5.2, 7.2, 7.3
ChatGPT	Figure formatting	Sec. 5.2, 7.2, 7.3
ChatGPT	LaTeX code debugging	Throughout
DeepSeek	Python code generation	All Python files
DeepSeek	LaTeX code debugging	Throughout
Github Copilot	LaTeX code debugging	Throughout
Github Copilot	Python code generation	All Python files