ADAPTIVE CERTAINTY-EQUIVALENT METHODS FOR GENERATION SCHEDULING UNDER UNCERTAINTY

Technical Report
Power Systems Laboratory
ETH Zurich

Tomás Andrés Tinoco De Rubira
June 2017

Abstract

Supplying a large percentage of the energy demand of countries or regions using renewable energy sources such as wind and solar poses many challenges to power system operations due to their highly variable and unpredictable nature. In particular, if not properly taken into account during the process of scheduling power generation, the uncertainty associated with these sources can result in high system operating costs or even security issues. Stochastic optimization provides techniques and methods for handling this uncertainty in generation scheduling, but the problem is that they are typically very computationally demanding. They typically either approximate expected-value functions with sample averages, which leads to very large problems, or rely excessively on random samples, which leads to high susceptibility to noise. However, a stochastic optimization algorithm has been proposed that has the particular property of being able to leverage initial deterministic approximations of expected-value functions. If these approximations are accurate, perhaps because they are constructed from problem-specific knowledge from practitioners, a solution can be found efficiently. In this work, the combination of this algorithm with initial approximations based on certainty-equivalent models, a combination which we refer to as the Adaptive Certainty-Equivalent method, is explored for solving generation scheduling problems under uncertainty. The performance of this methodology is first analyzed on simple two-stage stochastic economic dispatch problems, on which an interesting connection to another algorithm is found. Then, extensions of the algorithm are developed for handling expected-value constraints and multiple planning stages in order to solve risk-averse and multi-stage stochastic economic dispatch problems, respectively. Lastly, the applicability and performance of this methodology on generation scheduling problems with unit commitment decisions is analyzed. The overall results obtained in this work suggest that these Adaptive Certainty-Equivalent methods could be effective and efficient tools for obtaining generation schedules that are adequate for modern low-carbon power systems.

Acknowledgments

The work described in this report has been supported by the ETH Zurich Postdoctoral Fellowship FEL-11 15-1. This fellowship has not only allowed me to investigate and learn new ideas, but also to experience living in one of the most beautiful cities and countries in the world. For this, I am and always will be deeply grateful. I would like to thank Professor Gabriela Hug for giving me the opportunity to work at PSL and for all her support. In addition, I am grateful to all the doctoral students, master-thesis students, and semester-thesis students with whom I had the pleasure to work at PSL during these two years. I will never forget their amazing enthusiasm. Last but not least, I would like to acknowledge the support from my wife Lakkhana. She made an immense sacrifice to move to a new country with me, and I will never forget that.

Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

AC	Alternating Current
AdaCE	Adaptive Certainty-Equivalent
ADMM	Alternating Direction Method of Multipliers
AGC	Automatic Generation Control
CE	Certainty-Equivalent
CVaR	Conditional Value at Risk
DC	Direct Current
ED	Economic Dispatch
ERCOT	Electric Reliability Council of Texas
ETH	Swiss Federal Institute of Technology
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Non-Linear Programming
MIQP	Mixed-Integer Quadratic Programming
MISO	Midcontinent Independent System Operator
OPF	Optimal Power Flow
QCQP	Quadratically Constrained Quadratic Programming
QP	Quadratic Programming
SA	Stochastic Approximation
SAA	Sample-Average Approximation
SDDP	Stochastic Dual Dynamic Programming
SHA	Stochatic Hybrid Approximation
UC	Unit Commitment
VaR	Value at Risk

List of Mathematical Symbols

a_j	constant scalar
a	constant vector
A	constant sparse matrix
b_j	constant vector
B	positive definite matrix
$\mathcal{B}$	scenario tree branch map
c_{ij}	constant scalar
c	constant scalar
c_i^d	shut-down cost of generator i
c_i^u	start-up cost of generator i
$\mathcal{C}$	children map of node of scenario tree
d	vector of load powers
$\bar{d}$	vector of requested load powers
d_t	vector of load powers for stage t
d_{ij}	constant vector
D	constant sparse matrix
e	Euler's number
$\mathbb{E}[\,\cdot\,]$	expectation operator
$\mathbb{E}[\,\cdot\,\vert\,\cdot\,]$	conditional expectation operator
F	scalar-valued function
$\bar{F}$	w-independent part of the function F
$\hat{F}$	w-dependent part of the function F
F_t	scalar-valued cost function for stage t

$\mathcal{F}$	expected-value scalar-valued function
$\mathcal{F}_k$	deterministic approximation of $\mathcal{F}$ during iteration k
g_k	slope-correction vector during iteration k
$g_k(\,\cdot\,)$	gradient of $\mathcal{F}_k$
g_t^k	slope-correction function for stage t during iteration k
$\hat{g}_k$	noisy subgradient
G	constant sparse matrix or vector-valued function
$\bar{G}$	w-independent part of the function G
$\hat{G}$	w-dependent part of the function G
$\hat{G}_k$	noisy evaluation of $\mathcal{G}$
G_t	expected cost-to-go for stage t problem
$\widehat{G}_t$	initial deterministic approximation of G_t
$\widetilde{G}_t$	expected cost-to-go for stage t of SAA problem
$\widetilde{G}_t^k$	piece-wise linear approximation of $\widetilde{G}_t$
$\mathcal{G}$	expected-value vector-valued function
$\mathcal{G}_k$	deterministic approximation of $\mathcal{G}$ during iteration k
h_t^k	linear under-estimator of stage-$(t+1)$ cost of SAA problem
H	second-stage cost function
$\mathcal{H}^k$	approximation of expected second-stage cost during iteration k
H_t	optimal objective value of stage-t problem
$\widetilde{H}_t$	optimal objective value of stage-t SAA problem
H_t^k	optimal objective value of approximate stage-t problem during iteration k
J_k	Jacobian of $\mathcal{G}_k$
$\hat{J}_k$	matrix of noisy subgradients of $\mathcal{G}$
k	iteration counter
k_0	iteration counter offset
$\mathcal{K}$	feasible set of second-stage problem
$\mathcal{K}_t$	feasible set function for stage-t problem
L	noisy Lagrangian function or constant sparse matrix
$\mathcal{L}$	Lagrangian function

m	number of samples of random vector
N	correlation distance for renewable powers
$\mathbb{N}$	set of natural numbers
$\mathcal{N}$	normal distribution or node map for scenario tree
n	number of generators
n_b	number of buses
n_d	number of loads
n_e	number of edges
n_p	number of generators or generators with limited flexibility
n_q	number of fast-ramping flexible generators
n_r	number of renewables
n_s	number of line segments
p	vector of generator powers
$p_{i,t}$	power of generator i during time period t
p_t	vector of powers of generators with limited flexibility for stage t
$p^{\min}$	vector of lower limits of generator powers
$p^{\max}$	vector of upper limits of generator powers
P	constant sparse matrix
$\mathcal{P}$	set of possible vectors of generator powers
q	vector of generator power adjustments
q_t	vector of power of fast-ramping flexible generators for stage t
$q^{\min}$	vector of lower limits of powers from fast-ramping flexible generators
$q^{\max}$	vector of upper limits of powers from fast-ramping flexible generators
Q	second-stage cost function or constant sparse matrix
$\bar{Q}$	piece-wise linear approximation of the function Q
$Q^{\max}$	desired bound on second-stage cost
$\mathcal{Q}$	expected second-stage cost function
$\mathcal{Q}^k$	deterministic approximation of second-stage cost during iteration k
r	vector of powers from renewables
r_t	vector of powers from renewables for stage t

$\hat{r}_t$	vector of predicted powers from renewables for stage t
r_0	base powers from renewables
$r^{\max}$	power capacities of renewables
$\bar{r}_t$	base powers for renewables during stage t
R	constant sparse matrix
$\mathcal{R}$	set of capacity-limited powers from renewables
$\mathcal{R}_t$	observation history of renewable powers up to stage t
$\widehat{\mathcal{R}}_{t,\tau}$	renewable powers up to stage-τ predicted at stage t
$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	set of non-negative real numbers
$\mathbb{R}_{++}$	set of positive real numbers
s	vector of powers from renewables after curtailments
s_t	vector of powers from renewables after curtailments for stage t
S	scalar-valued function or constant sparse matrix
S_i^d	index set for enforcing minimum down time of generator i
S_i^u	index set for enforcing minimum up time of generator i
$\mathcal{S}$	expected-value scalar-valued function
t	auxiliary variable for CVaR constraint or stage or time index
T	transpose operator (as superscript) or number of time stages
T_i^d	minimum down time of generator i
T_i^u	minimum up time of generator i
$\mathcal{T}$	set of bounded auxiliary variables t
u	vector of generator on-off stages
$u_{i,t}$	on-off stage of generator i during time period t
$\mathcal{U}$	set of generator commitments that satisfy minimum up and down times
v_k	gradient-correction vector
w	general random vector or random vector of renewable powers
w_i	i-th sample of random vector of m samples
w_l	l-th sample of random vector of m samples
w_k	random vector sampled during iteration k

w_t	random vector for stage t
w_t^k	random vector for stage t sampled during iteration k
$\mathcal{W}_t$	observation history up to stage t
$\mathcal{W}_t^k$	observation history up to stage t sampled during iteration k
x	general vector of optimization variables
x^*	optimal primal point
x_k	solution candidate during iteration k
x_t	vector of optimization variables for stage t
x_t^k	solution candidate for stage t during iteration k
$\mathcal{X}$	compact convex set
$\mathcal{X}_t$	point-to-set mapping with convex compact outputs
Y	constant sparse matrix
$y^{\max}$	vector of generator ramp upper limits
$y^{\min}$	vector of generator ramp lower limits
$y_{i,t}$	auxiliary epigraph variable
z	auxiliary variable
$z^{\min}$	lower limited for thermal constraints
$z^{\max}$	upper limits for thermal constraints
$z_{i,t}$	auxiliary epigraph variable
$\mathbb{Z}$	set of integer
$\mathbb{Z}_+$	set of non-negative integers
$\mathbb{Z}_{++}$	set of positive integers
α_k	step length for iteration k
α_t^k	step-length function for stage t during iteration k
β_k	step length for iteration k
$\beta_{i,j}$	constant scalar for piece-wise linear generation cost function
γ	desired probability or load-shedding cost
γ_t	algorithm parameter
δ	subdifferential operator or random perturbation vector of renewable powers
δ_t	random perturbation vector of renewable powers for stage t

$\delta^{\max}$	vector of ramp upper limits for renewable powers
$\delta^{\min}$	vector of ramp lower limits for renewable powers
ϵ	probability of not enforcing renewable ramp bounds
ε	positive constant
ζ_k	gradient-correction vector
$\zeta_{i,j}$	constant scalar for piece-wise linear generation cost function
η_t^k	gradient of $\widehat{G}_t$ at $\left(x_t^k, \mathcal{W}_t^k\right)$
θ	vector of bus voltage angles
θ_t	vector of bus voltage angles for stage t
ϑ	auxiliary epigraph variable
λ	Lagrange multiplier
λ_k	Lagrange multiplier estimate during iteration k
λ^*	optimal dual point
$\lambda^{\max}$	Lagrange multiplier bound
Λ	set of bounded Lagrange multipliers
μ_t^k	Lagrange multipliers for stage-t ramping constraints
μ_i^k	Lagrange multipliers for ramping constraints, iteration k, sample i
ν_t	noisy subgradient for stage-$(t+1)$ cost of SAA problem
ξ_k	noisy subgradient obtained during iteration k
$\xi_{k,i}$	i-th noisy subgradient obtained during iteration k
π_t^k	Lagrange multipliers for stage-t ramping constraints
π_i^k	Lagrange multipliers for ramping constraints, iteration k, sample i
Π_x	projection operator onto $\mathcal{X}$
Π_r	projection operator onto set of capacity-limited renewable powers
Π_δ	projection operator onto set ramp-limited renewable powers
Π_λ	projection operator onto set of bounded multipliers
ρ	correlation coefficient for renewable powers
σ	positive constant
Σ	summation, or covariance matrix of renewable powers or noisy subgradients
Σ_m	covariance matrix of averaged noisy subgradients

Σ_t	covariance matrix of renewable powers for stage t
$\bar{\Sigma}_t$	covariance matrix of renewable powers for time period t
τ	stage index
ϕ_t^k	radial basis function
φ	cost function for generators with limited flexibility
$\bar{\varphi}$	piece-wise linear approximation of φ
φ_0	generation cost function
φ_1	generation adjustment cost function
ψ	cost function for fast-ramping flexible generators
Ω	set of possible random vectors
Ω_t	set of possible random vectors for stage t
$\in$	set membership
$\emptyset$	empty set
$\forall$	for all
∞	infinity
log	natural logarithm
∇	gradient operator
$\lvert \cdot \rvert$	set cardinality or absolute value of scalar
$\lVert \cdot \rVert_2$	Euclidean norm
$\lVert \cdot \rVert_B$	quadratic norm induced by positive definite matrix B
$\cup$	set union
$\cap$	set intersection

Chapter 1

Introduction

1.1 Generation Scheduling

In electric power systems, which are also known as power networks or grids, the operation of power plants or generators needs to be planned in advanced, *e.g.*, days or hours ahead. One reason for this is that some generation technologies, in particular those with low operating costs, have limited flexibility. For example, "baseload" units, which are typically large nuclear and coal-fired power plants, have slow ramp rates and relatively high minimum generation levels, and can take a long time to start or provide some flexible response [6] [52]. In practice, the on-off states of generators as well as tentative or financially-binding power production levels for every hour of the day are determined the day before by means of solving a Unit Commitment (UC) problem. For example, in a deregulated electricity market, this is done during day-ahead market clearing. The UC problem considers the various costs and constraints of each generator in the network, and aims to find a feasible schedule that reduces cost or maximizes social welfare. Then, shortly before delivery, *e.g.*, one hour ahead, adjustments to power production levels of generators are determined by means of solving an Economic Dispatch (ED) problem. This is done to ensure that the system remains balanced in an economically efficient way in the event of deviations from predicted conditions. Any last-minute disturbances and deviations are handled automatically through frequency control, with resources such as Automatic Generation Control (AGC) [57].

Until recently, day-ahead loading conditions of power systems as well as the availability of generation resources were known to system operators relatively well. Uncertainty was low and it was adequately handled by means of ensuring enough reserves during generation

scheduling. This meant that deterministic UC and ED models were sufficient. However, in modern low-carbon power systems, this is no longer the case.

1.2 Renewable Energy and Uncertainty

Driven by environmental regulations aimed at reducing carbon emissions from energy use, or by seeking energy independence, many countries and regions are setting aggressive targets for future levels of penetration of renewable energy into their power systems. For example, the European Union targets as a whole a 20% level of penetration by 2020, while some individual countries have more aggressive long term plans, such as Germany and Denmark, which target levels of 60% and 100%, respectively, by 2050 [72]. In the United States, many states have also established similar targets. California, being one of them, has set a target of 33% by 2020 [59]. However, achieving such high levels of renewable energy penetration poses serious challenges to power systems operations. For example, wind and solar energy, some of the most abundant renewable energy sources, is highly variable and unpredictable [49] [59] [81]. Hence, in order to ensure system reliability and economic competitiveness of this clean energy, power systems need to be much more flexible, and the decision making process needs to be improved in order to exploit this flexibility in an economic way.

With regards to generation scheduling, an improper treatment of the variability and uncertainty of wind and solar energy could result in schedules that are either overly conservative or rely excessively on flexible but more expensive generation resources, *e.g.*, peaking units [6]. For this reason, accurate stochastic scheduling models are needed, *e.g.* stochastic UC and ED models. Such models incorporate the probability distribution of the uncertain quantities, and allow finding schedules that take into account the different possibilities along with their likelihood. However, these models pose difficult computational challenges and require the use of special numerical optimization algorithm.

1.3 Stochastic Optimization

Stochastic optimization is a field that specializes in tools and methods for solving optimization problems with uncertain quantities, or stochastic optimization problems. These problems capture the key property that some decisions need to be made in advance with limited knowledge about the future, and typically involve expected-value functions either

in the objective, constraints, or both. These properties of stochastic optimization problems make them hard to solve in general. The reason is that these expected-value functions are in general defined by multi-dimensional integrals and often involve complex functions whose evaluation requires solving other optimization problems of performing computationally demanding simulations. Hence, accurate evaluations of these functions during the execution of an algorithm are impractical. For this reason, stochastic optimization algorithms rely on using approximations.

Roughly, the most widely studied and used approaches for solving stochastic optimization problems can be grouped into two families. The first one is based on *external sampling*. It consists of drawing samples of the uncertain quantities, forming a deterministic approximation of the problem, and then applying an optimization algorithm for solving the approximate problem. These samples are typically called *scenarios*, and the approximation is typically formed by replacing expected-value functions with sample averages. Hence, this approach is commonly known as *Sample-Average Approximation* (SAA) [36]. In theory, optimization algorithms for deterministic problems can be used to solve the SAA problems. However, these problems tend to be very large and specialized algorithms and techniques have been devised in order to exploit their structure. Examples of such techniques are *Benders decomposition* or *L-shaped method*, *Dantzig-Wolfe decomposition*, and *Lagrangian relaxation* [11] [58]. Despite these techniques, the difficulties in solving SAA problems due to their size imposes limits on the types of models used, *e.g.*, linear instead of nonlinear, and on the number of scenarios that can be used to model the uncertainty. Nevertheless, these external-sampling approaches have been a popular choice for solving stochastic generation scheduling problems [23] [64] [100]. The other family of approaches are those based on *internal sampling*. These algorithms periodically draw samples of the uncertain quantities during their execution. Widely known members of this family are *Stochastic Approximation* (SA) algorithms, which are characterized by making iterative improvements to solution estimates by taking steps along random search directions [36]. These SA algorithms have been widely used for solving problems in Machine Learning and Signal processing. They typically have low demand for computer memory, but they tend to be highly susceptible to noise and difficult to tune in practice.

Cheung and Powell proposed a *Stochastic Hybrid Approximation* (SHA) procedure for solving stochastic optimization problems that combines elements found in algorithms from

the two families described above [21]. More specifically, the algorithm takes an initial user-provided deterministic approximation of the stochastic (expected-value) objective function. Then, it simultaneously uses this approximation to generate solution estimates, and noisy slope observations at these estimates to improve the approximation. As the authors state, the strengths of this approach are the fact that it can leverage an accurate initial approximation, perhaps because it is constructed from problem-specific knowledge available to practitioners, and that the structure of this approximation does not change with the updates performed during the execution of the algorithm. This promising algorithm was motivated by problems arising in transportation involving the management of large fleets of vehicles, and it was designed for two-stage stochastic problems. To the best of our knowledge, its applicability and performance on generation scheduling problems from power systems with large penetration of renewable energy sources has not been studied. Perhaps for these problems, certain types of initial deterministic approximations of expected-value functions can be exploited to make the algorithm find a solution efficiently. If so, ways to extend this algorithm to handle multiple planning stages, expected-value constraints (to bound risk), and binary variables need to be explored, as they are important elements of generation scheduling problems in modern low-carbon power systems.

1.4 Contributions

In this work, the approach consisting of the combination of the *Stochastic Hybrid Approximation* procedure with initial deterministic approximations of expected-value functions based on *Certainty-Equivalent* (CE) models is explored for solving stochastic generation scheduling problems. With this methodology, which is referred to here as the *Adaptive Certainty-Equivalent* (AdaCE) method, we seek to solve stochastic generation scheduling problems more efficiently and effectively than with other methods used in practice or discussed in the literature. The ultimate goal is to be able to find generation schedules that are economical and that exploit the flexibility of available power generating resources to the maximum in order to allow large-scale penetration of wind and solar energy in power systems. This goal is quite challenging, and hence we take gradual steps towards it, extending the techniques proposed by Cheung and Powell in [21] in order to handle problems with increasing complexity. More specifically, the contributions made in this work are the following:

- **The Adaptive Certainty-Equivalent Method**: The AdaCE method, which combines the SHA algorithm with initial deterministic approximations of expected-value functions based on CE models, is introduced in the context of two-stage stochastic ED problems. For such problems, the performance of this method is compared against that of SAA-based and SA benchmark algorithms. The results obtained show that the initial deterministic approximations used are accurate and allow the algorithm to quickly produce better generation dispatches compared to the other algorithms considered. In addition, an interesting connection is found between the underlying SHA algorithm of the AdaCE method and a non-Euclidean stochastic proximal gradient algorithm on the problems considered. This work, which was done in collaboration with Professor Gabriela Hug, was presented in the 2016 European Control Conference in Aalborg, Denmark, and was later published in the conference proceedings [90].

- **A Primal-Dual Extension for Risk-Averse Dispatch**: Motivated by the fact that power system operators need to be risk-averse, an extension to the SHA algorithm is developed in order to apply the AdaCE method to two-stage stochastic ED problems with constraints that bound the risk of high system operating cost. A theoretical analysis of the convergence of the proposed "primal-dual" extension of the SHA algorithm is provided, along with experimental results that compare the performance of the resulting AdaCE method against SAA-based and SA algorithms. This work, which was also done in collaboration with Professor Gabriela Hug, was submitted for publication to the Journal of Computational Optimization and Application, and is currently undergoing the second round of review [91].

- **A Parameterized Extension for Multiple Stages**: Motivated by the fact that the operation of generators in power systems is a sequential decision-making process spanning multiple time periods, an extension to the SHA algorithm is developed for solving multi-stage stochastic ED problems. Such problems capture the inter-temporal constraints of generators as well as the fact that decisions at each time period or stage depend on previous decisions and have to be made with imperfect knowledge about the future. Mathematically, they present serious computational challenges due to the presence of nested expected-value functions. A theoretical analysis of the convergence of the proposed "parameterized" extension of the SHA algorithm is provided, along with experimental results that compare the performance of the resulting AdaCE

method against benchmark algorithms that include the popular Stochastic Dual Dynamic Programming (SDDP) algorithm. This work, which was done in collaboration with Professor Gabriela Hug and Doctor Line Roald, was submitted for publication to the Journal of Optimization Theory and Applications, and is currently undergoing the first round of review [92].

- **Applicability of AdaCE for Unit Commitment**: Generator or unit commitment decisions in generation scheduling are binary decisions, and they make the resulting stochastic optimization problem fail to satisfy the conditions the guarantee the convergence of the SHA algorithm on which the AdaCE method is based. Moreover, they also increase the complexity of the problem significantly. Hence, possible outcomes of the SHA algorithm on these problems are investigated as well as practical limitations imposed on the structure or properties of the initial deterministic approximations of expected-value functions. In this study, it is found that although the theoretical guarantees of the SHA algorithm are lost due to the binary decisions, the algorithm combined with a modified CE-based initial deterministic approximation is still able to find commitment schedules in reasonable times that are better than those obtained with some benchmark algorithms. This work was done in collaboration with José Espejo-Uribe as part of his Master Thesis at the Power Systems Laboratory of the Swiss Federal Institute of Technology in Zurich [27].

1.5 Report Outline

This report is structured as follows: Chapter 2 introduces the AdaCE method in the context of two-stage stochastic ED problems and analyzes its performance. Chapter 3 presents the primal-dual extension of the method for solving stochastic risk-averse ED problems. Chapter 4 presents the parameterized extension of the method for solving multi-stage stochastic ED problems. Chapter 5 analysis the applicability and performance of the AdaCE method for solving stochastic UC problems. Lastly, Chapter 6 summarizes the work and describes interesting future research directions. Convergence analyses of the various versions of the SHA algorithm considered in this work can be found in the Appendix.

Chapter 2

The Adaptive Certainty-Equivalent Method

2.1 Background and Overview

As already noted in Chapter 1, energy from renewable sources, in particular wind and solar, is intermittent, highly variable, and difficult to predict [59]. These properties pose many challenges to power grid operations that need to be overcome before high levels of renewable energy penetration can be attained without compromising grid reliability and economic efficiency. One particularly difficult challenge is that large forecast errors in available renewable energy can cause large power imbalances in the grid. These imbalances need to be resolved by making fast power adjustments that typically have a high cost. This cost may be associated with load curtailments or with the operational and environmental costs associated with overusing highly-flexible generators [59].

Stochastic optimization techniques have been studied extensively in recent years for considering uncertainty and balancing costs during generation scheduling. In particular, they have been studied in the context of determining optimal generator power production levels or dispatches for a single or multiple time periods in the near future, *e.g.*, sub-hourly to a few hours. Approaches proposed in the literature have been typically based on SAA, which consists of replacing expected values with sample averages using scenarios. The resulting deterministic problems, which are typically large, are then solved using techniques that exploit the specific structure of such problems, such as the Benders or L-shaped decomposition method [12] [98]. For example, the authors in [45] consider the ED problem

with contingencies and wind forecast errors, and model it as a two-stage stochastic convex problem. Their model includes frequency constraints and their solution approach consists of using a small number of scenarios and applying the L-shaped method. They test their approach on the system from the Electric Reliability Council of Texas (ERCOT), and show that it gives lower expected operating costs compared to a deterministic approach that uses fixed reserve requirements. The authors in [29] consider a similar problem in hourly and sub-hourly time-frames, model it as a two-stage linear program with recourse, and apply a sampling-based version of the L-shaped method known as stochastic decomposition [36]. They test their approach on a 96 and 1900-bus system and show its superiority in terms of solution quality and time compared to solving one large deterministic linear problem constructed from a fixed number of scenarios. Lastly, the authors in [64] also study a two-stage Optimal Power Flow (OPF) problem with uncertain renewable generation, but consider AC power flow constraints, which significantly increase the problem complexity. Their solution approach assumes zero duality gap of the second-stage problems [44], and is based on using a set of pre-selected scenarios and forming outer approximations in a way similar to generalized Benders decomposition [30]. They compare their approach with one based on the Alternating Direction Method of Multipliers (ADMM) [15], but only on IEEE and other small test systems having up to 300 buses.

In this chapter, an alternative family of stochastic optimization techniques that deal with uncertainty more directly than SAA-based approaches are explored for solving the stochastic ED problem. More specifically, SA techniques, which make iterative improvements to a solution estimate using noisy observations (typically gradients or subgradients), and hybrid techniques that use deterministic approximations of the original problem but improve these using noisy observations are explored. SA algorithms go back to the pioneering work of Robbins and Monro [75], and have been an active area of research [53] [88] [106]. The most well-known example, the stochastic subgradient algorithm, has been widely used for solving large-scale Machine Learning problems [14]. In a recent work, the authors in [53] compare the performance of robust SA algorithms against an efficient SAA-based algorithm, and find that the former are able to obtain similar solution accuracy with significantly lower computational requirements on certain problems. The hybrid approach that updates a deterministic approximation of the original problem using noisy observations is proposed in [21]. It is discussed in the context of two-stage stochastic linear problems with

recourse and is motivated by a dynamic vehicle allocation problem. The proposed deterministic approximation for such problem is a piece-wise linear separable function whose slope is updated using stochastic subgradients of the recourse function. The authors provide a convergence proof of this algorithm under certain assumptions as well as an illustration of the approach on a one-variable problem, but do not provide performance results on practical problems. Motivated by the fact that in power systems, operators typically obtain generator dispatches from solving CE problems [29], which are obtained by replacing uncertain quantities with their expected or predicted values, we explore here using and updating these deterministic approximations. For this reason, we refer to the hybrid approach considered here as the *Adaptive Certainty-Equivalent* method, or AdaCE. It is shown here that this adaptive algorithm is equivalent to the stochastic proximal gradient algorithm described in [79] with a non-Euclidean metric when applied to the stochastic ED problem. To assess its performance, this algorithm is compared against the stochastic subgradient and two SAA-based algorithms on four test cases from power networks having 2500 to 9200 buses.

2.2 Convex Stochastic Optimization Problems

The general stochastic optimization problems considered in this chapter are of the form

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad \mathcal{F}(x) := \mathbb{E}\left[F(x,w)\right], \tag{2.1}$$

where $F(\,\cdot\,,w)$ are convex functions, $\mathcal{X}$ is a nonempty compact convex set, w is a random vector from a compact set Ω, and $\mathbb{E}\left[\,\cdot\,\right]$ denotes expectation with respect to w. It is assumed that $\mathcal{F}$ is well-defined and continuous in $\mathcal{X}$. It can be shown that with these assumptions, problem (2.1) is a convex optimization problem.

Despite convexity, optimization problems of this form are challenging to solve in general. The reason for this is that evaluating the function $\mathcal{F}$ requires computing a multi-dimensional integral, and doing this with high accuracy is not practical when the vector w has more than a few dimensions [53]. For this reason, algorithms for solving (2.1) resort to using approximations.

2.3 Solution Algorithms

In this section, algorithms for solving problems of the form of (2.1) are described. In particular, an SA algorithm, namely stochastic subgradient, SAA-based algorithms, and the SHA algorithm are described.

In the following subsections, unless otherwise stated, all derivatives and subdifferentials are assumed to be with respect to the vector x of optimization variables of problem (2.1).

2.3.1 Stochastic Subgradient

The stochastic subgradient algorithm for solving problem (2.1) is an iterative *internal sampling* algorithm that performs the following update at each iteration $k \in \mathbb{Z}_+$:

$$x_{k+1} = \Pi_x \left(x_k - \alpha_k \xi_k \right), \tag{2.2}$$

where x_k denotes the solution estimate during the k-th iteration, α_k are step lengths, Π_x is the projection operator onto $\mathcal{X}$, $\xi_k \in \partial F(x_k, w_k)$, and w_k are independent samples of w. For the algorithm to work, the step lengths α_k are required to satisfy certain properties, *e.g.*, square-summability but not summability. The ones considered here are simple step lengths of the for $\sigma/(k + k_0)$, where $\sigma \in \mathbb{R}_{++}$, and $k_0 \in \mathbb{Z}_{++}$. For "modern" convergence proofs of this algorithm, the reader is referred to [53] and to Section 7.6.3 of [67].

As seen from (2.2), this algorithm is quite simple and only requires computing a noisy subgradient of $\mathcal{F}$ at each iteration. However, the downside is that finding step lengths that lead to adequate performance, *e.g.*, finding a suitable σ or k_0, can be quite challenging and problem-dependent. The fundamental difficulty lies in the fact that these step lengths need to be small enough to adequately suppress noise, while at the same time large enough to avoid slow convergence.

2.3.2 SAA-Based Algorithms

A widely-used *external sampling* approach for approximately solving stochastic optimization problems of the form of (2.1) is the SAA approach. This approach consists of sampling independent realizations of the random vector w, say $\{w_i\}_{i=1}^m$, where $m \in \mathbb{Z}_{++}$, and replace

expected values with sample averages. The resulting problem approximation is

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad m^{-1}\sum_{i=1}^{m} F(x,w_i), \tag{2.3}$$

which is a deterministic optimization problem. It can be shown that under certain conditions and sufficiently large m, the solutions of (2.3) are arbitrarily close to those of (2.1) with high probability [36].

Due to the requirement of choosing m sufficiently large in order to get an accurate estimate of a solution of (2.1), problem (2.3) can be difficult to solve with standard (deterministic) optimization algorithms. This is particularly true when the functions $F(\,\cdot\,,w)$ are computationally expensive to evaluate, *e.g.*, when they are defined in terms of optimal values of other optimization problems that depend on both x and w, as in the case of two-stage stochastic optimization problems. For these problems, a common strategy is to use a decomposition method, such as the L-shaped method [12] [98]. Two versions of this SAA-based method are considered here: a "single-cut" version and a "multi-cut" version [10]. These methods are described below for the problem obtained after expressing $F(\,\cdot\,,w)$ as

$$F(\,\cdot\,,w) = \bar{F} + \hat{F}(\,\cdot\,,w).$$

The "single-cut" L-shaped method consists of performing the following step at each iteration $k \in \mathbb{Z}_+$:

$$x_k = \arg\min\left\{ \bar{F}(x) + \max_{0\le j<k} a_j + b_j^T(x-x_j) \;\middle|\; x\in\mathcal{X} \right\}, \tag{2.4}$$

where

$$a_j := m^{-1}\sum_{i=1}^{m} \hat{F}(x_j,w_i) \quad \text{and} \quad b_j \in m^{-1}\sum_{i=1}^{m} \partial\hat{F}(x_j,w_i).$$

The "multi-cut" version of the method consists of performing the following step at each iteration $k \in \mathbb{Z}_+$:

$$x_k = \arg\min\left\{ \bar{F}(x) + m^{-1}\sum_{i=1}^{m} \max_{0\le j<k} c_{ij} + d_{ij}^T(x-x_j) \;\middle|\; x\in\mathcal{X} \right\}, \tag{2.5}$$

where

$$c_{ij} := \hat{F}(x_j,w_i) \quad \text{and} \quad d_{ij} \in \partial\hat{F}(x_j,w_i).$$

Some key properties of these methods are that the functions

$$\max_{0\leq j<k} a_j + b_j^T(x - x_j) \quad \text{and} \quad m^{-1}\sum_{i=1}^{m} \max_{0\leq j<k} c_{ij} + d_{ij}^T(x - x_j)$$

are gradually-improving piece-wise linear under-estimators of $m^{-1}\sum_{i=1}^{m} \hat{F}(\,\cdot\,, w_i)$, and that parallelization can be easily exploited when computing a_j, b_j, c_{ij}, and d_{ij}. However, on some nonlinear problems, it can take many iterations for these piece-wise linear under-estimators to become adequate approximations of the function $m^{-1}\sum_{i=1}^{m} \hat{F}(\,\cdot\,, w_i)$, and hence many iterations for the iterates (2.4) and (2.5) to be adequate candidate solutions of (2.3).

2.3.3 Stochastic Hybrid Approximation

As noted earlier, the authors in [21] propose an SHA algorithm for solving stochastic optimization problems of the form of (2.1) that combines elements found in algorithms based on both external and internal sampling, namely using deterministic approximations and noisy observations, respectively. The algorithm consists of repeating the following steps at each iteration $k \in \mathbb{Z}_+$: First, a candidate solution is obtained using

$$x_k = \arg\min_{x\in\mathcal{X}} \ \mathcal{F}_k(x), \tag{2.6}$$

where $\mathcal{F}_k$ is a deterministic strongly-convex differentiable approximation of $\mathcal{F}$. Then, $\xi_k \in \partial F(x_k, w_k)$ is computed after drawing an independent sample w_k of w, and then a new approximation $\mathcal{F}_{k+1}$ is constructed by incorporating this observation into $\mathcal{F}_k$. More specifically, $\mathcal{F}_{k+1}$ is obtained using

$$\mathcal{F}_{k+1}(x) = \mathcal{F}_k(x) + \alpha_k\big(\xi_k - \nabla\mathcal{F}_k(x_k)\big)^T x, \ \forall x, \tag{2.7}$$

where α_k are step lengths that satisfy similar conditions as in SA algorithms [8] [42], and $\mathcal{F}_0$ is an initial approximation provided by the user. The authors in [21] show that the iterates x_k convergence almost surely to an optimal point of (2.1) under the stated and other minor technical assumptions. They claim that the strengths of this approach are its ability to exploit an accurate initial approximation $\mathcal{F}_0$, and the fact that the slope corrections do not change the structure of the approximations. For completeness, the convergence proof from [21] of this SHA algorithm has been included in Appendix A.

An important observation that will be user later is that, from (2.7), it follows that the function approximations can be expressed

$$\mathcal{F}_k(x) = \mathcal{F}_0(x) + g_k^T x, \ \forall x, \tag{2.8}$$

where g_k satisfies $g_0 = 0$ and

$$g_{k+1} = g_k + \alpha_k\big(\xi_k - \nabla\mathcal{F}_0(x_k) - g_k\big) \tag{2.9}$$

for all $k \in \mathbb{Z}_+$.

2.4 Two-Stage Stochastic Economic Dispatch

In this section, a two-stage version of the stochastic ED problem is described and some important properties are derived. In addition, the application of the SHA algorithm for this problem is discussed and an interesting connection with another algorithm is shown. This problem is used in the next section to compare the performance of the algorithms considered in this chapter.

2.4.1 Problem Formulation

The stochastic ED problem considered here consists of determining generator powers for a specific time in the near future, say one hour ahead, taking into account the uncertainty associated with renewable energy and the cost of making power balancing adjustments. Mathematically, the problem is expressed as

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & \varphi_0(p) + \mathcal{Q}(p) \\ \text{subject to} \quad & p \in \mathcal{P}, \end{aligned} \tag{2.10}$$

where $p \in \mathbb{R}^{n_p}$ is the vector of planned generator powers, n_p is the number of generators in the network, φ_0 is a strongly convex separable quadratic function that quantifies generation cost, and the set

$$\mathcal{P} := \{\ \bar{p} \mid p^{\min} \leq \bar{p} \leq p^{\max}\ \}$$

represents generator power limits. The *recourse* function $\mathcal{Q}$ quantifies the expected cost of generation adjustments and is given by

$$\mathcal{Q} := \mathbb{E}\left[Q(\,\cdot\,, r)\right],$$

where $r \in \mathbb{R}^{n_r}_+$ is a vector of random powers from renewable energy sources, n_r is the number of such sources, $\mathbb{E}\left[\,\cdot\,\right]$ denotes expectation with respect to r, and $Q(p, r)$ is the optimal objective value of

$$\begin{aligned}
\underset{q,\theta,s}{\text{minimize}} \quad & \varphi_1(q) && \text{(2.11a)}\\
\text{subject to} \quad & G(p+q) + Rs - Dd - A\theta = 0 && \text{(2.11b)}\\
& p^{\min} \leq p + q \leq p^{\max} && \text{(2.11c)}\\
& z^{\min} \leq J\theta \leq z^{\max} && \text{(2.11d)}\\
& 0 \leq s \leq r. && \text{(2.11e)}
\end{aligned}$$

In problem (2.11), $q \in \mathbb{R}^{n_p}$ are generator power adjustments, $\theta \in \mathbb{R}^{n_b}$ are bus voltage angles, n_b is the number of non-slack buses, $s \in \mathbb{R}^{n_r}$ are powers from renewable energy sources after possible curtailments, $d \in \mathbb{R}^{n_d}$ are load power consumptions, n_d is the number of loads, and $\{G, R, D, A, J\}$ are constant sparse matrices. The function φ_1 is a strongly convex separable quadratic function that quantifies the cost of generation adjustments, which are assumed to be costly. Constraint (2.11b) enforces power balance at every bus of the network using a DC model [108], and constraint (2.11d) enforces branch flow limits due to thermal ratings. The notation $\mathcal{K}(p, r)$ is used to denote the feasible set of the second-stage problem (2.11) for a given (p, r). It is noted here that, although not explicitly shown by the notation, $\mathcal{Q}$, Q, and $\mathcal{K}$ also depend on the load power consumptions d.

This model, albeit simplified, captures key properties of the stochastic ED problem and hence make it an adequate model for performing an initial evaluation of algorithms. The key properties captured are that generator powers need to be planned in advance, especially for "baseload" units, which typically have limited flexibility, and that balancing adjustments typically require using resources that have a higher cost.

2.4.2 Assumptions

The following technical assumptions are made about the second-stage problem (2.11):

A1. $p^{\min} < p^{\max}$ and $z^{\min} < z^{\max}$ (element-wise).

A2. relint $\mathcal{K}(0,0) \neq \emptyset$.

A3. $[\ G \quad A\ ]$ is full row rank.

Assumption A1 ensures that bounds are non-trivial. In assumption A2, relint denotes relative interior, as defined in Section 2.1.3 of [16]. This assumption guarantees that the load can be served purely by adjusting generator dispatches in real-time and without relying on powers from renewable energy sources, even if generator and branch limits are slightly perturbed. Assumption A3 guarantees that any new small power injections and consumptions on arbitrary buses of the network can be balanced by adjusting generator powers and branch power flows. These assumptions will be used in the next subsection for deriving important properties about the recourse function.

2.4.3 Properties

In order to determine the properties of problem (2.10), the properties of the functions $\mathcal{Q}$ and $Q(\,\cdot\,, r)$ must be understood. Of particular importance are boundedness, convexity, and differentiability.

With regards to boundedness, assumption A2 gives $\mathcal{K}(p,r) \neq \emptyset$ for any p and any $r \geq 0$, so problem (2.11) is always feasible. Since the optimal q of problem (2.11) satisfies $q \in -p + \mathcal{P}$, and $\mathcal{P}$ is bounded, it follows that for any p there are constants $M_1(p)$ and $M_2(p)$ such that

$$-\infty < M_1(p) \leq Q(p,r) \leq M_2(p) < \infty$$

for all $r \geq 0$. This gives that $\mathcal{Q}(p) = \mathbb{E}\left[Q(p,r)\right]$ is finite for all p. Stochastic optimization problems with recourse that have this property are said to have *complete recourse* [102].

With regards to convexity, consider $r \geq 0$ and $p_i \in \mathcal{P}$ for $i \in \{1,2\}$. Then, for any $\varepsilon > 0$ there exists for each $i \in \{1,2\}$ a point $(q_i, \theta_i, s_i) \in \mathcal{K}(p_i, r)$ such that

$$\varphi_1(q_i) \leq Q(p_i, r) + \varepsilon.$$

It follows that for $\bar{p} := \alpha p_1 + (1-\alpha)p_2$, where $\alpha \in [0,1]$,

$$\begin{aligned} Q(\bar{p}, r) &= \inf\{\ \varphi_1(q) \mid (q, \theta, s) \in \mathcal{K}(\bar{p}, r)\ \} \\ &\leq \varphi_1(\alpha q_1 + (1-\alpha)q_2) \\ &\leq \alpha\varphi_1(q_1) + (1-\alpha)\varphi_1(q_2) \\ &\leq \alpha Q(p_1, r) + (1-\alpha)Q(p_2, r) + \varepsilon. \end{aligned}$$

The first inequality follows from the fact that

$$\alpha y_1 + (1-\alpha)y_2 \in \mathcal{K}(\bar{p}, r),$$

where $y_i := (q_i, \theta_i, s_i)$, $i \in \{1, 2\}$. Since ε is arbitrary, $Q(\,\cdot\,, r)$ is convex for every $r \geq 0$. It is straight forward to show that this implies that $\mathcal{Q}$ is convex [16].

Lastly, with regards to differentiability, since $Q(\,\cdot\,, r)$ is convex for all $r \geq 0$ and $\mathcal{Q}(p)$ is finite for all p, Proposition 2.10 of [103] gives

$$\partial\mathcal{Q} = \partial\mathbb{E}\left[Q(\,\cdot\,, r)\right] = \mathbb{E}\left[\partial Q(\,\cdot\,, r)\right],$$

where ∂f denotes the subdifferential of a function f. To characterize $\partial Q(\,\cdot\,, r)$, consider the Lagrangian of problem (2.11) for some (p, r),

$$\begin{aligned} L(q, \theta, s, \lambda, \mu, \pi) := \varphi_1(q) &- \\ &\lambda^T \Phi(q, \theta, s) + \\ &\mu^T (p + q - p^{\max}) - \\ &\pi^T (p + q - p^{\min}), \end{aligned}$$

with domain

$$\mathcal{L} := \{\ (q, \theta, s) \mid z^{\min} \leq J\theta \leq z^{\max},\ 0 \leq s \leq r\ \},$$

where $\Phi(q, \theta, s)$ denotes the left-hand side of (2.11b). The dual functional is then given by

$$g(\lambda, \mu, \pi) := \inf_{(q,\theta,s)\in\mathcal{L}} L(q, \theta, s, \lambda, \mu, \pi),$$

and the dual problem by

$$\sup_{(\lambda,\mu,\pi)\in\mathcal{T}} g(\lambda,\mu,\pi),$$

where

$$\mathcal{T} := \{\ (\lambda,\mu,\pi) \mid \mu \geq 0,\ \pi \geq 0\ \}.$$

From feasibility and Slater's condition [16], strong duality holds and hence

$$Q(p,r) = \sup_{\nu\in\mathcal{T}} -\psi(\nu), \tag{2.12}$$

where $\psi := -g$ and $\nu := (\lambda,\mu,\pi)$. A similar analysis using $(p+\Delta p, r)$, where Δp is any perturbation, gives

$$Q(p+\Delta p, r) = \sup_{\nu\in\mathcal{T}} \left\{\ \nu^T H\Delta p - \psi(\nu)\ \right\},$$

where $H := \left[\ -G^T\ I\ \ -I\ \right]^T$. From this and considering $\operatorname{dom}\psi = \mathcal{T}$, it follows that

$$Q(p+\Delta p, r) = \psi^*(H\Delta p), \tag{2.13}$$

where ψ^* is the conjugate function of ψ, as defined in [16] and [76]. The function ψ is convex and closed, and equation (2.12) together with the fact that $Q(p,r)$ is finite imply that ψ is proper. Hence, Theorem 27.1 of [76] gives

$$\partial\psi^*(0) = \arg\min_{\nu\in\mathcal{T}} \psi(\nu). \tag{2.14}$$

The function ψ^* is convex and proper, and (2.13) implies that $0 \in \operatorname{dom}\psi^*$. Now, since perturbing p by Δp is equivalent to injecting powers $G\Delta p$ into the network and reducing both generator upper and lower limits by Δp in (2.11), (2.13) implies that $\psi^*(H\Delta p)$ equals $Q(p,r)$ but for a perturbed version of problem (2.11). From a similar analysis, one gets that $\psi^*(\epsilon)$, where $\epsilon := (\epsilon_\lambda, \epsilon_\mu, \epsilon_\pi)$ is small, equals $Q(p,r)$ but for a perturbed version of problem (2.11) with extra power consumptions ϵ_λ (not necessarily at load buses), and generator upper and lower limits reduced by ϵ_μ and increased by ϵ_π, respectively. From assumptions A1, A2, and A3, this perturbed problem is still feasible, its optimal objective value is finite, and hence $\epsilon \in \operatorname{dom}\psi^*$. This implies that 0 is in the interior of $\operatorname{dom}\psi^*$. Hence, Theorem

23.9 of [76] applies and it gives that (2.13) implies

$$\partial Q(p + \Delta p, r) = H^T \partial \psi^*(H\Delta p).$$

Combining this with (2.14) results in

$$\begin{aligned} \partial Q(p, r) &= H^T \partial \psi^*(0) \\ &= H^T \operatorname*{arg\,min}_{\nu \in \mathcal{T}} \psi(\nu) \\ &= H^T \operatorname*{arg\,max}_{(\lambda,\mu,\pi) \in \mathcal{T}} g(\lambda, \mu, \pi). \end{aligned} \tag{2.15}$$

In other words, the subdifferential of $Q(\,\cdot\,, r)$ at p equals the set of optimal dual variables, each multiplied by H^T. From the first-order optimality conditions of problem (2.11), we know that

$$\nabla \varphi_1(q^*) = G^T \lambda^* - \mu^* + \pi^* = -H^T \nu^*, \tag{2.16}$$

where q^* and $\nu^* := (\lambda^*, \mu^*, \pi^*)$ are part of a primal-dual solution of (2.11) for (p, r). From the strong convexity of φ_1, q^* is unique and hence (2.15) and (2.16) give

$$\partial Q(p, r) = \{-\nabla \varphi_1(q^*)\}.$$

Therefore, $Q(\,\cdot\,, r)$ is differentiable with gradient $-\nabla \varphi_1(q^*)$ at p, and $\mathcal{Q}$ is differentiable with gradient $-\mathbb{E}\left[\nabla \varphi_1(q^*)\right]$ at p.

2.4.4 Model of Uncertainty

The random renewable power injections r are modeled using the expression

$$r = \Pi_r \left(r_0 + \delta\right),$$

where r_0 is a vector of base powers, $\delta \sim \mathcal{N}(0, \Sigma)$, Π_r is the projection operator onto the set

$$\mathcal{R} := \{\ \bar{r} \mid 0 \leq \bar{r} \leq r^{\max}\ \},$$

and the vector $r^{\max}$ represents power capacities of renewable energy sources.

The use of Gaussian random variables for modeling renewable energy uncertainty is

common in the literature [64] [74]. To model spatial correlation between nearby renewable energy sources, a sparse non-diagonal covariance matrix Σ is considered here. The off-diagonal entries of this matrix corresponding to pairs of renewable sources that are "nearby" are such that the correlation coefficient between their powers equals a pre-selected value ρ. In the absence of geographical information, a pair of sources are considered as being "nearby" if the network shortest path between the buses where they are connected is less than or equal to some pre-selected number of edges N. We refer to this maximum number of edges as the correlation distance.

2.4.5 Adaptive Certainty-Equivalent Method

In order to apply the SHA algorithm described in Section 2.3.3 to solve the two-stage stochastic ED problem (2.10), an initial deterministic strongly-convex differentiable function $\mathcal{F}_0$ is required. Here, motivated by the fact that CE models are common in power system operations, which are obtained by replacing random vectors with their expected values, and using the properties derived in Section 2.4.3, the following function is proposed:

$$\mathcal{F}_0 := \varphi_0 + Q(\,\cdot\,, \mathbb{E}\,[r]).$$

Since the first problem solved by the SHA algorithm using this initial function approximation consists of the CE problem

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & \varphi_0(p) + Q(p, \mathbb{E}\,[r]) \\ \text{subject to} \quad & p \in \mathcal{P}, \end{aligned}$$

and this is iteratively updated to better approximate the original problem (2.1), this algorithm is referred here as the *Adaptive Certainty-Equivalent* method, or AdaCE.

2.4.6 Connection of AdaCE with Proximal Algorithms

An interesting observation is that, when applied to the two-stage stochastic ED problem (2.10), the AdaCE algorithm is actually equivalent to a *stochastic proximal gradient* algorithm with a non-Euclidean metric. To see this, let B be a positive definite matrix, a a

vector, and c a scalar such that

$$\varphi_0(p) = \frac{1}{2}p^T Bp + a^T p + c, \ \forall p.$$

Then, ignoring constant terms, the approximation (2.8) used by the algorithm is given by

$$\mathcal{F}_k(p) = \frac{1}{2}p^T Bp + (a + g_k)^T p + Q(p, \bar{r}), \ \forall p,$$

where $\bar{r} := \mathbb{E}[r]$. By defining $\bar{p}_k := -B^{-1}(a + g_k)$ and again ignoring constant terms, approximation model is equivalent to

$$\begin{aligned}\mathcal{F}_k(p) &= \frac{1}{2}(p - \bar{p}_k)^T B(p - \bar{p}_k) + Q(p, \bar{r}) \\ &= \frac{1}{2}\|p - \bar{p}_k\|_B^2 + Q(p, \bar{r}),\end{aligned}$$

where $\| \cdot \|_B$ denotes the quadratic norm defined by B. Hence, step (2.6) of the algorithm consists of

$$\begin{aligned}p_k &= \arg\min_{p \in \mathcal{P}} \left\{ \frac{1}{2}\|p - \bar{p}_k\|_B^2 + Q(p, \bar{r}) \right\} \\ &= \text{prox}_{Q(\,\cdot\,, \bar{r})}(\bar{p}_k),\end{aligned}$$

where prox is the proximal operator defined by

$$\text{prox}_f(y) = \arg\min_{p \in \mathcal{P}} \left\{ \frac{1}{2}\|p - y\|_B^2 + f(p) \right\}$$

for any function f and vector y. Now, the update (2.9) can be expressed as

$$g_{k+1} = g_k + \alpha_k \left(\nabla Q(p_k, r_k) - \nabla Q(p_k, \bar{r}) - g_k \right). \tag{2.17}$$

Letting

$$\hat{p}_k := -B^{-1} \left(\nabla Q(p_k, r_k) - \nabla Q(p_k, \bar{r}) + a \right)$$

and using the definition of $\bar{p}_k$ in (2.17) gives

$$\bar{p}_{k+1} = (1 - \alpha_k)\bar{p}_k + \alpha_k \hat{p}_k.$$

Furthermore, by defining the functions $\mathcal{S}$ and $S(\,\cdot\,, r)$ by

$$\begin{aligned} S(p, r) &= Q(p, r) - Q(p, \bar{r}) + \varphi_0(p) \\ \mathcal{S}(p) &= \mathbb{E}\left[S(p, r)\right], \end{aligned}$$

it follows that

$$\hat{p}_k = p_k - B^{-1}\nabla S(p_k, r_k).$$

Hence, the AdaCE algorithm applied to problem (2.10) consists of repeating the steps

$$\begin{aligned} p_k &= \mathrm{prox}_{Q(\,\cdot\,,\bar{r})}(\bar{p}_k) \\ \hat{p}_k &= p_k - B^{-1}\nabla S(p_k, r_k) \\ \bar{p}_{k+1} &= (1 - \alpha_k)\bar{p}_k + \alpha_k \hat{p}_k \end{aligned}$$

at each iteration $k \in \mathbb{Z}_+$, which is equivalent to Algorithm 2.2 of [79], in a space with metric induced by $\|\,\cdot\,\|_B$, applied to problem (2.10) expressed as

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & \mathcal{S}(p) + Q(p, \bar{r}) \\ \text{subject to} \quad & p \in \mathcal{P}. \end{aligned}$$

2.5 Implementation

The stochastic ED problem described in Section 2.4 was constructed using and extending the C library `PFNET` and its Python wrapper [1]. The code for doing this construction has been included in the Python package `GRIDOPT` [2] along with the problem-specific SAA benchmark algorithms. The routine for constructing the sparse covariance matrix Σ was also implemented in C and incorporated into `PFNET`. For obtaining $\Sigma^{1/2}$ to sample $\delta \sim \mathcal{N}(0, \Sigma)$, the sparse Cholesky code `CHOLMOD` was used via its Python wrapper [3] [19]. An efficient interior-point solver for sparse Quadratic Programming (QP) problems based on Section 16.6 of [56] was implemented in Python for solving second-stage problems (2.11), modified CE problems (2.6), and L-shaped subproblems (2.4) and (2.5). The code for this algorithm

[1] `http://pfnet-python.readthedocs.io`
[2] `http://gridopt.readthedocs.io`
[3] `http://pythonhosted.org/scikits.sparse`

and the stochastic subgradient and SHA algorithms has been included in the Python package `OPTALG` [4].

2.6 Numerical Experiments

To compare the performance of the algorithms described in Section 2.3, four test cases were considered. One case was obtained from a North American system operator, while the rest were obtained from the package MATPOWER [109]. Important properties of these cases are shown in Table 2.1.

Table 2.1: Properties of test cases

name	buses	branches	generators
Case A	2454	2781	275
Case B	3012	3572	385
Case C	3120	3693	298
Case D	9241	16049	1445

The separable generation cost function φ_0 was constructed using uniformly random coefficients in $[0.01, 0.05]$ for the quadratic terms, and uniformly random coefficients in $[10, 50]$ for the linear terms. These coefficients assume that power quantities are in units of MW and are consistent with those found in several MATPOWER cases. For the separable generation adjustment cost function φ_1, the coefficients for the quadratic terms were set to equal those of φ_0 scaled by a constant $\gamma > 1$, and the coefficients of the linear terms were set to zero. The scaling factor γ was set large enough to make balancing costs with adjustments higher than with planned powers. The coefficients for the linear terms were set to zero to penalize both positive and negative power adjustments equally.

Renewable energy sources were added to each of the power networks at the generator buses. The capacity $r_i^{\max}$ of each source was set to $1^T d/n_r$, where d is the vector of load consumptions and n_r is the number of renewable sources. The base powers were set using $r_0 = 0.5 r^{\max}$ so that the base renewable energy penetration was 50% of the load, which is a high penetration scenario. The standard deviations of the renewable powers, *i.e.*, $\operatorname{diag} \Sigma^{1/2}$, were also set to $0.5 r^{\max}$, which corresponds to a high-variability scenario. For the off-diagonals of Σ, a correlation coefficient ρ of 0.05 and a correlation distance N of 5 were used. The sparsity pattern of the resulting Σ for Case A is shown in Figure 2.1.

[4] `http://optalg.readthedocs.io`

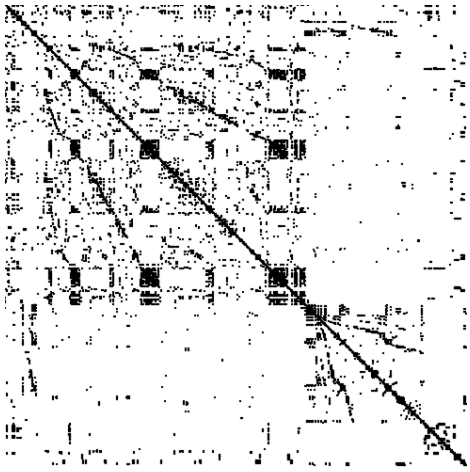

Figure 2.1: Sparsity pattern of Σ of Case A

The algorithms were applied to the stochastic ED problem associated with each of the test cases of Table 2.1 with the stated cost functions and uncertain renewable power injections. These algorithms were run for a fixed number of iterations and their progress was compared. More especially, every 20 iterations for the AdaCE algorithm, every 50 iterations of the stochastic subgradient algorithm, and every iteration of the L-shaped algorithms, an approximate expected cost associated with the current iterate was computed using 2000 fixed samples of renewable powers. For the L-shaped algorithms, 100 samples of renewable powers (scenarios) were used to form the SAA problem (2.3). For the stochastic subgradient and AdaCE algorithms, step lenghts given by $1/(k+1)$ were used. Figure 2.2 shows the normalized expected cost as a function of time for each of the algorithms and cases, where SG denotes stochastic subgradient, LS denotes L-shaped with single cut, LSMC denotes L-shaped with multiple cuts, and CE denotes certainty-equivalent. The CE line shows the expected cost of the solution of the certainty-equivalent problem, which was used as a starting point for the stochastic subgradient algorithm.

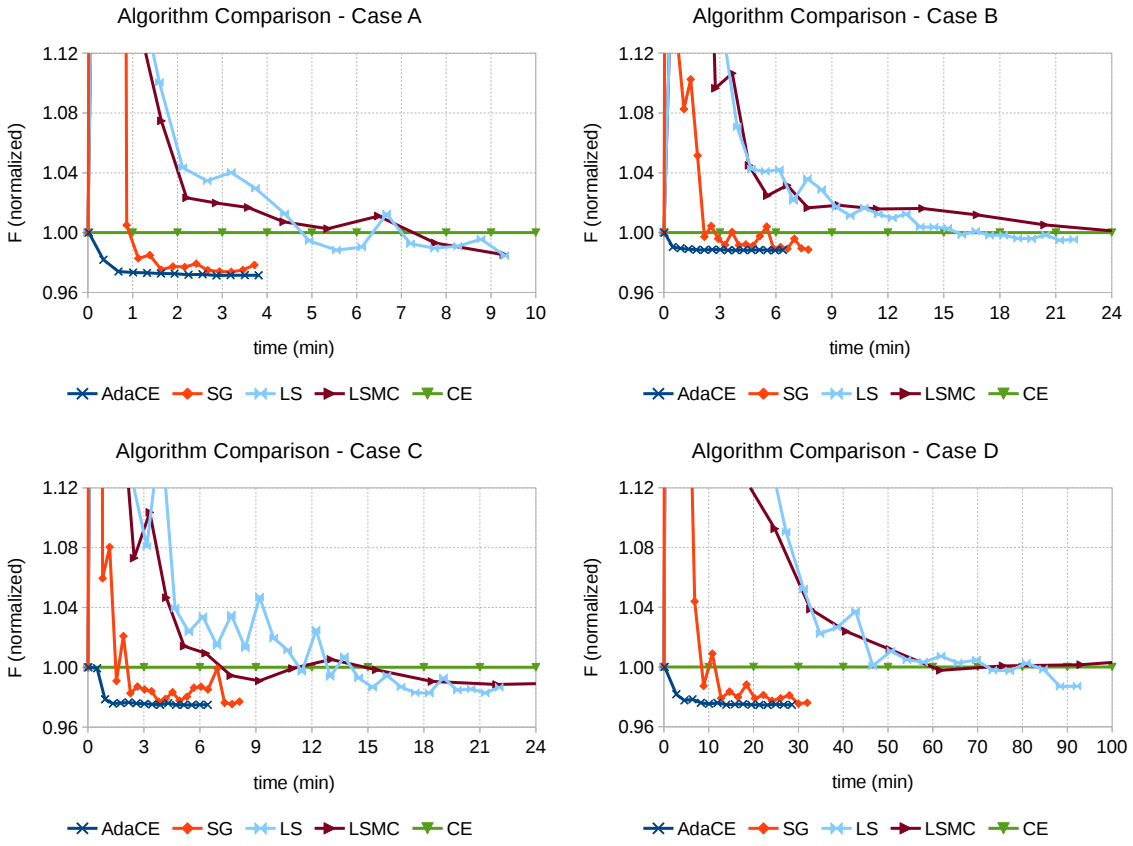

Figure 2.2: Performance of algorithms on test cases

As the figure shows, the expected cost associated with the iterates of the stochastic subgradient algorithm initially increases with respect to that of the CE solution on all the cases. This is a manifestation of the well-known fact that large step lengths, *e.g.*, the initial ones in our case, poorly suppress noise [50] [53]. The expected cost eventually decreases below the cost of the CE solution, but it experiences significant variations in some cases, *e.g.*, Case B and C. For the L-shaped algorithms, the expected cost starts high with respect to that of the CE solution. The reason is that these algorithms require several iterations to build an adequate model of the recourse function. Since each iteration of these algorithms is computationally expensive, they take a long time before producing iterates that are better than the CE solution. On the other hand, the AdaCE algorithm makes a more monotonic progress towards the solution and generates in much shorter times iterates that have a lower expected cost than the CE solution. This is in a way expected. As shown in Section 2.4.6,

the AdaCE algorithm is equivalent to a stochastic proximal gradient algorithm when applied to problem (2.10), and proximal-type stochastic algorithms are generally more stable and depend less on the choice of parameters, *e.g.* step lengths, than a stochastic subgradient algorithm [93]. Also, the AdaCE algorithm starts with a much better model of the recourse function compared to the L-shaped algorithms, and new information is incorporated into this model quickly after every iteration.

In these experiments, parallelization of computations has not been considered, and it is well-known that L-shaped algorithms can benefit significantly from this. Parallel versions of the L-shaped algorithms used here will be considered in future work as SAA-based benchmarks for assessing the performance of versions of the stochastic subgradient and AdaCE algorithms that also exploit parallelism.

2.7 Conclusions

In this chapter, stochastic optimization algorithms have been applied to a stochastic ED problem from power systems with high penetration of intermittent renewable energy such as wind and solar. The problem was formulated as a two-stage stochastic quadratic program with recourse, and important properties of the recourse function such as boundedness, convexity, and differentiability were analyzed. The SHA algorithm explored here was based on ideas proposed in the literature but considered using CE models for constructing initial expected-value function approximations, resulting in the AdaCE method. It was shown that when applied to the stochastic ED problem considered, this algorithm is equivalent to a stochastic proximal gradient algorithm equipped with a non-Euclidean metric. The performance of this algorithm was compared to that of a stochastic subgradient algorithm and two SAA-based algorithms on four test cases from power networks having 2500 to 9200 buses. The results showed that the AdaCE algorithm outperformed the other algorithms on the test cases considered, since its progress was less affected by noise and it produced better iterates.

Chapter 3

A Primal-Dual Extension for Risk-Averse Dispatch

3.1 Background and Overview

In many applications, optimization problems arise whose objective function or constraints have expectation functions. For example, in portfolio optimization, one is interested in selecting the portfolio that results in the lowest expected loss, and a bound on the risk of high losses may be enforced using Conditional Value-At-Risk (CVaR) [101]. In generation scheduling in modern low-carbon power systems, as stated in the previous chapters, an important task is to plan generator output powers in such a way that the expected system operation cost is minimized in the presence of uncertain inputs from renewable energy sources. In communication networks, one goal is to maximize throughput while bounding the expected packet loss or time delay in transmission [7]. In call centers, one is interested in minimizing cost while ensuring that the expected number of answered calls are above a certain level [3]. Solving optimization problems such as these that have expectation functions is in general a challenging task. The reason is that the multi-dimensional integrals that define the expectation functions cannot be computed with high accuracy in practice when the uncertainty has more than a few dimensions [53]. Hence, existing approaches used for solving these problems work with approximations instead of the exact expectation functions.

Two families of approaches have been studied for trying to solve optimization problems with expectation functions in a computationally tractable way. The first approach is

based on *external sampling*, which consists of sampling realizations of the uncertainty and replacing the stochastic problem with a deterministic approximation of it. This approach is also referred to as the SAA approach since the deterministic approximation is obtained by replacing expected values with sample averages [36]. The resulting deterministic problem, which is typically very large, is solved using techniques that exploit the problem structure, such as the Benders or L-shaped decomposition method [12] [98]. Several authors have analyzed the performance of SAA-based algorithms in the literature. The reader is referred to [36], [40], [84], and [101] for discussions on asymptotic convergence of solutions and optimal values, and on the accuracy obtained using finite sample sizes.

The second approach for solving optimization problems with expectation functions is based on *internal sampling*, which consists of periodically sampling realizations of the uncertainty during the execution of the algorithm. Some of the most studied algorithms of this family are SA algorithms. These algorithms update a solution estimate by taking steps along random directions. These random directions typically have the property that on expectation are equal to negative gradients or subgradients of the function to be minimized [36]. The advantage of these algorithms is that they typically have low demand for computer memory. Their convergence, however, is very sensitive to the choice of step lengths, *i.e.*, the lengths of the steps taken along the random directions. The fundamental issue is that the step lengths need to be small enough to adequately suppress noise, while at the same time large enough to avoid slow convergence. Since first proposed by Robbins and Monro [75], SA algorithms have been an active area of research. Most of the work has been done in the context of stochastic optimization problems with deterministic constraints. The reader is referred to [53] and [88] for important advancements and insights on these algorithms. To handle expectation functions in the constraints, to the best of our knowledge, SA approaches explored in the literature have been based on the Lagrangian method [7] [41] [42] [107]. More specifically, stochastic gradient or similar steps are used for minimizing the Lagrangian with respect to the primal variables and for maximizing the Lagrangian with respect to the dual variables.

Another important internal-sampling algorithm, which was investigated in Chapter 2, is the SHA algorithm proposed in [21] for solving stochastic problems with deterministic constraints. It consists of solving a sequence of deterministic approximations of the original problem whose objective functions are periodically updated using noisy subgradients. Its development was motivated by a dynamic resource allocation problem, and it belongs to

an important line of research that uses techniques from approximate dynamic programming for solving problems of such type [66] [68] [69]. The authors in [21] show that the iterates produced by the algorithm convergence almost surely to an optimal point under certain conditions. In Chapter 2, this algorithm is applied to a two-stage stochastic optimization problem with quadratic objectives and linear constraints for determining optimal generation dispatches in power systems with high penetration of renewable energy. The initial deterministic approximation used is based on the CE problem, *i.e.*, the problem obtained by replacing random variables with their expected values, resulting in the AdaCE method. The results obtained show the superiority of this method over benchmark algorithms that include stochastic subgradient descent and (sequential) algorithms based on SAA and the L-shaped decomposition method on the problem considered. In particular, the AdaCE algorithm is shown to produce better iterates than the other algorithms, and to be more robust against noise during the initial iterations compared to stochastic subgradient descent. These properties are attributed to the quality of the initial deterministic approximation used, and to an interesting connection between the AdaCE algorithm and a stochastic proximal gradient algorithm when applied to the problem considered.

In this chapter, the SHA algorithm proposed in [21] and further investigated in Chapter 2 is extended to also handle constraints with expectation functions. This is done by applying the hybrid procedure to the Lagrangian and combining it with dual stochastic gradient ascent. An analysis of the convergence of this new algorithm is included in Appendix B. Furthermore, numerical experience obtained from applying the resulting AdaCE method and benchmark algorithms to a risk-averse version of the stochastic ED problem considered in Chapter 2 is reported. The benchmark algorithms include a primal-dual SA algorithm and SAA-based algorithms with and without decomposition techniques. The test problem considered consists of a two-stage stochastic ED problem with quadratic objectives and a CVaR constraint that bounds the risk of high second-stage costs. The instances of the test problem used for the experiments come from real electric power networks having around 2500 and 3000 buses and hence several thousand combined first- and second-stage variables. The empirical results obtained suggest that the AdaCE method based on the extended SHA algorithm benefits from the accuracy of the CE-based initial deterministic approximations of the expectation functions and converges to a solution faster than the benchmark algorithms.

3.2 Convex Stochastic Optimization Problems with Expected-Value Constraints

The general stochastic optimization problems considered in this chapter are of the form

$$\begin{aligned} &\underset{x\in\mathcal{X}}{\text{minimize}} \quad \mathcal{F}(x) \\ &\text{subject to} \;\; \mathcal{G}(x) \leq 0, \end{aligned} \tag{3.1}$$

where $\mathcal{X}$ is a compact convex set, $\mathcal{F} := \mathbb{E}\,[F(\,\cdot\,, w)]$, $F(\,\cdot\,, w)$ is convex for each random vector $w \in \Omega$, $\mathcal{G} := \mathbb{E}\,[G(\,\cdot\,, w)]$, $G(\,\cdot\,, w)$ is a vector-valued function composed of convex functions for each $w \in \Omega$, Ω is a compact set, and $\mathbb{E}\,[\,\cdot\,]$ denotes expectation with respect to w. It is assumed that all functions are continuous in $\mathcal{X}$, and that Slater's condition and hence strong duality holds [16].

The Lagrangian of problem (3.1) is given by

$$\mathcal{L}(x, \lambda) := \mathcal{F}(x) + \lambda^T \mathcal{G}(x),$$

and the "noisy" Lagrangian is defined as

$$L(x, \lambda, w) := F(x, w) + \lambda^T G(x, w).$$

The vector x^* denotes a primal optimal point of problem (3.1). Similarly, the vector λ^* denotes a dual optimal point of problem (3.1). It is assumed that $\lambda^* \in \Lambda$, where

$$\Lambda := \{\, \lambda \mid 0 \leq \lambda < 1\lambda^{\max} \,\},$$

1 is the vector of ones, and $\lambda^{\max}$ is some known positive scalar.

In the following sections, unless otherwise stated, all derivatives and subdifferentials are assumed to be with respect to the primal variables x.

3.3 SAA-Based Algorithms

A widely-used *external-sampling* approach for solving stochastic optimization problems of the form of (3.1) is to sample independent realizations of the random vector w, say $\{w_l\}_{l=1}^m$,

where $m \in \mathbb{Z}_{++}$, and replace expected values with samples averages. The resulting deterministic problem approximation is

$$\begin{aligned} \underset{x \in \mathcal{X}}{\text{minimize}} \quad & m^{-1} \sum_{l=1}^{m} F(x, w_l) \\ \text{subject to} \quad & m^{-1} \sum_{l=1}^{m} G(x, w_l) \leq 0. \end{aligned} \tag{3.2}$$

It can be shown that under certain conditions and sufficiently large m the solutions of (3.2) are arbitrarily close to those of (3.1) with high probability [36] [101].

Due to the requirement of choosing m sufficiently large in order to get an accurate estimate of a solution of (3.1), problem (3.2) can be difficult to solve with standard (deterministic) optimization algorithms. This is particularly true when $F(\,\cdot\,, w)$ or $G(\,\cdot\,, w)$ are computationally expensive to evaluate, *e.g.*, when they are defined in terms of optimal values of other optimization problems that depend on both x and w. In this case, a common strategy for solving (3.2) is to use decomposition. One widely-used decomposition approach is based on cutting planes and consists of first expressing (3.2) as

$$\begin{aligned} \underset{x \in \mathcal{X}}{\text{minimize}} \quad & \bar{F}(x) + m^{-1} \sum_{l=1}^{m} \hat{F}(x, w_l) \\ \text{subject to} \quad & \bar{G}(x) + m^{-1} \sum_{l=1}^{m} \hat{G}(x, w_l) \leq 0, \end{aligned}$$

and then performing the following step at each iteration $k \in \mathbb{Z}_{+}$:

$$\begin{aligned} x_k = \arg\min \; \Big\{ \bar{F}(x) + \max_{0 \leq j < k} a_j + b_j^T (x - x_j) \;\Big|\; \\ x \in \mathcal{X},\; \bar{G}_i(x) + \max_{0 \leq j < k} c_{ij} + d_{ij}^T (x - x_j) \leq 0,\; \forall i \Big\}, \end{aligned} \tag{3.3}$$

where

$$\begin{aligned} a_j &= m^{-1} \sum_{l=1}^{m} \hat{F}(x_j, w_l), \qquad & b_j &\in m^{-1} \sum_{l=1}^{m} \partial \hat{F}(x_j, w_l), \\ c_{ij} &= m^{-1} \sum_{l=1}^{m} \hat{G}_i(x_j, w_l), \qquad & d_{ij} &\in m^{-1} \sum_{l=1}^{m} \partial \hat{G}_i(x_j, w_l), \end{aligned}$$

and $\bar{G}_i(x,w)$ and $\hat{G}_i(x,w)$ denote the i-th component of $\bar{G}(x,w)$ and $\hat{G}(x,w)$, respectively. Some key properties of this algorithm are that the functions

$$\max_{0\leq j<k} a_j + b_j^T(x - x_j) \quad \text{and} \quad \max_{0\leq j<k} c_{ij} + d_{ij}^T(x - x_j)$$

are gradually-improving piece-wise linear under-estimators of

$$m^{-1}\sum_{l=1}^{m} \hat{F}(\,\cdot\,, w_l) \quad \text{and} \quad m^{-1}\sum_{l=1}^{m} \hat{G}_i(\,\cdot\,, w_l),$$

respectively, and that parallelization can be easily exploited for computing a_j, b_j, c_{ij} and d_{ij} efficiently. The interested reader is referred to [12] and [98] for more details about decomposition techniques based on cutting planes.

3.4 Primal-Dual SA Algorithm

As already mentioned, the authors in [41] and [42] describe an *internal-sampling* primal-dual stochastic approximation algorithm for solving problems of the form of (3.1). It consists of performing the following steps at each iteration $k \in \mathbb{Z}_+$:

$$x_{k+1} = \Pi_x(x_k - \alpha_k \xi_k) \tag{3.4a}$$

$$\lambda_{k+1} = \Pi_\lambda\big(\lambda_k + \alpha_k G(x_k, w_k)\big), \tag{3.4b}$$

where $\xi_k \in \partial L(x_k, \lambda_k, w_k)$, α_k are scalar step lengths, Π_x is projection on $\mathcal{X}$, Π_λ is projection on Λ, and w_k are independent samples of the random vector w. Since ξ_k is a noisy subgradient of $\mathcal{L}(\,\cdot\,, \lambda_k)$ at x_k, and $G(x_k, w_k)$ is a noisy gradient of $\mathcal{L}(x_k, \,\cdot\,)$ at λ_k, *i.e.*,

$$\mathbb{E}\,[\,\xi_k \,|\, \mathcal{W}_{k-1}\,] \in \partial\mathcal{L}(x_k, \lambda_k) \quad \text{and} \quad \mathbb{E}\,[\,G(x_k, w_k) \,|\, \mathcal{W}_{k-1}\,] = \nabla_\lambda \mathcal{L}(x_k, \lambda_k),$$

where $\mathcal{W}_k$ denotes the observation history $(w_0, \ldots, w_k)$, it is clear that this algorithm performs stochastic subgradient descent to minimize the Lagrangian with respect to x, and stochastic gradient ascent to maximize the Lagrangian with respect to λ.

In [41], the authors provide a convergence proof of this algorithm under certain conditions based on the "ODE method". These conditions include, among others, $\mathcal{F}$ being strictly convex and continuously differentiable, the components of $\mathcal{G}$ being convex and continuously

differentiable, and the existence of a point $\tilde{x}$ such that $\mathcal{G}(\tilde{x}) < 0$.

3.5 Primal-Dual SHA Algorithm

In [21], the authors propose an *internal-sampling* SHA algorithm for solving stochastic optimization problems of the form

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad \mathbb{E}\left[F(x, w)\right], \tag{3.5}$$

where $\mathcal{X}$ and $F(\,\cdot\,, w)$ have the same properties as those of problem (3.1). The algorithm consists of generating iterates

$$x_k = \arg\min \left\{ \mathcal{F}_k(x) \mid x \in \mathcal{X} \right\}$$

at each iteration $k \in \mathbb{Z}_+$, where $\mathcal{F}_k$ are strongly convex deterministic differentiable approximations of $\mathbb{E}\left[F(\,\cdot\,, w)\right]$. The first approximation $\mathcal{F}_0$ is provided by the user, and the rest are obtained by performing "noisy" slope corrections at each iteration using

$$\mathcal{F}_{k+1}(x) = \mathcal{F}_k(x) + \alpha_k(\hat{g}_k - \nabla\mathcal{F}_k(x_k))^T x,$$

where $\hat{g}_k \in \partial F(x_k, w_k)$, w_k are independent samples of the random vector w, and α_k are step lengths that satisfy conditions that are common in stochastic approximation algorithms [8] [42]. The slope corrections performed at each iteration are "noisy" because they are based on subgradients associated with specific realizations of the random vector. Roughly speaking, the conditions satisfied by the step lengths ensure that this noise is eventually averaged out. The authors show that the iterates x_k convergence almost surely to an optimal point of (3.5) under the stated and other minor technical assumptions. They claim that the strengths of this approach are its ability to exploit an accurate initial approximation $\mathcal{F}_0$, and the fact that the slope corrections do not change the structure of the approximations.

In Chapter 2, this SHA algorithm was applied to a two-stage stochastic ED problem that arises in power systems with high penetration of renewable energy. Motivated by the fact that in power systems operations planning CE models are common, the function $\mathcal{F}_0 = F(\,\cdot\,, \mathbb{E}\left[w\right])$ is used as the initial function approximation, resulting in the AdaCE method. The results obtained show how the algorithm outperforms common SAA-based

algorithms on the problem considered due to its ability to exploit the accuracy of the given initial function approximation. The algorithm is also shown to be less susceptible to noise compared to stochastic subgradient descent, and this is attributed to an interesting connection between the algorithm and a stochastic proximal gradient algorithm when applied to the problem considered.

In order to apply the AdaCE method to solve problem (3.1), which also has expectation functions in the constraints as opposed to only in the objective, a primal-dual extension of the SHA algorithm is proposed here. The proposed algorithm works with deterministic approximations of the Lagrangian, and combines the SHA procedure with dual stochastic gradient ascent, the latter which is motivated from algorithm (3.4). More specifically, the proposed algorithm consists of performing the following steps at each iteration $k \in \mathbb{Z}_+$:

$$x_k = \arg\min \{ \mathcal{F}_k(x) + \lambda_k^T \mathcal{G}_k(x) \mid x \in \mathcal{X} \} \tag{3.6a}$$

$$\lambda_{k+1} = \Pi_\lambda\big(\lambda_k + \alpha_k \hat{G}_k\big) \tag{3.6b}$$

$$\mathcal{F}_{k+1}(x) = \mathcal{F}_k(x) + \alpha_k(\hat{g}_k - g_k(x_k))^T x \tag{3.6c}$$

$$\mathcal{G}_{k+1}(x) = \mathcal{G}_k(x) + \alpha_k\big(\hat{J}_k - J_k(x_k)\big)x, \tag{3.6d}$$

where $\mathcal{F}_k$ and $\mathcal{G}_k$ are deterministic convex differentiable approximations of the stochastic functions $\mathcal{F}$ and $\mathcal{G}$, respectively, $\hat{G}_k := G(x_k, w_k)$, $\hat{g}_k \in \partial F(x_k, w_k)$, $g_k := \nabla \mathcal{F}_k$, $\hat{J}_k$ is a matrix whose rows are subgradients (transposed) of the components of $G(\,\cdot\,, w_k)$ at x_k, $J_k := \frac{\partial}{\partial x}\mathcal{G}_k$, α_k are scalar step lengths, and w_k are independent samples of the random vector w. Step (3.6a) generates primal iterates by minimizing deterministic approximations of the Lagrangian. Step (3.6b) updates dual iterates using noisy measurements of feasibility, namely, $\hat{G}_k = G(x_k, w_k)$. Lastly, steps (3.6c) and (3.6d) perform slope corrections on the deterministic approximations $\mathcal{F}_k$ and $\mathcal{G}_k$, respectively, using noisy subgradients of the stochastic function $\mathcal{F}$ and of the components of $\mathcal{G}$.

3.5.1 Convergence

An analysis of the convergence of the primal-dual SHA algorithm (3.6) is provided in Appendix B. In particular, it is shown that under certain assumptions, the primal iterates produced by the algorithm have a subsequence that converges to an optimal point of problem (3.1). The assumptions are similar to those needed for the convergence of the SHA algorithm (Appendix A, except for two extra assumptions. The extra assumptions have to

do with Lagrange multipliers of regularized versions of the problem, and with the "drift" of the algorithm. This "drift", which needs to remain bounded above in order for the algorithm to work, represents the cumulative sum of uncontrolled side effects produced by the algorithm. At the end of Appendix B, an intuitive justification for why this may hold in practice is provided.

3.5.2 Illustrative Examples

Two simple deterministic two-dimensional examples are used to illustrate the key properties of algorithm (3.6). In particular, they are used to show the effects of the quality of the initial function approximation $\mathcal{G}_0$ and of the dual bound $\lambda^{\max}$ (used for defining Λ in Section 3.2) on the performance of the algorithm, and to illustrate the geometric ideas behind the algorithm. The effects of the quality of $\mathcal{F}_0$ are similar or less severe than those of $\mathcal{G}_0$ and hence are not shown. Both examples considered are problems of the form

$$\begin{aligned} \underset{x_1,\,x_2}{\text{minimize}} \quad & \mathcal{F}(x_1, x_2) && \text{(3.7a)} \\ \text{subject to} \quad & \mathcal{G}(x_1, x_2) \leq 0 && \text{(3.7b)} \\ & x^{\min} \leq x_i \leq x^{\max},\ i \in \{1, 2\}. && \text{(3.7c)} \end{aligned}$$

The first example consists of the QP problem obtained using

$$\begin{aligned} \mathcal{F}(x_1, x_2) &= x_1^2 + x_2^2 \\ \mathcal{G}(x_1, x_2) &= -0.5x_1 - x_2 + 2 \\ (x^{\min}, x^{\max}) &= (-6, 6). \end{aligned}$$

The primal optimal point of the resulting problem is $(x_1^*, x_2^*) = (0.8, 1.6)$, and the dual optimal point associated with constraint (3.7b) is $\lambda^* = 3.2$. For the objective function, the initial approximation

$$\mathcal{F}_0(x_1, x_2) = 0.3(x_1 - 1)^2 + 2(x_2 - 0.5)^2$$

is used. For the constraint function, the initial approximations

$$\mathcal{G}_0^1(x_1, x_2) = -1.5x_1 - 2x_2$$
$$\mathcal{G}_0^2(x_1, x_2) = 3x_1 + x_2$$
$$\mathcal{G}_0^3(x_1, x_2) = 4x_1 + 9x_2,$$

are used, which can be considered "good", "poor", and "very poor", respectively, according to how their gradients compare against that of the exact function at the optimal point. Note that for this case, the function approximations are of the same form as the exact functions, *i.e.*, the objective function approximation is quadratic and the constraint function approximation is linear. The algorithm parameters used in this example are $\lambda_0 = \lambda^*/8$ and $\alpha_k = 1/(k+3)$.

Figure 3.1 shows the primal error, dual error, and drift as a function of the iterations of the algorithm for the different initial constraint approximations and dual bounds. The drift shown here is the maximum of the partial sums in (B.2) using $\sigma = 1$. As the figure shows, the primal and dual errors approach zero relatively fast and the drift remains small when using $\mathcal{G}_0^1$ and $\mathcal{G}_0^2$. On the other hand, when using the "very poor" approximation $\mathcal{G}_0^3$ and a loose dual bound, the primal error, dual error and drift all become large during the initial iterations of the algorithm. In this case, a very large number of iterations is required by the algorithm to approach the solution. Compared to this case, the performance of the algorithm is significantly improved when using a tighter dual bound. This is because the better bound prevents the dual iterates from continuing to grow, and this allows the slope corrections to "catch up" and make the constraint approximation more consistent with the exact function.

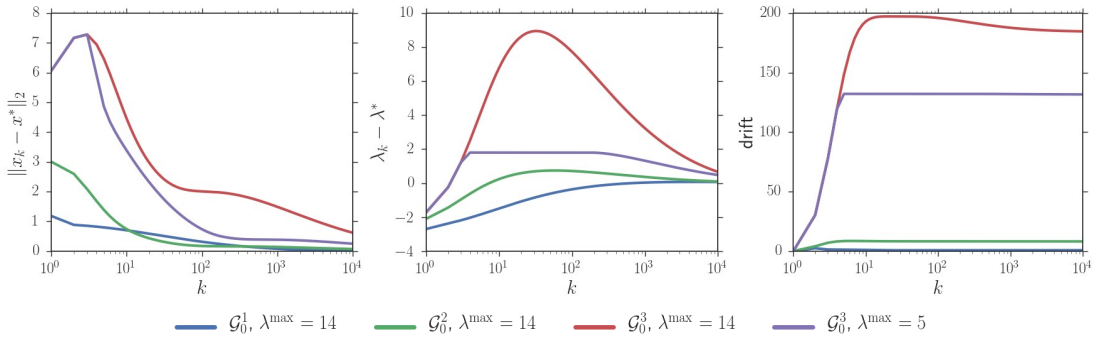

Figure 3.1: Algorithm performance on QP

Figure 3.2 shows the contours of the constraint and objective function approximations during certain iterations using $\mathcal{G}_0^2$ as well as those of the exact functions. The gradients of these functions are shown at the optimal primal points associated with the functions. As the figure shows, the initial approximations (top left) are poor since their gradients point in opposite directions as those of the exact functions (bottom right). As the number of iterations increases, the slope corrections performed by the algorithm improve the direction of these gradients and hence the quality of the iterates. An important observation is that the shape of the contours does not change since only the slope and not the curvature of the function approximations is affected. For example, the gradient of the objective function approximation at the solution matches well that of the exact function but its contours are elliptical and not circular.

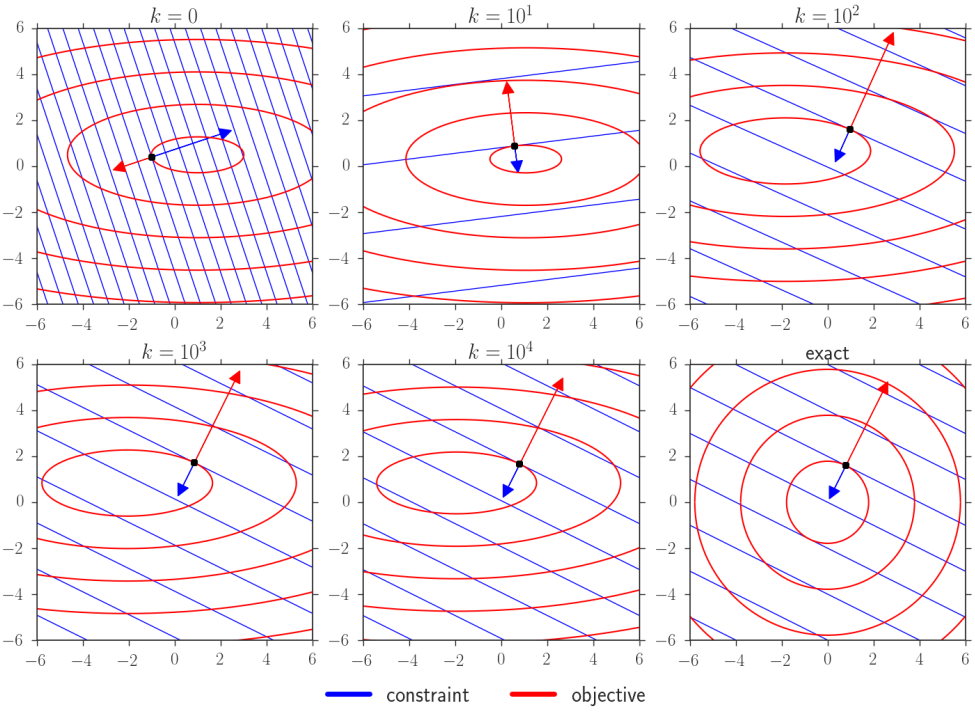

Figure 3.2: Contours of objective and constraint approximations of QP using $\mathcal{G}_0^2$ and $\lambda^{\max} = 14$

The second example consists of the Quadratically Constrained Quadratic Programming

(QCQP) problem obtained using

$$\begin{aligned}\mathcal{F}(x_1, x_2) &= x_1^2 + x_2^2 \\ \mathcal{G}(x_1, x_2) &= (x_1 - 1.8)^2 + (x_2 - 1.4)^2 - 1 \\ (x^{\min}, x^{\max}) &= (-6, 6).\end{aligned}$$

The primal optimal point for this problem is $(x_1^*, x_2^*) = (1, 0.8)$, and the dual optimal point associated with constraint (3.7b) is $\lambda^* = 1.3$. The initial objective function approximation used is

$$\mathcal{F}_0(x_1, x_2) = 0.3(x_1 - 1)^2 + 2(x_2 - 0.5)^2,$$

and the initial constraint function approximations used are

$$\begin{aligned}\mathcal{G}_0^1(x_1, x_2) &= -1.5x_1 - 2x_2 \\ \mathcal{G}_0^2(x_1, x_2) &= -1.5x_1 + 3x_2 \\ \mathcal{G}_0^3(x_1, x_2) &= 2x_1 + 4x_2,\end{aligned}$$

which again can be considered "good", "poor", and "very poor", respectively. Note that for this case, only the objective function approximation is of the same form as the exact function. The initial constraint approximation is not since it is linear while the exact function is quadratic. The algorithm parameters used in this example are $\lambda_0 = \lambda^*/6$ and $\alpha_k = 1/(k + 3)$.

Figures 3.3 and 3.4 show the performance of the algorithm and the evolution of the function approximations, respectively, on the second example. The results obtained are similar to those obtained on the first example. That is, the algorithm can perform well with reasonable initial function approximations. With very poor initial function approximations, the primal error, dual error and drift can all increase significantly during the initial iterations until the approximations become reasonable, resulting in the algorithm requiring a large number of iterations to approach the solution. Furthermore, when the initial function approximation is very poor, the quality of the dual bound plays an important role in the performance of the algorithm.

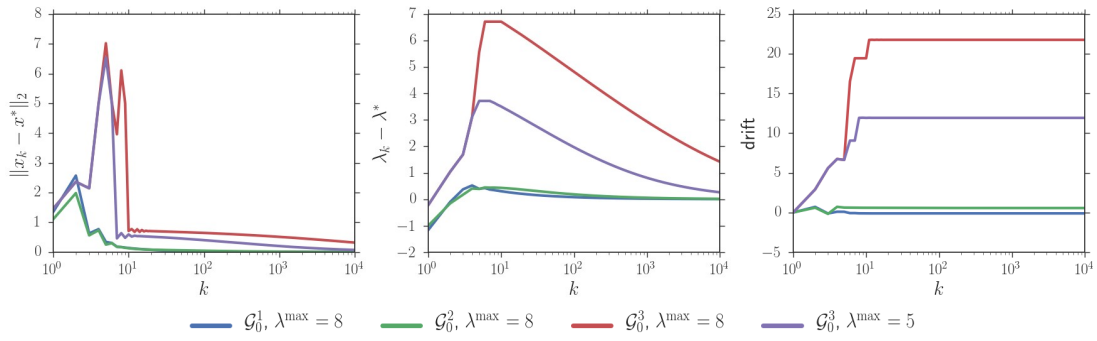

Figure 3.3: Algorithm performance on QCQP

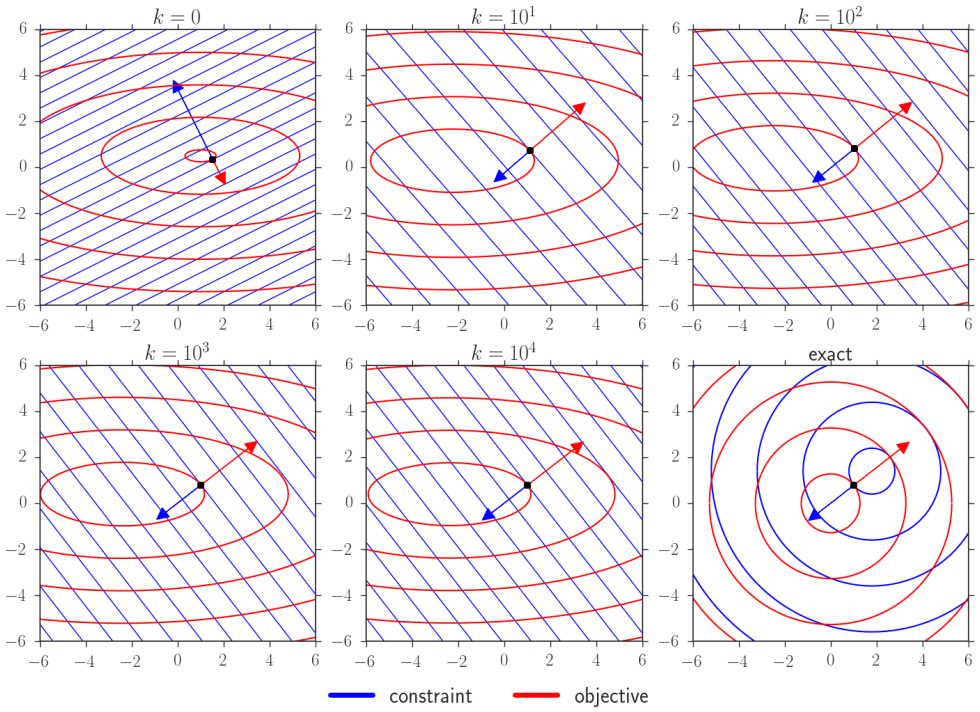

Figure 3.4: Contours of objective and constraint approximations of QCQP using $\mathcal{G}_0^2$ and $\lambda^{\max} = 8$

3.6 Two-Stage Stochastic Risk-Averse Economic Dispatch

In this section, the performance of the algorithms described in Sections 3.3, 3.4 and 3.5 is compared on a stochastic optimization problem that contains expectation functions in both the objective and constraints. This problem originates in power systems operations planning

under high penetration of renewable energy, where optimal risk-averse power generation dispatches are sought. Mathematically, the problem is formulated as a two-stage stochastic optimization problem with quadratic objectives and a CVaR constraint that limits the risk of high second-stage costs.

In electric power systems, the output powers of generators are typically planned in advance, *e.g.*, hours ahead. One reason for this is that low-cost generators typically have limited flexibility. Uncertainty in the operating conditions of a system, such as power injections from renewable energy sources, must be taken into account during generation scheduling. This is because real-time imbalances in power production and consumption are undesirable as they must be resolved with resources that typically have a high cost, such as flexible fast-ramping generators or load curtailments. Furthermore, due to the large quantities of money involved in the operation of a power system and society's dependence on reliable electricity supply, risk-averse operation policies are desirable. The problem of determining optimal risk-averse power generation policies considered here consists of determining generator output powers in advance that minimize expected operation cost, while ensuring that the risk of high balancing costs is below an acceptable limit.

3.6.1 Problem Formulation

Mathematically, the problem is formulated as

$$\underset{p \in \mathcal{P}}{\text{minimize}} \quad \varphi_0(p) + \mathbb{E}\left[Q(p, r)\right] \tag{3.8a}$$

$$\text{subject to} \quad \text{CVaR}_\gamma\left(Q(p, r) - Q^{\max}\right) \leq 0, \tag{3.8b}$$

where $p \in \mathbb{R}^{n_p}$ is a vector of planned generator output powers, n_p is the number of generators, $r \in \mathbb{R}^{n_r}_+$ is a vector of uncertain powers from renewable energy sources available in the near future, say a few hours ahead, n_r is the number of renewable energy sources, φ_0 is a strongly convex separable quadratic function that quantifies planned generation cost, $Q(p, r)$ quantifies the real-time power balancing cost for a given plan of generator outputs and realization of the uncertainty, $Q^{\max}$ is a positive constant that defines a desirable limit on the balancing cost, the set

$$\mathcal{P} := \{\ \bar{p} \mid p^{\min} \leq \bar{p} \leq p^{\max}\ \}$$

represents power limits of generators, and CVaR_γ is conditional value-at-risk with parameter $\gamma \in [0, 1]$. CVaR is a coherent risk measure that quantifies dangers beyond the Value-at-Risk (VaR) [77], and is given by the formula

$$\mathrm{CVaR}_\gamma\left(Z\right) = \inf_{t \in \mathbb{R}} \left(t + \frac{1}{1-\gamma} \mathbb{E}\left[(Z-t)_+\right] \right), \tag{3.9}$$

for any random variable Z, where $(\,\cdot\,)_+ := \max\{\,\cdot\,, 0\}$ [54] [101]. An important property of this risk measure is that constraint (3.8b) is a convex conservative approximation (when $Q(\,\cdot\,, r)$ is convex) of the chance constraint

$$\mathrm{Prob}\left(Q(p, r) \leq Q^{\max}\right) \geq \gamma,$$

and hence it ensures that the balancing cost is below the predefined limit with a desired probability [54]. The function $Q(p, r)$ is given by the optimal objective value of a second-stage optimization problem that represents real-time balancing of the power system, which is given by

$$\begin{aligned}
\underset{q,\,\theta,\,s}{\text{minimize}} \quad & \varphi_1(q) && \text{(3.10a)}\\
\text{subject to} \quad & Y(p+q) + Rs - A\theta - Dd = 0 && \text{(3.10b)}\\
& p^{\min} \leq p + q \leq p^{\max} && \text{(3.10c)}\\
& z^{\min} \leq J\theta \leq z^{\max} && \text{(3.10d)}\\
& 0 \leq s \leq r. && \text{(3.10e)}
\end{aligned}$$

Here, φ_1 is a strongly convex separable quadratic function that quantifies the cost of power balancing adjustments, $q \in \mathbb{R}^{n_p}$ are the power adjustments, which for simplicity are treated as changes in planned generator powers, $\theta \in \mathbb{R}^{n_b}$ are node voltage phase angles, n_b is the number of non-slack nodes, $s \in \mathbb{R}^{n_r}$ are powers from renewable sources after possible curtailments, $d \in \mathbb{R}^{n_d}$ are load power consumptions, n_d is the number of loads, $\{Y, R, A, D, J\}$ are sparse matrices, constraint (3.10b) enforces conservation of power using a DC power flow model [108], constraint (3.10c) enforces generator power limits, constraint (3.10d) enforces edge flow limits due to thermal ratings of transmission lines and transformers, and constraint (3.10e) enforces renewable power curtailment limits. Under the assumptions of Section 2.4, which are also made here, the functions $Q(\,\cdot\,, r)$ and $\mathbb{E}\left[Q(\,\cdot\,, r)\right]$ are convex and

differentiable.

Using the definition (3.9) of CVaR and adding bounds for the variable t, the first-stage problem (3.8) can be reformulated as

$$\underset{p,\,t}{\text{minimize}} \quad \mathbb{E}\left[\varphi_0(p) + Q(p,r)\right] \tag{3.11a}$$

$$\text{subject to} \quad \mathbb{E}\left[(1-\gamma)t + \left(Q(p,r) - Q^{\max} - t\right)_+\right] \leq 0 \tag{3.11b}$$

$$(p,t) \in \mathcal{P} \times \mathcal{T}, \tag{3.11c}$$

where $\mathcal{T} := [t^{\min}, t^{\max}]$. In order to avoid affecting the problem, $t^{\max}$ can be set to zero while $t^{\min}$ can be set to a sufficiently large negative number. The resulting problem is clearly of the form of (3.1).

3.6.2 Model of Uncertainty

As already noted, the random vector r represents available powers from renewable energy sources in the near future, say a few hours ahead. It is modeled here by the expression

$$r = \Pi_r\left(r_0 + \delta\right),$$

where r_0 is a vector of base powers, $\delta \sim \mathcal{N}(0,\Sigma)$ are perturbations, Π_r is the projection operator on the set

$$\mathcal{R} := \{\ \bar{r} \mid 0 \leq \bar{r} \leq r^{\max}\ \},$$

and the vector $r^{\max}$ represents the capacities of the renewable energy sources.

The use of Gaussian random variables for modeling renewable energy uncertainty is common in the power systems operations planning literature [64] [74]. To model spatial correlation between nearby renewable sources, a sparse non-diagonal covariance matrix Σ is considered here. The off-diagonal entries of this matrix that correspond to pairs of renewable energy sources that are "nearby" are such that the correlation coefficient between their powers equals a pre-selected value ρ. In the absence of geographical information, a pair of sources are considered as being "nearby" if the network shortest path between the nodes where they are connected is less than or equal to some pre-selected number of edges N. This maximum number of edges is referred here as the correlation distance.

3.6.3 Noisy Functions and Subgradients

By comparing problems (3.1) and (3.11), the functions $F(\,\cdot\,, w)$ and $G(\,\cdot\,, w)$ are given by

$$\begin{aligned} F(x, w) &= \varphi_0(p) + Q(p, r) \\ G(x, w) &= (1 - \gamma)t + (Q(p, r) - Q^{\max} - t)_+ \end{aligned}$$

for all $x = (p, t)$, where $w = r$. Subgradients of these functions are given by the expressions

$$\begin{bmatrix} \nabla\varphi_0(p) + \nabla Q(p, r) \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} 0 \\ 1 - \gamma \end{bmatrix} + 1\{Q(p, r) - Q^{\max} - t \geq 0\} \begin{bmatrix} \nabla Q(p, r) \\ -1 \end{bmatrix},$$

respectively, where $1\{\,\cdot\,\}$ is the indicator function defined by

$$1\{A\} = \begin{cases} 1 & \text{if } A \text{ is true,} \\ 0 & \text{otherwise,} \end{cases}$$

for any event or condition A. As shown in Section 2.4, $\nabla Q(p, r) = -\nabla\varphi_1(q^*)$, where q^* is part of the optimal point of problem (3.10).

3.6.4 Function Approximations

In order to apply the primal-dual SHA algorithm (3.6), initial function approximations $\mathcal{F}_0$ and $\mathcal{G}_0$ need to be defined. $\mathcal{F}_0$ needs to be strongly convex and continuously differentiable, while $\mathcal{G}_0$ needs to be component-wise convex and continuously differentiable. Motivated by the results obtained in Chapter 2 with the AdaCE method, the approximations used here are also based on the CE formulation of problem (3.1), namely, the problem obtained by replacing $\mathbb{E}\,[f(p, r)]$ with $f(p, \bar{r})$, where $\bar{r} := \mathbb{E}\,[r]$, for any stochastic function f. More specifically, $\mathcal{F}_0$ is defined by

$$\mathcal{F}_0(x) := \varphi_0(p) + Q(p, \bar{r}) + \frac{1}{2}\varepsilon t^2, \tag{3.12}$$

for all $x = (p, t)$, where $\varepsilon > 0$, and $\mathcal{G}_0$ is defined by

$$\mathcal{G}_0(x) := (1-\gamma)t + \left(Q(p, \bar{r}) - Q^{\max} - t\right)_+, \tag{3.13}$$

for all $x = (p, t)$. This choice of function $\mathcal{G}_0$ is differentiable everywhere except when $Q(p, \bar{r}) - Q^{\max} - t = 0$. As will be seen in the experiments, this deficiency does not cause problems on the test cases considered. For cases for which this may cause problems, *e.g.*, for cases for which $Q(p, \bar{r}) - Q^{\max} - t = 0$ occurs near or at the solution, the smooth function

$$(1-\gamma)t + \frac{1}{\nu}\log\left(1 + e^{\nu(Q(p,\bar{r}) - Q^{\max} - t)}\right)$$

may be used instead, where ν is a positive parameter.

During the k-th iteration of algorithm (3.6), the function approximations can be expressed as

$$\begin{aligned}\mathcal{F}_k(x) &= \mathcal{F}_0(x) + \zeta_k^T x \\ \mathcal{G}_k(x) &= \mathcal{G}_0(x) + \upsilon_k^T x,\end{aligned}$$

for all $x = (p, t)$, where the vectors ζ_k and υ_k follow the recursive relations

$$\begin{aligned}\zeta_{k+1} &= \zeta_k + \alpha_k\left(\hat{g}_k - g_0(x_k) - \zeta_k\right) \\ \upsilon_{k+1} &= \upsilon_k + \alpha_k\left(\hat{J}_k - J_0(x_k) - \upsilon_k^T\right)^T\end{aligned}$$

with $\zeta_0 = \upsilon_0 = 0$, and $\hat{g}_k$, g_0, $\hat{J}_k$ and J_0 are as defined in Section 3.5.

3.6.5 Subproblems

To perform step (3.6a) of the primal-dual SHA algorithm, the subproblem

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad \mathcal{F}_k(x) + \lambda_k^T \mathcal{G}_k(x)$$

needs to be solved, where $\lambda_k \geq 0$. For problem (3.11) with initial function approximations (3.12) and (3.13), this subproblem is given by

$$\begin{aligned}
\underset{p,\,t,\,z}{\text{minimize}} \quad & \varphi_0(p) + Q(p,\bar{r}) + \frac{1}{2}\varepsilon t^2 + \lambda_k(1-\gamma)t + \lambda_k z + (\zeta_k + \lambda_k \upsilon_k)^T \begin{bmatrix} p \\ t \end{bmatrix} \\
\text{subject to} \quad & Q(p,\bar{r}) - Q^{\max} - t - z \leq 0 \\
& (p,t) \in \mathcal{P} \times \mathcal{T} \\
& 0 \leq z.
\end{aligned}$$

By embedding the definition of $Q(p,\bar{r})$, this subproblem can be reformulated as

$$\begin{aligned}
\underset{p,\,t,\,q,\,\theta,\,s,\,z}{\text{minimize}} \quad & \varphi_0(p) + \varphi_1(q) + \frac{1}{2}\varepsilon t^2 + \lambda_k(1-\gamma)t + \lambda_k z + (\zeta_k + \lambda_k \upsilon_k)^T \begin{bmatrix} p \\ t \end{bmatrix} \\
\text{subject to} \quad & Y(p+q) + Rs - A\theta - Dd = 0 \\
& \varphi_1(q) - Q^{\max} - t - z \leq 0 \\
& p^{\min} \leq p + q \leq p^{\max} \\
& z^{\min} \leq J\theta \leq z^{\max} \\
& p^{\min} \leq p \leq p^{\max} \\
& t^{\min} \leq t \leq t^{\max} \\
& 0 \leq s \leq \bar{r} \\
& 0 \leq z,
\end{aligned} \tag{3.14}$$

which is a sparse QCQP problem.

3.6.6 Implementation

The benchmark primal-dual SA algorithm (3.4) and the new primal-dual SHA algorithm (3.6) were implemented in Python using the scientific computing packages `Numpy` and `Scipy` [38] [97]. The code for these algorithms has been included in the Python package `OPTALG` [1].

The stochastic risk-averse ED problem (3.11) was also constructed using Python and

[1] `http://optalg.readthedocs.io`

included in the Python package `GRIDOPT` [2]. In particular, the C library `PFNET` [3] was used through its Python wrapper for representing power networks and for constructing the required constraints and functions of the problem. For solving the sparse QP problems (3.10) and sparse QCQP problems (3.14), the modeling library `CVXPY` [24] was used together with the second-order cone solver `ECOS` [25]. The SAA-based benchmark algorithms of Section 3.3, namely the one that solves (3.2) directly and algorithm (3.3) that uses decomposition and cutting planes were also implemented using `CVXPY` and `ECOS`.

The covariance matrix Σ of the renewable power perturbations defined in Section 3.6.2 was constructed using routines available in `PFNET`. For obtaining $\Sigma^{1/2}$ to sample $\delta \sim \mathcal{N}(0, \Sigma)$, the sparse Cholesky code `CHOLMOD` was used via its Python wrapper [4] [19].

3.6.7 Numerical Experiments

The stochastic optimization algorithms described in Sections 3.3, 3.4 and 3.5 were applied to two instances of the stochastic risk-averse ED problem (3.11) in order to test and compare their performance. The test cases used were constructed from real power networks from North America (Canada) and Europe (Poland). Table 3.1 shows important information about each of the cases, including name, number of nodes in the network, number of edges, number of first-stage variables (dimension of (p, t) in (3.11)), number of second-stage variables (dimension of (q, θ, s) in (3.10)), and dimension of the uncertainty (dimension of the vector r in (3.11)).

Table 3.1: Information about test cases

name	nodes	edges	dim (p, t)	dim (q, θ, s)	dim r
Case A	2454	2781	212	2900	236
Case B	3012	3572	380	3688	298

The separable generation cost function φ_0 in (3.11) was constructed using uniformly random coefficients in $[0.01, 0.05]$ for the quadratic terms, and uniformly random coefficients in $[10, 50]$ for the linear terms. These coefficients assume that power quantities are in units of MW and are consistent with those found in several `MATPOWER` cases [109]. For the separable generation adjustment cost function φ_1 in (3.10), the coefficients for the quadratic terms

[2] `http://gridopt.readthedocs.io`
[3] `http://pfnet.readthedocs.io`
[4] `http://pythonhosted.org/scikits.sparse`

were set to equal those of φ_0 scaled by a factor greater than one, and the coefficients of the linear terms were set to zero. The scaling factor was set large enough to make balancing costs with adjustments higher than with planned powers. The coefficients for the linear terms were set to zero to penalize both positive and negative power adjustments equally. For the t-regularization added in (3.12) to make $\mathcal{F}_0$ strongly convex, $\varepsilon = 10^{-6}$ was used.

The renewable energy sources used to construct the test cases of Table 3.1 were added manually to each of the power networks at the nodes with adjustable or fixed generators. The capacity $r_i^{\max}$ of each source was set to $1^T d/n_r$, where d is the vector of load power consumptions and n_r is the number of renewable energy sources. The base powers were set using $r_0 = 0.5r^{\max}$ so that the base renewable energy penetration was 50% of the load, which is a high penetration scenario. The standard deviations of the renewable power perturbations, *i.e.*, $\operatorname{diag}\Sigma^{1/2}$, were also set to $0.5r^{\max}$, which corresponds to a high-variability scenario. For the off-diagonals of Σ, a correlation coefficient ρ of 0.05 and a correlation distance N of 5 edges were used.

For the risk-limiting constraint (3.11b), the desirable balancing cost limit $Q^{\max}$ was set to $0.8Q_0$, where $Q_0 := \mathbb{E}\,[Q(p_0, r)]$ and p_0 is the optimal power generation policy for the CE problem without risk-limiting constraint, *i.e.*,

$$p_0 := \operatorname*{arg\,min}_{p \in \mathcal{P}} \ \varphi_0(p) + Q(p, \mathbb{E}\,[r]).$$

The CVaR parameter γ defined in Section 3.6.1, which is related to the probability of satisfying the inequality $Q(p, r) \leq Q^{\max}$, was set to 0.95. For bounding the variable t in problem (3.11), $t^{\max}$ and $t^{\min}$ were set to 0 and $-0.1Q_0$, respectively. For choosing this value of $t^{\min}$, some experimentation was required. In particular, executing algorithms (3.4) and (3.6) for a few iterations was needed in order to check that $t^{\min}$ was sufficiently negative to allow for feasibility, but not too large in magnitude. The latter was needed to avoid encountering excessively large values of $G(x_k, w_k)$ (and hence large increases in λ_k) due to poor values of t in the initial iterations.

Each of the algorithms was applied to the test cases shown in Table 3.1 and its performance was recorded. In particular, the SAA cutting-plane algorithm (3.3) with 1000 scenarios (`SAA CP 1000`), the primal-dual SA algorithm (3.4) (`PD SA`), and the primal-dual SHA algorithm (3.6) considering and ignoring the risk-limiting constraint (`PD SHA` and `SHA`, respectively) were executed for a fixed time period. For the `PD SA` algorithm, $(p_0, t^{\min})$ was

used as the initial primal solution estimate. For both `PD SHA` and `PD SA`, 0 was used as the initial dual solution estimate λ_0, and the step lengths α_k used were of the form $(k_0 + k)^{-1}$, where k_0 is a constant. A k_0 value of 50 was used for the `PD SHA` algorithm, while a k_0 value of 350 was used for the `PD SA` algorithm. These values were chosen to reduce the susceptibility of the algorithms to noise during the initial iterations. Only for the `SAA CP 1000` algorithm, 24 parallel processes were used. This was done to compute values and subgradients of the sample-average second-stage cost function efficiently. For evaluating the progress of the algorithms as a function of time, the quantities $\mathbb{E}[F(x, w)]$, $\mathbb{E}[G(x, w)]$ and $\text{Prob}(Q(p, r) \leq Q^{\max})$ were evaluated using 2000 fixed scenarios at fixed time intervals. Figures 3.5 and 3.6 show the results obtained on a computer cluster at the Swiss Federal Institute of Technology (ETH) with computers equipped with Intel®Xeon®E5-2697 CPU (2.70 GHz), and running the operating system CentOS 6.8. In the figures, the objective values shown were normalized by $\varphi_0(p_0) + Q_0$, while the constraint function values shown were normalized by Q_0, which was defined in the previous paragraph. The dashed lines are meant to represent references, and correspond to the results obtained by solving the CE problem ignoring the risk-limiting constraint (`CE`), and the SAA-based algorithm that consists of solving problem (3.2) directly (without decomposition) using 1000 scenarios (`SAA 1000`). The times needed to compute these references are show in Table 3.2. Table 3.3 shows the (maximum) memory requirements of the algorithms on each of the test cases.

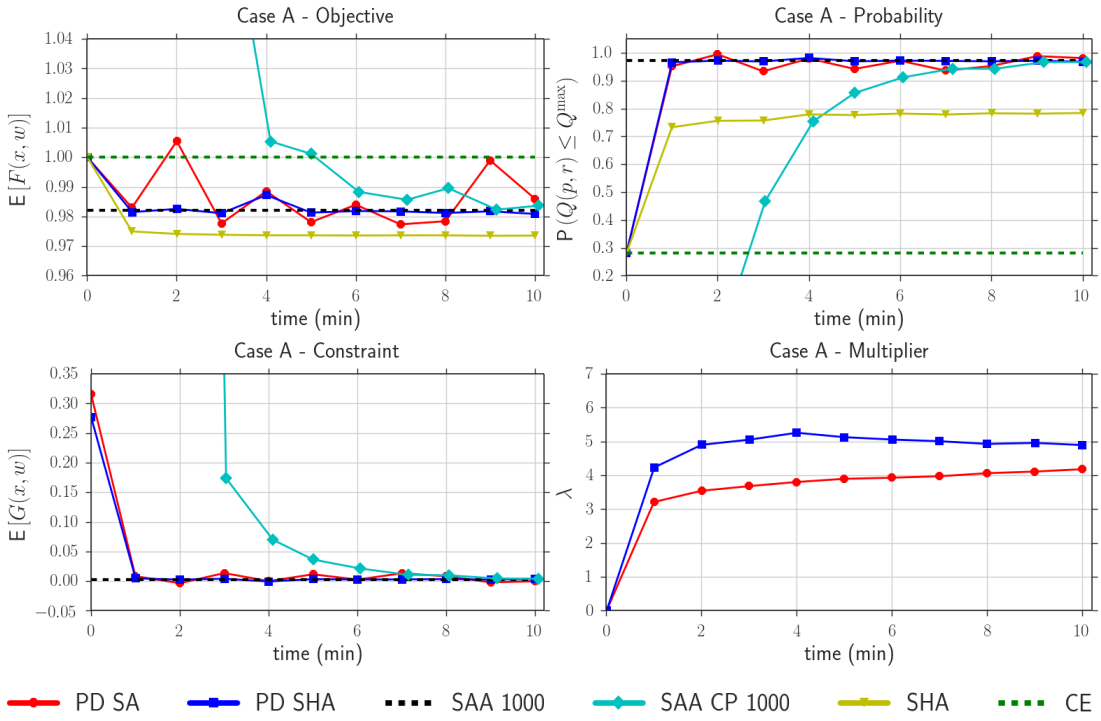

Figure 3.5: Performance of algorithms on Case A

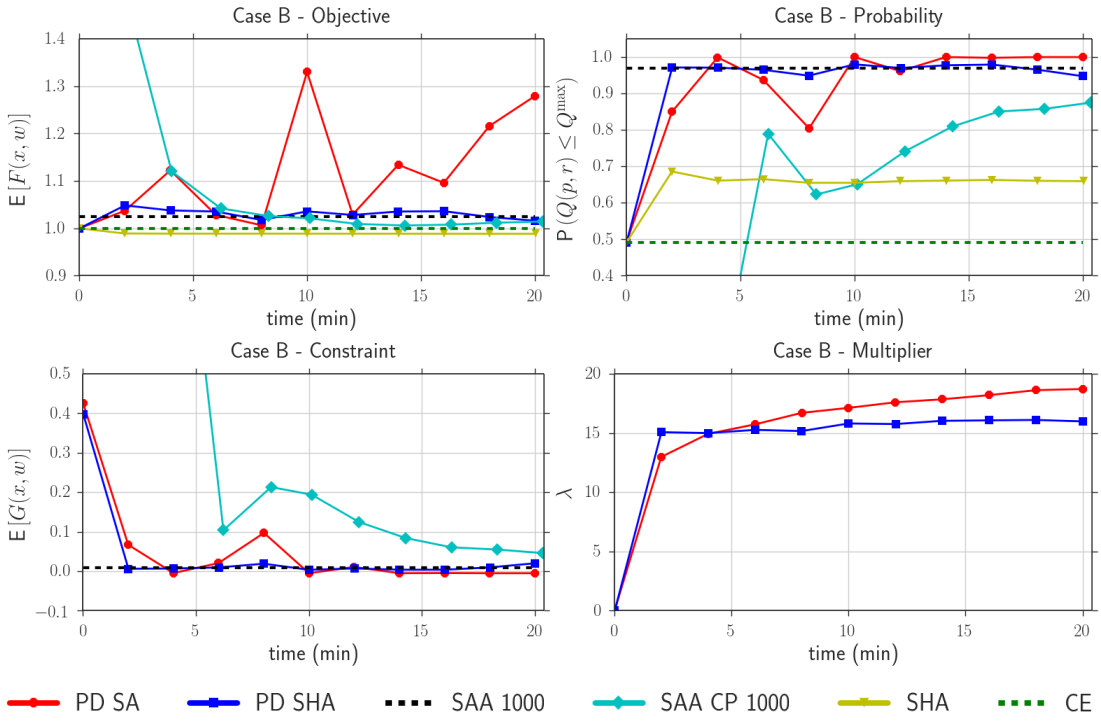

Figure 3.6: Performance of algorithms on Case B

Table 3.2: Computation times of references in minutes

algorithm	Case A	Case B
`CE`	0.0075	0.018
`SAA 1000`	15	92

Table 3.3: Memory requirements of algorithms in gigabytes

algorithm	Case A	Case B
`CE`	0.14	0.16
`SAA 1000`	8.4	13
`SAA CP 1000`	1.5	1.8
`PD SA`	0.12	0.20
`PD SHA`	0.14	0.16

From the plots shown in Figures 3.5 and 3.6, several important observations can be made: First, the `CE` solution can result in both poor expected cost and high risk of violating

$Q(p,r) \leq Q^{\max}$. On Case A, for example, all algorithms find power generation policies that result in lower expected cost and in $Q(p,r) \leq Q^{\max}$ being satisfied with significantly higher probability. Second, by ignoring the risk-limiting constraint, the `SHA` algorithm converges fast, *i.e.*, in a few minutes, and finds a power generation policy that results in the lowest expected cost. The probability of satisfying $Q(p,r) \leq Q^{\max}$, however, is only around 0.7 and 0.8 for both cases, and hence the policy found is not enough risk-averse. Third, by considering the risk-limiting constraint, the `PD SHA` and `PD SA` algorithms produce in only a few minutes power generation policies that are risk-averse, *i.e.*, $\text{Prob}(Q(p,r) \leq Q^{\max}) \geq \gamma$. This is, of course, at the expense of higher expected costs compared with those obtained with the `SHA` algorithm. The iterates produced by the `PD SA` algorithm experience significant variations and do not appear to settle during the execution period. This is particularly visible in the plot of expected cost for both test cases, and in the plot of probability for Case B. The `PD SHA` algorithm, on the other hand, produces iterates that experience much less variations (despite using larger step lengths) and converge during the execution period. The expected cost and risk associated with the policy found by the `PD SHA` algorithm match those obtained with the reference policy found with the `SAA 1000` algorithm, which requires more time and much more computer memory, as seen on Tables 3.2 and 3.3. Lastly, the other SAA-based algorithm, namely `SAA CP 1000`, also produces policies whose expected cost and risk approach those obtained using the `SAA 1000` algorithm. However, it requires more computer memory and, despite using 24 parallel processes, more time compared to the `PD SHA` algorithm.

The power generation policies associated with the last iterates produced by the primal-dual stochastic hybrid approximation algorithm considering and ignoring the risk-limiting constraint (`PD SHA` and `SHA`, respectively) were compared. The results are shown in Figure 3.7. The plots show the (sorted) power output differences in MW of the generators with respect to p_0, which is the solution of the CE problem without risk-limiting constraint (`CE`). As the plots show, the power generation policy obtained when ignoring the risk-limiting constraint shows a consistent negative bias on both test cases with respect to the risk-averse policy. This negative bias corresponds to lower planned generator powers, which is expected. The reason is that higher planned generator powers help reduce the risk of requiring generator output increases in real time when renewable power injections are low. On the other hand, when renewable power injections are high, renewable power curtailments can be used as a means for balancing instead of generator output adjustments.

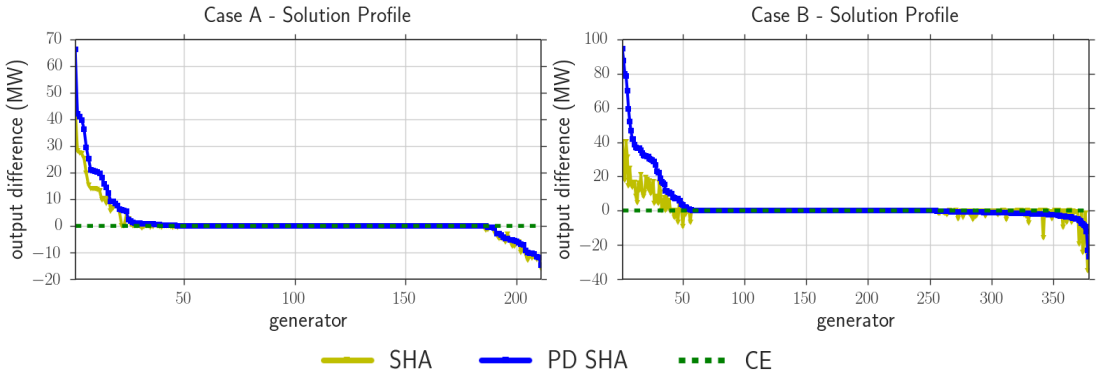

Figure 3.7: Power generation policies found by algorithms

3.7 Conclusions

In this chapter, a new algorithm for solving convex stochastic optimization problems with expectation functions in both the objective and constraints has been described and evaluated. The algorithm combines a stochastic hybrid procedure, which was originally designed to solve problems with expectation only in the objective, with dual stochastic gradient ascent. More specifically, the algorithm generates primal iterates by minimizing deterministic approximations of the Lagrangian that are updated using noisy subgradients, and dual iterates by applying stochastic gradient ascent to the true Lagrangian. A proof that the sequence of primal iterates produced by this algorithm has a subsequence that converges almost surely to an optimal point under certain conditions has been provided in the Appendix. The proof relies on a "bounded drift" assumption for which we have provided an intuitive justification for why it may hold in practice as well as examples that illustrate its validity using different initial function approximations. Furthermore, experimental results obtained from applying the new algorithm with CE-based initial function approximations to instances of a stochastic risk-averse ED problem that originates in power systems operations planning under high penetration of renewable energy have been reported. In particular, the performance of the new AdaCE algorithm has been compared with that of a primal-dual SA algorithm and algorithms based on SAA with and without decomposition. The results obtained showed that the new algorithm had superior performance compared to the benchmark algorithms on the test cases considered. In particular, the new algorithm was able

to leverage the accuracy of the CE-based initial function approximations and produce iterates that approached a solution fast without requiring large computer memory or parallel computing resources.

Chapter 4

A Parameterized Extension for Multiple Stages

4.1 Background and Overview

Many important applications, such as financial planning, supply chain management, and operations planning of modern low-carbon power systems, can be modeled as sequential decision-making processes under uncertainty [82]. A typical goal in these applications is to determine a decision policy that optimizes a certain performance measure, such as total expected cost. However, obtaining such a policy is in general an intractable task. The reason for this is that the decisions at each point in time can depend on all the decisions and observations made so far, and are subject to any remaining uncertainty present in the future.

Multi-stage stochastic optimization has been an invaluable tool for obtaining decision policies for this type of applications. By far the most common approach has been to represent or approximate the exogenous uncertainty with a finite number of scenarios, and use cutting planes to construct gradually-improving deterministic approximations of expected cost-to-go functions. For example, the algorithm proposed by Birge for solving problems with linear functions, which can be considered a multi-stage extension of Benders decomposition and the L-shaped method [4] [98], performs forward passes through a scenario tree to construct candidate solutions, and backward passes to construct cutting planes and update piece-wise linear cost-to-go function approximations [9]. An issue with approaches such as this one is that the number of scenarios can grow exponentially with the length of the

decision-making horizon, and hence processing all of them at each iteration is in general computationally impractical. Pereira & Pinto developed the SDDP algorithm for problems with stage-wise independent uncertainty, which works with a relatively small number of scenarios that are randomly sampled at each iteration [63] [85] [86]. This algorithm has become a popular tool used by practitioners, in particular for the optimization of hydro power systems [2] [71] [86] [99], and has also been analyzed and extended by several authors in the literature. For example, Philpott & Guan provided a rigorous analysis of the convergence of a broad class of sampling-based algorithms that includes SDDP, and discussed specific requirements for the sampling procedure [65]. Rebennack adapted SDDP to work with stage-wise dependent uncertainty, and showed the applicability of the algorithm on a hydro-thermal scheduling problem with various types of uncertainty [71]. Asamov & Powell, on the other hand, showed how to improve the convergence speed of SDDP by including quadratic regularization in a way that does not require obtaining exponentially-many incumbent solutions [2]. Another sampling-based decomposition algorithm related to SDDP is the multi-stage stochastic decomposition algorithm proposed by Sen & Zhou [83]. This algorithm, which is a dynamic extension of the sampled regularized algorithm proposed by Higle & Sen for two-stage problems [35], constructs statistical estimates of supports of cost-to-go functions that improve over time [34]. Lastly, a different family of algorithms explored in the literature for solving convex multi-stage stochastic optimization problems are algorithms based on the principle of progressive hedging [78]. These algorithms model the non-anticipativity of policies using linear constraints that are relaxed through the use of an approximate augmented Lagrangian function. Rockafellar & Wets were the first to describe an algorithm based on this idea [78]. In their algorithm, non-anticipative policies are obtained at each iteration through a projection operator. Mulvey & Ruszczyński proposed an algorithm based on similar ideas for solving large-scale problems exploiting parallel computing resources [51].

Most of the algorithms proposed in the literature for solving convex multi-stage stochastic optimization problems have specifically targeted problems having linear functions. For such problems, expected cost-to-go functions can be shown to be piece-wise linear and hence algorithms that build piece-wise linear approximations such as SDDP are a natural choice. Only a relatively few number of authors have studied the applicability of these algorithms to problems having nonlinear functions, or extended them to solve such problems more efficiently. For example, Girardeau, Leclere & Philpott studied the convergence of

sampling-based nested decomposition algorithms that include SDDP on problems having general convex objective functions [31]. They showed that these algorithms converge on such problems, but did not perform numerical experiments. From [55], it is known that the performance of cutting-plane algorithms can be very slow on instances of problems with general convex objectives, and hence these algorithms may not be the best choice for problems beyond ones having only linear functions. Another example is the work of Louveaux, which consists of a nested decomposition algorithm for problems with quadratic objectives and linear constraints [47]. The algorithm approximates cost-to-go functions by constructing and updating pieces of polyhedral piece-wise quadratic functions. However, the algorithm does not consider sampling and hence suffers from the same computational limitations as other early nested decomposition algorithms that require traversing the complete scenario tree. Lau & Womersley also proposed an algorithm for solving problems with quadratic objectives [43]. The algorithm uses a recursive generalized Newton procedure, and requires recursively solving entire subtrees of the scenario tree.

In this work, a new sampling-based algorithm is proposed for solving convex multi-stage stochastic optimization problems having nonlinear functions for which cutting-plane methods such as SDDP may not be the most efficient. The algorithm utilizes user-provided initial approximations of expected cost-to-go functions that are gradually improved using an SHA procedure. Hence, it can leverage application-specific knowledge and start with approximations that with cutting planes alone could be potentially time-consuming to obtain. The SHA procedure on which this algorithm is based consists of using noisy subgradients that are computed at each iteration to make slope corrections to the expected cost-to-go function approximations. This procedure was proposed by Cheung & Powell for solving two-stage stochastic optimization problems, and was motivated by a dynamic resource allocation problem [21]. In Chapter 2, this procedure was tested on a two-stage stochastic ED problem using an initial approximation based on the CE problem, resulting in the AdaCE method, and promising performance was observed. In Chapter 3, this procedure was extended to handle expected-value constraints and tested on a two-stage stochastic risk-averse ED problem. To use this procedure in the multi-stage setting, the algorithm proposed in this work parameterizes the step lengths, slope corrections, and the expected cost-to-go function approximations by the observation history of the exogenous uncertainty. Additionally, the slope corrections obtained for a function associated with a specific realization

of the uncertainty are generalized and applied to other functions associated with similar realizations of uncertainty using radial basis functions [1] [94]. This allows the algorithm to be applied directly to problems with continuous uncertainty without the need of discretization or scenario tree construction. An analysis of the convergence of the algorithm is provided for the case of finite-support uncertainty since the presence of nested expectations makes the continuous case theoretically intractable. Lastly, the performance of the algorithm is compared against that of three benchmark algorithms on several instances of a multi-stage stochastic ED problem from low-carbon power systems with high penetration of renewable energy. The benchmark algorithms consist of a greedy, a CE-based, and an SDDP algorithm. The results obtained suggest that the proposed algorithm can leverage accurate initial expected cost-to-go function approximations provided by the user and obtain better policies compared to the benchmark algorithms and in shorter times.

4.2 Convex Multi-Stage Stochastic Optimization Problems

The general problem considered in this chapter consists of the following nested optimization problems:

$$\begin{aligned} \underset{x_t}{\text{minimize}} \quad & F_t(x_t, w_t) + G_t(x_t, \mathcal{W}_t) \\ \text{subject to} \quad & x_t \in \mathcal{X}_t(x_{t-1}, w_t), \end{aligned} \tag{4.1}$$

for $t \in \{1, \ldots, T\}$, where $T \in \mathbb{N}$, $F_t(\,\cdot\,, w_t)$ are continuous convex functions, $\mathcal{X}_t(\,\cdot\,, w_t)$ are convex[1] point-to-set mappings with convex compact outputs, $w_t \in \Omega_t$ is an exogenous random vector observed at time t with w_1 being deterministic, Ω_t are compact sets, $\mathcal{W}_t := \{w_1, \ldots, w_t\}$ is the observation history up to time t, and x_0 is a constant vector. The cost-to-go functions $G_t(\,\cdot\,, \mathcal{W}_t)$ are defined by

$$G_t(x, \mathcal{W}_t) := \begin{cases} \mathbb{E}\,[\, H_{t+1}(x, \mathcal{W}_{t+1}) \,|\, \mathcal{W}_t \,] & \text{if } t \in \{1, \ldots, T-1\}, \\ 0 & \text{else,} \end{cases} \tag{4.2}$$

for all x, where the function H_t is such that $H_t(x, \mathcal{W}_t)$ is the optimal objective value of problem (4.1) with $x_{t-1} = x$, and $\mathbb{E}\,[\,\cdot\,|\,\mathcal{W}_t\,]$ is expectation with respect to $\mathcal{W}_{t+1}$ conditioned

[1] For all inputs x and y, and any $\lambda \in [0, 1]$, $\lambda\mathcal{X}_t(x, w_t) + (1-\lambda)\mathcal{X}_t(y, w_t) \subset \mathcal{X}_t(\lambda x + (1-\lambda)y, w_t)$.

on $\mathcal{W}_t$, which is assumed to be always well defined.

It is assumed that for all $t \in \{1, \ldots, T-1\}$ and $\mathcal{W}_{t+1}$, there exists an $\epsilon > 0$ such that for all $\{x_1, \ldots, x_t\}$ that satisfy $x_\tau \in \mathcal{X}_\tau(x_{\tau-1}, w_\tau)$ for each $\tau \in \{1, \ldots, t\}$, and for all δ that satisfy $\|\delta\|_2 < \epsilon$, it holds that $\mathcal{X}_{t+1}(x_t + \delta, w_{t+1}) \neq \emptyset$. This property is slightly stronger than *relatively complete recourse*, which guarantees the feasibility of problem (4.1) [2] [26]. In fact, it is similar to the *extended relatively complete recourse* condition introduced in [31].

From the assumed properties of $F_t(\,\cdot\,, w_t)$ and $\mathcal{X}_t(\,\cdot\,, w_t)$, it can be shown that $H_t(\,\cdot\,, \mathcal{W}_t)$ and $G_t(\,\cdot\,, \mathcal{W}_t)$ are real-valued convex continuous functions on the convex sets of stage-$(t-1)$ and stage-t feasible inputs, respectively, for each $t \in \{1, \ldots, T\}$.

4.3 SDDP Algorithm

4.3.1 Background and Overview

The SDDP algorithm is a sampling algorithm based on nested Benders decomposition. It works with a finite number of scenarios that represent the exogenous random process. Hence, for problems with continuous uncertainty, it solves and approximation of the problem. Since this algorithm is widely used in practice, especially on linear multi-stage problems that arise in the scheduling of hydro-thermal electricity systems [2] [71] [86] [99], it is used here as a benchmark to assess the performance of the proposed algorithm described in the next Section.

The first version of SDDP proposed by Pereira & Pinto assumed stage-wise independent uncertainty, which implies that expected cost-to-go functions do not depend on the observation history of the exogenous random process [63]. This property is exploited by the algorithm by accumulating information obtained at each stage, *e.g.*, cutting planes, to build a single expected cost-to-go function approximation for that stage. The problem described in Section 4.2 does not assume stage-wise independent uncertainty. Hence, the version of the SDDP algorithm considered here treats expected cost-to-go functions as "path-dependent", where path here refers to a realization of the exogenous random process, or equivalently, to a node in a scenario tree. This case is also considered by Girardeau, Leclere & Philpott [31], and Rebennack [71].

In the following, the notation used to represent the scenario tree is introduced, and an SAA of problem (4.1) based on this scenario tree is described. Then, the SDDP algorithm is presented followed by a list of references where its convergence is analyzed.

4.3.2 SAA Problem

For problems with continuous uncertainty, a commonly-used approach for constructing a scenario tree is conditional Monte Carlo sampling [36] [86]. The resulting scenario tree can be represented as follows: For each $t \in \{1, \ldots, T\}$, $\mathcal{N}(t) \subset \Omega_t$ denotes the set of nodes associated with stage t. The set $\mathcal{C}(w_t) \subset \mathcal{N}(t+1)$ denotes the children of node $w_t \in \mathcal{N}(t)$. For $w_T \in \mathcal{N}(T)$, $|\mathcal{C}(w_T)| = 0$, *i.e.*, nodes of stage T are leaf nodes, and $w_1 \in \mathcal{N}(1) = \{w_1\}$ is the root or deterministic node of the tree. An arbitrary tree branch of length t is denoted by $\mathcal{W}_t := \{w_1, \ldots, w_t\}$, where $w_{\tau+1} \in \mathcal{C}(w_\tau)$ for all τ such that $1 \leq \tau < t$. The set of all t-length branches is denoted by $\mathcal{B}(t)$.

Using this notation for representing the scenario tree, the SAA of problem (4.1) then consists of the following nested optimization problems:

$$\begin{aligned} \underset{x_t}{\text{minimize}} \quad & F_t(x_t, w_t) + \widetilde{G}_t(x_t, \mathcal{W}_t) \\ \text{subject to} \quad & x_t \in \mathcal{X}_t(x_{t-1}, w_t), \end{aligned} \tag{4.3}$$

for $t \in \{1, \ldots, T\}$ and $\mathcal{W}_t \in \mathcal{B}(t)$. The cost-to-go functions $\widetilde{G}_t(\,\cdot\,, \mathcal{W}_t)$ are defined by

$$\widetilde{G}_t(x, \mathcal{W}_t) := \begin{cases} |\mathcal{C}(w_t)|^{-1} \sum_{w \in \mathcal{C}(w_t)} \widetilde{H}_{t+1}(x, \mathcal{W}_t \cup \{w\}) & \text{if } t \in \{1, \ldots, T-1\}, \\ 0 & \text{else,} \end{cases} \tag{4.4}$$

for all x, where the function $\widetilde{H}_t$ is such that $\widetilde{H}_t(x, \mathcal{W}_t)$ is the optimal objective value of problem (4.3) with $x_{t-1} = x$.

4.3.3 Algorithm

At each iteration $k \in \mathbb{Z}_+$, the SDDP algorithm considered here performs the following steps: First, a tree branch $\mathcal{W}_t^k = \left\{w_1^k, \ldots, w_t^k\right\} \in \mathcal{B}(t)$ is selected uniformly at random for $t = T$, independently from samples drawn in previous iterations. Then, the problem

$$\begin{aligned} \underset{x_t}{\text{minimize}} \quad & F_t(x_t, w_t) + \widetilde{G}_t^k(x_t, \mathcal{W}_t) \\ \text{subject to} \quad & x_t \in \mathcal{X}_t(x_{t-1}, w_t) \end{aligned} \tag{4.5}$$

is solved *forward in time* for each $t = 1, \ldots, T$ with $x_{t-1} = x_{t-1}^k$ and $\mathcal{W}_t = \mathcal{W}_t^k$ to obtain a solution x_t^k. The vector $x_0^k := x_0$ is a constant and the function $\widetilde{G}_t^k(\,\cdot\,, \mathcal{W}_t)$ is a piece-wise

linear approximation of the expected cost-to-go function $\widetilde{G}_t(\,\cdot\,, \mathcal{W}_t)$ defined in (4.4) such that $\widetilde{G}_t^0 := 0$ for all t and $\widetilde{G}_T^k := 0$ for all k. After reaching the end of the branch, the branch is processed *backward in time*. More specifically, for each $t = T-1, \ldots, 1$, a subgradient

$$\nu_{t+1}(w) \in \partial \widetilde{H}_{t+1}^{k+1}\big(x_t^k, \mathcal{W}_t^k \cup \{w\}\big)$$

is computed for each $w \in \mathcal{C}(w_t^k)$, where the function $\widetilde{H}_t^k$ is such that $\widetilde{H}_t^k(x, \mathcal{W}_t)$ is the optimal objective value of problem (4.5) with $x_{t-1} = x$. The computed subgradients $\big\{\, \nu_{t+1}(w) \;\big|\; w \in \mathcal{C}(w_t^k) \,\big\}$, are used to construct a new function approximation of $\widetilde{G}_t\big(\,\cdot\,, \mathcal{W}_t^k\big)$ as follows:

$$\widetilde{G}_t^{k+1}\big(x, \mathcal{W}_t^k\big) = \begin{cases} h_t^k\big(x, \mathcal{W}_t^k\big) & \text{if } k = 0, \\ \max\Big\{\widetilde{G}_t^k\big(x, \mathcal{W}_t^k\big), h_t^k\big(x, \mathcal{W}_t^k\big)\Big\} & \text{else,} \end{cases}$$

for all x, where

$$h_t^k\big(x, \mathcal{W}_t^k\big) := \big|\mathcal{C}(w_t^k)\big|^{-1} \sum_{w \in \mathcal{C}(w_t^k)} \Big(\widetilde{H}_{t+1}^{k+1}\big(x_t^k, \mathcal{W}_t^k \cup \{w\}\big) + \nu_{t+1}(w)^T\big(x - x_t^k\big)\Big).$$

For all t-length branches $\mathcal{W}_t \neq \mathcal{W}_t^k$, the cost-to-go function approximations are kept the same, *i.e.*, $\widetilde{G}_t^{k+1}(\,\cdot\,, \mathcal{W}_t) = \widetilde{G}_t^k(\,\cdot\,, \mathcal{W}_t)$.

An important property of this algorithm is that the function $\widetilde{G}_t^k(\,\cdot\,, \mathcal{W}_t)$ is an underestimator of the expected cost-to-go function $\widetilde{G}_t(\,\cdot\,, \mathcal{W}_t)$ for each t, $\mathcal{W}_t$ and k. Hence, $F_1\big(x_1^k, \mathcal{W}_1\big) + \widetilde{G}_1^k\big(x_1^k, \mathcal{W}_1\big)$ is a lower bound of the optimal value of the nested problem (4.3) for stage $t = 1$. To obtain upper bounds, (4.5) can be solved forward for each node of the tree. These bounds can in theory be used to stop the algorithm when a desired accuracy has been reached. However, computing exact upper bounds is typically impractical since a scenario tree can have a very large number of nodes. A more tractable option is to use statistical estimates of upper bounds by solving (4.5) only for a relatively small number of sampled branches of the tree [86]. Nevertheless, as stated in [2], it is more common in practice to run the algorithm for a fixed number of iterations due to the difficulty of estimating appropriate convergence tolerances a priori.

4.3.4 Convergence

For the case of problems with linear functions and stage-wise independent uncertainty, Chen & Powell and Philpott & Guan provide convergence proofs for SDDP and related algorithms [20] [65]. For the more general case of problems having convex objective functions and potentially stage-wise dependent uncertainty, this is done by Girardeau, Leclere & Philpott [31]. The interested reader is referred to these works for more details.

4.4 Parameterized SHA Algorithm

4.4.1 Background and Overview

In [21], Cheung & Powell proposed an SHA procedure for solving two-stage stochastic optimization problems of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & F(x) + \mathbb{E}\left[H(x, w)\right] \\ \text{subject to} \quad & x \in \mathcal{X}, \end{aligned}$$

where $F + H(\,\cdot\,, w)$ are convex functions, $\mathcal{X}$ is a convex compact set, and $\mathbb{E}\left[\,\cdot\,\right]$ denotes expectation with respect to a random vector w. The procedure consists of generating iterates

$$x^k := \underset{x \in \mathcal{X}}{\arg\min}\, F(x) + \mathcal{H}^k(x),$$

where $\mathcal{H}^k$ are approximations of $\mathbb{E}\left[H(\,\cdot\,, w)\right]$ such that $F + \mathcal{H}^k$ are strongly convex and differentiable. The first approximation $\mathcal{H}^0$ is provided by the user, and the rest are obtained by performing slope corrections using noisy subgradients of $\mathbb{E}\left[H(\,\cdot\,, w)\right]$ as follows:

$$\mathcal{H}^{k+1}(x) = \mathcal{H}^k(x) + \alpha_k\big(\xi^k - \nabla\mathcal{H}^k\big(x^k\big)\big)^T x, \tag{4.6}$$

for all x, where $\xi^k \in \partial H(x^k, w^k)$, w^k are independent samples of w, and α_k are step lengths that satisfy conditions that are common in stochastic approximation algorithms [8] [42]. The strengths of this approach are its ability to exploit an accurate initial function approximation $\mathcal{H}^0$, and the fact that the slope corrections do not change the structure of this approximation.

In Chapter 2, this SHA procedure was applied to a two-stage stochastic ED problem.

Motivated by the fact that in power system operations planning CE models are common, an initial function approximation based on such a model was considered, *i.e.*, $\mathcal{H}^0 = H(\cdot, \mathbb{E}[w])$, resulting in the AdaCE method. The results obtained with this method were promising, since they were better than those obtained with a stochastic subgradient algorithm and two cutting-plane SAA-based algorithms. Motivated by these results, the SHA procedure was combined with dual stochastic gradient ascent in Chapter 3 in order to solve problems of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & F(x) + \mathbb{E}[H(x, w)] \\ \text{subject to} \quad & \mathbb{E}[G(x, w)] \leq 0 \\ & x \in \mathcal{X}, \end{aligned}$$

where $G(\cdot, w)$ are vector-valued functions composed of convex functions. The proposed algorithm was tested on a risk-averse version of the two-stage stochastic ED problem considered in Chapter 2 using initial approximations of $\mathbb{E}[H(x, w)]$ and $\mathbb{E}[G(x, w)]$ that were also based on the CE problem. The results obtained were again positive, and this suggests that at least for the application of generation dispatch under uncertainty, the SHA procedure with initial approximations based on CE models is an effective combination.

Motivated by the results obtained in previous chapters, the SHA procedure is extended here to solve convex multi-stage stochastic optimization problems of the form of (4.1). Unlike in the two-stage case, continuous uncertainty makes this problem intractable due to the presence of at least one level of nested expectations, which results in an uncountably infinite number of expected cost-to-go functions. For this reason, the aim here is to design an algorithm that can be applied directly to problems with continuous uncertainty, and that it is guaranteed to work on the theoretically-tractable case of finite-support uncertainty under some conditions that are likely to hold in practice.

The main idea of the proposed algorithm is to parameterize the SHA procedure by the exogenous random process. More specifically, the step lengths α_k, slope corrections, and cost-to-go function approximations are treated as functions of the exogenous random process, with their values being dependent on how close a given realization of the exogenous random process is to each of the realizations drawn so far during previous iterations. Radial basis functions, which are widely used in machine learning [1] [94], are used to control how slope corrections are shared or generalized to similar realizations of uncertainty.

In the following, a detailed description of the algorithm is provided followed by a discussion of its memory requirements and convergence.

4.4.2 Algorithm

At each iteration $k \in \mathbb{Z}_+$, the proposed algorithm performs the following steps: First, a realization $\mathcal{W}_t^k := \{w_1^k, \ldots, w_t^k\}$ of the exogenous random process is sampled for $t = T$. Then, the problem

$$\begin{aligned} \underset{x_t}{\text{minimize}} \quad & F_t(x_t, w_t) + \widehat{G}_t(x_t, \mathcal{W}_t) + g_t^k(\mathcal{W}_t)^T x_t \\ \text{subject to} \quad & x_t \in \mathcal{X}_t(x_{t-1}, w_t) \end{aligned} \tag{4.7}$$

is solved *forward in time* for each $t = 1, \ldots, T$ with $x_{t-1} = x_{t-1}^k$ and $\mathcal{W}_t = \mathcal{W}_t^k$ to obtain a solution x_t^k. The vector $x_0^k := x_0$ is a constant and g_t^k is a slope-correction function such that $g_t^0 := 0$ for all t and $g_T^k := 0$ for all k. The function $\widehat{G}_t(\,\cdot\,, \mathcal{W}_t)$ is a user-provided convex differentiable approximation of the expected cost-to-go function $G_t(\,\cdot\,, \mathcal{W}_t)$ defined in (4.2) such that $\widehat{G}_T := 0$. For each $t \neq T$, after obtaining x_t^k, the vectors

$$\xi_t^k \in \partial H_{t+1}^k\big(x_t^k, \mathcal{W}_{t+1}^k\big) \tag{4.8}$$

$$\eta_t^k = \nabla \widehat{G}_t\big(x_t^k, \mathcal{W}_t^k\big) \tag{4.9}$$

are computed, were H_t^k is the function such that $H_t^k(x, \mathcal{W}_t)$ is the optimal objective value of problem (4.7) with $x_{t-1} = x$. These vectors are then used to update each of the slope-correction functions g_t^k as follows:

$$g_t^{k+1}(\mathcal{W}_t) = g_t^k(\mathcal{W}_t) + \alpha_t^k(\mathcal{W}_t)\left(\xi_t^k - \eta_t^k - g_t^k\big(\mathcal{W}_t^k\big)\right), \tag{4.10}$$

for all $t \in \{1, \ldots, T-1\}$ and $\mathcal{W}_t$. The step length functions α_t^k are defined by

$$\alpha_t^k(\mathcal{W}_t) := \phi_t^k(\mathcal{W}_t)\beta_k \tag{4.11}$$

for all $t \in \{1, \ldots, T-1\}$, $\mathcal{W}_t$, and $k \in \mathbb{Z}_+$, where $\beta_k \in (0, 1)$, ϕ_t^k is the *radial basis function* [94] defined by

$$\phi_t^k(\mathcal{W}_t) := e^{-\gamma_t \|\mathcal{W}_t - \mathcal{W}_t^k\|_2}, \tag{4.12}$$

and $\gamma_t \in \mathbb{R}_{++}$ is an algorithm parameter. The role of this radial basis function is to weigh how much a slope correction obtained from a specific realization of the uncertainty, *e.g.*, $\mathcal{W}_t^k$, should count towards the slope correction applied to a function associated with a path $\mathcal{W}_t$ of the exogenous random process. The parameter γ_t controls how this weight decreases as the "distance" between paths of the exogenous random process increases.

4.4.3 Memory Requirements

From (4.10), the slope correction function g_t^k corresponding to stage $t \in \{1, \ldots, T-1\}$ and iteration $k \in \mathbb{Z}_+$ can be expressed

$$g_t^k(\mathcal{W}_t) := \begin{cases} \sum_{j=0}^{k-1} \alpha_t^j(\mathcal{W}_t) \left(\xi_t^j - \eta_t^j - g_t^j\big(\mathcal{W}_t^j\big) \right) & \text{if } k > 0, \\ 0 & \text{else,} \end{cases} \tag{4.13}$$

for all $\mathcal{W}_t$. In contrast to the two-stage case [21], in which g_t^k is a vector that gets updated at each iteration, here it is a function that determines a slope-correction vector suitable for an approximation associated with a specific path of the exogenous random process. From (4.13), its output depends on how similar its input is to each of the realizations of the random process sampled so far, *i.e.*, $\left\{ \mathcal{W}_t^j \;\middle|\; 0 \leq j < k \right\}$, as well as the slope-correction vectors computed so far, *i.e.*, $\left\{ \xi_t^j - \eta_t^j - g_t^j\big(\mathcal{W}_t^j\big) \;\middle|\; 0 \leq j < k \right\}$. Hence, the proposed algorithm requires storing all this information, which grows linearly with the number of iterations.

4.4.4 Convergence

For the case of finite-support uncertainty, for which the exogenous random process can be represented by a finite scenario tree, for specific parameters of the algorithm, namely $\gamma_t = \infty$ for each $t \in \{1, \ldots, T-1\}$, and under certain assumptions, the iterates produced by the proposed algorithm can be shown to have a subsequence that converges to the solutions of the nested problems (4.1). The details can be found in Appendix C. The main idea behind the proof is to show that the iterates produced by the algorithm, which are obtained using deterministic cost-to-go function approximations and are referred to as "level-0" solutions, "track" the solutions of approximate problems that have one level of exact expected values, or "level-1" solutions. These approximations with one level of exact expected values, say for stage t, are based on the exact expected value of the cost of

stage $t+1$ but assume deterministic approximations of the cost-to-go of the future stages. Then, show inductively that level-1 solutions track level-2 solutions, level-2 solutions track level-3 solutions, and so on. For a given stage t, level-$(T-t)$ solutions are exact, and hence the iterates produced by the proposed algorithm can be shown to track these exact solutions. Aside from mathematical induction, the proof uses techniques based on those used by Cheung & Powell for the two-stage case [21], and those used in Appendix B for the case with expected-value constraints. These consist (respectively) of showing that level-τ solutions for each τ get closer together as the number of iterations increases, and assuming that an infinite sum of uncontrolled side effects, or "drift", is bounded above almost surely. One condition required for the proof of the parameterized algorithm proposed here, which is not required by the algorithms analyzed in [21] and Chapter 3, is that the cost functions $F_t(\,\cdot\,, \mathcal{W}_t)$ need to be strongly convex. However, in practice, not having this property is not likely to be an issue since regularization can be added to the problem without significantly affecting accuracy.

4.5 Multi-Stage Stochastic Economic Dispatch

4.5.1 Background and Overview

In electric power systems, which are also known as power networks or grids, the operation of generators needs to be planned in advance, *e.g.*, days or hours ahead. One reason for this is that generators with low operating cost typically have limited flexibility [6] [52] [89]. In practice, the on-off states of generators as well as tentative or financially-binding production levels for every hour of the day are determined the day before. Then, adjustments to this schedule are typically made shortly before delivery, *e.g.*, an hour ahead, to ensure that the system remains balanced in an economically efficient way in the event of deviations from predicted conditions. Any last-minute disturbances and deviations are handled automatically through frequency control, with resources such as AGC [57]. Until recently, uncertainty in power systems came mainly from demand forecast errors and potential fault occurrences, was relatively low, and hence deterministic models with appropriately-defined reserve requirements produced acceptable results. However, with the projected large-scale penetration of renewable energy sources, in particular wind and solar, which are highly variable and unpredictable, uncertainty must be properly considered. If not, operating costs of the system could become excessively high and also security issues could arise [6] [73] [89].

In this section, the performance of the algorithms described in Sections 4.3 and 4.4 is compared on a multi-stage day-ahead stochastic ED problem from power systems with large-scale penetration of renewable energy such as wind and solar. The problem considered assumes that generator on-off states have been determined, and seeks to find power production levels taking into account the limited flexibility of certain generation technologies. This problem is formulated with the goal of capturing important properties of generation scheduling in networks with large-scale penetration of renewable energy, namely uncertainty, the sequential nature of the decision-making process, and generator limitations. In practice, the techniques evaluated here could be used to construct operational policies in a receding-horizon fashion. Lastly, the applicability and performance of the algorithms described in this work on more realistic multi-stage generator scheduling problems, *e.g.*, that consider unit commitment decisions or market-clearing practices, is subject of future work.

4.5.2 Problem Formulation

The problem considered consists of determining generator output powers for a sequence of time periods indexed by $t \in \{1, \ldots, T\}$, where $T \in \mathbb{N}$, in the presence of exogenous uncertain output powers of renewable energy sources. The power network is assumed to have $n_p \in \mathbb{N}$ generators with limited flexibility, *e.g.*, "baseload units" such as large nuclear and coal-fired power plants [6] [52], $n_q \in \mathbb{N}$ fast-ramping flexible generators, *e.g.*, "peaking units" such as gas-fired power plants [6] [52], $n_r \in \mathbb{N}$ renewable energy sources, $n_b \in \mathbb{N}$ buses, $n_e \in \mathbb{N}$ edges, and $n_d \in \mathbb{N}$ loads. At each stage $t \in \{1, \ldots, T\}$, $p_t \in \mathbb{R}^{n_p}$ denotes output powers of generators with limited flexibility, $q_t \in \mathbb{R}^{n_q}$ denotes output powers of fast-ramping flexible generators, $r_t \in \mathbb{R}^{n_r}_+$ denotes output powers of renewable energy sources, $s_t \in \mathbb{R}^{n_r}$ denotes powers of renewable energy sources after curtailments, $\theta_t \in \mathbb{R}^{n_b}$ denotes node voltage angles, where one is treated as a reference, and $d_t \in \mathbb{R}^{n_d}$ denotes powers consumed by loads. With this notation, the problem can be expressed mathematically as

the following nested optimization problem:

$$\begin{aligned}
\underset{p_t,q_t,\theta_t,s_t}{\text{minimize}} \quad & \varphi(p_t) + \psi(q_t) + G_t(p_t, \mathcal{R}_t) && \text{(4.14a)}\\
\text{subject to} \quad & Pp_t + Qq_t + Ss_t - A\theta_t - Dd_t = 0 && \text{(4.14b)}\\
& y^{\min} \le p_t - p_{t-1} \le y^{\max} && \text{(4.14c)}\\
& z^{\min} \le J\theta_t \le z^{\max} && \text{(4.14d)}\\
& p^{\min} \le p_t \le p^{\max} && \text{(4.14e)}\\
& q^{\min} \le q_t \le q^{\max} && \text{(4.14f)}\\
& 0 \le s_t \le r_t, && \text{(4.14g)}
\end{aligned}$$

for each $t \in \{1, \ldots, T\}$, where φ and ψ are separable strongly convex quadratic functions that quantify power generation costs of generators with limited flexibility and of fast-ramping flexible generators, respectively, $\{P, Q, S, A, D, J\}$ are sparse constant matrices, $\mathcal{R}_t := \{r_1, \ldots, r_t\}$, and p_0 is a constant vector. Constraint (4.14b) enforces power balance at each node of the network using a DC power flow model [108]. Constraint (4.14c) imposes strong ramping constraints on the generators with limited flexibility to model their limited ability to change power levels during operations due to physical, economic, and design reasons [6] [52]. Constraint (4.14d) enforces edge flow limits due to thermal ratings of transmission lines and transformers, constraint (4.14e) enforces output power limits of generators with limited flexibility, constraint (4.14f) enforces output power limits of fast-ramping flexible generators, and constraint (4.14g) enforces limits of powers of renewable energy sources after curtailments. The cost-to-go functions $G_t(\,\cdot\,, \mathcal{R}_t)$ are defined by

$$G_t(p, \mathcal{R}_t) := \begin{cases} \mathbb{E}\,[\, H_{t+1}(p, \mathcal{R}_{t+1}) \,|\, \mathcal{R}_t \,] & \text{if } t \in \{1, \ldots, T-1\}, \\ 0 & \text{else,} \end{cases} \tag{4.15}$$

for all p, where the function H_t is such that $H_t(p, \mathcal{R}_t)$ is the optimal objective value of problem (4.14) with $p_{t-1} = p$, and $\mathbb{E}\,[\,\cdot\,|\,\mathcal{R}_t\,]$ is expectation with respect to $\mathcal{R}_{t+1}$ conditioned on $\mathcal{R}_t$.

Due to the ramping restrictions (4.14c) on the output powers of generators with limited flexibility, and the output power limits (4.14e) and (4.14f), the nested problem (4.14) may not in general have the extended relatively complete recourse property described in Section

4.2. However, this can be typically ensured by modifying the problem slightly to allow for emergency load shedding at an appropriately high cost, or include other source of flexibility such as demand response.

4.5.3 Exogenous Random Process

The exogenous random output powers of renewable energy sources are assumed to be given by

$$r_t = \Pi_r\big(\Pi_\delta(\bar{r}_t + \delta_t, r_{t-1})\big), \tag{4.16}$$

for each $t \in \{1, \ldots, T\}$, where $\bar{r}_t \in \mathbb{R}^{n_r}_+$ are constant vectors, and $\delta_t \sim \mathcal{N}(0, \Sigma_t)$. Π_r is the projection operator given by

$$\Pi_r(z) := \arg\min \big\{ \|r - z\|_2 \;\big|\; 0 \le r \le r^{\max} \big\},$$

for all z, where $r^{\max} \in \mathbb{R}^{n_r}_{++}$ are the power capacities of the renewable energy sources. Π_δ is a randomized operator given by

$$\Pi_\delta(z_1, z_2) := \begin{cases} \arg\min \big\{ \|r - z_1\|_2 \;\big|\; -\delta^{\max} \le r - z_2 \le \delta^{\max} \big\} & \text{with probability } 1-\epsilon, \\ z_1 & \text{with probability } \epsilon, \end{cases} \tag{4.17}$$

for all (z_1, z_2), where $\delta^{\max} \in \mathbb{R}^{n_r}_{++}$ is a constant vector. Lastly, for $t \in \{1, \ldots, T\}$, the covariance matrix Σ_t is assumed to satisfy $\Sigma_t = (t-1)\Sigma/(T-1)$, for $T > 1$, where Σ is a positive definite matrix. This captures the fact that the output powers of renewable energy sources are known at time $t = 1$, and that the uncertainty increases with t.

The use of Gaussian random variables for modeling renewable energy uncertainty or forecast errors is common in the power systems literature [64] [70] [74]. The model considered here adds Gaussian noise δ_t to "base" power levels $\bar{r}_t$. The projection operator Π_r ensures that the resulting powers are non-negative and not greater than the capacities of the sources. The randomized operator Π_δ guarantees with probability $1 - \epsilon$ that the resulting powers have ramps bounded by a given $\delta^{\max}$. This is used to prevent having large ramps that occur unrealistically often in the planning horizon. Lastly, as is done in Chapters 2 and 3, spatial correlation between "nearby" renewable energy sources is considered here through a non-diagonal matrix Σ parameterized by a correlation coefficient ρ and a "correlation distance" N.

4.5.4 Application of Proposed Algorithm

In order to apply the parameterized SHA algorithm described in Section 4.4.2 to the nested problem (4.14), initial expected cost-to-go function approximations are required. Motivated by the results obtained in Chapters 2 and 3 using the AdaCE method, the approximations considered here consist of

$$\widehat{G}_t(p, \mathcal{R}_t) := \begin{cases} H^0_{t+1}(p, \mathbb{E}\,[\,\mathcal{R}_{t+1} \,|\, \mathcal{R}_t\,]) & \text{if } t \in \{1, \ldots, T-1\}, \\ 0 & \text{else,} \end{cases} \tag{4.18}$$

for all p, for each $t \in \{1, \ldots, T\}$ and $\mathcal{R}_t$, where the function H^k_t here is such that $H^k_t(p, \mathcal{R}_t)$ is the optimal objective value of problem (4.7) with $p_{t-1} = p$. An important observation is that this choice of approximation is not necessarily differentiable everywhere due to the inequality constraints in (4.14). However, as will be seen in Section 4.5.6, this does not cause problems possibly due to the fact that the lack of differentiability only occurs at certain points. For instances of problem (4.7) on which issues do arise due to this lack of differentiability everywhere, one may consider using instead of $H^0_t(p, \mathcal{R}_t)$ an approximation that replaces inequality constraints with barrier terms in the objective.

For this choice of initial expected cost-to-go function approximations, the subproblems (4.7) solved by the proposed algorithm can be expressed as

$$\underset{\{p_\tau, q_\tau, \theta_\tau, s_\tau\}_{\tau=t}^T}{\text{minimize}} \quad \varphi(p_t) + \psi(q_t) + g_t^m\big(\mathcal{R}_t\big)^T p_t \; + \sum_{\tau=t+1}^{T} \Big(\varphi(p_\tau) + \psi(q_\tau) + g_\tau^0\big(\widehat{\mathcal{R}}_{t,\tau}\big)^T p_\tau\Big) \tag{4.19a}$$

$$\text{subject to} \quad (p_t, q_t, \theta_t, s_t) \in \mathcal{K}_t\big(p_{t-1}, r_t\big) \tag{4.19b}$$

$$(p_\tau, q_\tau, \theta_\tau, s_\tau) \in \mathcal{K}_\tau\big(p_{\tau-1}, \hat{r}_\tau\big), \;\; \tau = t+1, \ldots, T, \tag{4.19c}$$

for each $t \in \{1, \ldots, T\}$ and $k \in \mathbb{Z}_+$ with $p_{t-1} = p^k_{t-1}$, $\mathcal{R}_t = \mathcal{R}^k_t$, and $m = k$. Here, p^k_{t-1} denotes the solution of the previous-stage subproblem, $\mathcal{R}^k_t := \big\{r^k_1, \ldots, r^k_t\big\}$ are the sampled realizations of renewable powers up to stage t during iteration k, constraints (4.19b) and (4.19c) correspond to constraints (4.14b)-(4.14g) for $\tau \in \{t, \ldots, T\}$, and $\widehat{\mathcal{R}}_{t,\tau} := \big\{\hat{r}_1, \ldots, \hat{r}_\tau\big\}$

is defined recursively as

$$\widehat{\mathcal{R}}_{t,\tau} = \begin{cases} \mathbb{E}\left[\mathcal{R}_\tau \,\middle|\, \widehat{\mathcal{R}}_{t,\tau-1} \right] & \text{if } \tau > t, \\ \mathcal{R}_t & \text{else,} \end{cases} \tag{4.20}$$

for each $\tau \in \{t, \ldots, T\}$. From the properties of φ, ψ and $\mathcal{K}_\tau$, these optimization problems are deterministic multi-stage convex QP problems.

From (4.8), the vector ξ_t^k can be constructed from the dual solution of subproblem (4.19) for stage $t+1$ with $p_t = p_t^k$, $\mathcal{R}_{t+1} = \mathcal{R}_{t+1}^k$, and $m = k$, which is solved during the "forward pass" of the algorithm. More specifically, it can be shown that $\xi_t^k = -\mu_{t+1}^k + \pi_{t+1}^k$, where μ_{t+1}^k and π_{t+1}^k denote the Lagrange multipliers associated with the upper and lower ramping constraints (4.14c), respectively. Similarly, from (4.9) and (4.18), the vector η_t^k can be constructed (estimated) using an analogous formula from the dual solution of subproblem (4.19) for stage $t+1$ with $p_t = p_t^k$, $\mathcal{R}_{t+1} = \mathbb{E}\left[\mathcal{R}_{t+1} \,\middle|\, \mathcal{R}_t^k \right]$, and $m = 0$. This subproblem is also solved during the "forward pass" of the algorithm but only implicitly, *i.e.*, as part of the solution of another subproblem. In particular, its primal solution is obtained during this process, and this can be exploited when computing the dual solution.

As mentioned above, the proposed initial function approximations (4.18) result in the algorithm having to solve the deterministic multi-stage subproblems (4.19) during the forward pass. More specifically, at each iteration, the algorithm has to solve a subproblem with $T-\tau+1$ stages for each $\tau \in \{1, \ldots, T\}$. Hence, this choice of initial function approximations is not suitable for problems with a large number of stages since the computational requirements of the algorithm would be high. For such problems, a "depth-limited" modification of (4.18) that ignores the cost-to-go after a certain number of stages may be considered. The investigation of this idea will be the subject of future work.

4.5.5 Implementation

The proposed and benchmark algorithms as well as the multi-stage stochastic ED problem were implemented using the Python programming language. More specifically, the SDDP and parameterized SHA algorithms were implemented within the Python package `OPTALG` [2] using the scientific-computing packages `Numpy` and `Scipy` [38] [97]. The multi-stage optimal

[2] `http://optalg.readthedocs.io`

generator dispatch problem was implemented within the Python package `GRIDOPT` [3] using the modeling library `PFNET` [4]. For solving problems (4.19), the sparse interior-point QP solver of `OPTALG` was used. For obtaining $\Sigma^{1/2}$, where Σ is the spatial covariance matrix defined in Section 4.5.3, the sparse Cholesky code `CHOLMOD` was used via its Python wrapper [5] [19]. Lastly, to construct a suitable scenario tree for SDDP, the clustering routines from the Python package `scikit-learn` [62] were used.

4.5.6 Numerical Experiments

The proposed and benchmark algorithms were applied to several instances of the multi-stage stochastic ED problem in order to compare their performance. The performance of these two algorithms was also compared against that of a greedy and a CE-based algorithm. The greedy algorithm ignores the expected cost-to-go for subsequent stages at each stage, and hence is equivalent to the SDDP algorithm executed for 0 iterations. This follows from the fact that $\widetilde{G}_t^0 = 0$ for all t, as stated in Section 4.3.3. On the other hand, the CE-based algorithm solves the problem by replacing expected costs by costs associated with expected realizations of the uncertainty. Due to the particular choice (4.18) of initial function approximations, the CE-based algorithm is equivalent to the parameterized SHA algorithm before any slope corrections are made, *i.e.*, executed for 0 iterations.

4.5.6.1 Test Cases

The power networks used to construct instances of the multi-stage stochastic ED problem consisted of the small `IEEE14` network [6] and two mid-scale networks from North America and Europe. For all of these networks, a 24-hour scheduling period divided into 8 stages was considered. Table 4.1 shows the properties of the test cases constructed. The labels of the columns correspond to the quantities defined in Section 4.5.2. In practice, higher resolutions, *e.g.*, hourly, are typically used for generator scheduling but with deterministic models. Multi-stage stochastic ED problems with such resolution will be considered in future work along with a suitable choice of initial function approximations, such ones based on "depth-limited" CE models, as suggested at the end of Section 4.5.4.

[3] `http://gridopt.readthedocs.io`
[4] `http://pfnet.readthedocs.io`
[5] `http://pythonhosted.org/scikits.sparse`
[6] `http://www2.ee.washington.edu/research/pstca/`

Table 4.1: Test Cases

name	n_b	n_e	n_p	n_q	n_r	n_d	T
Case A	14	20	5	5	5	11	8
Case B	2454	2781	211	211	236	1149	8
Case C	3120	3693	278	278	248	2314	8

The generators with limited flexibility were taken as the adjustable generators originally present in the network data. Due to the absence of cost data, the separable cost function φ for these generators was constructed using uniformly random coefficients in $[0.01, 0.05]$ for the quadratic terms, and uniformly random coefficients in $[10, 50]$ for the linear terms. These coefficients assume that power quantities are in units of MW, and are consistent with those found in several `MATPOWER` cases [109]. For these generators, the maximum ramping rates were set to 1% of their power capacities per hour in order to model their limited ability to change power levels during operations due to physical, economic, and design reasons. The fast-ramping flexible generators were new generators added to the networks in the same locations as the generators with limited flexibility. The separable cost function ψ for these generators was set to satisfy $\psi = 10\varphi$. This cost difference is consistent with the difference in marginal costs between baseload and peaking generators analyzed in [105]. Lastly, the known constant output powers of generators with limited flexibility before stage 1, *i.e.*, p_0, were set to the least-cost powers that balanced the network for stage 1.

Three daily load profiles were obtained from a North American power marketing administration. Each load profile was normalized and used to modulate the (static) loads of one of the power networks. Figure 4.1 shows the resulting aggregate load profiles for each of the test cases.

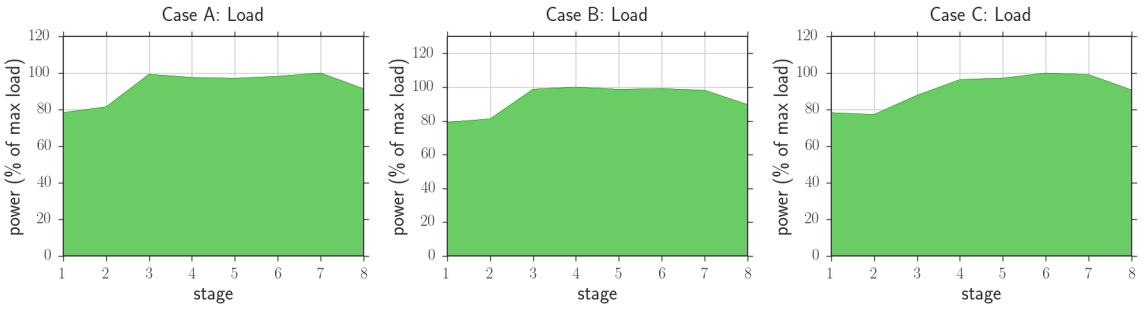

Figure 4.1: Load profiles

The renewable energy sources used to construct the test cases of Table 4.1 were added manually to each of the power networks at the nodes with generators. The capacity $r_i^{\max}$ of each source was set to $\max_t 1^T d_t/n_r$, *i.e.*, to the maximum aggregate load divided by the number of renewable energy sources. The base power profiles $\{\bar{r}_t\}$ were obtained by multiplying $0.5r^{\max}$ by a normalized daily wind power profile obtained from the same North American power marketing administration from which the load data was obtained. This resulted in a maximum base renewable energy penetration of 50% of the maximum aggregate load, which is a high penetration scenario. A different normalized wind power profile was used for each network. For the covariance matrix Σ, its diagonal was set to the element-wise square of $r^{\max}$, and its off-diagonal elements were set in such a way that powers from sources connected to nodes that are at most $N = 10$ edges ways had a correlation coefficient of $\rho = 0.1$, and zero otherwise. For these sources, the maximum ramping rates were set to 30% of their power capacities per hour, and these limits were enforced with a probability of 0.7, *i.e.*, $\epsilon = 0.3$ in (4.17). Figure 4.2 shows sample realizations of the aggregate powers from the renewable energy sources for each of the test cases. An important observation is that the aggregate power from renewable sources is much more variable for Case A than for the other cases. The reason for this is that Cases B and C have many more renewable energy sources that are uncorrelated and hence there is more cancellation.

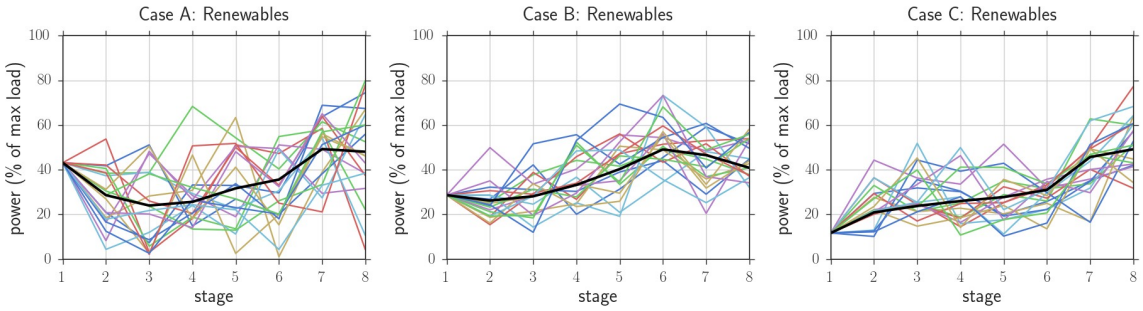

Figure 4.2: Sample realizations of powers from renewable energy sources

4.5.6.2 SDDP Validation

In order to validate the implementation of the SDDP algorithm, convergence to zero of the difference between upper and lower bounds computed at every iteration was checked. This was done for the smallest case, namely, Case A, using a small scenario tree obtained using Monte Carlo sampling with branching factors given by $(4, 3, 2, 1, 1, 1, 1)$, which gives 137

nodes and 24 scenarios. The upper and lower bounds were obtained at each iteration $k \in \mathbb{Z}_+$ of the algorithm by evaluating $F_1(x_1^k, w_1) + \widetilde{G}_1(x_1^k, \mathcal{W}_1)$ and $F_1(x_1^k, w_1) + \widetilde{G}_1^k(x_1^k, \mathcal{W}_1)$, respectively. The results obtained are shown in Figure 4.3.

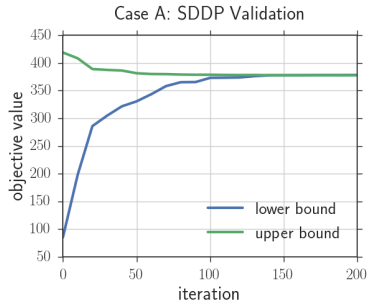

Figure 4.3: SDDP validation

4.5.6.3 Algorithm Comparison

The algorithms were executed for various numbers of iterations and the policies obtained were evaluated. More specifically, the parameterized SHA algorithm was executed for 0, 25, 50 and 100 iterations. As already noted, for 0 iterations the policy obtained with this algorithm is the same as the obtained from solving CE problems at each stage. For the radial basis functions used by the algorithm, the parameter γ_t was set to $\gamma(n_r t)^{-1/2}$ for each $t \in \{1, \ldots, T\}$, where $\gamma = 1$, and n_r is the number of renewable energy sources. On the other hand, the SDDP algorithm was executed for 0, 25, 100 and 400 iterations. As already noted, for 0 iterations the policy obtained with this algorithm is the same as the greedy policy obtained by solving problem (4.5) at each stage ignoring the cost-to-go function approximation. The scenario trees used for this algorithm were constructed using conditional Monte Carlo sampling and K-means clustering at each stage. Information about these scenario trees including their construction time is shown in Table 4.2. To evaluate the policies produced by the algorithms, the policies were simulated using 500 sampled realizations of the exogenous random process. The average total costs obtained are shown Figure 4.4 as a function of the execution time of the algorithm, where GR denotes greedy.

Table 4.2: Scenario trees for SDDP

case	stages	samples/stage	clusters/stage	nodes	scenarios	time (min)
Case A	8	1000	3	3280	2187	1.65
Case B	8	1000	3	3280	2187	7.02
Case C	8	1000	3	3280	2187	7.26

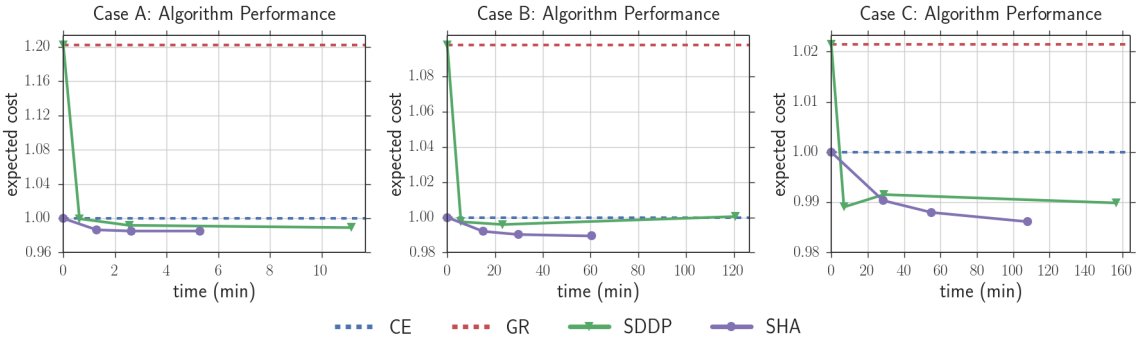

Figure 4.4: Algorithm performance

As the plots of Figure 4.4 suggest, the proposed algorithm is able to exploit the accuracy of the initial expected cost-to-go function approximations provided, which in this case are the CE models, and quickly generate policies that achieve lower expected costs compared to the ones obtained using SDDP. Another important observation is that for Cases A and B, the greedy model used by SDDP at the beginning of its execution can be much worse than the CE model. For Case B, the expected cost associated with the SDDP policy obtained after 400 iterations is slightly worse than with the one obtained after 100 iterations. This phenomenon may be attributed to a slight overfitting of the policy to the scenario tree or to the small errors obtained by computing expected costs using a finite number of sampled realizations. We note here that the execution times of SDDP shown in these plots do not include the times spent constructing the scenario trees, whose details are provided in Table 4.2.

In addition to the average total costs obtained by simulating the policies obtained with the algorithms, the average aggregate power generation profiles were also plotted. This was done to obtain a simple visualization of the strategy used by each policy. The policies considered were those obtained with the proposed algorithm after 0 iterations, *i.e.*, CE, and 100 iterations, and with SDDP after 0 iterations, *i.e.*, greedy, and 400 iterations. The

results obtained are shown in Figure 4.5, where again GR denotes greedy. From the plots, it can be seen that the greedy policy starts by using all the renewable energy available, and has in general less renewable energy left on average during all stages. The other three policies are visually similar but the ones obtained with the proposed algorithm after 100 iterations and with SDDP after 400 iterations have in general a smaller "red area" than the CE policy. This corresponds to using less energy on average from fast-ramping flexible generators, which have a higher operating cost. In other words, the policies obtained using stochastic optimization exploit better the limited flexibility of the resources that have lower operating cost.

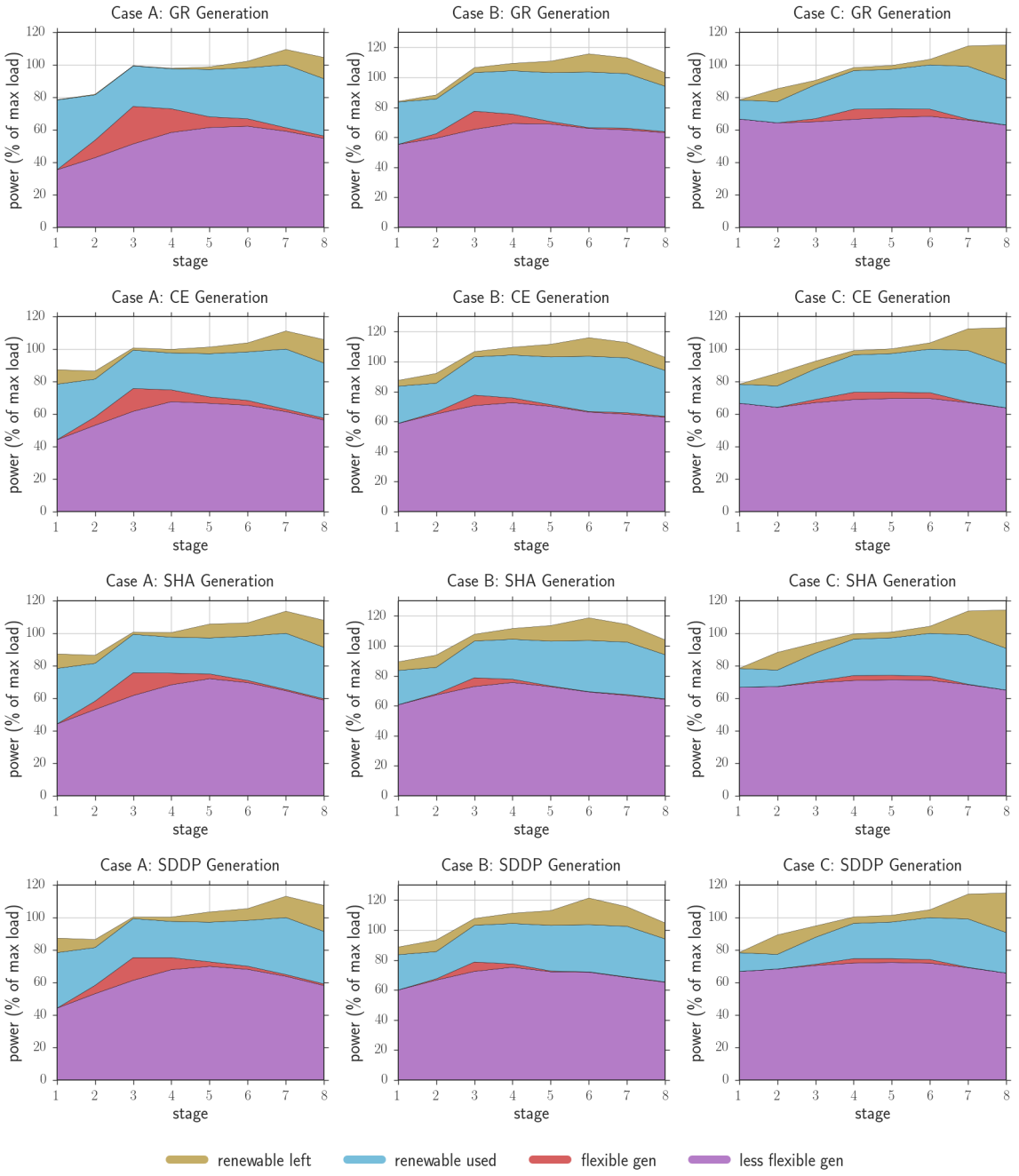

Figure 4.5: Generation profiles

4.5.6.4 Generalization Analysis

An experiment was also performed to determine the effects of the choice of generalization parameters γ_t used by the radial basis functions on the performance of the parameterized SHA algorithm. This experiment consisted of executing the algorithm for 0, 25, 50, and 100 iterations, as in the previous experiment, using $\gamma_t = \gamma(n_r t)^{-1/2}$ for each $t \in \{1, \ldots, T\}$ and $\gamma \in \left\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\right\}$. The results obtained are shown in Figure 4.6.

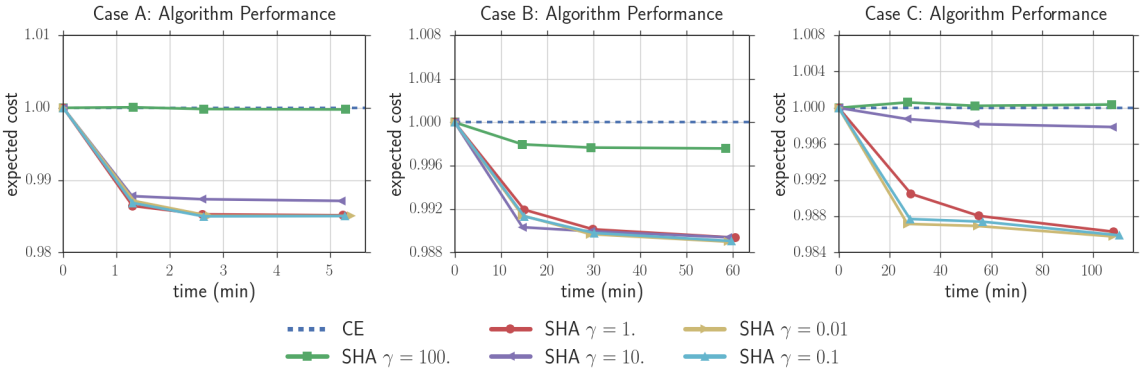

Figure 4.6: Generalization analysis

As the plots of Figure 4.6 show, for large values of γ, *i.e.*, for little or no generalization, the performance of the algorithm is similar to that of the CE-based algorithm. This is expected since for γ large, the same path of the random process needs to be sampled again in order to see the effects of a slope correction. Another important and perhaps surprising observation is that for all $\gamma \in \left\{10^{-2}, 10^{-1}, 10^{0}\right\}$, the performance of the algorithm is equally promising. This shows that the algorithm is benefiting similarly from generalizing slope corrections to neighboring paths of the random process for a wide range of neighborhood "sizes".

4.6 Conclusions

In this chapter, a new algorithm for solving convex multi-stage stochastic optimization problems has been presented. The algorithm is based on a parameterized extension of a stochastic hybrid approximation procedure that allows it to exploit accurate initial expected cost-to-go function approximations provided by the user. The algorithm can be applied directly to problems with continuous uncertainty since it uses radial basis functions

for generalizing information learned from specific realizations of the random process. An analysis of the convergence of the algorithm has been provided for the tractable case of finite-support uncertainty. In particular, conditions have been identified that guarantee that the iterates produced by the algorithm have a subsequence that convergences to the solution of the problem. The performance of the algorithm has been compared against that of a greedy, a CE-based, and an SDDP algorithm on a multi-stage stochastic ED problem from power system operations planning under high penetration of renewable energy. The results obtained showed that for this problem the proposed algorithm was able to exploit the accuracy of the initial approximations of expected cost-to-go functions and outperform the benchmark algorithms.

Chapter 5

Applicability of AdaCE for Unit Commitment

5.1 Background and Overview

The UC problem aims to schedule the most cost-effective combination of generating units to meet the demand while taking into account technical and operational constraints [80]. Due to strong temporal limitations of some units, *e.g.*, long start-up times, the schedule must be planned in advance. Usually, this is done the day before for a time horizon of 24 hours with hourly resolution. Due to its non-convexity and scale, the UC problem poses serious computational challenges, especially since the time available to solve it is limited in practice [89]. For example, the Midcontinent Independent System Operator (MISO) needs to solve a UC problem with around 1400 generating units in about 20 minutes [18]. Until recently, uncertainty came mainly from demand forecast errors and potential fault occurrences, was relatively low, and hence deterministic UC models were acceptable. However, with the projected large-scale penetration of renewable energy sources, in particular wind and solar, which are highly variable and unpredictable, uncertainty must be properly considered. If not, operating costs of the system could become excessively high and also security issues could arise [37] [6]. For this reason, algorithms that can solve UC problems with high levels of uncertainty efficiently are highly desirable.

For the past 40 years, there has been extensive research by the scientific community on the UC problem. In particular, research has focused on analyzing the practical issues and economic impacts of UC in vertically integrated utilities and in deregulated markets,

and on leveraging advances in optimization algorithms. With regards to algorithms, the most explored approaches have been based on the following: Dynamic Programming [87], Mixed-Integer Linear Programming (MILP) [60] [95], and heuristics [48]. Approaches for dealing with uncertainty in UC have been mostly based on robust optimization [5], chance-constrained optimization [96], and stochastic optimization. Those based on the latter have typically used scenarios, or SAA-based approaches, combined with Benders decomposition [100] or with Lagrangian Relaxation [104]. For example, in [95], the authors explore the benefits of stochastic optimization over a deterministic approach for solving a UC problem with large amounts of wind power. Uncertainty is represented with a scenario tree and the resulting problem is solved directly using an MILP solver. The authors in [100] also use scenarios to model wind uncertainty in UC but use Benders decomposition to exploit the structure of the resulting two-stage problem combined with heuristics to improve its performance. In [61], the authors use Lagrangian Relaxation to decompose a multi-stage stochastic UC problem into single-scenario problems and exploit parallel computing resources. For more information on approaches for solving the UC problem, the interested reader is referred to [89].

SHA algorithms are a family of algorithms for solving stochastic optimization problems. They were initially proposed in [21] and are characterized by working directly with stochastic quantities at every iteration, and using these to improve deterministic models of expected-value functions. They have the key property that they can leverage accurate initial approximations of expected-value functions provided by the user, and that the improvements to these functions made during the execution of the algorithm do not change their structure. In previous chapters, algorithms of this type are applied to various stochastic ED problems, which are closely related to the stochastic UC problem but focus exclusively on determining generator power levels and assume that on-off states are given. In particular, in Chapter 2, the SHA algorithm proposed in [21] is applied to a two-stage stochastic ED problem by using a CE model for constructing the initial approximation of the expected second-stage cost. The resulting AdaCE approach is shown to have superior performance compared to some widely-used approaches on the test cases considered. This approach is then extended in Chapters 3 and 4 to solve more complex versions of the stochastic ED problem that include expected-value constraints and multiple planning stages, and promising performances are shown. The close connection between UC and ED problems suggests that perhaps the combination of the SHA procedure and CE models may also lead to a promising approach

for solving stochastic UC problems. However, the theoretical and practical issues that could arise due to the binary variables present in the UC problem warrant further investigation in order to validate this hypothesis.

In this chapter, the applicability and performance of the AdaCE algorithm, which combines the SHA procedure with initial expected-value function approximations based on CE models, is studied. More specifically, the stochastic UC problem is formulated as a two-stage mixed-integer optimization problem that captures the essential properties of the problem that make it challenging to solve, namely, binary variables, causality, and uncertainty. The theoretical and practical challenges associated with the use of the AdaCE algorithm for solving this problem are explored. In particular, possible outcomes and practical limitations on the types of approximations are investigated. Additionally, ideas inspired from the mini-batch technique used in Machine Learning are considered for improving the performance of the algorithm on this problem. Furthermore, the performance of the algorithm is compared against that of a deterministic approach and that of the widely-used SAA or scenario approach combined with Benders decomposition on test cases constructed from several power networks and wind power distributions. The results obtained show that despite a lack of theoretical guarantees, the AdaCE method can find commitment schedules in reasonable times that are better than the ones obtained with the benchmark algorithms considered.

5.2 Two-Stage Stochastic Unit Commitment

As already mentioned, the stochastic UC problem consists of determining the most cost-effective schedule of power generating units to meet the demand while taking into account device limits, operational constraints, and uncertainty. Typically, the scheduling period of interest is 24 hours with a resolution of one hour. In a vertically integrated utility, the schedule defines periods of time during which each generating unit will be on or off, respecting its minimum up and down times. Power production levels are typically determined shortly before delivery in a process known as ED, and are therefore not part of the unit commitment schedule. On the other hand, in a deregulated electricity market, both on-off states and power production levels are determined jointly during market clearing [22]. The schedules are chosen to maximize profit or social welfare, and special attention is paid to energy pricing and payments [28]. Traditionally, the uncertainty in a system has been small

and it has come mainly from the demand forecast errors and potential contingencies. However, with the projected large-scale penetration of renewable energy in modern low-carbon power systems, in particular wind and solar, the uncertainty is expected to be quite high, and hence a proper treatment of this is crucial for ensuring system reliability and economic efficiency.

5.2.1 Problem Formulation

In this chapter, the stochastic UC problem is modeled as a two-stage stochastic mixed-integer optimization problem. This model, albeit simplified, allows capturing the following key properties that make this problem computationally challenging:

- The commitment decisions are restricted to be either on or off, *i.e.*, they are binary.
- The commitment schedule needs to be determined ahead of operation.
- The commitment schedule needs to respect the inter-temporal constraints of the generating units.
- The operation of the system is strongly affected by the availability of renewable energy, which is highly uncertain.
- Network security constraints can impose restrictions on the utilization of the available generation resources.

Hence, it serves as an adequate model for performing an initial assessment of the applicability and performance of an algorithm for solving the stochastic UC problem.

Mathematically, the problem is given by

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & F(u) + \mathbb{E}\left[Q(u, w)\right] \\ \text{subject to} \quad & u \in \mathcal{U} \cap \{0,1\}^{nT}, \end{aligned} \tag{5.1}$$

where u is the vector of on-off states $u_{i,t}$ for each power generating unit $i \in \{1, \ldots, n\}$ and time $t \in \{1, \ldots, T\}$, w is the vector of random powers from renewable energy sources during the operation period, and $\mathbb{E}\left[\,\cdot\,\right]$ denotes expectation with respect to w. The function F is an LP-representable cost function, *i.e.*, it can be represented as a linear program. It is given

by

$$F(u) = \sum_{i=1}^{n} \sum_{t=1}^{T} \left(c_i^u \max\{u_{i,t} - u_{i,t-1}, 0\} + c_i^d \max\{u_{i,t-1} - u_{i,t}, 0\} \right),$$

for all $u \in \{0,1\}^{nT}$, where c_i^u and c_i^d are the start-up and shut-down costs for generator i, respectively, and $u_{i,0} \in \{0,1\}$ are known constants. The constraint $u \in \mathcal{U}$ represents the minimum up and down time restrictions of the generating units, which serve to prevent the erosion caused by frequent changes of thermal stress. They are given by

$$\begin{aligned} u_{i,t} - u_{i,t-1} &\leq u_{i,\tau}, && \forall i \in \{1,\dots,n\}, \forall (t,\tau) \in S_i^u \\ u_{i,t-1} - u_{i,t} &\leq 1 - u_{i,\tau}, && \forall i \in \{1,\dots,n\}, \forall (t,\tau) \in S_i^d, \end{aligned}$$

where

$$\begin{aligned} S_i^u &:= \left\{ (t,\tau) \;\middle|\; t \in \{1,\dots,T\}, \tau \in \{t+1,\dots,\min\{t+T_i^u-1,T\}\} \right\} \\ S_i^d &:= \left\{ (t,\tau) \;\middle|\; t \in \{1,\dots,T\}, \tau \in \{t+1,\dots,\min\{t+T_i^d-1,T\}\} \right\}, \end{aligned}$$

and T_i^u and T_i^d are the minimum up and down times for unit i, respectively.

The "recourse" function $Q(u,w)$ quantifies the operation cost of the system for a given commitment schedule u of generating units and available powers w from renewable energy sources. It is modeled as the optimal objective value of the following multi-period DC OPF problem:

$$\begin{aligned} \underset{p,\theta,s,d}{\text{minimize}} \quad & \varphi(p) + \gamma \left\| d - \bar{d} \right\|_1 && \text{(5.2a)} \\ \text{subject to} \quad & Pp + Ss - Ld - A\theta = 0 && \text{(5.2b)} \\ & \operatorname{diag}(p^{\min})u \leq p \leq \operatorname{diag}(p^{\max})u && \text{(5.2c)} \\ & y^{\min} \leq Dp \leq y^{\max} && \text{(5.2d)} \\ & z^{\min} \leq J\theta \leq z^{\max} && \text{(5.2e)} \\ & 0 \leq d \leq \bar{d} && \text{(5.2f)} \\ & 0 \leq s \leq w, && \text{(5.2g)} \end{aligned}$$

where p are generator powers, s are the utilized powers from renewable energy sources, $\bar{d}$ and d are the requested and supplied load powers, respectively, and θ are bus voltage angles

during the operation period. The function φ is a separable convex quadratic function that quantifies the generation cost, and $\gamma\|d-\bar{d}\|_1$ quantities the cost of load not served ($\gamma \geq 0$). Constraint (5.2b) enforces power flow balance using a DC network model [108], constraint (5.2c) enforces generator power limits, constraint (5.2d) enforces generator ramping limits, constraint (5.2e) enforces branch flow limits due to thermal ratings, and constraints (5.2g) and (5.2f) impose limits on the utilized renewable powers and on the supplied load. Lastly, P, S, L, A, D, J are sparse matrices. It can be shown that the function $Q(\,\cdot\,, w)$ is convex for all w [11].

It is assumed for simplicity that the second-stage problem (5.2) is feasible for all $u \in \mathcal{U} \cap \{0,1\}^{nT}$ and almost all w. This property is commonly referred to as *relatively complete recourse*, and guarantees that $\mathbb{E}\left[Q(u,w)\right] < \infty$ for all $u \in \mathcal{U} \cap \{0,1\}^{nT}$. Typically, allowing emergency load curtailment, as in problem (5.2), or other sources of flexibility such as demand response, are enough to ensure this property in practical problems.

5.2.2 Model of Uncertainty

The vector of available powers from renewable energy sources for each time $t \in \{1,\ldots,T\}$ is modeled by

$$r_t := \Pi_r(\bar{r}_t + \delta_t),$$

where $\bar{r}_t$ is a vector of non-negative "base" powers and $\delta_t \sim \mathcal{N}(0, \Sigma_t)$. Π_r is the projection operator given by

$$\Pi_r(z) := \arg\min\left\{ \|z-r\|_2 \;\middle|\; 0 \leq r \leq r^{\max} \right\},\ \forall z,$$

where $r^{\max}$ is the vector of power capacities of the renewable energy sources. The random vector w in (5.1) is therefore composed of $(r_1,\ldots,r_T)$. The covariance matrix Σ_t is assumed to be given by $\Sigma_t := (t/T)\bar{\Sigma}_t$, where $\bar{\Sigma}_t$ is a positive semidefinite matrix whose diagonal is equal to the element-wise square of $\bar{r}_t$, and off-diagonals are such that the correlation coefficient between powers of sources at most N branches away equals some pre-defined ρ and zero otherwise. With this model, the forecast uncertainty increases with time and with base power levels.

5.3 SAA-Based Algorithm

Since evaluating the function $\mathbb{E}\left[Q(\,\cdot\,, w)\right]$ in (5.1) accurately is computationally expensive, algorithms for solving problems of the form of (5.1) resort to approximations. A widely used approximation consists of the sample average $m^{-1}\sum_{i=1}^{m} Q(\,\cdot\,, w_i)$, where $m \in \mathbb{Z}_{++}$ and w_i are realizations of the random vector w, which are often called scenarios. Hence, this approach is typically referred to as *sample-average approximation* or *scenario approach* [36]. The resulting approximate problem is given by

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & F(u) + \frac{1}{m}\sum_{i=1}^{m} Q(u, w_i) \\ \text{subject to} \quad & u \in \mathcal{U} \cap \{0,1\}^{nT}, \end{aligned} \tag{5.3}$$

which is a deterministic optimization problem. It can be shown that by making m sufficiently large, the solutions of problem (5.3) approach those of problem (5.1) [36]. However, as m increases, problem (5.3) becomes very difficult to solve due to its size. For this reason it becomes necessary to exploit the particular structure of this problem.

5.3.1 Benders Decomposition

A widely-used strategy for dealing with the complexity of problem (5.3) is to use decomposition combined with a cutting-plane method [11]. This approach, which is known as *Benders decomposition*, replaces the sample-average function $m^{-1}\sum_{i=1}^{m} Q(\,\cdot\,, w_i)$ with a gradually-improving piece-wise linear approximation. The "pieces" of this piece-wise linear approximation are constructed from values and subgradients of the sample-average function, and these can be computed efficiently using parallel computing resources [11].

More specifically, the algorithm performs the following steps at each iteration $k \in \mathbb{Z}_{+}$:

$$u_k = \arg\min \left\{ F(u) + \mathcal{Q}^k(u) \;\middle|\; u \in \mathcal{U} \cap \{0,1\}^{nT} \right\} \tag{5.4a}$$

$$\mathcal{Q}^{k+1}(u) = \begin{cases} a_k + b_k^T(u - u_k) & \text{if } k = 0, \\ \max\left\{ \mathcal{Q}^k(u),\ a_k + b_k^T(u - u_k) \right\} & \text{else}, \end{cases} \tag{5.4b}$$

where

$$a_k := \frac{1}{m}\sum_{i=1}^{m} Q(u_k, w_i), \quad b_k \in \frac{1}{m}\sum_{i=1}^{m} \delta Q(u_k, w_i), \quad \mathcal{Q}^0 := 0.$$

From (5.2), it can be shown that b_k can be chosen as

$$b_k = -\frac{1}{m}\sum_{i=1}^{m}\left(\text{diag}(p^{\max})\mu_i^k - \text{diag}(p^{\min})\pi_i^k\right),$$

where μ_i^k and π_i^k are the Lagrange multipliers associated with the right and left inequality of constraint (5.2c), respectively, for $(u, w) = (u_k, w_i)$.

From the properties of subgradients, it can be shown that

$$\mathcal{Q}^k(u) \leq \frac{1}{m}\sum_{i=1}^{m} Q(u, w_i)$$

for all $u \in \mathcal{U} \cap \{0,1\}^{nT}$. Hence, $F(u_k) + \mathcal{Q}^k(u_k)$ gives a lower bound of the optimal objective value of problem (5.3). Moreover, since u_k is a feasible point, $F(u_k) + m^{-1}\sum_{i=1}^{m} Q(u_k, w_i)$ gives an upper bound. The gap between these lower and upper bounds in theory goes to zero as k increases, and hence it can be used as a stopping criteria for the algorithm [11]. Note that the gap approaching zero suggests that u_k is close to being a solution of (5.3), but not necessarily of (5.1).

In practice, in order to avoid dealing in with the non-differentiability of the functions F and $\mathcal{Q}^k$, step (5.4a) is carried out by solving instead the following equivalent optimization problem, which uses an *epigraph* representation of the functions:

$$\begin{aligned}
\underset{u,x,y,z}{\text{minimize}} \quad & \sum_{i=1}^{n}\sum_{t=1}^{T}\left(c_i^u y_{i,t} + c_i^d z_{i,t}\right) + \vartheta \\
\text{subject to} \quad & a_j + b_j^T(u - u_j) \leq \vartheta, \;\; j = 0, \ldots, k-1 \\
& u_{i,t} - u_{i,t-1} \leq y_{i,t}, \qquad i = 1, \ldots, n, \;\; t = 1, \ldots, T \\
& u_{i,t-1} - u_{i,t} \leq z_{i,t}, \qquad i = 1, \ldots, n, \;\; t = 1, \ldots, T \\
& 0 \leq y_{i,t}, \qquad\qquad\;\; i = 1, \ldots, n, \;\; t = 1, \ldots, T \\
& 0 \leq z_{i,t}, \qquad\qquad\;\; i = 1, \ldots, n, \;\; t = 1, \ldots, T \\
& u \in \mathcal{U} \cap \{0,1\}^{nT}.
\end{aligned}$$

5.4 SHA Algorithm

Motivated by problems arising in transportation, the authors in [21] proposed a *stochastic hybrid approximation* algorithm to solve two-stage stochastic optimization problems. For problems of the form of

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & F(u) + \mathbb{E}\left[Q(u, w)\right] \\ \text{subject to} \quad & u \in \mathcal{U}, \end{aligned} \tag{5.5}$$

where $F + Q(\,\cdot\,, w)$ are convex functions and $\mathcal{U}$ is a convex compact set, the procedure consist of producing iterates

$$u_k := \arg\min \left\{ F(u) + \mathcal{Q}^k(u) \;\middle|\; u \in \mathcal{U} \right\} \tag{5.6}$$

for each $k \in \mathbb{Z}_+$, where $\mathcal{Q}^k$ are deterministic approximations of $\mathbb{E}\left[Q(\,\cdot\,, w)\right]$ such that $F + \mathcal{Q}^k$ are strongly convex and differentiable. The initial approximation $\mathcal{Q}^0$ is given by the user. Then, at every iteration, the approximation is updated with a linear correction term based on the difference between a noisy subgradient of $\mathbb{E}\left[Q(\,\cdot\,, w)\right]$ and the gradient of the approximation at the current iterate. That is, for each $k \in \mathbb{Z}_+$,

$$\mathcal{Q}^{k+1}(u) = \mathcal{Q}^k(u) + \alpha_k\big(\xi_k - \nabla \mathcal{Q}^k(u_k)\big)^T u, \tag{5.7}$$

for all $u \in \mathcal{U}$, where $\xi_k \in \delta Q(u_k, w_k)$, w_k are independent samples of the random vector w, and α_k are step lengths that satisfy conditions that are common in SA algorithms [42]. As discussed in [21], the strengths of this algorithm are its ability to exploit a potentially accurate initial approximation $\mathcal{Q}^0$, and the fact the structure of the approximation is not changed by the linear correction terms.

5.4.1 AdaCE

In Chapters 2, 3, and 4, the SHA algorithm outlined above and extensions of it are applied to different versions of stochastic ED problems. Inspired by the fact that in power systems operations and planning, CE models are common, these models are considered for constructing initial approximations of the expected-cost functions. For example, for problem

(5.5), the CE model is given by

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & F(u) + Q(u, \mathbb{E}[w]) \\ \text{subject to} \quad & u \in \mathcal{U}, \end{aligned}$$

and hence the initial function approximation based on this model is given by $\mathcal{Q}^0 := Q(\cdot, \mathbb{E}[w])$. Using this type of initial approximations, promising performance of the resulting AdaCE algorithms is shown in the previous chapters on several instances and types of stochastic ED problems compared to some widely-used benchmark algorithms.

For the stochastic UC problem (5.1), the function Q is defined as the optimal objective value of problem (5.2), and hence it can be shown that the initial function approximation $\mathcal{Q}^0 = Q(\cdot, \mathbb{E}[w])$ for the AdaCE approach is a piece-wise quadratic function [46].

5.4.2 Applicability to Stochastic Unit Commitment

The authors in [21] show that the SHA algorithm is guaranteed to solve problems of the form of (5.5). However, for the stochastic UC problem (5.1), theoretical and practical issues arise due to the binary restrictions $u \in \{0,1\}^{nT}$.

The first issue has to do with the use of slope corrections (5.7). These *local* corrections aim to make the slope of the approximations $\mathcal{Q}^k$ closer to that of $\mathbb{E}[Q(\cdot, w)]$ at the current iterate. This allows the algorithm to improve its knowledge about neighboring feasible points around the iterates u_k. This knowledge gradually becomes more accurate since the distance between consecutive iterates decreases and hence the slope corrections accumulate. However, when binary restrictions exist such as in the stochastic UC problem, neighboring feasible points of the iterates are not necessarily close enough for local slope information to be helpful to assess their goodness. Moreover, the distance between consecutive iterates is no longer guaranteed to decrease, and hence slope corrections are no longer guaranteed to accumulate.

Figure 5.1 illustrates qualitatively the possible outcomes of the SHA algorithm on a simple one-dimensional deterministic problem with convex objective function f and binary restrictions. The exact objective function is shown in blue while the (convex) approximations constructed by the algorithm are shown in green. The two feasible points are represented with vertical dashed lines. Three possible outcomes are shown: the algorithm gets stuck in a sub-optimal point (left column), the algorithm cycles between two points

(one of which is sub-optimal) (middle column), and the algorithm finds an optimal point (right column). The progression of the algorithm for each of these outcomes is shown after (roughly) k_1, k_2, and k_3 iterations, where $k_1 < k_2 < k_3$. In the left column of Figure 5.1, after k_1 iterations, the slopes of the approximation and exact function are equal at the current point u_k and this point is optimal with respect to the approximation, but due to curvature discrepancies, it is sub-optimal with respect to the exact function. Hence, the algorithm remains stuck in this sub-optimal point in later iterations k_2 and k_3. In the middle column of Figure 5.1, after k_1 iterations, the slopes match at the current point u_k and this point is sub-optimal for both functions. Hence, u_k moves to the other point and after k_2 iterations the slopes match again at this other point. However, due to curvature discrepancies, the new point becomes sub-optimal with respect to the approximation and hence after k_3 iterations u_k is back at the previous point, resulting in a cycling behavior. Lastly, in the right column of Figure 5.1, after k_1 iterations, the slopes of the approximation and exact function match at the current point u_k and this point is sub-optimal for both functions. Hence, u_k moves to the other point and after k_2 iterations the slopes match again at this other point. This new point is now optimal for both functions and u_k remains there after k_3 iterations. These examples illustrate that even for the simplest of problems with binary restrictions, the SHA algorithm can get stuck, cycle, or find an optimal point. Unlike in the continuous case analyzed in [21], the resulting outcome when binary restrictions are present depends heavily on the quality of the initial function approximation.

The second issue has to do with the strong convexity requirement for the initial function approximation $F + \mathcal{Q}^0$. This property is required to ensure convergence in the continuous case [21], but it makes step (5.6) computationally heavy in the case with binary restrictions. In particular, it makes step (5.6) consist of solving a Mixed-Integer Quadratic Programming (MIQP) problem at best. This is clearly too computationally demanding for a task that needs to be executed in every iteration of the SHA algorithm. For this reason, and since theoretical guarantees are already lost due to the binary restrictions regardless of the properties of $F + \mathcal{Q}^0$, this initial function approximation should be limited to be a piece-wise linear function. This practical restriction also results in the initial approximation also being in general not necessarily differentiable, and hence requires using in general subgradients instead of gradients in the update (5.7).

For the specific case of the stochastic UC problem (5.1), the initial function approximation for the AdaCE approach is $\mathcal{Q}^0 = Q(\,\cdot\,, \mathbb{E}\,[w])$, which is a convex piece-wise quadratic

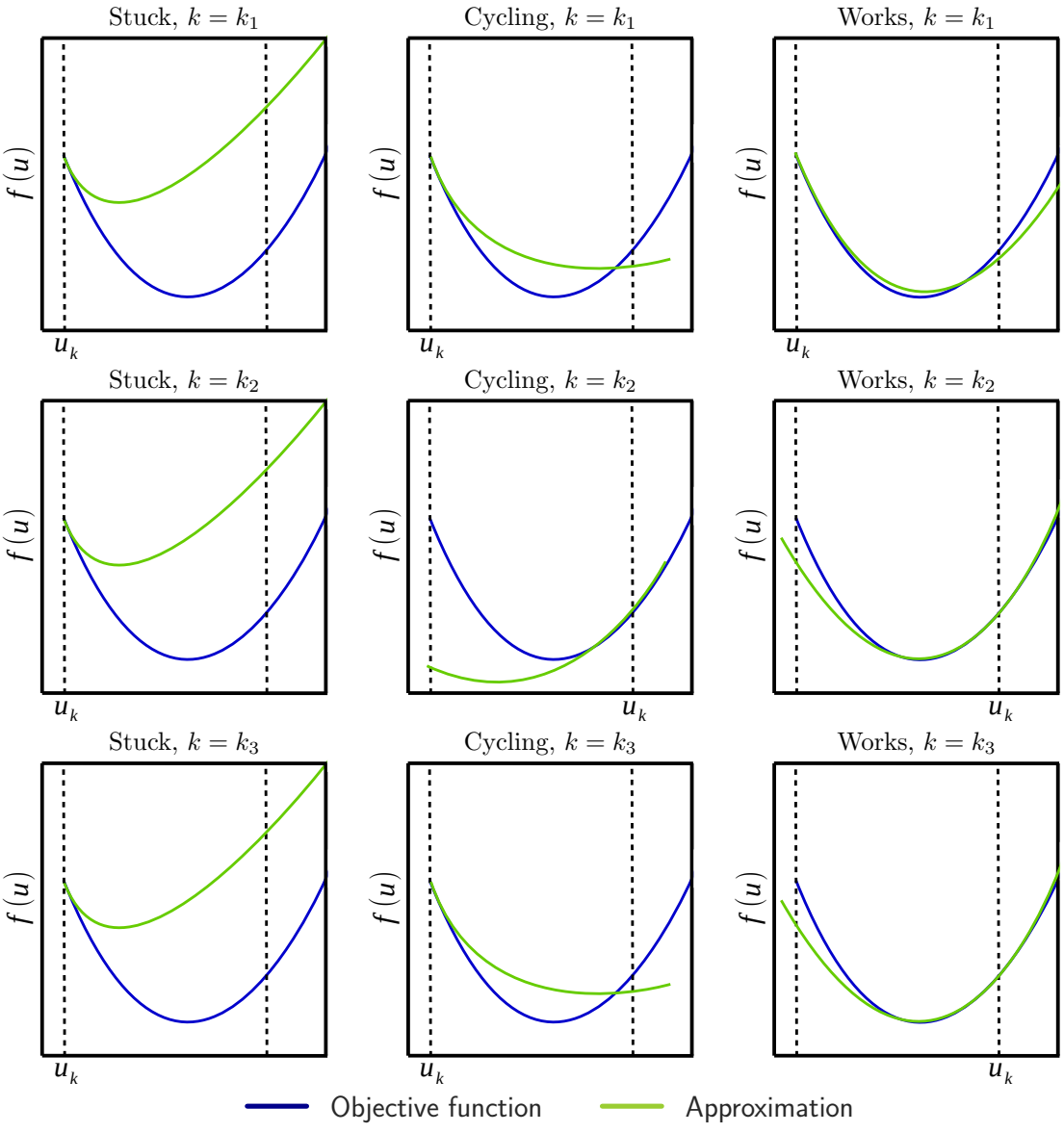

Figure 5.1: Possible outcomes of SHA with binary restrictions

function in general. In order to make it piece-wise linear, the approach proposed here is to approximate the separable convex quadratic generation cost function φ with a separable convex piece-wise linear function $\bar{\varphi}$. More specifically, the proposed function $\bar{\varphi}$ is given by

$$\bar{\varphi}(p) := \sum_{i=1}^{n} \sum_{t=1}^{T} \max_{1 \leq j \leq n_s} (\zeta_{i,j} + \beta_{i,j} p_{i,t}), \ \forall p, \tag{5.8}$$

where $n_s \in \mathbb{Z}_{++}$, $\zeta_{i,j}$ and $\beta_{i,j}$ are constant scalars, and $p_{i,t}$ denotes the output power of generator i during time t. Hence, the proposed alternative initial function approximation for the AdaCE algorithm is $\mathcal{Q}^0 = \bar{Q}(\cdot, \mathbb{E}[w])$, where $\bar{Q}(u, w)$ is the optimal objective value of the optimization problem obtained by replacing φ with $\bar{\varphi}$ in problem (5.2).

5.4.3 Noise Reduction

As discussed above, a piece-wise linear function $\mathcal{Q}^0$ based on the CE model is proposed for applying the AdaCE algorithm to the stochastic UC problem. With this, the resulting step (5.6) of the SHA algorithm consists of solving an MILP problem. Although less computationally demanding than solving MIQP or Mixed-Integer Non-Linear Programming (MINLP) problems, this still constitutes a severe computational bottleneck for the approach since this has to be done in every iteration. Furthermore, since only a single noisy subgradient $\xi_k \in \partial Q(u_k, w_k)$ is observed at each iteration, many iterations are required in order to average out the noise and get some accurate information about the slope of $\mathbb{E}[Q(\cdot, w)]$. Hence, to alleviate this, we propose in this chapter applying the mini-batch technique from Machine Learning [17], and replace (5.7) with the following update for each $k \in \mathbb{Z}_+$:

$$\mathcal{Q}^{k+1}(u) = \mathcal{Q}^k(u) + \alpha_k \left(\frac{1}{m} \sum_{i=1}^{m} \xi_{k,i} - \nabla \mathcal{Q}^k(u_k) \right)^T u, \tag{5.9}$$

for all $u \in \mathcal{U}$, where $m \in \mathbb{Z}_{++}$, $\xi_{k,i} \in \delta Q(u_k, w_{k,i})$, and $w_{k,i}$ are independent samples of the random vector w drawn at iteration k. As shown below, this averaging reduces the noise in the noisy subgradient of $\mathbb{E}[Q(\cdot, w)]$ used in iteration k. In addition, it can be done efficiently since $\xi_{k,i}$, $i \in \{1, \ldots, m\}$, can be computed in parallel.

Lemma 5.1. *The covariance matrix Σ_m of the noisy subgradient $m^{-1} \sum_{i=1}^{m} \xi_{k,i}$ used in (5.9) satisfies $\Sigma_m = \Sigma/m$, where Σ is the covariance of the noisy subgradient ξ_k used in (5.7).*

Proof. Letting $\mu := \mathbb{E}[\xi_k] = \mathbb{E}[\xi_{k,i}]$ and using the definition of covariance, it follows that

$$\begin{aligned}
\Sigma_m &= \mathbb{E}\left[\left(\frac{1}{m}\sum_{i=1}^{m}\xi_{k,i} - \frac{1}{m}\sum_{i=1}^{m}\mu\right)\left(\frac{1}{m}\sum_{j=1}^{m}\xi_{k,j} - \frac{1}{m}\sum_{j=1}^{m}\mu\right)^T\right] \\
&= \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\left(\xi_{k,i}-\mu\right)\frac{1}{m}\sum_{j=1}^{m}\left(\xi_{k,j}-\mu\right)^T\right] \\
&= \mathbb{E}\left[\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(\xi_{k,i}-\mu\right)\left(\xi_{k,j}-\mu\right)^T\right] \\
&= \frac{1}{m^2}\mathbb{E}\left[\sum_{i=1}^{m}\left(\xi_{k,i}-\mu\right)\left(\xi_{k,i}-\mu\right)^T + \sum_{i=1}^{m}\sum_{j\neq i}^{m}\left(\xi_{k,i}-\mu\right)\left(\xi_{k,j}-\mu\right)^T\right] \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\mathbb{E}\left[\left(\xi_{k,i}-\mu\right)\left(\xi_{k,i}-\mu\right)^T\right] + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j\neq i}^{m}\mathbb{E}\left[\left(\xi_{k,i}-\mu\right)\left(\xi_{k,j}-\mu\right)^T\right].
\end{aligned}$$

Since $\xi_{k,i}$ are independent, it holds that

$$\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j\neq i}^{m}\mathbb{E}\left[\left(\xi_{k,i}-\mu\right)\left(\xi_{k,j}-\mu\right)^T\right] = 0.$$

Therefore, it follows that

$$\Sigma_m = \frac{1}{m^2}\sum_{i=1}^{m}\mathbb{E}\left[\left(\xi_{k,i}-\mu\right)\left(\xi_{k,i}-\mu\right)^T\right] = \frac{1}{m^2}\sum_{i=1}^{m}\Sigma = \Sigma/m.$$

□

5.5 Numerical Experiments

This section describes the numerical experiments carried out to assess and compare the performance of the AdaCE algorithm and the algorithm based on SAA and Benders decomposition. The experimental results provide important information about the computational requirements of the algorithms, their efficiency, and the properties of the solutions obtained. Furthermore, they also provide information about the impact of the noise reduction technique described in Section 5.4.3 for the AdaCE algorithm on a network of realistic size.

The test cases used for the experiments consist of several instances of the stochastic UC problem described in Section 5.2 constructed from three power networks and five different load profiles and wind distributions. Details regarding the implementation of the algorithms as well as a validation of the algorithm based on Benders decomposition are also included in this section.

5.5.1 Implementation

The two algorithms were implemented in the Python programming language using different packages and frameworks. The modeling of the power networks was done with the Python wrapper of the library `PFNET` [1]. The optimization problems were modeled via `CVXPY` [24]. For solving the MILP problems, the solver `GUROBI` 6.5.1 was used [33]. For solving the QP problems, the second-order cone solver `ECOS` was used [25]. The spatial covariance matrix of the powers of renewable energy sources was constructed using routines available in `PFNET`.

5.5.2 Test Cases

Three different power networks were used to construct the test cases for the experiments. More specifically, the power networks used were a simple approximation of the American power system (`IEEE 14` [2]), a mid-scale network (`IEEE 300` [2]), and a accurate representation of the European high voltage network (`PEGASE 1354` [39]). A 24-hour time horizon with hourly resolution was considered, and all generators were assumed to be off before the start of the operation period. Table 5.1 shows important information of the cases, including name, number of buses in the network, number of branches, number of conventional generation units ("gens"), time horizon, number of binary variables in problem (5.1), and number of renewables energy sources ("res").

Table 5.1: Properties of test cases

name	buses	branches	gens	horizon	binary	res
Case A	14	20	5	24	235	5
Case B	300	411	69	24	3243	69
Case C	1354	1991	260	24	12220	260

[1] `http://pfnet-python.readthedocs.io`
[2] `http://www2.ee.washington.edu/research/pstca`

Due to the absence of inter-temporal limits and costs of generators, data from five different generation technologies was obtained from the `IEEE RTS 96` test case [32]. Table 5.2 shows this data, including technology name, generation cost function ($\varphi(p)$ in \$), minimum and maximum powers ($p^{\min}$ and $p^{\max}$ in MW), maximum ramping rates ($y^{\min}$ and $y^{\max}$ in MW/hour), minimum up and down times (T^u and T^d in hours), and start-up costs (c^u in \$). Shut-down costs (c^d) were assumed to be zero for every technology. The ramping rates used here are consistent with those described in [13]. Additionally, the load shedding cost γ was set to 10 times the highest marginal cost of generation. This ensured that load shedding was regarded by the algorithms as an emergency resource.

Table 5.2: Properties of generators by technology

name	$\varphi(p)$	$p^{\min}$	$p^{\max}$	$y^{\min}$	$y^{\max}$	T^d	T^u	c^u
nuclear	$0.02p^2 + 3.07p$	100	400	-280	280	24	168	40000
IGCC	$0.25p^2 + 10.6p$	54	155	-70	80	16	24	2058
CCGT	$0.14p^2 + 7.72p$	104	197	-310	310	3	4	230
OCGT	$2.26p^2 + 13.7p$	8	20	-90	100	1	2	46
coal	$0.11p^2 + 12.2p$	140	350	-140	140	5	8	12064

For each test case, each generator was assigned the properties associated with a technology (except $p^{\max}$ and $p^{\min}$) in order to get a distribution of capacity per technology that was approximately consistent with that of a target generation mix. This was done by assigning generators to technologies in order of decreasing $p^{\max}$ until the capacity share of that technology was reached, and then moving to the technology with the next largest capacity share. Tables 5.3 and 5.4 show the number of generating units of each technology and the resulting generation mix, respectively, for the three test cases.

Table 5.3: Number of generating units per technology

name	nuclear	IGCC	CCGT	OCGT	Coal
Case A	1	1	3	0	0
Case B	12	12	26	16	3
Case C	27	25	152	50	6

Five daily load profiles were obtained from a North American power marketing administration. Each load profile was normalized and used to modulate the (static) loads of each of the power networks. Figure 5.2 shows the resulting aggregate load profiles.

Table 5.4: Total capacity per generation technology in %

name	nuclear	IGCC	CCGT	OCGT	Coal
Case A	44	18	38	0	0
Case B	48	18	23	5	6
Case C	48	18	23	5	6

5.5.3 Renewables

For each of the test cases, renewable energy sources were added to the network at the buses with conventional generators. The capacity $r_i^{\max}$ for each source $i \in \{1, \ldots, n_r\}$, where n_r is the number of sources, was set to the peak total load divided by n_r. The base powers $\bar{r}_t$ were obtained by multiplying $0.5r^{\max}$ by a normalized daily wind power profile obtained from a North American power marketing administrator, yielding a high penetration setting. For constructing $\bar{\Sigma}_t$, $N = 5$ and $\rho = 0.1$ were used. Figure 5.3 shows sampled realizations (gray) and expected realizations (red) of aggregate powers from renewable energy sources for five different days for Case A.

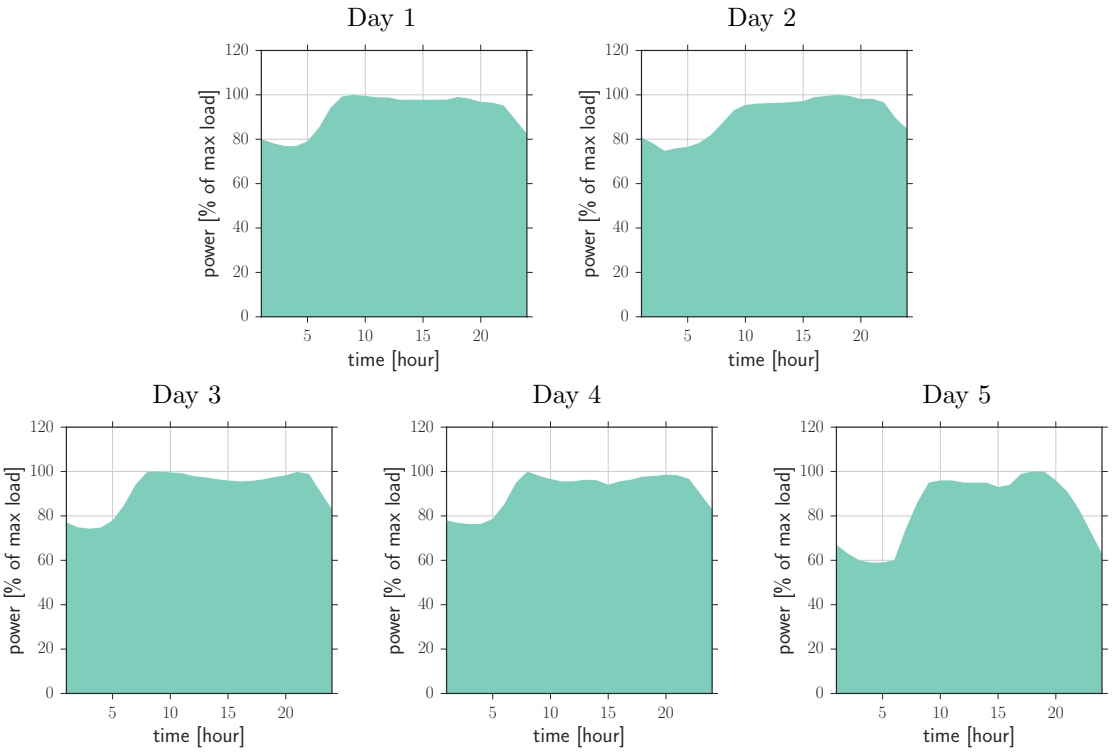

Figure 5.2: Load profiles

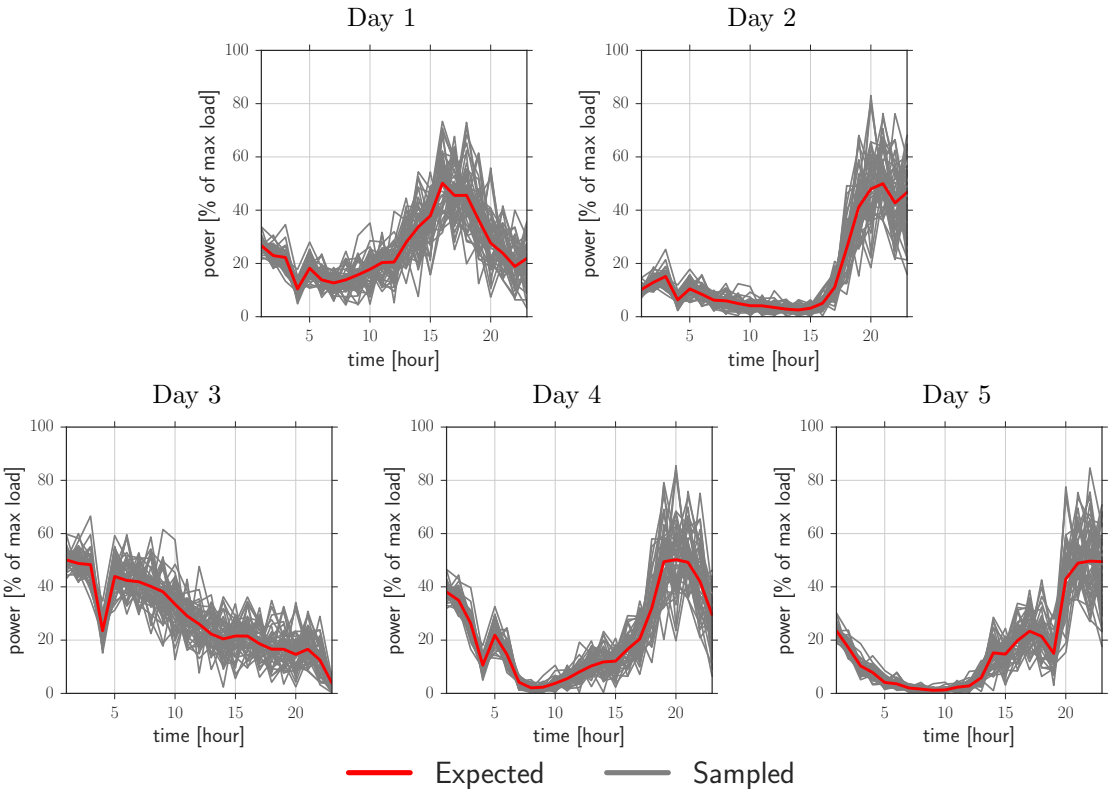

Figure 5.3: Powers from renewable energy sources for Case A

5.5.4 Benders Validation

In order to validate the implementation of the algorithm based on Benders decomposition, the algorithm was tested on Case A without renewable energy sources. Theoretically, as stated in Section 5.3.1, as the number of iterations increases, the "gap" between the upper and lower bounds produced by the algorithm of the optimal objective value of the SAA problem (5.3) approaches zero. Experimental results testing this property are shown in Figure 5.4.

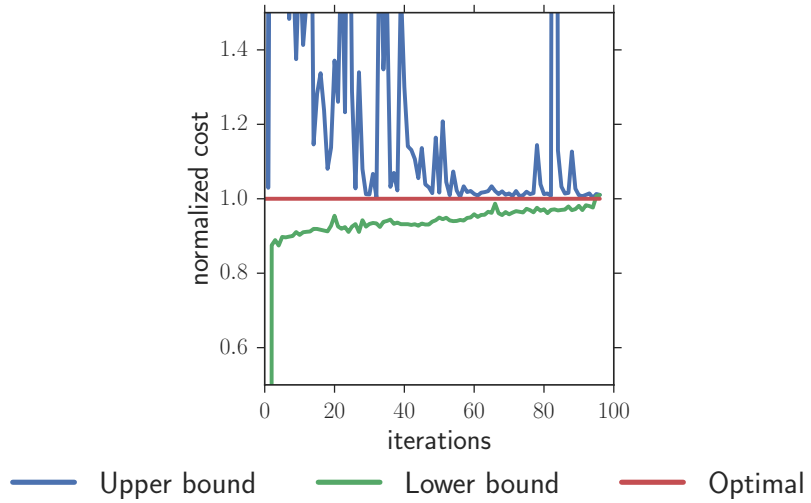

Figure 5.4: Benders validation on deterministic Case A

As Figure 5.4 shows, the lower bound gradually increases towards the optimal objective value. The upper bound has a general trend of decreasing towards the optimal objective value but it shows significant variations on this validation case. In particular, it can be seen that even after many iterations the upper bound has sudden "jumps" to very high values. The reason for these jumps is that the piece-wise linear approximation built by the algorithm fails to fully capture the properties of the exact recourse function, resulting in the algorithm making "mistakes". Some of these mistakes consist of choosing generator commitments that are poor with respect to the exact recourse function as they result in load shedding, which incurs a very high cost. It is interesting to note that, if the option of load shedding was not available, then the second-stage problem could be infeasible for such poor generator commitments, resulting in an operation cost of infinity.

5.5.5 Performance and Solutions

To evaluate the performance of the AdaCE algorithm and the benchmark algorithm based on Benders decomposition, or "Benders algorithm" for simplicity, the algorithms were tested on each test case with five different load profiles and wind distributions. As a termination criteria for the algorithms, a maximum number of iterations was used. Details for this are shown in Table 5.5. For Benders, a candidate solution u_k was evaluated every 25 iterations for Case A, and every 5 iterations for Case B. This algorithm was not applied to Case C since its performance was already not satisfactory on Cases A and B, which are significantly smaller. The number m of scenarios used for Benders was 300, and the evaluation of the recourse function for these scenarios at each iteration of the algorithm was done using 24 and 10 parallel processors for Cases A and B, respectively. For AdaCE, a candidate solution u_k was evaluated every 5 iterations for Case A, every 2 iterations for Case B, and every iteration for Case C. The number n_s of linear segments in the individual piece-wise linear generation cost functions in (5.8) was set to 3 for each test case. For Case C only, the noise reduction strategy was applied using 10 random samples at each iteration, and the evaluation of the recourse function for these samples was done using 10 parallel processors. For evaluating the candidates solutions from the algorithms, a set of 1000 new independent samples of the random powers from renewable energy sources was used. That is, the expected first stage cost associated with a candidate solution u_k was approximated with $F(u_k) + \frac{1}{1000}\sum_{i=1}^{1000} Q(u_k, w_i)$. The results obtained are shown and discussed in the subsections below. The computer system used for the experiments was the ETH Euler cluster, which is equipped with computing nodes having Intel Xeon E5-2697 processor and running CentOS 6.8.

Table 5.5: Maximum number of iterations

name	Benders	AdaCE
Case A	400	100
Case B	100	30
Case C	-	10

5.5.5.1 Case A

Figure 5.5 shows the performance of the algorithms on Case A. The expected costs shown are relative to the cost obtained with the solution of the CE problem for the corresponding

day (dashed line). From the figure, several observations can be made: First, the AdaCE algorithm is able to exploit user-provided knowledge of the problem in the form of an initial approximation of the expected resource function, and achieve cost reductions ranging from 0.5% (Day 3) to 2.8% (Day 1) relatively fast. This also shows that the CE solution is a relatively good solution for the original problem under the given conditions. On the other hand, the Benders algorithm struggles to find a candidate solution that is better than the CE solution on each day, with its best performance being a cost reduction of -0.5% achieved in Day 5. This outcome may be attributed to the fact that in Problem (5.1), the expected recourse function is a fairly complex function that plays a critical role in the determination of the total cost, and the algorithm requires many cutting planes (and hence iterations) to form an adequate approximation. Moreover, the algorithm only "knows" about the 300 scenarios, and these may not adequately represent the uncertainty.

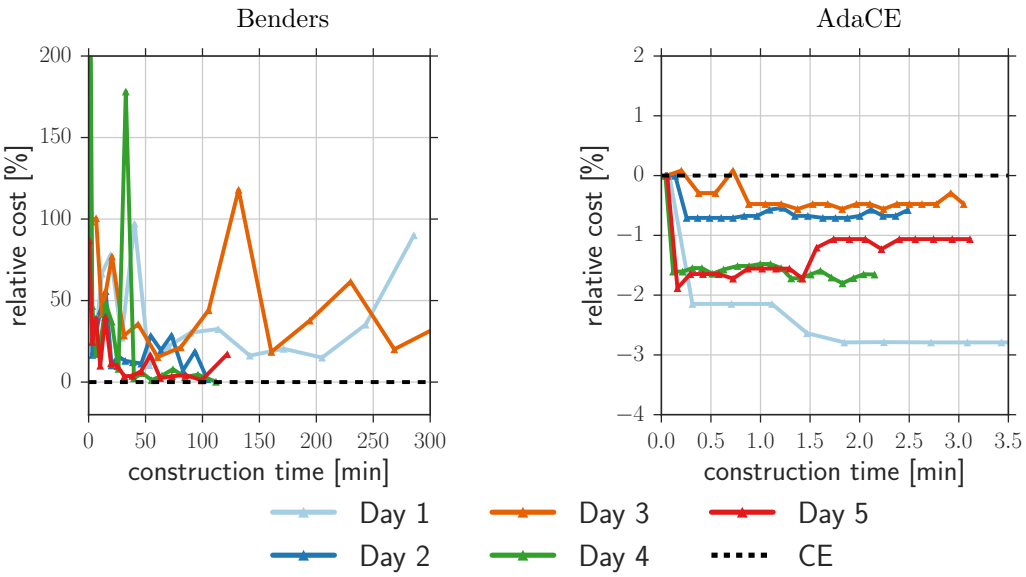

Figure 5.5: Performance of algorithms on Case A

Figure 5.6 shows the expected aggregate power generation profiles associated with the CE solution and the last iterate produced by the AdaCE algorithm on Case A, Day 5. This is the day for which AdaCE achieves a cost reduction of 1% over the CE solution. It can be seen that the CE solution, which is obtained by ignoring the uncertainty, fails to foresee that the scheduled generating units do not supply the demand under some possible scenarios (hour 5), resulting in higher expected operational cost. On the other hand, the

AdaCE algorithm is able to account for this and avoid load shedding. The generation mixes associated with the CE and AdaCE solutions are shown in Figure 5.7.

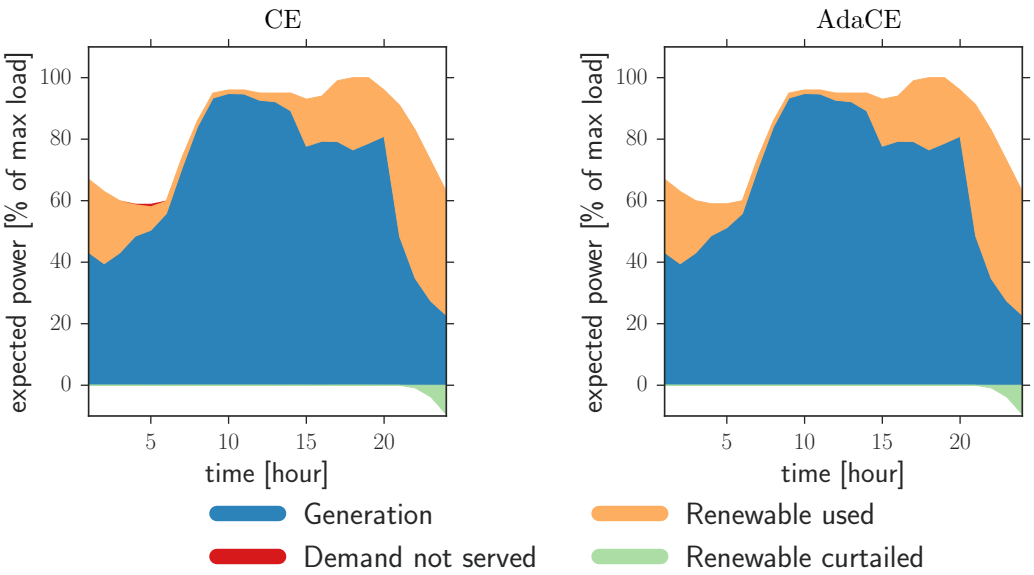

Figure 5.6: Expected generation profile for Case A, Day 5

5.5.5.2 Case B

Figure 5.8 shows the performance of the algorithms on Case B. As the figure shows, the AdaCE algorithm is able to find a better solution than the CE solution for four out of the five evaluated days in around 12 minutes. The cost reductions range from 0% to around 0.55%, which are less than the results obtained on Case A. On the other hand, the Benders algorithm fails to obtain a candidate solution that is better or even close in terms of cost to the CE solution in the allowed number of iterations. As discussed earlier, this could be attributed to the complexity of the expected recourse function, which may require a large number of cutting planes to approximate, and to a potential inadequacy of the 300 scenarios for representing the uncertainty.

Figure 5.9 shows the expected aggregate generation profiles associated with the CE solution and the last iterate produced by the AdaCE algorithm on Case B, Day 1. As the figure shows, neither of these candidate solutions results in load shedding. Furthermore, as in Case A, the scheduled generation is quite flexible and is able to accommodate a high utilization of renewable energy. The key differences between these two candidate solutions

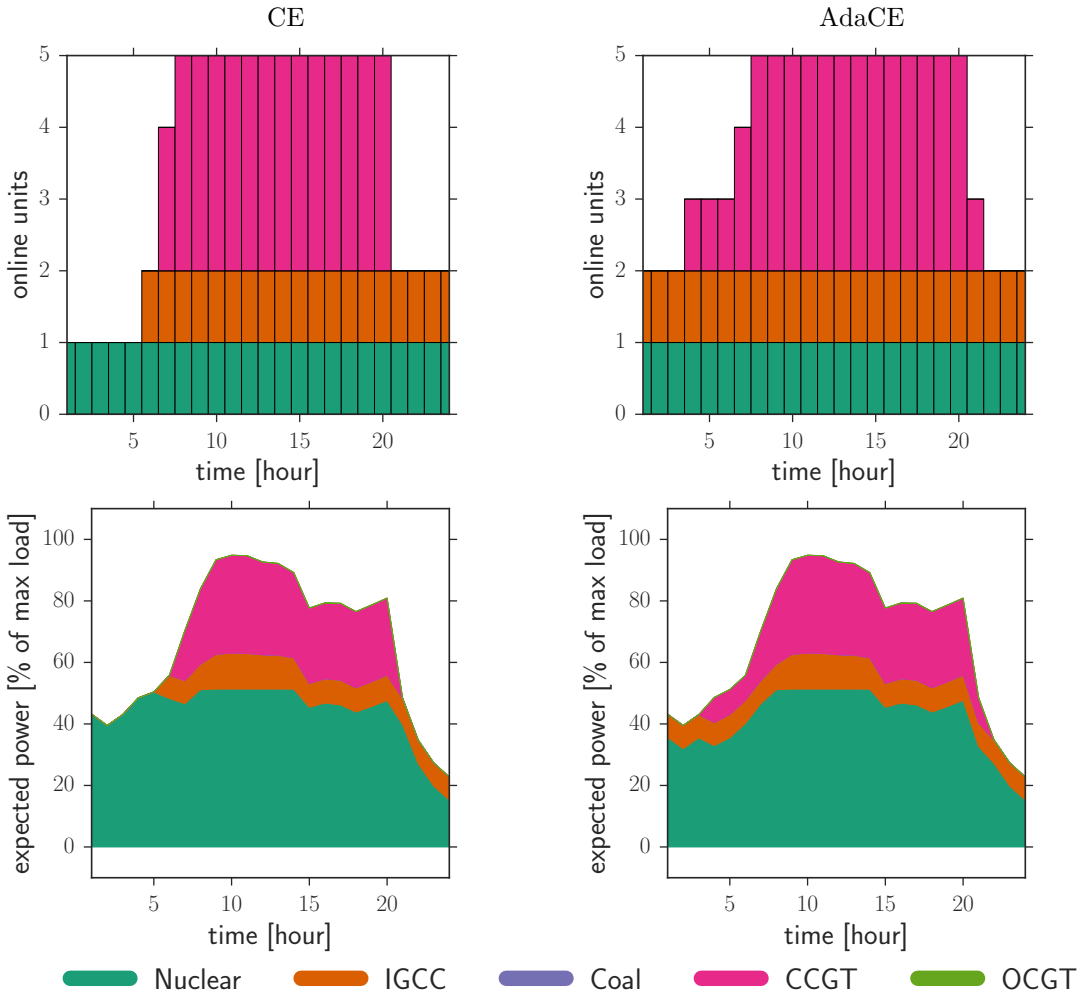

Figure 5.7: Generation mix for Case A, Day 5

becomes clear when the production by technology is considered. This is shown in Figure 5.10. As this and Figure 5.3 show, around hour 10, when wind production is low, and around hour 20, when wind uncertainty is high and there are sudden drops, the AdaCE solution has more flexible generating units scheduled and hence is able to accommodate these changes more efficiently.

5.5.5.3 Case C

As stated before, the Benders algorithm was not tested on Case C due to its poor performance on the other two test cases, which are much smaller. Instead, the impact of using the

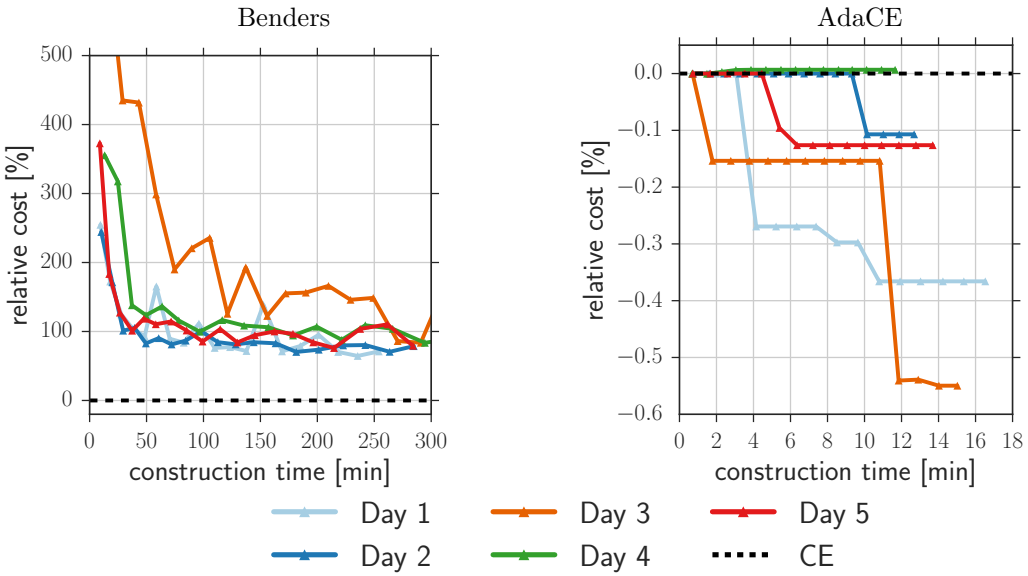

Figure 5.8: Performance of algorithms on Case B

noise reduction strategy in the AdaCE algorithm was analyzed on this case. Figure 5.11 shows the performance of the AdaCE algorithm with and without noise reduction (NR) on this case under five different load profiles and wind distributions. It can be seen that the general trend is that the noise reduction strategy eventually makes the AdaCE algorithm find better commitment schedules than without it. Another important observation is that, unlike on Cases A and B, not all candidate solutions obtained with the AdaCE algorithm have a equal or lower cost compared with the CE solution, especially without noise reduction.

5.6 Conclusions

In this chapter, the applicability and performance of the AdaCE algorithm for solving the stochastic UC problem was explored. To do this, the problem was first modeled as a stochastic two-stage mixed-integer optimization problem, which captured the essential properties that make the problem computationally challenging. Then, theoretical and practical issues associated with the applicability of the AdaCE algorithm for solving this problem were investigated. In particular, it was determined that the binary restrictions present in the UC problem could make the algorithm cycle or get stuck in a sub-optimal point. Through some illustrative qualitative examples, it was shown that the outcome depends heavily on

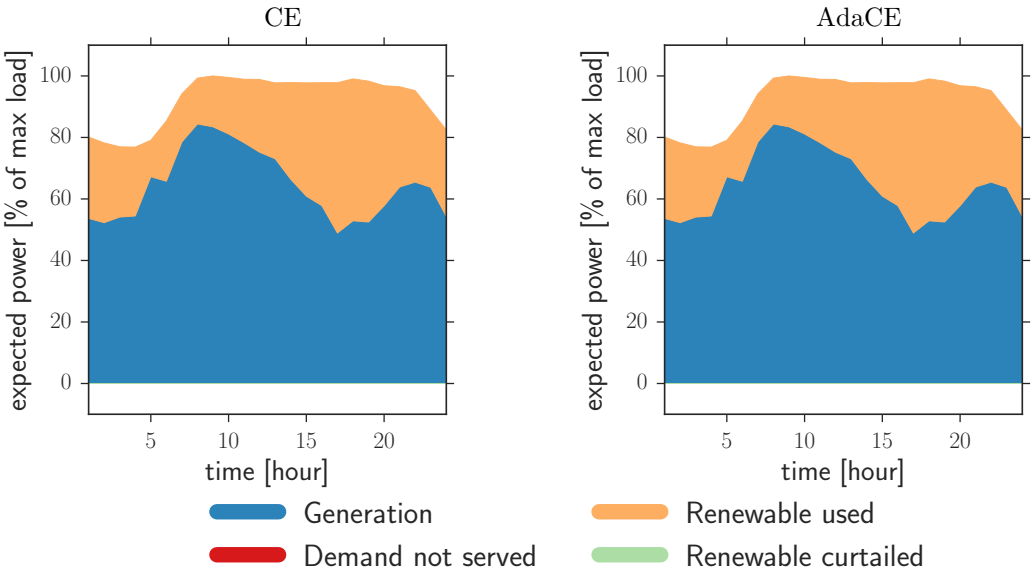

Figure 5.9: Expected generation profile for Case B, Day 1

the quality of the initial approximation of the expected second-stage cost function provided to the algorithm. Furthermore, the computational challenges of the algorithm were investigated. It was concluded that the initial approximations of the expected second-stage cost should be limited to be piece-wise linear, deviating from the requirements that make the algorithm work for problems with continuous variables. To try to alleviate the computational burden of solving MILP problems during potentially many iterations of the algorithm, the use of noise-reduction techniques was proposed. The performance of the algorithm was investigated along with that of a widely-used algorithm consisting of SAA combined with Benders decomposition on test cases constructed from three different power networks and five different load profiles and wind power distributions. The results obtained showed that despite a lack of theoretical guarantees, the AdaCE algorithm could find commitment schedules that are better than ones obtained with deterministic models and with the Benders-based algorithm.

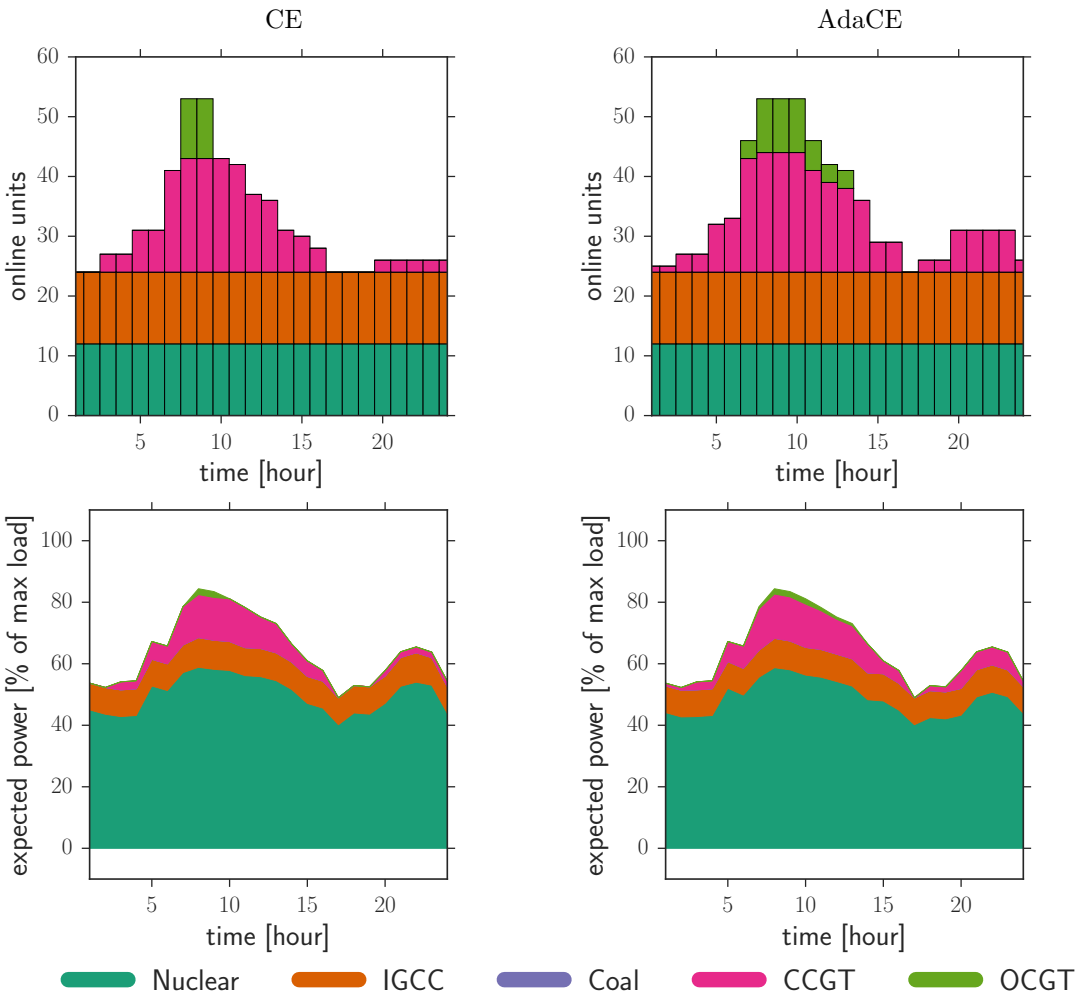

Figure 5.10: Generation mix for Case B, Day 1

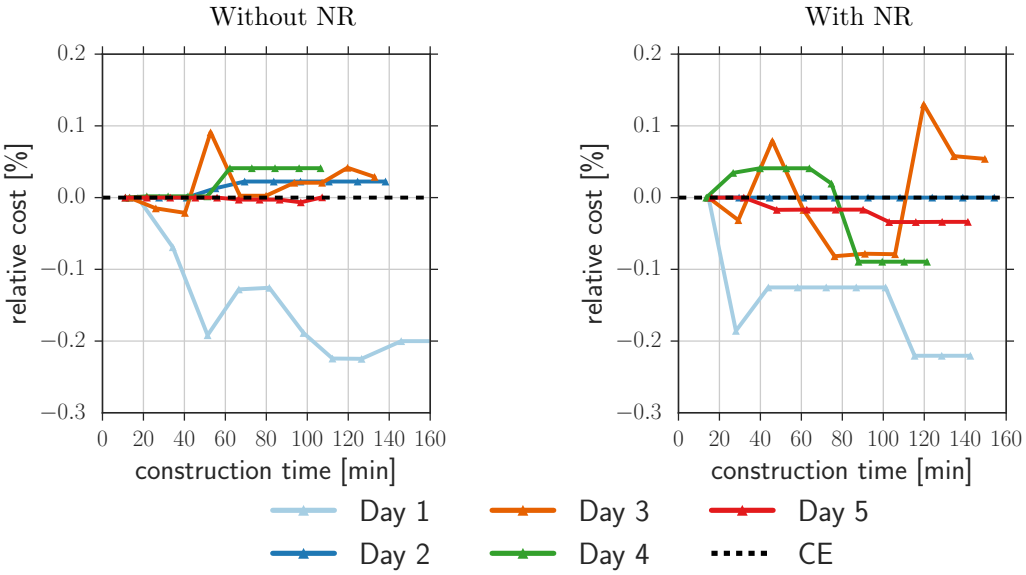

Figure 5.11: Performance of AdaCE on Case C

Chapter 6

Conclusions

6.1 Summary

In this work, *Adaptive Certainty-Equivalent* methods have been explored for solving generation scheduling problems in power systems with high penetration of renewable energy, such as wind and solar. This is an important application since generation schedules that can be computed in practice, properly take into account high levels of uncertainty, and exploit the limited flexibility of low-cost generation technologies, are crucial. Without these, economic viability of clean energy sources such as wind and solar could not be achieved for supplying a large part of the demand, and also system reliability could be jeopardized. The methods explored consisted of a combination of *Stochastic Hybrid Approximation* procedures with initial deterministic approximations of expected-value functions based on *Certainty-Equivalent* models. In Chapter 2, this approach was tested on a simple two-stage stochastic economic dispatch problem, and an interesting connection with stochastic proximal gradient algorithms was found. In Chapter 3, this approach was extended to handle expected-value constraints in order to determine generator dispatches that guaranteed a desired bound on the risk of high system balancing cost. In Chapter 4, the approach was extended to handle the multi-stage nature of the generation scheduling process. Motivated by the promising results obtained with the proposed methodology in these chapters, its applicability and performance was analyzed in Chapter 5 for solving generation scheduling problems involving unit commitment decisions, which are binary. The results obtained were also promising for these problems, but theoretical guarantees were lost.

6.2 Next Research Directions

Several interesting research directions can be considered for extending this work:

- As described in Chapter 2, the stochastic hybrid approximation procedure was found to be equivalent to a stochastic proximal gradient algorithm equipped with a non-Euclidean metric on the two-stage stochastic economic dispatch problem considered. An interesting direction is to determine whether a similar connection can be established on more general stochastic optimization problems, perhaps by considering a more general notion of distance beyond that induced by a quadratic norm. Other interesting connections to be found and formalized could be those with gradient boosting, and approximate dynamic programming.

- All the power network models considered in this work have been based on the DC approximation. This approximation does not adequately capture important properties of the network such as voltage magnitude deviations and reactive power flow. Hence, AC models need to be considered, but these are in general non-convex. An interesting direction is therefore to explore the applicability of the methods considered here on non-convex stochastic optimization problems. Interesting questions include whether approximations of expected-value functions can be convex, and whether theoretical guarantees such as local optimality can be obtained. If convex approximations can be used, then an interesting idea consists of trying to leverage work on convex relaxations of optimal power flow problems, which is an active area of research, to construct these approximations.

- The stochastic hybrid approximation procedures used and extended in this work have been based on updating deterministic approximations of expected-value functions using noisy slope corrections. An interesting idea consists of exploring richer types of updates, possibly involving curvature or structure. These are particularly interesting in the context of unit commitment since in Chapter 5 it was determined that slope corrections are not enough to obtain any theoretical guarantees on this problem.

- For ensuring the existence of a subsequence of the iterates produced by the proposed primal-dual and parameterized extensions of the stochastic hybrid approximation algorithm that converges to a solution, "bounded drift" assumptions were needed. These assumptions required that certain sums of uncontrolled side effects of the algorithms

remained bounded above. Only an intuitive justification was provided in this work suggesting why these conditions are likely to hold when the proposed algorithms are applied to practical problems. An interesting and needed research task consists of finding sufficient conditions that ensure these "bounded drifts", and trying to find examples of problems on which the algorithms have unbounded drift.

- Last but not least, the CE-based initial deterministic approximations of expected cost-to-go functions used for the multi-stage problems considered in Chapter 4 are not suitable for problems with a large number of stages. As stated in that chapter, "depth-limited" CE-based approximations that ignore costs-to-go after a certain number of stages could potentially lead to accurate approximations that keep the computational requirements of the algorithm manageable on problems with a large number of stages.

Appendix A

Convergence Analysis of SHA

Almost-sure convergence of the iterates produced by the SHA algorithm described in Section 2.3.3 to an optimal point of (2.1) is shown. The proof follows that of Cheung & Powell [21] and has been included here for completeness. Aside from the problem-specific assumptions stated in Section 2.2, the following additional assumptions are made:

Assumption A.1 (Initial Function Approximation)**.** *The initial objective function approximation $\mathcal{F}_0$ is strongly convex and continuously differentiable in $\mathcal{X}$.*

Assumption A.2 (Sampled Subgradients)**.** *The subgradients ξ_k are uniformly bounded for all $k \in \mathbb{Z}_+$.*

Assumption A.3 (Step Lengths)**.** *The step lengths α_k lie inside the open interval $(0, 1)$, and satisfy $\sum_{k=0}^{\infty} \alpha_k = \infty$ almost surely and $\sum_{k=0}^{\infty} \mathbb{E}\left[\alpha_k^2\right] < \infty$.*

The results are now derived through a sequence of lemmas, corollaries and a theorem.

Lemma A.1. *The functions $\mathcal{F}_k$ and their gradients are uniformly Lipschitz for all $k \in \mathbb{Z}_+$ and hence also uniformly bounded in $\mathcal{X}$.*

Proof. From (2.7), the function $\mathcal{F}_k$ can be expressed as

$$\mathcal{F}_k(x) = \mathcal{F}_0(x) + r_k^T x, \ \forall x \in \mathcal{X}, \tag{A.1}$$

for $k \in \mathbb{Z}_+$, where r_k is a vector such that $r_0 = 0$. It follows that for all $k \in \mathbb{Z}_+$ and $x \in \mathcal{X}$,

$$\begin{aligned}\mathcal{F}_{k+1}(x) &= \mathcal{F}_k(x) + \alpha_k\big(\xi_k - \nabla\mathcal{F}_k(x_k)\big)^T x \\ &= \mathcal{F}_0(x) + r_k^T x + \alpha_k\big(\xi_k - \nabla\mathcal{F}_0(x_k) - r_k\big)^T x \\ &= \mathcal{F}_0(x) + \big((1-\alpha_k)r_k + \alpha_k\big(\xi_k - \nabla\mathcal{F}_0(x_k)\big)\big)^T x.\end{aligned}$$

Hence

$$r_{k+1} = (1-\alpha_k)r_k + \alpha_k\big(\xi_k - \nabla\mathcal{F}_0(x_k)\big).$$

From Assumptions A.1 and A.2, there is some constant M_1 such that

$$\|\xi_k - \nabla\mathcal{F}_0(x_k)\|_2 \le M_1$$

for all $k \in \mathbb{Z}_+$. Hence, Assumption A.3 gives that $\|r_k\|_2 \le M_1$ implies

$$\begin{aligned}\|r_{k+1}\|_2 &= \big\|(1-\alpha_k)r_k + \alpha_k\big(\xi_k - \nabla\mathcal{F}_0(x_k)\big)\big\|_2 \\ &\le (1-\alpha_k)\|r_k\|_2 + \alpha_k\|\xi_k - \nabla\mathcal{F}_0(x_k)\|_2 \\ &\le (1-\alpha_k)M_1 + \alpha_k M_1 \\ &= M_1.\end{aligned}$$

Since $r_0 = 0$, it holds by induction that $\|r_k\|_2 \le M_1$ for all $k \in \mathbb{Z}_+$. It follows that for any x and $y \in \mathcal{X}$ and $k \in \mathbb{Z}_+$,

$$\begin{aligned}\|\mathcal{F}_k(x) - \mathcal{F}_k(y)\|_2 &= \big\|\mathcal{F}_0(x) + r_k^T x - \mathcal{F}_0(y) - r_k^T y\big\|_2 \\ &\le \|r_k\|_2\|x-y\|_2 + \|\mathcal{F}_0(x) - \mathcal{F}_0(y)\|_2 \\ &\le (M_1 + M_2)\,\|x-y\|_2,\end{aligned}$$

where $M_2 > 0$ exists because Assumption A.1 implies that $\mathcal{F}_0$ is Lipschitz in the compact set $\mathcal{X}$.

From (A.1), the gradient of $\mathcal{F}_k$ is given by

$$\nabla\mathcal{F}_k(x) = \nabla\mathcal{F}_0(x) + r_k,\ \forall x \in \mathcal{X},$$

for each $k \in \mathbb{Z}_+$. Assumption A.1, the compactness of $\mathcal{X}$, and $\|r_k\|_2 \le M_1$ imply that the

functions $\nabla\mathcal{F}_k$, $k \in \mathbb{Z}_+$, are uniformly Lipschitz in $\mathcal{X}$. □

Lemma A.2. *The functions $\mathcal{F}_k$, $k \in \mathbb{Z}_+$, are uniformly strongly convex, and hence there is a constant $C > 0$ such that*

$$\big(\nabla\mathcal{F}_k(y) - \nabla\mathcal{F}_k(x)\big)^T(y-x) \geq C\|y-x\|_2^2 \tag{A.2}$$

for all y and $x \in \mathcal{X}$, and $k \in \mathbb{Z}_+$.

Proof. From (2.7) and Lemma A.1, $\mathcal{F}_k$ can be expressed as

$$\mathcal{F}_k(x) = \mathcal{F}_0(x) + r_k^T x,\ \forall x \in \mathcal{X},$$

for $k \in \mathbb{Z}_+$, where r_k are uniformly bounded vectors with $r_0 = 0$. Assumption A.1 gives that $\mathcal{F}_0$ is strongly convex, and therefore that there is a constant $C > 0$ such that

$$\mathcal{F}_k(y) - \mathcal{F}_k(x) \geq \nabla\mathcal{F}_k(x)^T(y-x) + \frac{C}{2}\|y-x\|_2^2 \tag{A.3}$$

for all y and $x \in \mathcal{X}$, and $k \in \mathbb{Z}_+$. The inequality (A.2) follows by adding (A.3) to the inequality obtained from (A.3) by interchanging x and y. □

Lemma A.3. *There exists an $M > 0$ such that $\|x_{k+1} - x_k\|_2 \leq \alpha_k M$ for all $k \in \mathbb{Z}_+$. For convenience, we say that $\|x_{k+1} - x_k\|_2$ is $\mathcal{O}(\alpha_k)$.*

Proof. Equation (2.6) implies that

$$\nabla\mathcal{F}_k(x_k)^T(x - x_k) \geq 0,\ \forall x \in \mathcal{X} \tag{A.4}$$

$$\nabla\mathcal{F}_{k+1}(x_{k+1})^T(x - x_{k+1}) \geq 0,\ \forall x \in \mathcal{X}, \tag{A.5}$$

for all $k \in \mathbb{Z}_+$. In particular, (A.4) holds with $x = x_{k+1}$, and (A.5) with $x = x_k$. Hence, combining these gives

$$\big(\nabla\mathcal{F}_{k+1}(x_{k+1}) - \nabla\mathcal{F}_k(x_k)\big)^T\delta_k \leq 0$$

for all $k \in \mathbb{Z}_+$, where $\delta_k := x_{k+1} - x_k$. Using (2.7) gives

$$\big(\nabla\mathcal{F}_k(x_{k+1}) - \nabla\mathcal{F}_k(x_k)\big)^T\delta_k \leq -\alpha_k\big(\xi_k - \nabla\mathcal{F}_k(x_k)\big)^T\delta_k$$

for all $k \in \mathbb{Z}_+$. Then, it follows from this, Assumption A.2, Lemmas A.1 and A.2 (inequality (A.2)) that there exist $C > 0$ and $M_1 > 0$ such that

$$\begin{aligned} C\|\delta_k\|_2^2 &\leq \big(\nabla \mathcal{F}_k(x_{k+1}) - \nabla \mathcal{F}_k(x_k)\big)^T \delta_k \\ &\leq -\alpha_k \big(\xi_k - \nabla \mathcal{F}_k(x_k)\big)^T \delta_k \\ &\leq \alpha_k M_1 \|\delta_k\|_2 \end{aligned}$$

for all $k \in \mathbb{Z}_+$, or equivalently, that

$$\|\delta_k\|_2 \leq \alpha_k M_1 / C$$

for all $k \in \mathbb{Z}_+$. □

Lemma A.4. *The sequence $\{T_k\}_{k\in\mathbb{Z}_+}$, defined by*

$$T_k := \mathcal{F}_k(x^*) - \mathcal{F}_k(x_k),$$

where x^* is an optimal point of (2.1), convergences almost-surely to a finite random variable.

Proof. From the definition of T_k and (2.7), it follows that

$$\begin{aligned} T_{k+1} - T_k &= \mathcal{F}_{k+1}(x^*) - \mathcal{F}_{k+1}(x_{k+1}) - \mathcal{F}_k(x^*) + \mathcal{F}_k(x_k) \\ &= \mathcal{F}_k(x^*) - \mathcal{F}_k(x_{k+1}) - \mathcal{F}_k(x^*) + \mathcal{F}_k(x_k) + \alpha_k \big(\xi_k - \nabla \mathcal{F}_k(x_k)\big)^T (x^* - x_{k+1}) \\ &= \mathcal{F}_k(x_k) - \mathcal{F}_k(x_{k+1}) + \alpha_k \big(\xi_k - \nabla \mathcal{F}_k(x_k)\big)^T (x^* - x_{k+1}) \end{aligned}$$

for all $k \in \mathbb{Z}_+$. From this, (2.6), which fives $\mathcal{F}_k(x_k) \leq \mathcal{F}_k(x_{k+1})$, Assumption A.2, and Lemmas A.1 and A.3, it holds that there exists some $M > 0$ such that

$$T_{k+1} - T_k \leq \alpha_k \xi_k^T (x^* - x_k) - \alpha_k \nabla \mathcal{F}_k(x_k)^T (x^* - x_k) + \alpha_k^2 M$$

for all $k \in \mathbb{Z}_+$. This, (2.6), which gives $\nabla \mathcal{F}_k(x_k)^T (x^* - x_k) \geq 0$, and the fact that $\xi_k \in \partial F(x_k, w_k)$, then gives that

$$T_{k+1} \leq T_k + \alpha_k \big(F(x^*, w_k) - F(x_k, w_k)\big) + \alpha_k^2 M \tag{A.6}$$

for all $k \in \mathbb{Z}_+$. Letting $\mathcal{W}_k := \{w_0, \ldots, w_{k-1}\}$ for $k > 0$ and $W_0 := \emptyset$, it follows from (A.6)

that

$$\mathbb{E}\left[\, T_{k+1} \,|\, W_k \,\right] \leq T_k + \alpha_k\big(\mathcal{F}(x^*) - \mathcal{F}(x_k)\big) + \alpha_k^2 M$$

for all $k \in \mathbb{Z}_+$. It then follows from the optimality of x^* with respect to $\mathcal{F}$ that

$$\mathbb{E}\left[\, T_{k+1} \,|\, W_{k-1} \,\right] \leq T_k + \alpha_k^2 M$$

for all $k \in \mathbb{Z}_+$. It holds from this and the non-negativity of T_k, which follows from (2.6), that the sequence $\{S_k\}_{k\in\mathbb{Z}_+}$ defined by $S_k := T_k + \sum_{i=k}^{\infty} \alpha_i^2 M$ is a positive supermartingale. Hence, it converges to a finite random variable T^* almost surely. From Assumption A.3, the term $\sum_{i=k}^{\infty} \alpha_i^2 M$ vanishes almost-surely as $k \to \infty$, and hence T_k also converges to T^* almost surely as as $k \to \infty$. □

Lemma A.5. *The sequence $\{T_k\}_{k\in\mathbb{Z}_+}$ defined in Lemma A.4 converges to zero almost surely.*

Proof. Summing (A.6) over k from 0 to K gives

$$\sum_{k=0}^{K} \alpha_k\big(F(x_k, w_k) - F(x^*, w_k)\big) \leq T_0 - T_{K+1} + \sum_{k=0}^{K} \alpha_k^2 M.$$

Taking expectation on both sides gives

$$\sum_{k=0}^{K} \mathbb{E}\left[\alpha_k\big(\mathcal{F}(x_k) - \mathcal{F}(x^*)\big)\right] \leq \mathbb{E}\left[T_0 - T_{K+1}\right] + \sum_{k=0}^{K} \mathbb{E}\left[\alpha_k^2\right] M.$$

Taking $K \to \infty$ and using the finiteness of T_k and Assumption A.3 gives that

$$\sum_{k=0}^{\infty} \mathbb{E}\left[\alpha_k\big(\mathcal{F}(x_k) - \mathcal{F}(x^*)\big)\right] < \infty.$$

Assumption A.3 and $\mathcal{F}(x_k) - \mathcal{F}(x^*) \geq 0$ then give that there exists a subsequence $\{x_{n_k}\}_{k\in\mathbb{Z}_+}$ such that $\mathcal{F}(x_{n_k}) - \mathcal{F}(x^*) \to 0$ almost surely as $k \to \infty$. Since the sequence $\{x_{n_k}\}_{k\in\mathbb{Z}_+}$ lies in the compact set $\mathcal{X}$, it can be assumed without loss of generality that $x_{n_k} \to x^*$ almost surely as $k \to \infty$, where x^* is an optimal point of (2.1). It follows from this and Lemma A.1 that $T_{n_k} = \mathcal{F}_{n_k}(x^*) - \mathcal{F}_{n_k}(x_{n_k}) \to 0$ almost surely as $k \to \infty$. This and Lemma A.4 then imply that $\{T_k\}_{k\in\mathbb{Z}_+}$ converges zero almost surely. □

Theorem A.1. *The sequence of iterates $\{x_k\}_{k\in\mathbb{Z}_+}$ produced by the SHA algorithm described in Section 2.3.3 converges almost surely to a solution of problem* (2.1).

Proof. From Lemma A.2, in particular, inequality (A.3), and (2.6), it follows that there exists a $C > 0$ such that

$$\begin{aligned}\frac{C}{2}\|x^* - x_k\|_2^2 &\leq \mathcal{F}_k(x^*) - \mathcal{F}_k(x_k) - \nabla\mathcal{F}_k(x_k)^T(x^* - x_k)\\ &\leq \mathcal{F}_k(x^*) - \mathcal{F}_k(x_k)\\ &= T_k\end{aligned}$$

for all $k \in \mathbb{Z}_+$. Lemma A.5 then gives that $x_k \to x^*$ almost surely as $k \to \infty$. □

Appendix B

Convergence Analysis of Primal-Dual SHA

Almost-sure convergence of a subsequence of the primal iterates produced by algorithm (3.6) to an optimal point of problem (3.1) is shown under certain assumptions. The assumptions include the problem-specific ones stated in Section 3.2 as well as the additional ones stated below. The proof uses approximate primal and dual solutions x_k^σ and λ_k^σ, respectively, of problem (3.1) that solve the regularized problem

$$\begin{aligned} \underset{x\in\mathcal{X}}{\text{minimize}} \quad & \mathcal{F}(x) + \frac{\sigma}{2}\|x - x_k\|_2^2 \\ \text{subject to} \quad & \mathcal{G}(x) \leq 0 \end{aligned} \tag{B.1}$$

for $\sigma > 0$ and $k \in \mathbb{Z}_+$. The proof also uses the approximate Lagrangian defined by

$$\mathcal{L}_k(x, \lambda) := \mathcal{F}_k(x) + \lambda^T \mathcal{G}_k(x).$$

Assumption B.1 (Initial Function Approximations)**.** *The initial objective function approximation $\mathcal{F}_0$ is strongly convex and continuously differentiable in $\mathcal{X}$. The components of the initial constraint function approximation $\mathcal{G}_0$ are convex and continuously differentiable in $\mathcal{X}$.*

Assumption B.2 (Sampled Subgradients)**.** *The subgradients $\hat{g}_k$ and matrices of subgradients $\hat{J}_k$ defined above are uniformly bounded for all $k \in \mathbb{Z}_+$.*

Assumption B.3 (Step Lengths)**.** *The step lengths α_k lie inside the open interval $(0,1)$, and satisfy $\sum_{k=0}^{\infty}\alpha_k = \infty$ almost surely and $\sum_{k=0}^{\infty}\mathbb{E}\left[\alpha_k^2\right] < \infty$.*

Assumption B.4 (Approximate Dual Solutions)**.** *For each $\sigma > 0$ there exists a constant $M > 0$ such that the approximate dual solutions λ_k^σ satisfy*

$$\|\lambda_{k+1}^\sigma - \lambda_k^\sigma\|_2 \leq M\big(\|x_{k+1} - x_k\|_2 + \|x_{k+1}^\sigma - x_k^\sigma\|_2\big)$$

for all $k \in \mathbb{Z}_+$. Furthermore, for all $\sigma > 0$ small enough, $\lambda_k^\sigma \in \Lambda$ for all $k \in \mathbb{Z}_+$.

Assumption B.5 (Drift)**.** *For all $\sigma > 0$ small enough, the sequences of partial sums*

$$\sum_{k=0}^{K}(\lambda_{k+1} - \lambda_k)^T\Gamma_k, \quad \sum_{k=0}^{K}(\lambda_{k+1}^\sigma - \lambda_k^\sigma)^T\Psi_k, \quad \sum_{k=0}^{K}(x_{k+1}^\sigma - x_k^\sigma)^T\Upsilon_k, \tag{B.2}$$

where $\Gamma_k := \mathcal{G}_k(x_k^\sigma) - \mathcal{G}_k(x_k)$, $\Psi_k := \lambda_k^\sigma - \lambda_k$ and $\Upsilon_k := \nabla\mathcal{L}_k(x_k^\sigma, \lambda_k)$, are almost-surely bounded above.

Assumptions B.1, B.2, and B.3 made on the initial function approximations, subgradients, and step lengths, respectively, follow those made in [21]. Assumption B.4 is reasonable from (B.1), the definition of Λ, and $\lambda^* \in \Lambda$. Assumption B.5 is more obscure and hence a discussion providing an intuitive justification for why this may hold in practice has been added at the end of this Appendix. Furthermore, the examples in Section 3.5.2 show the behavior of the partial sums in (B.2) when using initial function approximations and dual bounds $\lambda^{\max}$ of various quality.

Lemma B.1. *The functions $\mathcal{F}_k$ and $\mathcal{G}_k$ and their derivatives are uniformly Lipschitz for all $k \in \mathbb{Z}_+$ and hence also uniformly bounded in $\mathcal{X}$.*

Proof. From (3.6d), the function $\mathcal{G}_k$ can be expressed as

$$\mathcal{G}_k(x) = \mathcal{G}_0(x) + R_k x, \ \forall x \in \mathcal{X}, \tag{B.3}$$

for $k \in \mathbb{Z}_+$, where R_k is a matrix and $R_0 = 0$. It follows that for all $k \in \mathbb{Z}_+$ and $x \in \mathcal{X}$,

$$\begin{aligned}
\mathcal{G}_{k+1}(x) &= \mathcal{G}_k(x) + \alpha_k\big(\hat{J}_k - J_k(x_k)\big)x \\
&= \mathcal{G}_0(x) + R_k x + \alpha_k\big(\hat{J}_k - J_0(x_k) - R_k\big)x \\
&= \mathcal{G}_0(x) + \big((1-\alpha_k)R_k + \alpha_k\big(\hat{J}_k - J_0(x_k)\big)\big)x.
\end{aligned}$$

Hence

$$R_{k+1} = (1-\alpha_k)R_k + \alpha_k\big(\hat{J}_k - J_0(x_k)\big).$$

From Assumptions B.1 and B.2, there is some constant M_1 such that

$$\|\hat{J}_k - J_0(x_k)\|_2 \leq M_1$$

for all $k \in \mathbb{Z}_+$. Hence, Assumption B.3 gives that $\|R_k\|_2 \leq M_1$ implies

$$\begin{aligned}
\|R_{k+1}\|_2 &= \big\|(1-\alpha_k)R_k + \alpha_k\big(\hat{J}_k - J_0(x_k)\big)\big\|_2 \\
&\leq (1-\alpha_k)\|R_k\|_2 + \alpha_k\|\hat{J}_k - J_0(x_k)\|_2 \\
&\leq (1-\alpha_k)M_1 + \alpha_k M_1 \\
&= M_1.
\end{aligned}$$

Since $R_0 = 0$, it holds by induction that $\|R_k\|_2 \leq M_1$ for all $k \in \mathbb{Z}_+$. It follows that for any x and $y \in \mathcal{X}$ and $k \in \mathbb{Z}_+$,

$$\begin{aligned}
\|\mathcal{G}_k(x) - \mathcal{G}_k(y)\|_2 &= \|\mathcal{G}_0(x) + R_k x - \mathcal{G}_0(y) - R_k y\|_2 \\
&\leq \|R_k\|_2\|x-y\|_2 + \|\mathcal{G}_0(x) - \mathcal{G}_0(y)\|_2 \\
&\leq (M_1 + M_2)\,\|x-y\|_2,
\end{aligned}$$

where $M_2 > 0$ exists because Assumption B.1 implies that $\mathcal{G}_0$ is Lipschitz in the compact set $\mathcal{X}$.

From (B.3), the Jacobian of $\mathcal{G}_k$ is given by

$$J_k(x) = J_0(x) + R_k,\ \forall x \in \mathcal{X},$$

for each $k \in \mathbb{Z}_+$. Assumption B.1 and $\|R_k\|_2 \leq M_1$ imply that the functions J_k, $k \in \mathbb{Z}_+$, are uniformly Lipschitz in $\mathcal{X}$.

The derivation of these properties for the functions $\mathcal{F}_k$, $k \in \mathbb{Z}_+$, is similar and hence omitted. □

Lemma B.2. *The functions $\mathcal{F}_k$, $k \in \mathbb{Z}_+$, are uniformly strongly convex.*

Proof. From (3.6c) and Lemma B.1, $\mathcal{F}_k$ can be expressed as

$$\mathcal{F}_k(x) = \mathcal{F}_0(x) + r_k^T x, \ \forall x \in \mathcal{X},$$

for $k \in \mathbb{Z}_+$, where r_k are uniformly bounded vectors with $r_0 = 0$. Assumption B.1 gives that $\mathcal{F}_0$ is strongly convex, and therefore that $\mathcal{F}_k$ are uniformly strongly convex since their strong-convexity constant is independent of k. □

Lemma B.3. *There exists an $M > 0$ such that $\|x_{k+1} - x_k\|_2 \leq \alpha_k M$ for all $k \in \mathbb{Z}_+$. For convenience, we say that $\|x_{k+1} - x_k\|_2$ is $\mathcal{O}(\alpha_k)$.*

Proof. Equation (3.6a) implies that

$$\left(g_k(x_k) + J_k(x_k)^T \lambda_k\right)^T (x - x_k) \geq 0, \ \forall x \in \mathcal{X} \tag{B.4}$$

$$\left(g_{k+1}(x_{k+1}) + J_{k+1}(x_{k+1})^T \lambda_{k+1}\right)^T (x - x_{k+1}) \geq 0, \ \forall x \in \mathcal{X}, \tag{B.5}$$

for all $k \in \mathbb{Z}_+$. In particular, (B.4) holds with $x = x_{k+1}$, and (B.5) holds with $x = x_k$. Hence,

$$\left(g_{k+1}(x_{k+1}) - g_k(x_k)\right)^T \delta_k \leq \left(J_k(x_k)^T \lambda_k - J_{k+1}(x_{k+1})^T \lambda_{k+1}\right)^T \delta_k,$$

where $\delta_k := x_{k+1} - x_k$. Using (3.6c) and (3.6d) gives

$$\left(g_k(x_{k+1}) - g_k(x_k)\right)^T \delta_k \leq \left(J_k(x_k)^T \lambda_k - J_k(x_{k+1})^T \lambda_{k+1}\right)^T \delta_k - \alpha_k \left(\Delta_k^T \lambda_{k+1} + \eta_k\right)^T \delta_k, \tag{B.6}$$

where

$$\eta_k := \hat{g}_k - g_k(x_k) \tag{B.7}$$

$$\Delta_k := \hat{J}_k - J_k(x_k). \tag{B.8}$$

From Lemma B.2, $\mathcal{F}_k$ is uniformly strongly convex so there exists a $C > 0$ independent of k such that

$$C\|\delta_k\|_2^2 \leq \left(g_k(x_{k+1}) - g_k(x_k)\right)^T \delta_k$$

for all $k \in \mathbb{Z}_+$. It follows from this and (B.6) that

$$\begin{aligned} C\|\delta_k\|_2^2 &\leq \left(\lambda_k^T J_k(x_k) - \lambda_{k+1}^T J_k(x_{k+1})\right)\delta_k - \alpha_k \left(\Delta_k^T \lambda_{k+1} + \eta_k\right)^T \delta_k \\ &\leq \left(\lambda_k^T J_k(x_k) - \lambda_{k+1}^T J_k(x_{k+1})\right)\delta_k + \alpha_k M_1 \|\delta_k\|_2 \end{aligned} \tag{B.9}$$

for all $k \in \mathbb{Z}_+$, where $M_1 > 0$ exists because of the uniform boundedness of λ_k (by construction), and that of Δ_k and η_k (by Assumption B.2 and Lemma B.1). From the convexity of the component functions of $\mathcal{G}_k$ and the fact that $\lambda_k \geq 0$ and $\lambda_{k+1} \geq 0$, it holds that

$$\begin{aligned} \lambda_{k+1}^T \left(\mathcal{G}_k(x_k) - \mathcal{G}_k(x_{k+1})\right) &\geq -\lambda_{k+1}^T J_k(x_{k+1})\delta_k \\ \lambda_k^T \left(\mathcal{G}_k(x_{k+1}) - \mathcal{G}_k(x_k)\right) &\geq \lambda_k^T J_k(x_k)\delta_k. \end{aligned}$$

Adding these inequalities gives

$$\begin{aligned} \left(\lambda_k^T J_k(x_k) - \lambda_{k+1}^T J_k(x_{k+1})\right)\delta_k &\leq \left(\lambda_{k+1} - \lambda_k\right)^T \left(\mathcal{G}_k(x_k) - \mathcal{G}_k(x_{k+1})\right) \\ &\leq \|\lambda_{k+1} - \lambda_k\|_2 M_2 \|\delta_k\|_2 \\ &\leq \alpha_k \|\hat{G}_k\|_2 M_2 \|\delta_k\|_2 \\ &\leq \alpha_k M_2 M_3 \|\delta_k\|_2, \end{aligned}$$

where $M_2 > 0$ and $M_3 > 0$ exist due to the uniform boundedness of $\hat{G}_k$ and Lemma B.1. It follows from this and (B.9) that

$$C\|\delta_k\|_2^2 \leq \alpha_k (M_1 + M_2 M_3)\|\delta_k\|_2,$$

or equivalently,

$$\|\delta_k\|_2 \leq \alpha_k (M_1 + M_2 M_3)/C$$

for all $k \in \mathbb{Z}_+$. □

Corollary B.1. *For each $\sigma > 0$, $\|x_{k+1}^\sigma - x_k^\sigma\|_2$ and $\|\lambda_{k+1}^\sigma - \lambda_k^\sigma\|_2$ are $\mathcal{O}(\alpha_k)$.*

Proof. For each $\sigma > 0$ and $k \in \mathbb{Z}_+$, the optimality of x_k^σ implies that there exists a $g_k^\sigma \in \partial\mathcal{F}(x_k^\sigma)$ such that

$$\left(g_k^\sigma + \sigma(x_k^\sigma - x_k)\right)^T (x - x_k^\sigma) \geq 0$$

for all $x \in \mathcal{X}$ such that $\mathcal{G}(x) \leq 0$. It follows from this that

$$\begin{aligned}\left(g_k^\sigma + \sigma(x_k^\sigma - x_k)\right)^T (x_{k+1}^\sigma - x_k^\sigma) \geq 0 \\ \left(g_{k+1}^\sigma + \sigma(x_{k+1}^\sigma - x_{k+1})\right)^T (x_k^\sigma - x_{k+1}^\sigma) \geq 0.\end{aligned}$$

Adding these inequalities, rearranging, and using $(g_{k+1}^\sigma - g_k^\sigma)^T(x_{k+1}^\sigma - x_k^\sigma) \geq 0$ gives

$$\begin{aligned}\sigma\|x_{k+1}^\sigma - x_k^\sigma\|_2^2 &\leq \sigma(x_{k+1} - x_k)^T(x_{k+1}^\sigma - x_k^\sigma) + (g_k^\sigma - g_{k+1}^\sigma)^T(x_{k+1}^\sigma - x_k^\sigma) \\ &\leq \sigma\|x_{k+1} - x_k\|_2\|x_{k+1}^\sigma - x_k^\sigma\|_2.\end{aligned}$$

It follows from this and Lemma B.3 that $\|x_{k+1}^\sigma - x_k^\sigma\|_2$ is $\mathcal{O}(\alpha_k)$. Assumption B.4 then gives that $\|\lambda_{k+1}^\sigma - \lambda_k^\sigma\|_2$ is also $\mathcal{O}(\alpha_k)$. □

Lemma B.4. *For all $\sigma > 0$ small enough, there exists an $M > 0$ such that*

$$\frac{1}{2}\|\lambda_{k+1} - \lambda_{k+1}^\sigma\|_2^2 \leq \frac{1}{2}\|\lambda_k - \lambda_k^\sigma\|_2^2 + (\lambda_{k+1}^\sigma - \lambda_k^\sigma)^T\Psi_k + \alpha_k(\lambda_k - \lambda_k^\sigma)^T\hat{G}_k + \alpha_k^2 M$$

for all $k \in \mathbb{Z}_+$, where Ψ_k is as defined in Assumption B.5.

Proof. From equation (3.6b), the fact that $\lambda_k^\sigma \in \Lambda$ for small enough σ, Corollary B.1, and the uniform boundedness of $\hat{G}_k$, it follows that there exist positive scalars M_1 and M_2 independent of k such that

$$\begin{aligned}\frac{1}{2}\|\lambda_{k+1} - \lambda_{k+1}^\sigma\|_2^2 &= \frac{1}{2}\left\|\Pi_\lambda\left(\lambda_k + \alpha_k\hat{G}_k\right) - \lambda_{k+1}^\sigma\right\|_2^2 \\ &\leq \frac{1}{2}\|\lambda_k + \alpha_k\hat{G}_k - \lambda_{k+1}^\sigma\|_2^2 \\ &= \frac{1}{2}\|\lambda_k - \lambda_{k+1}^\sigma\|_2^2 + \alpha_k(\lambda_k - \lambda_{k+1}^\sigma)^T\hat{G}_k + \frac{1}{2}\|\alpha_k\hat{G}_k\|_2^2 \\ &\leq \frac{1}{2}\|\lambda_k - \lambda_{k+1}^\sigma\|_2^2 + \alpha_k(\lambda_k - \lambda_k^\sigma)^T\hat{G}_k + \alpha_k^2 M_1 \\ &\leq \frac{1}{2}\|\lambda_k - \lambda_k^\sigma\|_2^2 + (\lambda_{k+1}^\sigma - \lambda_k^\sigma)^T\Psi_k + \alpha_k(\lambda_k - \lambda_k^\sigma)^T\hat{G}_k + \alpha_k^2 M_2.\end{aligned}$$

□

Lemma B.5. *For all $\sigma > 0$ small enough, the sequence $\{T_k^\sigma\}_{k\in\mathbb{Z}_+}$ defined by*

$$T_k^\sigma := \mathcal{F}_k(x_k^\sigma) + \lambda_k^T\mathcal{G}_k(x_k^\sigma) - \mathcal{F}_k(x_k) - \lambda_k^T\mathcal{G}_k(x_k) + \frac{1}{2}\|\lambda_k - \lambda_k^\sigma\|_2^2$$

satisfies

$$\begin{aligned} T_{k+1}^{\sigma} - T_k^{\sigma} \leq\ & (\lambda_{k+1} - \lambda_k)^T \Gamma_k + (\lambda_{k+1}^{\sigma} - \lambda_k^{\sigma})^T \Psi_k + (x_{k+1}^{\sigma} - x_k^{\sigma})^T \Upsilon_k + \\ & \alpha_k \left(L(x_k^{\sigma}, \lambda_k^{\sigma}, w_k) - L(x_k, \lambda_k^{\sigma}, w_k) + \frac{\sigma}{2} \|x_k^{\sigma} - x_k\|_2^2 \right) + \\ & \alpha_k (\lambda_k - \lambda_k^{\sigma})^T G(x_k^{\sigma}, w_k) + \\ & \mathcal{O}(\alpha_k^2) \end{aligned}$$

for all $k \in \mathbb{Z}_+$, where Γ_k, Ψ_k and Υ_k are as defined in Assumption B.5.

Proof. From the definition of T_k^{σ}, it follows that

$$\begin{aligned} T_{k+1}^{\sigma} - T_k^{\sigma} = & \mathcal{F}_{k+1}(x_{k+1}^{\sigma}) + \lambda_{k+1}^T \mathcal{G}_{k+1}(x_{k+1}^{\sigma}) - \mathcal{F}_{k+1}(x_{k+1}) - \lambda_{k+1}^T \mathcal{G}_{k+1}(x_{k+1}) - \\ & \mathcal{F}_k(x_k^{\sigma}) - \lambda_k^T \mathcal{G}_k(x_k^{\sigma}) + \mathcal{F}_k(x_k) + \lambda_k^T \mathcal{G}_k(x_k) + \\ & \frac{1}{2} \|\lambda_{k+1} - \lambda_{k+1}^{\sigma}\|_2^2 - \frac{1}{2} \|\lambda_k - \lambda_k^{\sigma}\|_2^2. \end{aligned}$$

Using equations (3.6c) and (3.6d), and adding and subtracting $\lambda_k^T \mathcal{G}_k(x_{k+1})$ as well as $\lambda_k^T \mathcal{G}_k(x_{k+1}^{\sigma})$ to the right-had side of the above equation gives

$$\begin{aligned} T_{k+1}^{\sigma} - T_k^{\sigma} = & \mathcal{F}_k(x_k) + \lambda_k^T \mathcal{G}_k(x_k) - \mathcal{F}_k(x_{k+1}) - \lambda_k^T \mathcal{G}_k(x_{k+1}) + \\ & \mathcal{F}_k(x_{k+1}^{\sigma}) + \lambda_k^T \mathcal{G}_k(x_{k+1}^{\sigma}) - \mathcal{F}_k(x_k^{\sigma}) - \lambda_k^T \mathcal{G}_k(x_k^{\sigma}) + \\ & (\lambda_{k+1} - \lambda_k)^T (\mathcal{G}_k(x_{k+1}^{\sigma}) - \mathcal{G}_k(x_{k+1})) + \\ & \alpha_k \eta_k^T (x_{k+1}^{\sigma} - x_{k+1}) + \\ & \alpha_k \lambda_{k+1}^T \Delta_k (x_{k+1}^{\sigma} - x_{k+1}) + \\ & \frac{1}{2} \|\lambda_{k+1} - \lambda_{k+1}^{\sigma}\|_2^2 - \frac{1}{2} \|\lambda_k - \lambda_k^{\sigma}\|_2^2, \end{aligned}$$

where η_k and Δ_k are defined in (B.7) and (B.8), respectively. From (3.6a), it holds that

$$\mathcal{F}_k(x_k) + \lambda_k^T \mathcal{G}_k(x_k) - \mathcal{F}_k(x_{k+1}) - \lambda_k^T \mathcal{G}_k(x_{k+1}) \leq 0.$$

Using this, Lemma B.4, the convexity of $\mathcal{L}_k(\,\cdot\,,\lambda_k)$, and the definitions of η_k and Δ_k gives

$$\begin{aligned}T_{k+1}^\sigma - T_k^\sigma \leq\ & (\lambda_{k+1}-\lambda_k)^T(\mathcal{G}_k(x_{k+1}^\sigma)-\mathcal{G}_k(x_{k+1}))\ + \\ & (\lambda_{k+1}^\sigma-\lambda_k^\sigma)^T(\lambda_k^\sigma-\lambda_k)\ + \\ & (x_{k+1}^\sigma-x_k^\sigma)^T\nabla\mathcal{L}_k(x_{k+1}^\sigma,\lambda_k)\ + \\ & \alpha_k\big(\hat{g}_k+\hat{J}_k^T\lambda_{k+1}\big)^T(x_{k+1}^\sigma-x_{k+1})\ - \\ & \alpha_k\big(g_k(x_k)+J_k(x_k)^T\lambda_{k+1}\big)^T(x_{k+1}^\sigma-x_{k+1})\ + \\ & \alpha_k(\lambda_k-\lambda_k^\sigma)^T\hat{G}_k\ + \\ & \mathcal{O}(\alpha_k^2).\end{aligned}$$

From equation (3.6b) and the uniform boundedness of $\hat{G}_k$, it holds that $\|\lambda_{k+1}-\lambda_k\|_2$ is $\mathcal{O}(\alpha_k)$. From Lemma B.3 and Corollary B.1, it holds that $\|x_{k+1}-x_k\|_2$, $\|x_{k+1}^\sigma-x_k^\sigma\|_2$ and $\|\lambda_{k+1}^\sigma-\lambda_k^\sigma\|_2$ are $\mathcal{O}(\alpha_k)$. From these bounds, Lemma B.1 and Assumption B.2, it follows that

$$\begin{aligned}T_{k+1}^\sigma - T_k^\sigma \leq\ & (\lambda_{k+1}-\lambda_k)^T\Gamma_k+(\lambda_{k+1}^\sigma-\lambda_k^\sigma)^T\Psi_k+(x_{k+1}^\sigma-x_k^\sigma)^T\Upsilon_k\ + \\ & \alpha_k\big(\hat{g}_k+\hat{J}_k^T\lambda_k\big)^T(x_k^\sigma-x_k)\ - \\ & \alpha_k\big(g_k(x_k)+J_k(x_k)^T\lambda_k\big)^T(x_k^\sigma-x_k)\ + \\ & \alpha_k(\lambda_k-\lambda_k^\sigma)^T\hat{G}_k\ + \\ & \mathcal{O}(\alpha_k^2).\end{aligned}$$

where Γ_k, Ψ_k and Υ_k are as defined in Assumption B.5. From equation (3.6a), it holds that

$$\big(g_k(x_k)+J_k(x_k)^T\lambda_k\big)^T(x_k^\sigma-x_k)\geq 0.$$

Using this and the fact that $\hat{g}_k+\hat{J}_k^T\lambda_k$ is a subgradient of $L(x,\lambda_k,w_k)+\frac{\sigma}{2}\|x-x_k\|_2^2$ at x_k gives

$$\begin{aligned}T_{k+1}^\sigma - T_k^\sigma \leq\ & (\lambda_{k+1}-\lambda_k)^T\Gamma_k+(\lambda_{k+1}^\sigma-\lambda_k^\sigma)^T\Psi_k+(x_{k+1}^\sigma-x_k^\sigma)^T\Upsilon_k\ + \\ & \alpha_k\left(L(x_k^\sigma,\lambda_k,w_k)-L(x_k,\lambda_k,w_k)+\frac{\sigma}{2}\|x_k^\sigma-x_k\|_2^2\right)\ + \\ & \alpha_k(\lambda_k-\lambda_k^\sigma)^T\hat{G}_k\ + \\ & \mathcal{O}(\alpha_k^2).\end{aligned}$$

Then, adding and subtracting $\alpha_k \lambda_k^\sigma G(x_k^\sigma, w_k)$, and using $\hat{G}_k = G(x_k, w_k)$ and the definition of $L(\,\cdot\,,\,\cdot\,, w_k)$ gives

$$\begin{aligned} T_{k+1}^\sigma - T_k^\sigma \leq\ & (\lambda_{k+1} - \lambda_k)^T \Gamma_k + (\lambda_{k+1}^\sigma - \lambda_k^\sigma)^T \Psi_k + (x_{k+1}^\sigma - x_k^\sigma)^T \Upsilon_k + \\ & \alpha_k \left(L(x_k^\sigma, \lambda_k^\sigma, w_k) - L(x_k, \lambda_k^\sigma, w_k) + \frac{\sigma}{2} \|x_k^\sigma - x_k\|_2^2 \right) + \\ & \alpha_k (\lambda_k - \lambda_k^\sigma)^T G(x_k^\sigma, w_k) + \\ & \mathcal{O}(\alpha_k^2). \end{aligned}$$

□

Lemma B.6. *For all $\sigma > 0$ small enough, the sequence $\{x_k - x_k^\sigma\}_{k \in \mathbb{Z}_+}$ has a subsequence that converges to zero almost surely.*

Proof. From Lemma B.5, for all $\sigma > 0$ small enough it holds that

$$\begin{aligned} T_{k+1}^\sigma - T_k^\sigma \leq\ & (\lambda_{k+1} - \lambda_k)^T \Gamma_k + (\lambda_{k+1}^\sigma - \lambda_k^\sigma)^T \Psi_k + (x_{k+1}^\sigma - x_k^\sigma)^T \Upsilon_k + \\ & \alpha_k \left(L(x_k^\sigma, \lambda_k^\sigma, w_k) - L(x_k, \lambda_k^\sigma, w_k) + \frac{\sigma}{2} \|x_k^\sigma - x_k\|_2^2 \right) + \\ & \alpha_k (\lambda_k - \lambda_k^\sigma)^T G(x_k^\sigma, w_k) + \\ & \mathcal{O}(\alpha_k^2) \end{aligned}$$

for all $k \in \mathbb{Z}_+$. Hence, summing over k from 0 to K gives

$$\begin{aligned} T_{K+1}^\sigma - T_0^\sigma \leq\ & \sum_{k=0}^{K} (\lambda_{k+1} - \lambda_k)^T \Gamma_k + \sum_{k=1}^{K} (\lambda_{k+1}^\sigma - \lambda_k^\sigma)^T \Psi_k + \sum_{k=1}^{K} (x_{k+1}^\sigma - x_k^\sigma)^T \Upsilon_k + \\ & \sum_{k=0}^{K} \alpha_k \left(L(x_k^\sigma, \lambda_k^\sigma, w_k) - L(x_k, \lambda_k^\sigma, w_k) + \frac{\sigma}{2} \|x_k^\sigma - x_k\|_2^2 \right) + \\ & \sum_{k=0}^{K} \alpha_k (\lambda_k - \lambda_k^\sigma)^T G(x_k^\sigma, w_k) + \\ & \sum_{k=0}^{K} \mathcal{O}(\alpha_k^2). \end{aligned}$$

From this, Assumptions B.3 and B.5, there exists an $M > 0$ independent of K such that

$$T_{K+1}^{\sigma} - T_0^{\sigma} - M \leq \sum_{k=0}^{K} \alpha_k \left(L(x_k^{\sigma}, \lambda_k^{\sigma}, w_k) - L(x_k, \lambda_k^{\sigma}, w_k) + \frac{\sigma}{2} \|x_k^{\sigma} - x_k\|_2^2 \right) + \sum_{k=0}^{K} \alpha_k (\lambda_k - \lambda_k^{\sigma})^T G(x_k^{\sigma}, w_k)$$

for all K almost surely. By rearranging, taking expectation, and using the properties $(\lambda_k^{\sigma})^T \mathcal{G}(x_k^{\sigma}) = 0$ and $\lambda_k^T \mathcal{G}(x_k^{\sigma}) \leq 0$, which follow from the fact that the primal-dual point $(x_k^{\sigma}, \lambda_k^{\sigma})$ solves (B.1), it follows that

$$\sum_{k=0}^{K} \mathbb{E}\left[\alpha_k \left(\mathcal{L}(x_k, \lambda_k^{\sigma}) - \mathcal{L}(x_k^{\sigma}, \lambda_k^{\sigma}) - \frac{\sigma}{2} \|x_k^{\sigma} - x_k\|_2^2 \right) \right] \leq M + \mathbb{E}\left[T_0^{\sigma}\right] - \mathbb{E}\left[T_{K+1}^{\sigma}\right].$$

Taking $K \to \infty$ and using the boundedness of the terms on the right-hand side gives

$$\sum_{k=0}^{\infty} \mathbb{E}\left[\alpha_k \left(\mathcal{L}(x_k, \lambda_k^{\sigma}) - \mathcal{L}(x_k^{\sigma}, \lambda_k^{\sigma}) - \frac{\sigma}{2} \|x_k^{\sigma} - x_k\|_2^2 \right) \right] < \infty. \tag{B.10}$$

Again, since $(x_k^{\sigma}, \lambda_k^{\sigma})$ solves (B.1),

$$\begin{aligned} \mathcal{L}(x_k, \lambda_k^{\sigma}) &= \mathcal{F}(x_k) + (\lambda_k^{\sigma})^T \mathcal{G}(x_k) \\ &= \mathcal{F}(x_k) + (\lambda_k^{\sigma})^T \mathcal{G}(x_k) + \frac{\sigma}{2} \|x_k - x_k\|_2^2 \\ &\geq \mathcal{F}(x_k^{\sigma}) + (\lambda_k^{\sigma})^T \mathcal{G}(x_k^{\sigma}) + \frac{\sigma}{2} \|x_k^{\sigma} - x_k\|_2^2 \\ &= \mathcal{L}(x_k^{\sigma}, \lambda_k^{\sigma}) + \frac{\sigma}{2} \|x_k^{\sigma} - x_k\|_2^2, \end{aligned}$$

and hence

$$\mathcal{L}(x_k, \lambda_k^{\sigma}) - \mathcal{L}(x_k^{\sigma}, \lambda_k^{\sigma}) - \frac{\sigma}{2} \|x_k^{\sigma} - x_k\|_2^2 \geq 0.$$

It follows from this last inequality, (B.10), and Assumption B.3 that there exists a subsequence $\{(x_{n_k}, x_{n_k}^{\sigma}, \lambda_{n_k}^{\sigma})\}_{k \in \mathbb{Z}_+}$ such that

$$\mathcal{L}(x_{n_k}, \lambda_{n_k}^{\sigma}) - \mathcal{L}(x_{n_k}^{\sigma}, \lambda_{n_k}^{\sigma}) - \frac{\sigma}{2} \|x_{n_k}^{\sigma} - x_{n_k}\|_2^2 \to 0 \quad \text{as} \quad k \to \infty \tag{B.11}$$

almost surely. Since $(x_k^\sigma, \lambda_k^\sigma)$ solves (B.1), there exists a $\eta_k^\sigma \in \partial\mathcal{L}(x_k^\sigma, \lambda_k^\sigma)$ such that

$$\big(\eta_k^\sigma + \sigma(x_k^\sigma - x_k)\big)^T (x - x_k^\sigma) \geq 0$$

for all $x \in \mathcal{X}$, and hence that

$$(\eta_{n_k}^\sigma)^T (x_{n_k} - x_{n_k}^\sigma) \geq \sigma \|x_{n_k}^\sigma - x_{n_k}\|_2^2$$

for all $k \in \mathbb{Z}_+$. It follows from this that

$$\begin{aligned}
\mathcal{L}(x_{n_k}, \lambda_{n_k}^\sigma) - \mathcal{L}(x_{n_k}^\sigma, \lambda_{n_k}^\sigma) - \frac{\sigma}{2}\|x_{n_k}^\sigma - x_{n_k}\|_2^2 &\geq (\eta_{n_k}^\sigma)^T (x_{n_k} - x_{n_k}^\sigma) - \frac{\sigma}{2}\|x_{n_k}^\sigma - x_{n_k}\|_2^2 \\
&\geq \sigma \|x_{n_k}^\sigma - x_{n_k}\|_2^2 - \frac{\sigma}{2}\|x_{n_k}^\sigma - x_{n_k}\|_2^2 \\
&= \frac{\sigma}{2}\|x_{n_k}^\sigma - x_{n_k}\|_2^2.
\end{aligned}$$

This and (B.11) then give that $(x_{n_k}^\sigma - x_{n_k}) \to 0$ as $k \to \infty$ almost surely. □

Theorem B.1. *The sequence $\{x_k\}_{k\in\mathbb{Z}_+}$ of primal iterates produced by algorithm* (3.6) *has a subsequence that converges almost surely to an optimal point of problem* (3.1).

Proof. Since x_k^σ is an optimal primal point of (B.1), it follows that $\mathcal{G}(x_k^\sigma) \leq 0$ and

$$\begin{aligned}
\mathcal{F}(x_k^\sigma) &\leq \mathcal{F}(x^*) + \frac{\sigma}{2}\left(\|x^* - x_k\|_2^2 - \|x_k^\sigma - x_k\|_2^2\right) \\
&\leq \mathcal{F}(x^*) + \sigma M_1
\end{aligned}$$

for all $k \in \mathbb{Z}_+$ and $\sigma > 0$, where x^* is an optimal primal point of problem (3.1), and $M_1 > 0$ exists due to x^*, x_k^σ and x_k all being inside the compact convex set $\mathcal{X}$. Letting $m_{-1} := 0$, it follows from the above inequalities, Lemma B.6, and the continuity of $\mathcal{F}$ and $\mathcal{G}$ in $\mathcal{X}$, that for each $k \in \mathbb{Z}_+$ there exist $m_k > m_{k-1}$ and $\sigma_k > 0$ small enough with probability one such that

$$\begin{aligned}
|\mathcal{F}(x_{m_k}) - \mathcal{F}(x_{m_k}^{\sigma_k})| &\leq 1/k \\
\|\mathcal{G}(x_{m_k}) - \mathcal{G}(x_{m_k}^{\sigma_k})\|_2 &\leq 1/k \\
\mathcal{F}(x_{m_k}^{\sigma_k}) &\leq \mathcal{F}(x^*) + M_1/k.
\end{aligned}$$

It follows that the resulting sequence $\{x_{m_k}\}_{k\in\mathbb{Z}_+}$, which is a subsequence of $\{x_k\}_{k\in\mathbb{Z}_+}$,

satisfies

$$\begin{aligned}\mathcal{F}(x_{m_k}) &= \mathcal{F}(x_{m_k}^{\sigma_k}) + \mathcal{F}(x_{m_k}) - \mathcal{F}(x_{m_k}^{\sigma_k}) \\ &\leq \mathcal{F}(x_{m_k}^{\sigma_k}) + |\mathcal{F}(x_{m_k}) - \mathcal{F}(x_{m_k}^{\sigma_k})| \\ &\leq \mathcal{F}(x_{m_k}^{\sigma_k}) + 1/k \\ &\leq \mathcal{F}(x^*) + (M_1 + 1)/k\end{aligned}$$

and

$$\begin{aligned}\mathcal{G}(x_{m_k})^T e_i &= \mathcal{G}(x_{m_k}^{\sigma_k})^T e_i + (\mathcal{G}(x_{m_k}) - \mathcal{G}(x_{m_k}^{\sigma}))^T e_i \\ &\leq \mathcal{G}(x_{m_k}^{\sigma_k})^T e_i + \|\mathcal{G}(x_{m_k}) - \mathcal{G}(x_{m_k}^{\sigma})\|_2 \\ &\leq 1/k\end{aligned}$$

for each k, where e_i is any standard basis vector. These inequalities and the boundedness of $\{x_{m_k}\}_{k\in\mathbb{Z}_+}$ imply that the sequence $\{x_k\}_{k\in\mathbb{Z}_+}$ has a subsequence that convergences almost surely to a point $\bar{x} \in \mathcal{X}$ that satisfies $\mathcal{G}(\bar{x}) \leq 0$ and $\mathcal{F}(\bar{x}) = \mathcal{F}(x^*)$. □

Assumption B.5 states that for all $\sigma > 0$ small enough, the sequences of partial sums given in (B.2) are almost-surely bounded above. This assumption is only needed for Lemma B.6 and Theorem B.1. To provide a practical justification for this assumption, it is enough to consider the scalar case, *i.e.*, $\lambda_k \in \mathbb{R}$ and $x_k \in \mathbb{R}$.

By the compactness of Λ and $\mathcal{X}$, and Assumption B.4, the sequences $\{\lambda_k\}$, $\{\lambda_k^\sigma\}$ and $\{x_k^\sigma\}$ are bounded. From this and Lemma B.1, it holds that the sequences $\{\Gamma_k\}$, $\{\Psi_k\}$ and $\{\Upsilon_k\}$ are also bounded. Furthermore, equation (3.6b), the uniform boundedness of $\hat{G}_k$, and Corollary B.1 give that there exists an $M_1 > 0$ such that

$$|\lambda_{k+1} - \lambda_k| \leq \alpha_k M_1, \quad |\lambda_{k+1}^\sigma - \lambda_k^\sigma| \leq \alpha_k M_1, \quad |x_{k+1}^\sigma - x_k^\sigma| \leq \alpha_k M_1$$

for all $k \in \mathbb{Z}_+$. Similarly, these inequalities, equations (3.6c) and (3.6d), the boundedness of $\{\lambda_k\}$, $\{\lambda_k^\sigma\}$ and $\{x_k^\sigma\}$, Lemma B.1, and Assumption B.2 imply that there exists an $M_2 > 0$ such that

$$|\Gamma_{k+1} - \Gamma_k| \leq \alpha_k M_2, \quad |\Psi_{k+1} - \Psi_k| \leq \alpha_k M_2, \quad |\Upsilon_{k+1} - \Upsilon_k| \leq \alpha_k M_2$$

for all $k \in \mathbb{Z}_+$. Hence, the partial sums in (B.2) (in the scalar case) are all of the form

$$S_n := \sum_{k=1}^{n} (a_{k+1} - a_k) b_k,$$

where $\{a_k\}$ and $\{b_k\}$ are bounded, and $|a_{k+1} - a_k| \leq \alpha_k M$ and $|b_{k+1} - b_k| \leq \alpha_k M$ for all $k \in \mathbb{Z}_+$ for some $M > 0$. Unfortunately, these properties alone of a_k and b_k, which can be assumed to be non-negative without loss of generality, are not enough to ensure that the partial sums S_n are bounded above in the general case, as counterexamples for this can be constructed. However, due to Assumption B.3, all counterexamples require a high degree of synchronization between a_k and b_k in order to make b_k favor consistently (and infinitely often) increases in a_k, *i.e.*, $a_{k+1} - a_k > 0$, over decreases. Due to this strong synchronization requirement, it seems reasonable to expect that in practical applications the sequences of partial sums in (B.2) are bounded above and hence that Assumption B.5 holds.

Appendix C

Convergence Analysis of Parameterized SHA

In this section, a convergence analysis of the parameterized stochastic hybrid approximation algorithm described in Section 4.4.2 is presented for the theoretically tractable case of finite-support uncertainty. In particular, conditions are identified that guarantee that the iterates produced by the algorithm have a subsequence that converges to the solutions of the nested optimization problems (4.1).

This section is structured as follows: First, key functions and notation needed for the analysis are introduced. Then, the assumptions under which convergence is analyzed are described followed by a discussion regarding their justification and implications. Lastly, Lemmas, Corollaries and Theorems are presented that gradually derive the theoretical results and properties of the algorithm.

For $t \in \{1, \ldots, T\}$ and $k \in \mathbb{Z}_+$, let $H^k_{t,0}(x, \mathcal{W}_t)$ and $\mathcal{H}^k_{t,0}(x, \mathcal{W}_t)$ be the optimal value and an optimal point of

$$\begin{aligned} \underset{x_t}{\text{minimize}} \quad & F_t(x_t, w_t) + \widehat{G}^k_t(x_t, \mathcal{W}_t) \\ \text{subject to} \quad & x_t \in \mathcal{X}_t(x, w_t), \end{aligned}$$

respectively, where the function $\widehat{G}^k_t(\,\cdot\,, \mathcal{W}_t)$ is defined by

$$\widehat{G}^k_t(x, \mathcal{W}_t) := \widehat{G}_t(x, \mathcal{W}_t) + g^k_t(\mathcal{W}_t)^T x, \ \forall x. \tag{C.1}$$

For $t \in \{1, \ldots, T-1\}$ and $\tau \in \{1, \ldots, T-t\}$, let $H^k_{t,\tau}(x, \mathcal{W}_t)$ and $\mathcal{H}^k_{t,\tau}(x, \mathcal{W}_t)$ be the optimal value and an optimal point of

$$\begin{aligned} \underset{x_t}{\text{minimize}} \quad & F_t(x_t, w_t) + \mathbb{E}\left[H^k_{t+1,\tau-1}(x_t, \mathcal{W}_{t+1}) \,\middle|\, \mathcal{W}_t \right] \\ \text{subject to} \quad & x_t \in \mathcal{X}_t(x, w_t), \end{aligned}$$

respectively. Hence, the parameter τ denotes the number of exact expected values present in the nested problem, or the "depth" of the approximation. It follows that for $t \in \{1, \ldots, T\}$ and $\tau = T - t$, $H^k_{t,\tau}(x, \mathcal{W}_t)$ and $\mathcal{H}^k_{t,\tau}(x, \mathcal{W}_t)$ are independent of k and are equal to the optimal value and an optimal point of problem (4.1), respectively, with $x_{t-1} = x$.

Also, let $\widehat{\mathcal{H}}^k_{t,\tau}(x, \mathcal{W}_t)$ be an optimal point of the "expanded" problem

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & L^k_{t,\tau}(y, \mathcal{W}_t) \\ \text{subject to} \quad & y \in \mathcal{Y}_t(x, w_t), \end{aligned}$$

where the objective function $L^k_{t,\tau}(\,\cdot\,, \mathcal{W}_t)$ is given by

$$\begin{aligned} L^k_{t,\tau}(y, \mathcal{W}_t) := F_t(y_t(w_t), w_t) \; + \\ \mathbb{E}\Big[F_{t+1}(y_{t+1}(w_{t+1}), w_{t+1}) + \cdots \\ \mathbb{E}\Big[F_{t+\tau}(y_{t+\tau}(w_{t+\tau}), w_{t+\tau}) + \widehat{G}^k_{t+\tau}(y_{t+\tau}(w_{t+\tau}), \mathcal{W}_{t+\tau}) \Big| \mathcal{W}_{t+\tau-1} \Big] \cdots \Big| \mathcal{W}_t \Big], \end{aligned} \tag{C.2}$$

the argument y is given by

$$y := \left\{ y_\varsigma(w_\varsigma) \;\middle|\; \forall \varsigma \in \{t, \ldots, t+\tau\}, \forall w_\varsigma \in \Omega_\varsigma \right\},$$

and the constraint $y \in \mathcal{Y}_t(x, w_t)$ enforces $y_t(w_t) \in \mathcal{X}_t(x, w_t)$, $y_{t+1}(w_{t+1}) \in \mathcal{X}_{t+1}(y_t(w_t), w_{t+1})$ and so on. For convenience, let

$$y^k_{t,\tau} := \widehat{\mathcal{H}}^k_{t,\tau}(x^k_{t-1}, \mathcal{W}^k_t), \tag{C.3}$$

where x^k_t denotes the iterate produced by the algorithm for stage t and sample path $\mathcal{W}^k_t$ during iteration k. An important relationship between x^k_t and $y^k_{t,\tau}$ is that $x^k_t = y^k_{t,0}$ for all

$t \in \{1, \ldots, T\}$ and $k \in \mathbb{Z}_+$. For each $t \in \{1, \ldots, T\}$, $\tau \in \{0, \ldots, T-t\}$, and $k \in \mathbb{Z}_+$, $y_{t,\tau}^k$ is referred to as the "level-τ" solution for the path $\mathcal{W}_t^k$ sampled during iteration k. Using this terminology, for each stage $t \in \{1, \ldots, T\}$, level-0 solutions are the ones produced by the algorithm while level-$(T-t)$ are exact solutions of the corresponding nested problem (4.1).

To prevent the notation from becoming more complicated, expressions of the form

$$\widehat{\mathcal{H}}_{t,\tau_1}^{k_1}(x, \mathcal{W}_t) - \widehat{\mathcal{H}}_{t,\tau_2}^{k_2}(x, \mathcal{W}_t)$$

for $\tau_1 \neq \tau_2$ are assumed to be valid despite the difference in dimension. In this case the subtraction is assumed to be over the vector of smallest size. Similarly, expressions of the form $L_{t,\tau_1}^{k_1}\big(\widehat{\mathcal{H}}_{t,\tau_2}^{k_2}(x, \mathcal{W}_t), \mathcal{W}_t\big)$ for $\tau_1 \neq \tau_2$ are also assumed to be valid. For $\tau_2 > \tau_1$, decisions in $\widehat{\mathcal{H}}_{t,\tau_2}^{k_2}(x, \mathcal{W}_t)$ beyond stage $t + \tau_1$ are ignored. For $\tau_2 < \tau_1$, decisions for stages $\{t + \tau_2 + 1, \ldots, t + \tau_1\}$ are assumed to be any feasible decisions. Lastly, the notation $\mathcal{O}(a_k)$ is used to express that the elements of a sequence of scalars indexed by k are uniformly bounded by a constant multiple of a_k for each $k \in \mathbb{Z}_+$.

In addition to the assumptions made in Section 4.2 about the properties of problem (4.1), the following assumptions are also made in order to analyze the convergence of the algorithm:

Assumption C.1 (Strong Convexity)**.** *The function $F_t(\cdot, w_t)$ is strongly convex for each $t \in \{1, \ldots, T\}$ and $w_t \in \Omega_t$.*

Assumption C.2 (Uncertainty)**.** *The exogenous random process $\{w_t\}_{t=1}^T$ has finite support and hence can be represented by a finite scenario tree. The children nodes of each node of the scenario tree are assumed to occur with equal probability for notation simplicity.*

Assumption C.3 (Parameters)**.** *The algorithm parameter γ_t used in the radial basis function (4.12) satisfies $\gamma_t = \infty$ for all $t \in \{1, \ldots, T\}$.*

Assumption C.4 (Initial Function Approximations)**.** *The initial function approximation $\widehat{G}_t(\cdot, \mathcal{W}_t)$ is continuously differentiable and convex for all $t \in \{1, \ldots, T-1\}$ and $\mathcal{W}_t$.*

Assumption C.5 (Step Lengths)**.** *The step lengths β_k satisfy $\beta_k \in (0, 1)$ for all $k \in \mathbb{Z}_+$, $\sum_{k=0}^{\infty} \beta_k = \infty$ almost surely, and $\sum_{k=0}^{\infty} \mathbb{E}\left[\beta_k^2\right] < \infty$.*

Assumption C.6 (Sampled Subgradients)**.** *The sampled subgradients ξ_t^k defined in (4.8) have norms that are bounded above by a constant multiple of $1 + \left\|g_{t+1}^k(\mathcal{W}_{t+1}^k)\right\|_2$ for all $t \in \{1, \ldots, T-1\}$ and $k \in \mathbb{Z}_+$.*

Assumption C.7 (Neighbor Subgradients)**.** *For each $t \in \{1, \ldots, T-1\}$ and $\tau \in \{0, \ldots, T-t-1\}$, there exists an M such that for each $k \in \mathbb{Z}_+$ there is a $\bar{\xi}^k_{t+\tau} \in \partial H^k_{t+\tau+1}\big(z^k_{t,\tau}(\mathcal{W}^k_{t+\tau}), \mathcal{W}^k_{t+\tau+1}\big)$ that satisfies*

$$\big\|\bar{\xi}^k_{t+\tau} - \xi^k_{t+\tau}\big\|_2 \leq M\big\|x^k_{t+\tau} - z^k_{t,\tau}(\mathcal{W}^k_{t+\tau})\big\|_2,$$

where $\xi^k_{t+\tau}$ is defined in (4.8), and $z^k_{t,\tau}(\mathcal{W}^k_{t+\tau})$ denotes the component of $y^k_{t,\tau}$ associated with stage $t+\tau$ and sample path $\mathcal{W}^k_{t+\tau}$.

Assumption C.8 (Inter-Stage Drift)**.** *For each $t \in \{1, \ldots, T-1\}$ and $\tau \in \{0, \ldots, T-t-1\}$, if the quantities $\big\|x^{k+1}_{t-1} - x^k_{t-1}\big\|_2$ and $\big\|\widehat{\mathcal{H}}^{k+1}_{t,\tau+1}(x^k_{t-1}, \mathcal{W}^k_t) - \widehat{\mathcal{H}}^k_{t,\tau+1}(x^k_{t-1}, \mathcal{W}^k_t)\big\|_2$ are $\mathcal{O}(\beta_k)$, then the sequences of partial sums $\sum_{k=0}^K \Gamma^k_{t,\tau}$, $\sum_{k=0}^K \Upsilon^k_{t,\tau}$, and $\sum_{k=0}^K \Psi^k_{t,\tau}$ are almost-surely bounded above, where*

$$\Gamma^k_{t,\tau} := L^k_{t,\tau}\big(\widehat{\mathcal{H}}^{k+1}_{t,\tau+1}(x^{k+1}_{t-1}, \mathcal{W}^k_t), \mathcal{W}^k_t\big) - L^k_{t,\tau}\big(\widehat{\mathcal{H}}^{k+1}_{t,\tau+1}(x^k_{t-1}, \mathcal{W}^k_t), \mathcal{W}^k_t\big) \tag{C.4}$$

$$\Upsilon^k_{t,\tau} := L^k_{t,\tau}\big(\widehat{\mathcal{H}}^{k+1}_{t,\tau+1}(x^k_{t-1}, \mathcal{W}^k_t), \mathcal{W}^k_t\big) - L^k_{t,\tau}\big(\widehat{\mathcal{H}}^k_{t,\tau+1}(x^k_{t-1}, \mathcal{W}^k_t), \mathcal{W}^k_t\big) \tag{C.5}$$

$$\Psi^k_{t,\tau} := L^k_{t,\tau}\big(\widehat{\mathcal{H}}^{k+1}_{t,\tau}(x^k_{t-1}, \mathcal{W}^k_t), \mathcal{W}^k_t\big) - L^k_{t,\tau}\big(\widehat{\mathcal{H}}^{k+1}_{t,\tau}(x^{k+1}_{t-1}, \mathcal{W}^k_t), \mathcal{W}^k_t\big). \tag{C.6}$$

Assumption C.9 (Slope Drift)**.** *For each $t \in \{1, \ldots, T-1\}$ and $\tau \in \{0, \ldots, T-t-1\}$, if the infinite sums $\sum_{k=0}^\infty \beta_k\big\|\bar{\xi}^k_{t+\tau} - \xi^k_{t+\tau}\big\|_2^2$ and $\sum_{k=0}^\infty \beta_k\big\|\nabla\widehat{G}^k_{t+\tau}\big(z^k_{t,\tau}(\mathcal{W}^k_{t+\tau}), \mathcal{W}^k_{t+\tau}\big) - \nabla\widehat{G}^k_{t+\tau}\big(x^k_{t+\tau}, \mathcal{W}^k_{t+\tau}\big)\big\|_2^2$ are finite almost surely, then the sequences of partial sums $\sum_{k=0}^K \Theta^k_{t,\tau}$ and $\sum_{k=0}^K \Xi^k_{t,\tau}$ are almost-surely bounded above, where*

$$\Theta^k_{t,\tau} := p^k_{t,\tau}\beta_k\big(\xi^k_{t+\tau} - \bar{\xi}^k_{t+\tau}\big)^T\Delta z^k_{t,\tau} \tag{C.7}$$

$$\Xi^k_{t,\tau} := p^k_{t,\tau}\beta_k\Big(\nabla\widehat{G}^k_{t+\tau}\big(z^k_{t,\tau}(\mathcal{W}^k_{t+\tau}), \mathcal{W}^k_{t+\tau}\big) - \nabla\widehat{G}^k_{t+\tau}\big(x^k_{t+\tau}, \mathcal{W}^k_{t+\tau}\big)\Big)^T\Delta z^k_{t,\tau}, \tag{C.8}$$

$p^k_{t,\tau}$ denotes the probability that $\mathcal{W}_{t+\tau} = \mathcal{W}^k_{t+\tau}$ given $\mathcal{W}^k_t$, $\Delta z^k_{t,\tau} := z^k_{t,\tau+1}(\mathcal{W}^k_{t+\tau}) - z^k_{t,\tau}(\mathcal{W}^k_{t+\tau})$, and the vectors $z^k_{t,\tau+1}(\mathcal{W}^k_{t+\tau})$ and $z^k_{t,\tau}(\mathcal{W}^k_{t+\tau})$ denote components of $y^k_{t,\tau+1}$ and $y^k_{t,\tau}$, respectively, associated with stage $t+\tau$ and sample path $\mathcal{W}^k_{t+\tau}$.

Assumption C.1 is a strong assumption that is not present in the convergence analyses of stochastic hybrid approximation algorithms for two-stage stochastic problems [21] (Appendix A) and problems with expected-value constraints (Appendix B). It is needed here in order to obtain that the "expanded" objective functions $L^k_{t,\tau}(\,\cdot\,, \mathcal{W}_t)$ are strongly convex, and that the "expanded" iterates $y^k_{t,\tau}$ defined in (C.3) get closer together as k increases. In practice, for problems of the form of (4.1) for which $F_t(\,\cdot\,, w_t)$ is not strongly convex,

small regularization terms can be added to the objective without significantly affecting the solution quality.

From Assumption C.2, the exogenous random process can be represented by a finite scenario tree. This tree can be characterized by some functions $\mathcal{N}$, $\mathcal{C}$ and $\mathcal{B}$ that give the nodes for each stage, the child nodes of each node, and the branches of a given length, respectively. The finiteness of the tree gives that for each $t \in \{1, \ldots, T\}$, there is a finite number of cost-to-go functions $G_t(\,\cdot\,, \mathcal{W}_t)$ (see equation (4.2)) parameterized by $\mathcal{W}_t \in \mathcal{B}(t)$ that need to be approximated. Since in a scenario tree, a t-length branch $\mathcal{W}_t = \{w_1, \ldots, w_t\}$ can be uniquely associated with its stage-t node w_t, the notation is simplified in the remaining part of this section and w_t is used instead of $\mathcal{W}_t$ as the argument for functions that depend on the realization of the random process up to stage t.

Assumption C.3 states that there is no generalization of information associated with a sample path to neighboring paths of the random process. This is justified since the scenario tree is finite and each scenario has a positive probability of being sampled. It follows from this assumption and (4.12) that the radial basis functions satisfy

$$\phi_t^k(w_t) = \begin{cases} 1 & \text{if } w_t = w_t^k, \\ 0 & \text{else,} \end{cases}$$

for each $t \in \{1, \ldots, T\}$, $k \in \mathbb{Z}_+$, and $w_t \in \mathcal{N}(t)$, where w_t^k is the stage-t node of the tree branch sampled at the beginning of iteration k. This and (4.11) give that the step lengths can be expressed as

$$\alpha_t^k(w_t) = \begin{cases} \beta_k & \text{if } w_t = w_t^k, \\ 0 & \text{else.} \end{cases} \tag{C.9}$$

Assumptions C.4, C.5 and C.6 are similar to those made in the convergence analysis of other stochastic hybrid approximation algorithms (Appendices A and B). One difference is that here convexity and not strong convexity is assumed for the initial function approximations $\widehat{G}_t(\,\cdot\,, w_t)$. The reason for this is that strong convexity here is already provided by $F_t(\,\cdot\,, w_t)$. The second difference is that the norm of the sampled subgradients ξ_t^k are allowed to increase with increasing $\big\|g_{t+1}^k(w_{t+1}^k)\big\|_2$. The reason for this is that $\xi_t^k \in \partial H_{t+1}^k\big(x_t^k, w_{t+1}^k\big)$ from (4.8), and $g_{t+1}^k\big(w_{t+1}^k\big)$ is a varying component that affects the objective function of the optimization problem that defines $H_{t+1}^k\big(\,\cdot\,, w_{t+1}^k\big)$.

Assumption C.7 is a technical condition that ensures that if the iterates produced by

the algorithm get close to the components of $y^k_{t,\tau}$, then subgradients associated with these points can be found that also get close to each other.

Lastly, Assumptions C.8 and C.9 are similar to the "drift" assumption used in the convergence analysis of the primal-dual stochastic hybrid approximation algorithm (Appendix B). Both assumptions involve obscure quantities but can be interpreted in simpler terms: Assumption C.8 can be interpreted as assuming that sums of $\mathcal{O}(\beta_k)$ changes between past and future stages, as measured with respect to the present stage, are bounded. In practice, a lack of synchronization or correlation among these uncontrolled changes at different iterations would suffice to make the sums bounded. Similarly, Assumption C.9 can be interpreted as assuming that slope errors, when they approach zero, they do so either fast enough or are not fully synchronized or directionally correlated to solution errors.

In the proofs of the following Lemmas, Corollaries and Theorems, the notation is further simplified by dropping node arguments w_t from the notation of functions whenever possible.

Lemma C.1. *The slope corrections $g^k_t(w_t)$ are uniformly bounded over all $t \in \{1, \dots, T\}$, $w_t \in \mathcal{N}(t)$, and $k \in \mathbb{Z}_+$.*

Proof. From (4.10) it holds that

$$\begin{aligned} g^{k+1}_t &= g^k_t + \alpha^k_t(\xi^k_t - \eta^k_t - g^k_t) \\ &= (1 - \alpha^k_t)g^k_t + \alpha^k_t(\xi^k_t - \eta^k_t). \end{aligned}$$

Hence,

$$\left\|g^{k+1}_t\right\|_2 \leq (1 - \alpha^k_t)\left\|g^k_t\right\|_2 + \alpha^k_t \left(\left\|\xi^k_t\right\|_2 + \left\|\eta^k_t\right\|_2\right).$$

Assumptions C.4 and C.6, and the compactness of the feasible sets then imply that there exist M_1 and M_2 such that

$$\left\|g^{k+1}_t\right\|_2 \leq (1 - \alpha^k_t)\left\|g^k_t\right\|_2 + \alpha^k_t \left(M_1 + M_1\left\|g^k_{t+1}(w^k_{t+1})\right\|_2 + M_2\right)$$

for all t and k. Since $g^k_T = 0$ for all k, it holds that

$$\left\|g^{k+1}_{T-1}\right\|_2 \leq (1 - \alpha^k_{T-1})\left\|g^k_{T-1}\right\|_2 + \alpha^k_{T-1}N_{T-1}$$

for all k, where $N_{T-1} := M_1 + M_2$. It follows from this and $0 \leq \alpha_t^k \leq 1$ that

$$\left\|g_{T-1}^k\right\|_2 \leq N_{T-1} \implies \left\|g_{T-1}^{k+1}\right\|_2 \leq N_{T-1}$$

for all k. Hence, $g_{T-1}^0 = 0$ and induction on k give that $\left\|g_{T-1}^k\right\|_2 \leq N_{T-1}$ for all k. Using this bound and repeating the analysis recursively for decreasing t gives that

$$\left\|g_t^k\right\|_2 \leq N_t \implies \left\|g_t^{k+1}\right\|_2 \leq N_t$$

for all k, where N_t is defined recursively by

$$N_t = \begin{cases} 0 & \text{if } t = T, \\ M_1 + M_1 N_{t+1} + M_2 & \text{else,} \end{cases}$$

for $t \in \{1, \ldots, T\}$. Hence, $g_t^0 = 0$ and induction on k give that $\left\|g_t^k\right\|_2 \leq N_t$ for all k and t. It can be concluded from this that $\left\|g_t^k\right\|_2 \leq \max\left\{ N_\tau \;\middle|\; \tau = 1, \ldots, T \right\}$ for all t and k. □

Corollary C.1. *The function approximations $\widehat{G}_t^k(\,\cdot\,, w_t)$ defined in* (C.1) *and their gradients are uniformly Lipschitz for stage-t feasible inputs over all $t \in \{1, \ldots, T\}$, $w_t \in \mathcal{N}(t)$, and $k \in \mathbb{Z}_+$.*

Proof. This follows directly from Lemma C.1, the compactness of the feasible sets, and Assumption C.4. □

Corollary C.2. *The slope correction vectors defined by $\Delta g_t^k := \xi_t^k - \eta_t^k - g_t^k(w_t^k)$ are uniformly bounded over all $t \in \{1, \ldots, T-1\}$ and $k \in \mathbb{Z}_+$.*

Proof. This follows directly from Lemma C.1, the compactness of the feasible sets, and Assumptions C.4 and C.6. □

Lemma C.2. *The functions $L_{t,\tau}^k(\,\cdot\,, w_t)$ defined by* (C.2) *are uniformly strongly convex over all $t \in \{1, \ldots, T\}$, $\tau \in \{0, \ldots, T-t\}$, $w_t \in \mathcal{N}(t)$, and $k \in \mathbb{Z}_+$.*

Proof. This follows directly from Assumptions C.1, C.2, and C.4. □

Lemma C.3. *There exists an M such that for all $t \in \{1, \ldots, T\}$, $\tau \in \{0, \ldots, T-t\}$, $w_t \in \mathcal{N}(t)$, $k \in \mathbb{Z}_+$, and stage-$(t-1)$ feasible x,*

$$\left\|\widehat{\mathcal{H}}_{t,\tau}^{k+1}(x, w_t) - \widehat{\mathcal{H}}_{t,\tau}^k(x, w_t)\right\|_2 \leq M\beta_k.$$

Proof. Letting $y_0 := \widehat{\mathcal{H}}_{t,\tau}^k(x, w_t)$ and $y_1 := \widehat{\mathcal{H}}_{t,\tau}^{k+1}(x, w_t)$, and using the definition of $\widehat{\mathcal{H}}_{t,\tau}^k$ and $L_{t,\tau}^k$ gives that there exist $\zeta_0 \in \partial L_{t,\tau}^k(y_0, w_t)$ and $\zeta_1 \in \partial L_{t,\tau}^{k+1}(y_1, w_t)$ such that $\zeta_0^T(y_1 - y_0) \geq 0$ and $\zeta_1^T(y_0 - y_1) \geq 0$. From (4.10), (C.1), and (C.2), it holds that

$$0 \leq \zeta_1^T(y_0 - y_1) = \bar{\zeta_1}^T(y_0 - y_1) + \mathbb{E}\left[\alpha_t^k(w_{t+\tau})\big(\Delta g_{t+\tau}^k\big)^T\big(z_0(w_{t+\tau}) - z_1(w_{t+\tau})\big) \,\middle|\, w_t \right],$$

where $\bar{\zeta_1} \in \partial L_{t,\tau}^k(y_1, w_t)$, $z_0(w_{t+\tau})$ and $z_1(w_{t+\tau})$ denote the components of y_0 and y_1, respectively, associated with node $w_{t+\tau}$, and the expectation is with respect to $w_{t+\tau}$ given w_t. Combining this and $\zeta_0^T(y_1 - y_0) \geq 0$ gives

$$\big(\bar{\zeta_1} - \zeta_0\big)^T(y_1 - y_0) \leq \mathbb{E}\left[\alpha_t^k(w_{t+\tau})\big(\Delta g_{t+\tau}^k\big)^T\big(z_0(w_{t+\tau}) - z_1(w_{t+\tau})\big) \,\middle|\, w_t \right].$$

From this, (C.9), Corollary C.2, Lemma C.2, and the compactness of the feasible sets, it follows that there exist positive constants C and M_1 independent of x, t, w_t, k and τ such that

$$\begin{aligned} C\|y_1 - y_0\|_2^2 &\leq \big(\bar{\zeta_1} - \zeta_0\big)^T(y_1 - y_0) \\ &\leq \mathbb{E}\left[\alpha_t^k(w_{t+\tau})\big(\Delta g_{t+\tau}^k\big)^T\big(z_0(w_{t+\tau}) - z_1(w_{t+\tau})\big) \,\middle|\, w_t \right] \\ &\leq M_1\beta_k\|y_1 - y_0\|_2, \end{aligned}$$

and hence that

$$\big\|\widehat{\mathcal{H}}_{t,\tau}^{k+1}(x, w_t) - \widehat{\mathcal{H}}_{t,\tau}^k(x, w_t)\big\|_2 \leq M_1\beta_k/C.$$

□

Lemma C.4. *The function $\widehat{\mathcal{H}}_{t,\tau}^k(\,\cdot\,, w_t)$ is uniformly Lipschitz continuous for stage-$(t-1)$ feasible inputs over all $t \in \{1, \ldots, T\}$, $\tau \in \{0, \ldots, T-t\}$, $w_t \in \mathcal{N}(t)$, and $k \in \mathbb{Z}_+$.*

Proof. From the assumptions of Section 4.2 and Lemma C.1, the function $\widehat{\mathcal{H}}_{t,\tau}^k(\,\cdot\,, w_t)$ is real-valued and convex on an open convex set that contains all stage-$(t-1)$ feasible points. Hence, it is continuous. Since the set of stage-$(t-1)$ feasible inputs is compact, the function is Lipschitz continuous on this set. From this, Lemma C.1, and the fact that there are finitely many t, τ, and w_t, there exists an M_1 such that for all t, τ, k, w_t, and stage-$(t-1)$ feasible inputs x_0 and x_1, it holds that

$$\big\|\widehat{\mathcal{H}}_{t,\tau}^k(x_1, w_t) - \widehat{\mathcal{H}}_{t,\tau}^k(x_0, w_t)\big\|_2 \leq M_1\|x_1 - x_0\|_2.$$

□

Lemma C.5. *There exists an M such that for all $t \in \{1, \dots, T\}$, $\tau \in \{0, \dots, T-t\}$, and $k \in \mathbb{Z}_+$,*

$$\left\| y_{t,\tau}^{k+1} - y_{t,\tau}^{k} \right\|_2 \leq M\beta_k,$$

where $y_{t,\tau}^k$ is as defined in (C.3).

Proof. For all $t \in \{1, \dots, T\}$, $\tau \in \{0, \dots, T-t\}$, and $k \in \mathbb{Z}_+$, it holds by definition that

$$\begin{aligned}\left\| y_{t,\tau}^{k+1} - y_{t,\tau}^{k} \right\|_2 &= \left\| \widehat{\mathcal{H}}_{t,\tau}^{k+1}(x_{t-1}^{k+1}) - \widehat{\mathcal{H}}_{t,\tau}^{k}(x_{t-1}^{k}) \right\|_2 \\ &\leq \left\| \widehat{\mathcal{H}}_{t,\tau}^{k+1}(x_{t-1}^{k+1}) - \widehat{\mathcal{H}}_{t,\tau}^{k+1}(x_{t-1}^{k}) \right\|_2 + \left\| \widehat{\mathcal{H}}_{t,\tau}^{k+1}(x_{t-1}^{k}) - \widehat{\mathcal{H}}_{t,\tau}^{k}(x_{t-1}^{k}) \right\|_2.\end{aligned}$$

From this and Lemmas C.3 and C.4, there exist positive constants M_1 and M_2 such that

$$\left\| y_{t,\tau}^{k+1} - y_{t,\tau}^{k} \right\|_2 \leq M_1 \|x_{t-1}^{k+1} - x_{t-1}^{k}\|_2 + M_2\beta_k.$$

Since $y_{t,0}^k = x_t^k$ for all k and t by definition, the above inequality gives that

$$\|x_t^{k+1} - x_t^k\|_2 \leq M_1 \|x_{t-1}^{k+1} - x_{t-1}^{k}\|_2 + M_2\beta_k.$$

Since, $x_0^{k+1} = x_0^k$ for all k, the above inequality implies that

$$\|x_t^{k+1} - x_t^k\|_2 \leq \sum_{\varsigma=0}^{t-1} M_1^{\varsigma} M_2 \beta_k \leq \sum_{\varsigma=0}^{T-1} M_1^{\varsigma} M_2 \beta_k.$$

It follows that

$$\left\| y_{t,\tau}^{k+1} - y_{t,\tau}^{k} \right\|_2 \leq \sum_{\varsigma=0}^{T} M_1^{\varsigma} M_2 \beta_k$$

for all t, τ, and k. □

Lemma C.6. *For each $t \in \{1, \dots, T-1\}$ and $\tau \in \{0, \dots, T-t-1\}$, the sequence $\{T_{t,\tau}^k\}_{k \in \mathbb{Z}_+}$ defined by*

$$T_{t,\tau}^k := L_{t,\tau}^k\big(y_{t,\tau+1}^k, w_t^k\big) - L_{t,\tau}^k\big(y_{t,\tau}^k, w_t^k\big)$$

satisfies

$$\mathbb{E}\left[T_{t,\tau}^{k+1} \,\middle|\, w_t^k \right] - T_{t,\tau}^k \leq \mathbb{E}\left[\Gamma_{t,\tau}^k + \Upsilon_{t,\tau}^k + \Psi_{t,\tau}^k + \Theta_{t,\tau}^k + \Xi_{t,\tau}^k \,\middle|\, w_t^k \right] + p_{t,\tau}^k \beta_k \big(L_{t,\tau+1}^k(y_{t,\tau+1}^k, w_t^k) - L_{t,\tau+1}^k(y_{t,\tau}^k, w_t^k) \big) + \mathcal{O}\left(\mathbb{E}\left[\beta_k^2 \,\middle|\, w_t^k \right] \right)$$

for all $k \in \mathbb{Z}_+$*, where* $p_{t,\tau}^k > 0$ *denotes the probability that* $w_{t+\tau} = w_{t+\tau}^k$ *given* w_t^k*.*

Proof. From (4.10) and the definitions of $T_{t,\tau}^k$ and $L_{t,\tau}^k$, it holds that

$$\begin{aligned} T_{t,\tau}^{k+1} - T_{t,\tau}^k &= L_{t,\tau}^{k+1}\big(y_{t,\tau+1}^{k+1}\big) - L_{t,\tau}^{k+1}\big(y_{t,\tau}^{k+1}\big) - L_{t,\tau}^k\big(y_{t,\tau+1}^k\big) + L_{t,\tau}^k\big(y_{t,\tau}^k\big) \\ &= L_{t,\tau}^k\big(y_{t,\tau+1}^{k+1}\big) - L_{t,\tau}^k\big(y_{t,\tau}^{k+1}\big) - L_{t,\tau}^k\big(y_{t,\tau+1}^k\big) + L_{t,\tau}^k\big(y_{t,\tau}^k\big) + \\ &\quad \mathbb{E}\left[\alpha_{t+\tau}^k(w_{t+\tau}) \big(\Delta g_{t+\tau}^k\big)^T \big(z_{t,\tau+1}^{k+1}(w_{t+\tau}) - z_{t,\tau}^{k+1}(w_{t+\tau}) \big) \,\middle|\, w_t^k \right], \end{aligned}$$

where $z_{t,\tau+1}^{k+1}(w_{t+\tau})$ and $z_{t,\tau}^{k+1}(w_{t+\tau})$ are the components of $y_{t,\tau+1}^{k+1}$ and $y_{t,\tau}^{k+1}$, respectively, associated with node $w_{t+\tau}$, $\Delta g_{t+\tau}^k$ is as defined in Corollary C.2, and the expectation is with respect to $w_{t+\tau}$ given w_t^k. From Corollary C.2, Lemma C.5, and (C.9), it follows that

$$\begin{aligned} T_{t,\tau}^{k+1} - T_{t,\tau}^k &\leq L_{t,\tau}^k\big(y_{t,\tau+1}^{k+1}\big) - L_{t,\tau}^k\big(y_{t,\tau}^{k+1}\big) - L_{t,\tau}^k\big(y_{t,\tau+1}^k\big) + L_{t,\tau}^k\big(y_{t,\tau}^k\big) + \\ &\quad \mathbb{E}\left[\alpha_{t+\tau}^k(w_{t+\tau}) \big(\Delta g_{t+\tau}^k\big)^T \big(z_{t,\tau+1}^k(w_{t+\tau}) - z_{t,\tau}^k(w_{t+\tau}) \big) \,\middle|\, w_t^k \right] + \\ &\quad \mathcal{O}\big(\beta_k^2\big). \end{aligned}$$

Then, using (C.3) and adding and subtracting terms $L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau+1}^{k+1}\big(x_{t-1}^k\big)\big)$ and $L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau}^{k+1}\big(x_{t-1}^k\big)\big)$, it holds that

$$\begin{aligned} T_{t,\tau}^{k+1} - T_{t,\tau}^k &\leq L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau+1}^{k+1}\big(x_{t-1}^{k+1}\big)\big) - L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau+1}^{k+1}\big(x_{t-1}^k\big)\big) + \\ &\quad L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau+1}^{k+1}\big(x_{t-1}^k\big)\big) - L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau+1}^k\big(x_{t-1}^k\big)\big) + \\ &\quad L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau}^k\big(x_{t-1}^k\big)\big) - L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau}^{k+1}\big(x_{t-1}^k\big)\big) + \\ &\quad L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau}^{k+1}\big(x_{t-1}^k\big)\big) - L_{t,\tau}^k\big(\widehat{\mathcal{H}}_{t,\tau}^{k+1}\big(x_{t-1}^{k+1}\big)\big) + \\ &\quad \mathbb{E}\left[\alpha_{t+\tau}^k(w_{t+\tau}) \big(\Delta g_{t+\tau}^k\big)^T \big(z_{t,\tau+1}^k(w_{t+\tau}) - z_{t,\tau}^k(w_{t+\tau}) \big) \,\middle|\, w_t^k \right] + \\ &\quad \mathcal{O}\big(\beta_k^2\big). \end{aligned}$$

Using (C.4), (C.5) and (C.6), and the inequality

$$L_{t,\tau}^{k}(\widehat{\mathcal{H}}_{t,\tau}^{k}(x_{t-1}^{k})) \leq L_{t,\tau}^{k}(\widehat{\mathcal{H}}_{t,\tau}^{k+1}(x_{t-1}^{k})),$$

which follows from the optimality of $\widehat{\mathcal{H}}_{t,\tau}^{k}\big(x_{t-1}^{k}\big)$ with respect to $L_{t,\tau}^{k}\big(\,\cdot\,, w_t^k\big)$, gives

$$\begin{aligned} T_{t,\tau}^{k+1} - T_{t,\tau}^{k} \leq \Gamma_{t,\tau}^{k} + \Upsilon_{t,\tau}^{k} + \Psi_{t,\tau}^{k} + \\ \mathbb{E}\left[\, \alpha_{t+\tau}^{k}(w_{t+\tau})\big(\Delta g_{t+\tau}^{k}\big)^{T}\big(z_{t,\tau+1}^{k}(w_{t+\tau}) - z_{t,\tau}^{k}(w_{t+\tau})\big) \,\middle|\, w_t^k \,\right] + \\ \mathcal{O}\big(\beta_k^2\big). \end{aligned}$$

The definitions of Δg_t^k and $\widehat{G}_t^k(\,\cdot\,, w_t)$ then give

$$\begin{aligned} T_{t,\tau}^{k+1} - T_{t,\tau}^{k} \leq \Gamma_{t,\tau}^{k} + \Upsilon_{t,\tau}^{k} + \Psi_{t,\tau}^{k} + \\ \mathbb{E}\left[\, \alpha_{t+\tau}^{k}(w_{t+\tau})\big(\xi_{t+\tau}^{k}\big)^{T}\big(z_{t,\tau+1}^{k}(w_{t+\tau}) - z_{t,\tau}^{k}(w_{t+\tau})\big) \,\middle|\, w_t^k \,\right] - \\ \mathbb{E}\left[\, \alpha_{t+\tau}^{k}(w_{t+\tau})\big(\nabla\widehat{G}_{t+\tau}^{k}\big(x_{t+\tau}^{k}, w_{t+\tau}^{k}\big)\big)^{T}\big(z_{t,\tau+1}^{k}(w_{t+\tau}) - z_{t,\tau}^{k}(w_{t+\tau})\big) \,\middle|\, w_t^k \,\right] + \\ \mathcal{O}\big(\beta_k^2\big), \end{aligned}$$

where $\xi_{t+\tau}^{k} \in \partial H_{t+\tau+1}^{k}\big(x_{t+\tau}^{k}, w_{t+\tau+1}^{k}\big)$, as defined in (4.8). From (C.9) and Assumption C.2, it follows that

$$\begin{aligned} T_{t,\tau}^{k+1} - T_{t,\tau}^{k} \leq \Gamma_{t,\tau}^{k} + \Upsilon_{t,\tau}^{k} + \Psi_{t,\tau}^{k} + \\ p_{t,\tau}^{k}\beta_k\big(\xi_{t+\tau}^{k}\big)^{T}\big(z_{t,\tau+1}^{k}\big(w_{t+\tau}^{k}\big) - z_{t,\tau}^{k}\big(w_{t+\tau}^{k}\big)\big) - \\ p_{t,\tau}^{k}\beta_k\big(\nabla\widehat{G}_{t+\tau}^{k}\big(x_{t+\tau}^{k}, w_{t+\tau}^{k}\big)\big)^{T}\big(z_{t,\tau+1}^{k}\big(w_{t+\tau}^{k}\big) - z_{t,\tau}^{k}\big(w_{t+\tau}^{k}\big)\big) + \\ \mathcal{O}\big(\beta_k^2\big), \end{aligned}$$

where $p_{t,\tau}^{k} > 0$ denotes the probability that $w_{t+\tau} = w_{t+\tau}^{k}$ given w_t^k. Adding and subtracting terms, and using (C.7) and (C.8), it holds that

$$\begin{aligned} T_{t,\tau}^{k+1} - T_{t,\tau}^{k} \leq \Gamma_{t,\tau}^{k} + \Upsilon_{t,\tau}^{k} + \Psi_{t,\tau}^{k} + \Theta_{t,\tau}^{k} + \Xi_{t,\tau}^{k} + \\ p_{t,\tau}^{k}\beta_k\big(\bar{\xi}_{t+\tau}^{k}\big)^{T}\big(z_{t,\tau+1}^{k}\big(w_{t+\tau}^{k}\big) - z_{t,\tau}^{k}\big(w_{t+\tau}^{k}\big)\big) - \\ p_{t,\tau}^{k}\beta_k\big(\nabla\widehat{G}_{t+\tau}^{k}\big(z_{t,\tau}^{k}\big(w_{t+\tau}^{k}\big), w_{t+\tau}^{k}\big)\big)^{T}\big(z_{t,\tau+1}^{k}\big(w_{t+\tau}^{k}\big) - z_{t,\tau}^{k}\big(w_{t+\tau}^{k}\big)\big) + \\ \mathcal{O}\big(\beta_k^2\big), \end{aligned}$$

where $\bar{\xi}^k_{t+\tau} \in \partial H^k_{t+\tau+1}\big(z^k_{t,\tau}(w^k_{t+\tau}), w^k_{t+\tau+1}\big)$, as in Assumption C.7. Taking expectation conditioned on w^k_t, and adding and subtracting terms gives

$$\begin{aligned}\mathbb{E}\left[T^{k+1}_{t,\tau} \,\middle|\, w^k_t\right] - T^k_{t,\tau} \leq \mathbb{E}\left[\Gamma^k_{t,\tau} + \Upsilon^k_{t,\tau} + \Psi^k_{t,\tau} + \Theta^k_{t,\tau} + \Xi^k_{t,\tau} \,\middle|\, w^k_t\right] \; + \\ p^k_{t,\tau}\beta_k\big(\zeta^k_{t,\tau+1}\big)^T\big(y^k_{t,\tau+1} - y^k_{t,\tau}\big) \; - \\ p^k_{t,\tau}\beta_k\big(\zeta^k_{t,\tau}\big)^T\big(y^k_{t,\tau+1} - y^k_{t,\tau}\big) \; + \\ \mathcal{O}\left(\mathbb{E}\left[\beta^2_k \,\middle|\, w^k_t\right]\right),\end{aligned}$$

where $\zeta^k_{t,\tau+1} \in \partial L^k_{t,\tau+1}(y^k_{t,\tau}, w^k_t)$ and $\zeta^k_{t,\tau} \in \partial L^k_{t,\tau}(y^k_{t,\tau}, w^k_t)$. We note here that for each term added and subtracted, the former was used to construct $p^k_{t,\tau}\beta_k\big(\zeta^k_{t,\tau+1}\big)^T\big(y^k_{t,\tau+1} - y^k_{t,\tau}\big)$, while the latter was used to construct $-p^k_{t,\tau}\beta_k\big(\zeta^k_{t,\tau}\big)^T\big(y^k_{t,\tau+1} - y^k_{t,\tau}\big)$. The chosen subgradient $\zeta^k_{t,\tau}$ obtained is assumed to be one such that, because of the optimality of $y^k_{t,\tau}$, results in $\big(\zeta^k_{t,\tau}\big)^T\big(y^k_{t,\tau+1} - y^k_{t,\tau}\big) \geq 0$. From this and the convexity of $L^k_{t,\tau+1}\big(\,\cdot\,, w^k_t\big)$, it follows that

$$\begin{aligned}\mathbb{E}\left[T^{k+1}_{t,\tau} \,\middle|\, w^k_t\right] - T^k_{t,\tau} \leq \mathbb{E}\left[\Gamma^k_{t,\tau} + \Upsilon^k_{t,\tau} + \Psi^k_{t,\tau} + \Theta^k_{t,\tau} + \Xi^k_{t,\tau} \,\middle|\, w^k_t\right] \; + \\ p^k_{t,\tau}\beta_k\big(L^k_{t,\tau+1}\big(y^k_{t,\tau+1}, w^k_t\big) - L^k_{t,\tau+1}\big(y^k_{t,\tau}, w^k_t\big)\big) \; + \\ \mathcal{O}\left(\mathbb{E}\left[\beta^2_k \,\middle|\, w^k_t\right]\right).\end{aligned}$$

□

Lemma C.7. *For each $t \in \{1, \ldots, T-1\}$ and $\tau \in \{0, \ldots, T-t-1\}$, if $\sum_{k=0}^{\infty}\beta_k\big\|x^k_{t+\tau} - z^k_{t,\tau}\big(w^k_{t+\tau}\big)\big\|_2^2$ is finite almost surely, where $z^k_{t,\tau}\big(w^k_{t+\tau}\big)$ denotes the components of $y^k_{t,\tau}$ associated with node $w^k_{t+\tau}$, then $\sum_{k=0}^{\infty}\beta_k\big\|y^k_{t,\tau} - y^k_{t,\tau+1}\big\|_2^2$ is finite almost surely.*

Proof. From Lemma C.6, it holds that

$$\begin{aligned}\mathbb{E}\left[T^{k+1}_{t,\tau} \,\middle|\, w^k_t\right] - T^k_{t,\tau} \leq \mathbb{E}\left[\Gamma^k_{t,\tau} + \Upsilon^k_{t,\tau} + \Psi^k_{t,\tau} + \Theta^k_{t,\tau} + \Xi^k_{t,\tau} \,\middle|\, w^k_t\right] \; + \\ p^k_{t,\tau}\beta_k\big(L^k_{t,\tau+1}\big(y^k_{t,\tau+1}, w^k_t\big) - L^k_{t,\tau+1}\big(y^k_{t,\tau}, w^k_t\big)\big) \; + \\ \mathcal{O}\left(\mathbb{E}\left[\beta^2_k \,\middle|\, w^k_t\right]\right).\end{aligned}$$

Taking expectation, rearranging, and summing over k from 0 to K gives

$$p_{t,\tau}\mathbb{E}\left[\sum_{k=0}^{K}\beta_k\big(L_{t,\tau+1}^k(y_{t,\tau}^k, w_t^k) - L_{t,\tau+1}^k(y_{t,\tau+1}^k, w_t^k)\big)\right] \leq \mathbb{E}\left[T_{t,\tau}^0 - T_{t,\tau}^{K+1}\right] + \mathbb{E}\left[\sum_{k=0}^{K}\left(\Gamma_{t,\tau}^k + \Upsilon_{t,\tau}^k + \Psi_{t,\tau}^k\right)\right] + \mathbb{E}\left[\sum_{k=0}^{K}\left(\Theta_{t,\tau}^k + \Xi_{t,\tau}^k\right)\right] + \mathcal{O}\left(\mathbb{E}\left[\sum_{k=0}^{K}\beta_k^2\right]\right),$$

where $p_{t,\tau} := \min_{k\in\mathbb{Z}_+} p_{t,\tau}^k > 0$. If $\sum_{k=0}^{\infty}\beta_k\big\|x_{t+\tau}^k - z_{t,\tau}^k\big(w_{t+\tau}^k\big)\big\|_2^2$ is finite almost surely then this together with Assumptions C.5, C.7, C.8 and C.9, Corollary C.1, Lemmas C.3 and C.5, and the uniform boundedness of $T_{t,\tau}^k$ imply that there exists an M_1 such that

$$\sum_{k=0}^{K}\mathbb{E}\left[\beta_k\big(L_{t,\tau+1}^k(y_{t,\tau}^k, w_t^k) - L_{t,\tau+1}^k(y_{t,\tau+1}^k, w_t^k)\big)\right] < M_1 < \infty$$

for all $K \in \mathbb{Z}_+$. From the uniform strong convexity of $L_{t,\tau+1}^k(\,\cdot\,, w_t^k)$ (Lemma C.2), and the optimality of $y_{t,\tau+1}^k$ with respect to $L_{t,\tau+1}^k\big(\,\cdot\,, w_t^k\big)$, it holds that there exists an $M_2 > 0$ such that

$$M_2\big\|y_{t,\tau}^k - y_{t,\tau+1}^k\big\|_2^2 \leq L_{t,\tau+1}^k(y_{t,\tau}^k, w_t^k) - L_{t,\tau+1}^k(y_{t,\tau+1}^k, w_t^k)$$

for all $k \in \mathbb{Z}_+$. It follows that

$$\sum_{k=0}^{\infty}\mathbb{E}\left[\beta_k\big\|y_{t,\tau}^k - y_{t,\tau+1}^k\big\|_2^2\right] < M_1/M_2 < \infty,$$

and hence that $\sum_{k=0}^{\infty}\beta_k\big\|y_{t,\tau}^k - y_{t,\tau+1}^k\big\|_2^2$ is finite almost surely. □

Lemma C.8. *For each $t \in \{1,\ldots,T-1\}$ and $\tau \in \{0,\ldots,T-t-1\}$, $\sum_{k=0}^{\infty}\beta_k\big\|y_{t,\tau}^k - y_{t,\tau+1}^k\big\|_2^2$ is finite almost surely.*

Proof. For $t \in \{1,\ldots,T-1\}$, it holds by definition that $x_t^k = z_{t,0}^k\big(w_t^k\big)$, where $z_{t,\tau}^k(w)$ denotes the component of $y_{t,\tau}^k$ associated with node w. Hence, $\sum_{k=0}^{\infty}\beta_k\big\|x_t^k - z_{t,0}^k\big(w_t^k\big)\big\|_2^2 = 0$ and Lemma C.7 implies that $\sum_{k=0}^{\infty}\big\|y_{t,0}^k - y_{t,1}^k\big\|_2^2$ is finite almost surely. Now let $\tau > 0$ and

assume that for each ς such that $0 \leq \varsigma < \tau$, $\sum_{k=0}^{\infty} \|y_{t,\varsigma}^k - y_{t,\varsigma+1}^k\|_2^2$ is finite almost surely. From Lemma C.4 and $\|x_t^k - z_{t,0}^k(w_t^k)\|_2 = 0$, it follows that there exists an $M_1 > 1$ such that

$$\begin{aligned}
\|x_{t+\tau}^k - z_{t,\tau}^k(w_{t+\tau}^k)\|_2 &= \|\widehat{\mathcal{H}}_{t+\tau,0}^k(x_{t+\tau-1}^k, w_{t+\tau}^k) - \widehat{\mathcal{H}}_{t+\tau,0}^k(z_{t,\tau}^k(w_{t+\tau-1}^k), w_{t+\tau}^k)\|_2 \\
&\leq M_1\|x_{t+\tau-1}^k - z_{t,\tau}^k(w_{t+\tau-1}^k)\|_2 \\
&\leq M_1\|x_{t+\tau-1}^k - z_{t,\tau-1}^k(w_{t+\tau-1}^k)\|_2 + M_1\|z_{t,\tau-1}^k(w_{t+\tau-1}^k) - z_{t,\tau}^k(w_{t+\tau-1}^k)\|_2 \\
&\vdots \\
&\leq \sum_{\varsigma=0}^{\tau-1} M_1^{\tau-\varsigma}\|z_{t,\varsigma}^k(w_{t+\varsigma}^k) - z_{t,\varsigma+1}^k(w_{t+\varsigma}^k)\|_2 \\
&\leq M_1^\tau \sum_{\varsigma=0}^{\tau-1} \|z_{t,\varsigma}^k(w_{t+\varsigma}^k) - z_{t,\varsigma+1}^k(w_{t+\varsigma}^k)\|_2 \\
&\leq M_1^\tau \sum_{\varsigma=0}^{\tau-1} \|y_{t,\varsigma}^k - y_{t,\varsigma+1}^k\|_2
\end{aligned}$$

for all $k \in \mathbb{Z}_+$. From this and the equivalence of norms in finite-dimensional spaces, there exists an $M_2 > 0$ such that

$$\|x_{t+\tau}^k - z_{t,\tau}^k(w_{t+\tau}^k)\|_2 \leq M_1^\tau M_2 \sqrt{\sum_{\varsigma=0}^{\tau-1} \|y_{t,\varsigma}^k - y_{t,\varsigma+1}^k\|_2^2}$$

for all $k \in \mathbb{Z}_+$, and hence that

$$\sum_{k=0}^{\infty} \beta_k \|x_{t+\tau}^k - z_{t,\tau}^k(w_{t+\tau}^k)\|_2^2 \leq (M_1^\tau M_2)^2 \sum_{\varsigma=0}^{\tau-1} \sum_{k=0}^{\infty} \beta_k \|y_{t,\varsigma}^k - y_{t,\varsigma+1}^k\|_2^2.$$

This and the induction hypothesis then gives that $\sum_{k=0}^{\infty} \beta_k \|x_{t+\tau}^k - z_{t,\tau}^k(w_{t+\tau}^k)\|_2^2$ is finite almost surely. Lemma C.7 then gives that $\sum_{k=0}^{\infty} \beta_k \|y_{t,\tau}^k - y_{t,\tau+1}^k\|_2^2$ is finite almost surely. By strong induction, this holds for all $\tau \in \{0, \ldots, T-t-1\}$. □

Theorem C.1. *There exists a set of strictly increasing indices denoted by* $\{ n_k \in \mathbb{Z}_+ \mid k \in \mathbb{Z}_+ \}$ *such that for all* $t \in \{1, \ldots, T\}$, $\|y_{t,0}^{n_k} - y_{t,T-t}^{n_k}\|_2 \to 0$ *almost surely as* $k \to \infty$.

Proof. For $t = T$, the statement is trivial. For $t \in \{1, \ldots, T-1\}$, Lemma C.8 gives that the infinite sum $\sum_{k=0}^{\infty} \beta_k \|y_{t,\tau}^k - y_{t,\tau+1}^k\|_2^2$ is finite almost surely for each $\tau \in \{0, \ldots, T-t-1\}$.

Hence,

$$\sum_{k=0}^{\infty} \beta_k \sum_{t=1}^{T-1} \sum_{\tau=0}^{T-t-1} \left\| y_{t,\tau}^k - y_{t,\tau+1}^k \right\|_2^2 < \infty$$

almost surely. Since $\sum_{k=0}^{\infty} \beta_k = \infty$ almost surely from Assumption C.5, it follows that there exists a set of strictly increasing indices $\{ n_k \in \mathbb{Z}_+ \mid k \in \mathbb{Z}_+ \}$ such that

$$\sum_{t=1}^{T-1} \sum_{\tau=0}^{T-t-1} \left\| y_{t,\tau}^{n_k} - y_{t,\tau+1}^{n_k} \right\|_2^2 \to 0$$

almost surely as $k \to \infty$. It follows from this that $\left\| y_{t,\tau}^{n_k} - y_{t,\tau+1}^{n_k} \right\|_2 \to 0$ almost surely for $t \in \{1, \ldots, T-1\}$ and $\tau \in \{0, \ldots, T-t-1\}$ as $k \to \infty$. This combined with

$$\left\| y_{t,0}^{n_k} - y_{t,T-t}^{n_k} \right\|_2 \leq \sum_{\tau=0}^{T-t-1} \left\| y_{t,\tau}^{n_k} - y_{t,\tau+1}^{n_k} \right\|_2$$

gives that $\left\| y_{t,0}^{n_k} - y_{t,T-t}^{n_k} \right\|_2 \to 0$ almost surely as $k \to \infty$. □

Bibliography

[1] S. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[2] T. Asamov and W. Powell. Regularized decomposition of high-dimensional multistage stochastic programs with Markov uncertainty. *arXiv:1505.02227*, May 2015.

[3] J. Atlason, M. Epelman, and S. Henderson. Call center staffing with simulation and cutting plane methods. *Annals of Operations Research*, 127(1-4):333–358, 2004.

[4] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[5] D. Bertsimas, E. Litvinov, X. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, February 2013.

[6] R. Bessa, C. Moreira, B. Silva, and M. Matos. Handling renewable energy variability and uncertainty in power systems operation. *Wiley Interdisciplinary Reviews: Energy and Environment*, 3(2):156–178, 2014.

[7] S. Bhatnagar, N. Hemachandra, and V. Mishra. Stochastic approximation algorithms for constrained optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, 21(3):1–22, February 2011.

[8] S. Bhatnagar, H. Prasad, and L. Prashanth. *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*, chapter Stochastic Approximation Algorithms, pages 17–28. Springer London, 2013.

[9] J. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007, September 1985.

[10] J. Birge and F. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392, 1988.

[11] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2011.

[12] J. Birge and H. Tang. L-shaped method for two stage problems of stochastic convex programming. Technical report, University of Michigan, Department of Industrial and Operations Engineering, August 1993.

[13] Black and Veatch. Cost and performance data for power generation technologies. Technical report, Prepared for the National Renewable Energy Laboratory, 2012. `https://www.bv.com/docs/reports-studies/nrel-cost-report.pdf`.

[14] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier and G. Saporta, editors, *Proceedings of Computational Statistics*, pages 177–186. Physica-Verlag HD, 2010.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, January 2011.

[16] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[17] R. Byrd, G. Chin, J. Nocedal, and Y. Wu. Sample size selection in optimization methods for machine learning. *Mathematical Programming*, 134(1):127–155, 2012.

[18] Y. Chen, A. Casto, X. Wang, J. Wan, and F. Wang. Day-ahead market clearing software performance improvement. FERC Technical Conference on Increasing Real-Time and Day-Ahead Market Efficiency through Improved Software, 2015. `https://www.ferc.gov/june-tech-conf/2015/abstracts/m1-2.html`.

[19] Y. Chen, T. Davis, W. Hager, and S. Rajamanickam. Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software*, 35(3):1–14, October 2008.

[20] Z. Chen and W. Powell. Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse. *Journal of Optimization Theory and Applications*, 102(3):497–524, 1999.

[21] R. Cheung and W. Powell. SHAPE - a stochastic hybrid approximation procedure for two-stage stochastic programs. *Operations Research*, 48(1):73–79, 2000.

[22] H. Dai, N. Zhang, and W. Su. A literature review of stochastic programming and unit commitment. *Journal of Power and Energy Engineering*, 3:206–214, 2015.

[23] D. Dentcheva and W. R omisch. Optimal power generation under uncertainty via stochastic programming. In K. Marti and P. Kall, editors, *Stochastic Programming Methods and Technical Applications*, volume 458 of *Lecture Notes in Economics and Mathematical Systems*, pages 22–56. Springer Berlin Heidelberg, 1998.

[24] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[25] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.

[26] C. Donohue and J. Birge. The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithmic Operations Research*, 1(1), 2006.

[27] J. Espejo-Uribe. Analysis of adaptive certainty-equivalent techniques for the stochastic unit commitment problem. Master's thesis, Swiss Federal Institute of Technology (ETH), Zurich, April 2017.

[28] European Commission. Launching the public consultation process on a new energy market design. Technical Report SWD(2015) 142 Final, January 2015. `https://ec.europa.eu/energy/sites/ener/files/publication/web_1_EN_ACT_part1_v11_en.pdf`.

[29] H. Gangammanavar, S. Sen, and V. Zavala. Stochastic optimization of sub-hourly economic dispatch with wind energy. *IEEE Transactions on Power Systems*, PP(99):1–11, March 2015.

[30] A. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[31] P. Girardeau, V. Leclere, and A. Philpott. On the convergence of decomposition methods for multistage stochastic convex programs. *Mathematics of Operations Research*, 40(1):130–145, July 2014.

[32] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Li, R. Mukerji, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour, and C. Singh. The ieee reliability test system-1996. *IEEE Transactions on Power Systems*, 14(3):1010–1020, August 1999.

[33] Gurobi Optimization. Gurobi optimizer reference manual, 2016. `http://www.gurobi.com`.

[34] J. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research*, 16(3):650–669, 1991.

[35] J. Higle and S. Sen. Finite master programs in regularized stochastic decomposition. *Mathematical Programming*, 67(1):143–168, 1994.

[36] T. Homem De Mello and G. Bayraksan. Monte Carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85, 2014.

[37] International Energy Agency. System integration of renewables: Implications for electricity security. Technical report, G7 Germany, February 2016.

[38] E. Jones, T. Oliphant, and P. Peterson. SciPy: Open source scientific tools for Python. `http://www.scipy.org`, 2001.

[39] C. Josz, S. Fliscounakis, J. Maeght, and P. Panciatici. AC power flow data in MATPOWER and QCQP format: iTesla, RTE snapshots, and PEGASE, 2016. arXiv:1603.01533.

[40] A. King and R. Rockafellar. Asymptotic theory for solutions in statistical estimation and stochastic programming. *Mathematics of Operations Research*, 18(1):148–162, February 1993.

[41] H. Kushner and D. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Applied Mathematical Sciences. Springer-Verlag, 1978.

[42] H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Stochastic Modelling and Applied Probability. Springer-Verlag, 2003.

[43] K. Lau and R. Womersley. Multistage quadratic stochastic programming. *Journal of Computational and Applied Mathematics*, 129(12):105–138, 2001. Nonlinear Programming and Variational Inequalities.

[44] J. Lavaei and S. Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1):92–107, February 2012.

[45] Y. Lee and R. Baldick. A frequency-constrained stochastic economic dispatch model. *IEEE Transactions on Power Systems*, 28(3):2301–2312, August 2013.

[46] F. Louveaux. Piecewise convex programs. *Mathematical Programming*, 15(1):53–62, 1978.

[47] F. Louveaux. A solution method for multistage stochastic programs with recourse with application to an energy investment problem. *Operations Research*, 28(4):889–902, July 1980.

[48] R. Ma, Y. Huang, and M. Li. Unit commitment optimal research based on the improved genetic algorithm. In *Fourth International Conference on Intelligent Computation Technology and Automation*, volume 1, pages 291–294, March 2011.

[49] M. Milligan and B. Kirby. Utilizing load response for wind and solar integration and power system reliability. In *Wind Power Conference*, pages 1–18, July 2010.

[50] E. Moulines and F. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 451–459. Curran Associates, Inc., 2011.

[51] J. Mulvey and A. Ruszczyński. A new scenario decomposition method for large-scale stochastic optimization. *Operations Research*, 43(3):477–490, 1995.

[52] National Renewable Energy Laboratory. The importance of flexible electricity supply. Technical Report GO-102011-3201, United States Department of Energy, 2011.

[53] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[54] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

[55] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer US, 2004.

[56] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[57] North American Electric Reliability Corporation. Balancing and frequency control. Technical report, Prepared by the NERC Resources Subcommittee, January 2011.

[58] M. Nowak and W. Römisch. Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. *Annals of Operations Research*, 100(1):251–272, 2000.

[59] S. Oren. Renewable energy integration and the impact of carbon regulation on the electric grid. In *IEEE Power and Energy Society General Meeting*, pages 1–2, July 2012.

[60] A. Ott. Evolution of computing requirements in the PJM market: Past and future. In *IEEE PES General Meeting*, pages 1–4, July 2010.

[61] A. Papavasiliou, S. Oren, and B. Rountree. Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration. *IEEE Transactions on Power Systems*, 30(3):1109–1120, May 2015.

[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[63] M. Pereira and L. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(1):359–375, May 1991.

[64] D. Phan and S. Ghosh. Two-stage stochastic optimization for optimal power flow under renewable generation uncertainty. *ACM Transactions on Modeling and Computer Simulation*, 24(1):1–22, January 2014.

[65] A. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.

[66] W. Powell. The optimizing-simulator: Merging simulation and optimization using approximate dynamic programming. In *Winter Simulation Conference*, pages 96–109, 2005.

[67] W. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley Series in Probability and Statistics. Wiley, 2011.

[68] W. Powell, A. Ruszczyński, and H. Topaloglu. Learning algorithms for separable approximations of discrete stochastic optimization problems. *Mathematics of Operations Research*, 29(4):814–836, 2004.

[69] W. Powell and H. Topaloglu. Stochastic programming in transportation and logistics. In *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 555–635. Elsevier, 2003.

[70] R. Rajagopal, E. Bitar, P. Varaiya, and F. Wu. Risk-limiting dispatch for integrating renewable power. *International Journal of Electrical Power and Energy Systems*, 44(1):615–628, January 2013.

[71] S. Rebennack. Combining sampling-based and scenario-based nested Benders decomposition methods: application to stochastic dual dynamic programming. *Mathematical Programming*, 156(1):343–389, March 2016.

[72] REN21. Renewables global futures report, 2013. `http://www.ren21.net/REN21Activities/GlobalFuturesReport.aspx`.

[73] L. Roald, S. Misra, T. Krause, and G. Andersson. Corrective control to handle forecast uncertainty: A chance constrained optimal power flow. *IEEE Transactions on Power Systems*, 32(2):1626–1637, March 2017.

[74] L. Roald, F. Oldewurtel, T. Krause, and G. Andersson. Analytical reformulation of security constrained optimal power flow with probabilistic constraints. In *IEEE PowerTech Conference*, pages 1–6, June 2013.

[75] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[76] R. Rockafellar. *Convex Analysis*. Princeton landmarks in mathematics and physics. Princeton University Press, 1997.

[77] R. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, pages 1443–1471, 2002.

[78] R. Rockafellar and Roger J. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, February 1991.

[79] L. Rosasco, S. Villa, and B. Vũ. Convergence of stochastic proximal gradient algorithm. *ArXiv e-prints*, March 2014.

[80] P. Ruiz, C. Philbrick, E. Zak, K. Cheung, and P. Sauer. Uncertainty management in the unit commitment problem. *IEEE Transactions on Power Systems*, 24(2):642–651, May 2009.

[81] T. Samad and E. Koch. Automated demand response for energy efficiency and emissions reduction. In *IEEE Transmission and Distribution Conference and Exposition*, pages 1–3, May 2012.

[82] S. Sen. Stochastic programming: Computational issues and challenges. *Encyclopedia of operations research and management science*, pages 1–11, 2001.

[83] S. Sen and Z. Zhou. Multistage stochastic decomposition: A bridge between stochastic programming and approximate dynamic programming. *SIAM Journal on Optimization*, 24(1):127–153, 2014.

[84] A. Shapiro. Asymptotic behavior of optimal solutions in stochastic programming. *Mathematics of Operations Research*, 18(4):829–845, 1993.

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

[86] A. Shapiro, W. Tekaya, J. Da Costa, and M. Pereira. Risk neutral and risk averse stochastic dual dynamic programming method. *European Journal of Operational Research*, 224(2):375–391, January 2013.

[87] P. Singhal and R. Sharma. Dynamic programming approach for large scale unit commitment problem. In *International Conference on Communication Systems and Network Technologies*, pages 714–717, June 2011.

[88] J. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control*, 45(10):1839–1853, October 2000.

[89] M. Tahanan, W. Van Ackooij, A. Frangioni, and F. Lacalandra. Large-scale unit commitment under uncertainty. *4OR*, 13(2):115–171, 2015.

[90] T. Tinoco De Rubira and G. Hug. Adaptive certainty-equivalent approach for optimal generator dispatch under uncertainty. In *European Control Conference (ECC)*, pages 1215–1222, June 2016.

[91] T. Tinoco De Rubira and G. Hug. Primal-dual stochastic hybrid approximation algorithm. *Computational Optimization and Applications (under review)*, January 2017.

[92] T. Tinoco De Rubira, L. Roald, and G. Hug. Multi-stage stochastic optimization via parameterized stochastic hybrid approximation. *Journal of Optimization Theory and Applications (under review)*, June 2017.

[93] P. Toulis and E. Airoldi. Scalable estimation strategies based on stochastic approximations: classical results and new insights. *Statistics and Computing*, 25(4):781–795, 2015.

[94] J. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1):59–94, January 1996.

[95] A. Tuohy, P. Meibom, E. Denny, and M. O'Malley. Unit commitment for systems with significant wind penetration. *IEEE Transactions on Power Systems*, 24(2):592–601, May 2009.

[96] W. Van Ackooij. Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit commitment. *Mathematical Methods of Operations Research*, 80(3):227–253, 2014.

[97] S. Van Der Walt, S. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30, March 2011.

[98] R. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

[99] S. Wallace and S. Fleten. Stochastic programming models in energy. In *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 637–677. Elsevier, 2003.

[100] J. Wang, J. Wang, C. Liu, and J. Ruiz. Stochastic unit commitment with sub-hourly dispatch constraints. *Applied Energy*, 105:418–422, 2013.

[101] W. Wang and S. Ahmed. Sample average approximation of expected value constrained stochastic programs. *Operations Research Letters*, 36(5):515–519, 2008.

[102] R. Wets. Stochastic programs with fixed recourse: The equivalent deterministic program. *SIAM Review*, 16(3):309–339, 1974.

[103] R. Wets. Chapter VIII: Stochastic programming. In *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science*, pages 573–629. Elsevier, 1989.

[104] L. Wu, M. Shahidehpour, and T. Li. Stochastic security-constrained unit commitment. *IEEE Transactions on Power Systems*, 22(2):800–811, May 2007.

[105] R. Yang and G. Hug. Potential and efficient computation of corrective power flow control in cost vs. risk trade-off. *IEEE Transactions on Smart Grid*, 5(4):2033–2043, July 2014.

[106] F. Yousefian, A. Nedic, and U. Shanbhag. On stochastic gradient and subgradient methods with adaptive steplength sequences. *Automatica*, 48(1):56–67, 2012.

[107] J. Zhang and D. Zheng. A stochastic primal-dual algorithm for joint flow control and MAC design in multi-hop wireless networks. In *Conference on Information Sciences and Systems*, pages 339–344, March 2006.

[108] J. Zhu. *Optimization of Power System Operation.* John Wiley & Sons, Inc., 2009.

[109] R. Zimmerman, C. Murillo-Sánchez, and R. Thomas. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, February 2011.