ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

eeh power systems laboratory

José Sebastián Espejo-Uribe

Analysis of Adaptive Certainty-Equivalent Techniques for the Stochastic Unit Commitment Problem

Master Thesis
PSL1707

EEH – Power Systems Laboratory
Swiss Federal Institute of Technology (ETH) Zurich

Examiner: Prof. Dr. Gabriela Hug
Supervisor: Dr. Tomás Tinoco De Rubira

Zurich, April 21, 2017

Abstract

The unit commitment problem aims to schedule the most cost-effective combination of generating units to meet the demand while taking into account technical and operational constraints. With the projected large-scale penetration of renewable energy sources, in particular wind and solar, which are highly variable and intermittent, uncertainty must be properly considered. In this thesis, the applicability and performance of the Adaptive Certainty-Equivalent algorithm for solving the stochastic unit commitment problem with large penetration of renewable energy is studied. This approach has been previously applied to the stochastic economic dispatch problem with promising results, but the presence of binary restrictions in the stochastic unit commitment problem pose both practical and theoretical challenges that warrant investigation. It is shown in this work that despite a lack of theoretical guarantees, the Adaptive Certainty-Equivalent algorithm can find commitment schedules in reasonable times that are better than those obtained with benchmark algorithms.

Contents

List of Acronyms

AdaCE	Adaptive Certainty-Equivalent
CE	Certainty-Equivalent
ED	Economic Dispatch
LP	Linear Program
MILP	Mixed-Integer Linear Programming
MIQP	Mixed-Integer Quadratic Programming
MINLP	Mixed-Integer Non-Linear Programming
MISO	Midcontinent Independent System Operator
NR	Noise Reduction
SAA	Sample-Average Approximation
SHA	Stochastic Hybrid Approximation
UC	Unit Commitment

List of Symbols

Latin Letters

a_k	Constant term of under estimator at iteration k
A	Sparse matrix
b_k	Linear term of under estimator at iteration k
c_i^d	Shut-down cost of generator i
c_i^u	Start-up cost of generator i
d	Vector of supplied load powers
$\bar{d}$	Vector of requested load powers
D	Sparse matrix
$\mathbb{E}[\cdot]$	Expectation with respect to w
F	First stage start-up and shut-down cost function
J	Sparse matrix
k	Algorithm iterations
L	Sparse matrix
m	Number of realizations drawn of renewable power vectors
n	Number of conventional generation units
n_r	Number of renewable sources
N	Node distance parameter
p	Vector of generator output powers
$p_{i,t}$	Output power of generator i during time t
$p^{\max}$	Vector of maximum generator power limits
$p^{\min}$	Vector of minimum generator power limits
P	Sparse matrix
Q	Recourse function that quantifies total operating cost

$\bar{Q}$	Recourse function with piece-wise linear generating costs
$\mathcal{Q}^0$	Initial function approximation
$\mathcal{Q}^k$	Function approximation at iteration k
r	Number of linear segments
s	Vector of utilized powers from renewable sources
S	Sparse matrix
T	Time horizon
T_i^u	Minimum up time for generator i
T_i^d	Minimum down time for generator i
u	Vector of on-off states of generators
$u_{i,0}$	Initial on-off state of generator i
$u_{i,t}$	On-off state of generator i during time t
u_k	Candidate solution at iteration k
$\mathcal{U}$	Set of minimum up and down time restrictions
w	Vector of powers from renewable energy sources
$w^{\max}$	Vector of power capacities of renewable energy sources
w_k	Sample of random vector w drawn at iteration k
$w_{k,i}$	One of many samples of random vector w drawn at iteration k
$\bar{w}_t$	Vector of non-negative base powers
x	Second-stage cost epigraph auxiliary variable
$y_{i,t}$	Start-up cost epigraph auxiliary variable
$z_{i,t}$	Shut-down cost epigraph auxiliary variable
$z^{\max}$	Vector of maximum branch flow limits
$z^{\min}$	Vector of minimum branch flow limits

Greek Letters

α_k	Step length at iteration k
$\beta_{i,j}$	Constant of piecewise linear approximation
γ	Load not serve cost
$\delta^{\max}$	Vector of maximum generator ramping limits
$\delta^{\min}$	Vector of minimum generator ramping limits

$\zeta_{i,j}$	Constant of piecewise linear approximation
θ	Vector of bus voltage angles
μ	Expectation of noisy subgradient
μ_i^k	Lagrange multiplier
ξ_k	Noisy subgradient of recourse function at iteration k
$\xi_{k,i}$	One of many noisy subgradients of recourse function at iteration k
π_i^k	Lagrange multiplier
Π_c	Projection operator
ρ	Correlation coefficient parameter
Σ	Covariance of noisy subgradient
Σ_m	Covariance of averaged noisy subgradients
Σ_t	Covariance of renewable powers
$\bar{\Sigma}_t$	Positive semidefinite matrix
φ	Generation cost function
$\bar{\varphi}$	Piece-wise linear generation cost function

Chapter 1

Introduction

The Unit Commitment (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 [1]. 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 [2]. For example, the Midcontinent Independent System Operator (MISO) needs to solve a UC problem with around 1400 generating units in about 20 minutes [3]. Until recently, uncertainty came mainly from demand variations 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 intermittent, uncertainty must be properly considered. If not, operating costs of the system could become excessively high and also security issues could arise [4] [5]. 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 [6], Mixed-Integer Linear Programming (MILP) [7] [8], and heuristics [9]. Approaches for dealing with uncertainty in UC have been mostly based on robust optimization [10], chance-constrained optimization [11], and stochastic optimization. Those based on the latter have typically used scenarios, or Sample-Average Approximations (SAA), combined with Benders decomposition [12] or with Lagrangian relaxation [13]. For example, in [8], 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 [12] 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 [14], 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 [2].

Stochastic Hybrid Approximation (SHA) algorithms are a family of algorithms for solving stochastic optimization problems. They were initially proposed in [15] 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. Algorithms of this type have been recently applied to stochastic Economic Dispatch (ED) problems, which are closely related to the stochastic UC problem but focus exclusively on determining generator power levels and not on-off states. In particular, in [16], the authors apply the SHA algorithm proposed in [15] to a two-stage stochastic ED problem by using a certainty-equivalent model for constructing the initial approximation of the expected second-stage cost. The resulting "Adaptive Certainty-Equivalent" (AdaCE) approach is shown to have superior performance compared to some widely-used approaches on the test cases considered. The same authors also later extend this approach to solve more complex versions of the stochastic ED problem that include expected-value constraints and multiple planning stages [17] [18], and show promising performance. The close connection between UC and ED problems suggests that perhaps the combination of SHA and certainty-equivalent 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 thesis, the applicability and performance of the AdaCE algorithm, which combines SHA with initial expected-value function approximations based on certainty-equivalent 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 renewable energy distributions. The results obtained show that despite a lack of theoretical guarantees, the AdaCE algorithm can find commitment schedules in reasonable times that are better than the ones obtained with the benchmark algorithms considered.

This thesis is organized as follows: In Chapter 2, the two-stage mixed-integer model of the stochastic UC problem used in this work is presented. In Chapter 3, a benchmark algorithm based on SAA and Benders decomposition is described. In Chapter 4, the AdaCE algorithm is introduced, and potential theoretical and practical issues and solutions are discussed. In Chapter 5, the experiments performed to compare the performance of the algorithms described in this work are presented and the results obtained are discussed. Lastly, in Chapter 6, key findings are summarized and next research directions are outlined.

Chapter 2

Stochastic Unit Commitment Problem

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 [19]. The schedules are chosen to maximize profit or social welfare, and special attention is paid to energy pricing and payments [20]. Traditionally, the uncertainty in a system has been small and it has come mainly from the demand and potential contingencies. However, with the projected large-scale penetration of renewable energy in power systems, in particular of 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 in next-generation power systems.

2.1 Problem Formulation

In this thesis, the stochastic UC problem is modeled as a stochastic two-stage 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

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & F(u) + \mathbb{E}\left[Q(u,w)\right] \\ \text{subject to} \quad & u \in \mathcal{U} \cap \{0,1\}^{nT}, \end{aligned} \tag{2.1}$$

where u is the vector of on-off states $u_{i,t}$ for each power generating unit $i \in \{1,\dots,n\}$ and time $t \in \{1,\dots,T\}$, 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

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

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 Optimal Power Flow (OPF) problem:

$$\begin{aligned}
\underset{p,\theta,s,d}{\text{minimize}} \quad & \varphi(p) + \gamma \left\| d - \bar{d} \right\|_1 && \text{(2.2a)}\\
\text{subject to} \quad & Pp + Ss - Ld - A\theta = 0 && \text{(2.2b)}\\
& \operatorname{diag}(p^{\min})u \le p \le \operatorname{diag}(p^{\max})u && \text{(2.2c)}\\
& \delta^{\min} \le Dp \le \delta^{\max} && \text{(2.2d)}\\
& z^{\min} \le J\theta \le z^{\max} && \text{(2.2e)}\\
& 0 \le d \le \bar{d} && \text{(2.2f)}\\
& 0 \le s \le w, && \text{(2.2g)}
\end{aligned}$$

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 ($\gamma \ge 0$). Constraint (2.2b) enforces power flow balance using a DC network model [21], constraint (2.2c) enforces generator power limits, constraint (2.2d) enforces generator ramping limits, constraint (2.2e) enforces branch flow limits due to thermal ratings, and constraints (2.2g) and (2.2f) 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 [22].

It is assumed for simplicity that the second-stage problem (2.2) 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 (2.2), or other sources of flexibility such as demand response, are enough to ensure this property in practical problems.

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

$$w_t := \Pi_c(\bar{w}_t + \delta_t),$$

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

$$\Pi_c(z) := \arg\min \left\{ \|z - w\|_2 \;\middle|\; 0 \le w \le w^{\max} \right\}, \ \forall z,$$

where $w^{\max}$ is the vector of power capacities of the renewable energy sources. The random vector w in (2.1) is therefore composed of $(w_1, \ldots, w_T)$. 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 $\bar{w}_t$, 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.

Chapter 3

Sample-Average Approximation

Since evaluating the function $\mathbb{E}\left[Q(\,\cdot\,, w)\right]$ in (2.1) accurately is computationally expensive, algorithms for solving problems of the form of (2.1) 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* [23]. The resulting approximate problem is given by

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & F(u) + \frac{1}{m}\sum_{i=1}^{m} Q(u, w_i) \\ \text{subject to} \quad & u \in \mathcal{U} \cap \{0,1\}^{nT}, \end{aligned} \tag{3.1}$$

which is a deterministic optimization problem. It can be shown that by making m sufficiently large, the solutions of problem (3.1) approach those of problem (2.1) [23]. However, as m increases, problem (3.1) becomes very difficult to solve due to its size. For this reason it becomes necessary to exploit the particular structure of this problem.

3.1 Benders Decomposition

A widely used strategy for dealing with the complexity of problem (3.1) is to use decomposition combined with a cutting-plane method [22]. 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 [22].

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

$$u_k = \arg\min \left\{ F(u) + \mathcal{Q}^k(u) \;\middle|\; u \in \mathcal{U} \cap \{0,1\}^{nT} \right\} \tag{3.2a}$$

$$\mathcal{Q}^{k+1}(u) = \begin{cases} a_k + b_k^T(u - u_k) & \text{if } k = 0, \\ \max\left\{ \mathcal{Q}^k(u),\ a_k + b_k^T(u - u_k) \right\} & \text{else,} \end{cases} \tag{3.2b}$$

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 (2.2), it can be shown that b_k can be chosen as

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

where μ_i^k and π_i^k are the Lagrange multipliers associated with the right and left inequality of constraint (2.2c), respectively, for $(u, w) = (u_k, w_i)$.

From the properties of subgradients, it can be shown that

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

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 (3.1). 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 [22]. Note that the gap approaching zero suggests that u_k is close to being a solution of (3.1), but not necessarily of (2.1).

In practice, in order to avoid dealing in with the non-differentiability of the functions F and $\mathcal{Q}^k$, step (3.2a) is carried out by solving instead the following equivalent optimization problem, which uses an *epigraph* representation of the functions:

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

Chapter 4

Stochastic Hybrid Approximation

Motivated by problems arising in transportation, the authors in [15] proposed a *stochastic hybrid approximation* algorithm to solve two-stage stochastic optimization problems. For problems of the form of

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & F(u) + \mathbb{E}\left[Q(u, w)\right] \\ \text{subject to} \quad & u \in \mathcal{U}, \end{aligned} \tag{4.1}$$

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

$$u_k := \arg\min \left\{ F(u) + \mathcal{Q}^k(u) \;\middle|\; u \in \mathcal{U} \right\} \tag{4.2}$$

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 $k \in \mathbb{Z}_+$,

$$\mathcal{Q}^{k+1}(u) = \mathcal{Q}^k(u) + \alpha_k\big(\xi_k - \nabla \mathcal{Q}^k(u_k)\big)^T u, \tag{4.3}$$

for all $u \in \mathcal{U}$, where $\xi_k \in \delta Q(u_k, w_k)$, w_k are independent and identically distributed (i.i.d) samples of the random vector w, and α_k are step lengths that satisfy conditions that are common in stochastic approximation algorithms [24]. As discussed in [15], 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.

4.1 Adaptive Certainty-Equivalent

In [16], [17], and [18], the authors apply the SHA algorithm outlined above and extensions of it to different versions of the stochastic ED problem. Inspired by the fact that in power systems operations and planning, certainty-equivalent models are common, the authors consider using these models for constructing initial approximations of the expected-cost functions. For example, for problem (4.1), the certainty-equivalent 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, the aforementioned authors obtain promising performance of the resulting AdaCE algorithms on several instances and types of stochastic ED problems compared to some widely-used benchmark algorithms.

For the stochastic UC problem (2.1), the function Q is defined as the optimal objective value of problem (2.2), 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 [25].

4.2 Applicability to Stochastic Unit Commitment

The authors in [15] show that the SHA algorithm is guaranteed to solve problems of the form of (4.1). However, for the stochastic UC problem (2.1), 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 (4.3). 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 4.1 illustrates qualitatively the possible outcomes of the SHA algorithm on a simple one-dimensional 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 4.1, 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 4.1, 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 4.1, 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 [15], 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 [15], but it makes step (4.2) computationally heavy in the case with binary restrictions. In particular, it makes step (4.2) 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 (4.3).

For the specific case of the stochastic UC problem (2.1), the initial function approximation for the AdaCE approach is $\mathcal{Q}^0 = Q(\,\cdot\,, \mathbb{E}\,[w])$, which is a convex piece-wise quadratic function in general. In order to make it piece-wise linear, the approach proposed here is to approximate the sepa-

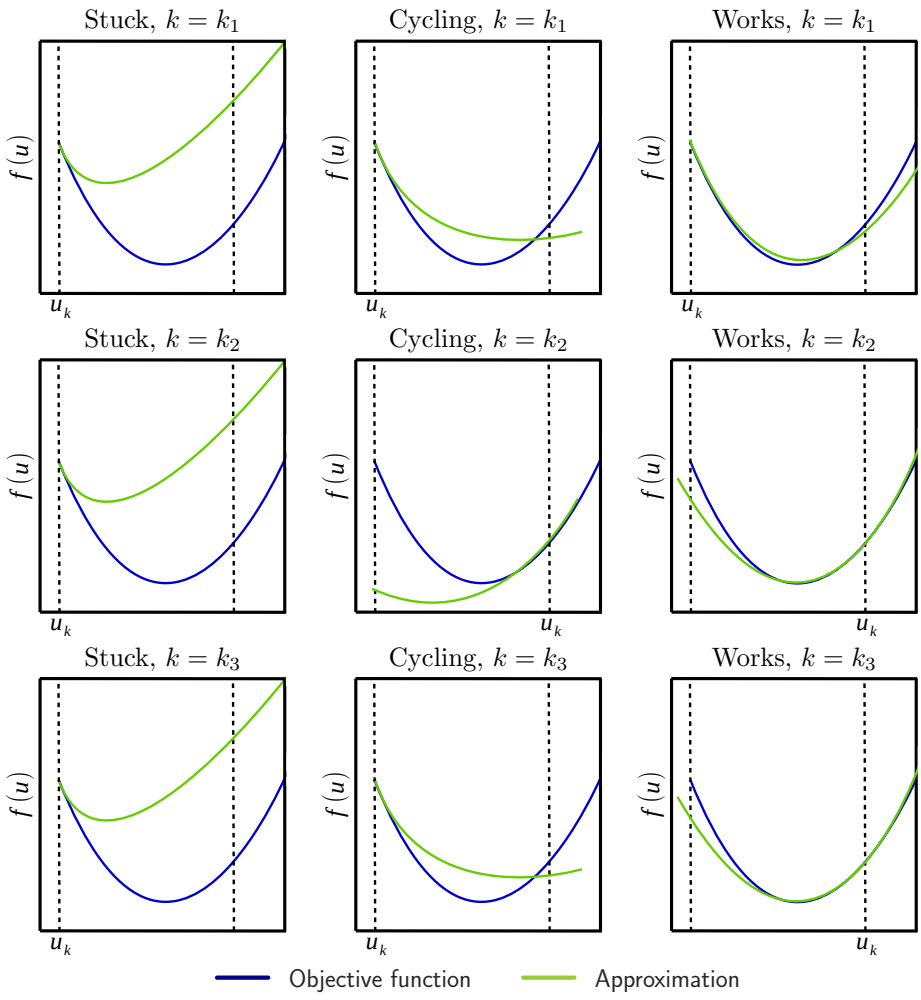

Figure 4.1: SHA possible outcomes with binary restrictions

rable 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

$$\bar{\varphi}(p) := \sum_{i=1}^{n} \sum_{t=1}^{T} \max_{1 \leq j \leq r} (\zeta_{i,j} + \beta_{i,j} p_{i,t}), \ \forall p, \tag{4.4}$$

where $r \in \mathbb{Z}_{++}$, $\zeta_{i,j}$ 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 (2.2).

4.3 Noise Reduction

As discussed above, a piece-wise linear function $\mathcal{Q}^0$ based on the certainty-equivalent model is proposed for applying the AdaCE algorithm to the stochastic UC problem. With this, the resulting step (4.2) 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 thesis applying the mini-batch technique from Machine Learning [26], and replace (4.3) with the following update for each $k \in \mathbb{Z}_+$:

$$\mathcal{Q}^{k+1}(u) = \mathcal{Q}^k(u) + \alpha_k \left(\frac{1}{m} \sum_{i=1}^{m} \xi_{k,i} - \nabla \mathcal{Q}^k(u_k) \right)^T u, \tag{4.5}$$

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 i.i.d 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 1. *The covariance Σ_m of the noisy subgradient $m^{-1} \sum_{i=1}^{m} \xi_{k,i}$ used in (4.5) satisfies $\Sigma_m = \Sigma/m$, where Σ is the covariance of the noisy subgradient ξ_k used in (4.3).*

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 i.i.d., 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

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

□

Chapter 5

Numerical Experiments

This chapter describes the numerical experiments carried out in this work to assess and compare the performance of the AdaCE algorithm and the algorithm based on sample-average approximation 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 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 Chapter 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 chapter.

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` [27]. For solving the MILP problems, the solver `GUROBI` 6.5.1 was used [28]. For solving the quadratic optimization problems, the second-order cone solver `ECOS` was used [29]. The spatial covariance matrix of the powers of renewable energy sources was constructed using routines available in `PFNET`.

[1] http://pfnet-python.readthedocs.io

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` [30]). 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 (2.1), 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

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 [31]. 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 ($\delta^{\min}$ and $\delta^{\max}$ 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 [32]. 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	$\varphi(p)$	$p^{\min}$	$p^{\max}$	$\delta^{\min}$	$\delta^{\max}$	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

[2] http://www2.ee.washington.edu/research/pstca/

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

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

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 5.1 shows the resulting aggregate load profiles.

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 $w_i^{\max}$ 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 $\bar{w}_t$ were obtained by multiplying $0.5w^{\max}$ 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 5.2 shows sampled realizations (gray) and expected realizations (red) of aggregate powers from renewable energy sources for five different days for Case A.

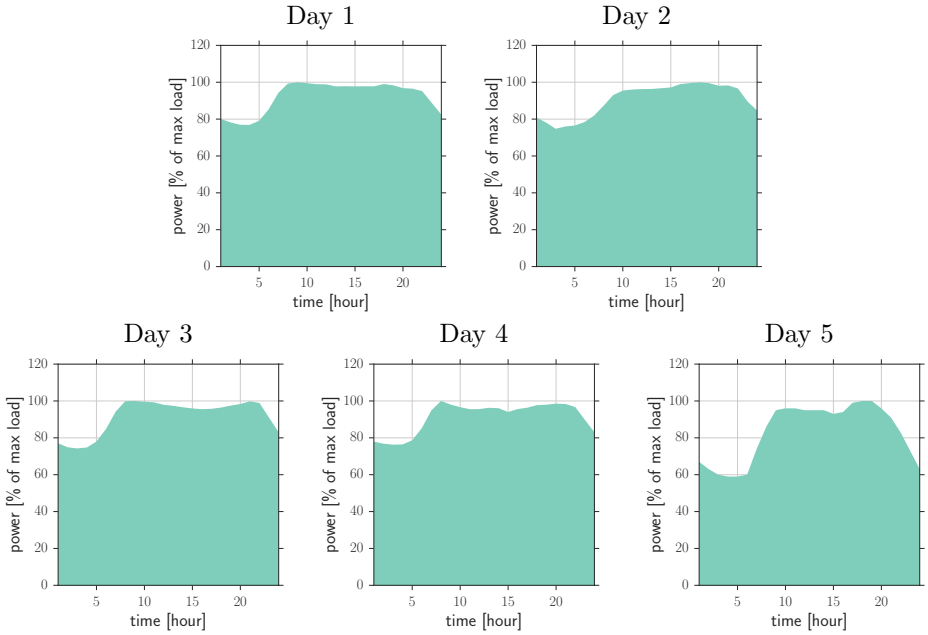

Figure 5.1: Load profiles

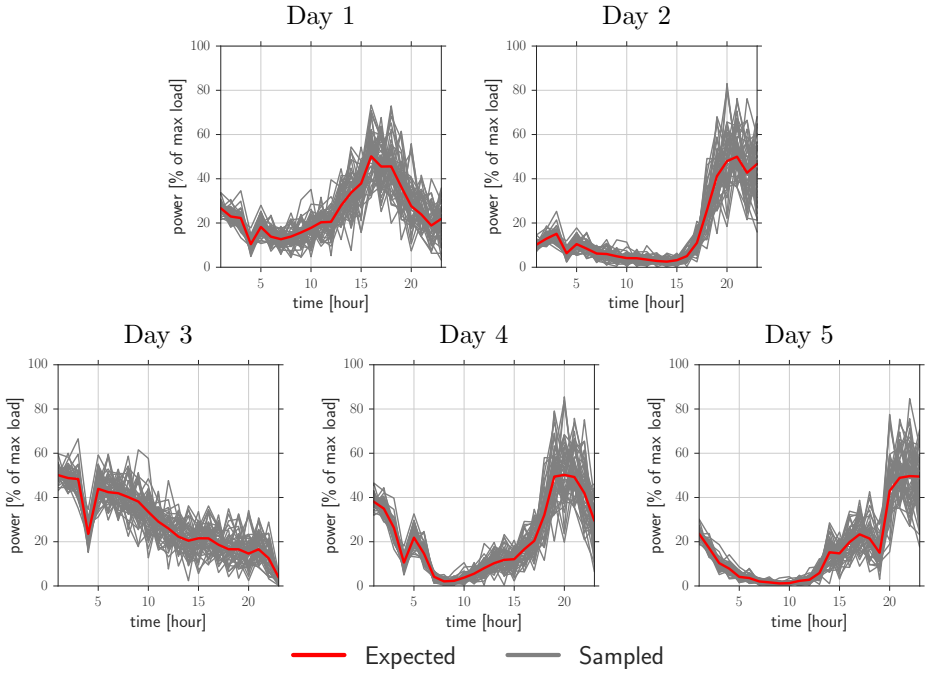

Figure 5.2: Powers from renewable energy sources for Case A

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 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 (3.1) approaches zero. Experimental results testing this property are shown in Figure 5.3.

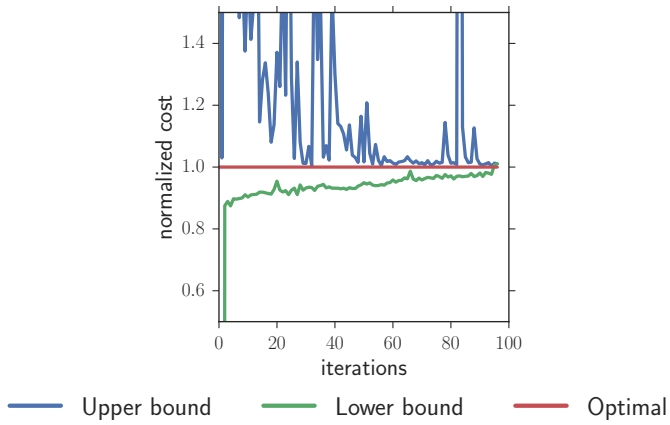

Figure 5.3: Benders validation on deterministic Case A

As Figure 5.3 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 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 r of linear segments in the individual piece-wise linear generation cost functions in (4.4) 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

5.5.1 Case A

Figure 5.4 shows the performance of the algorithms on Case A. The expected costs shown are relative to the cost obtained with the solution of the certainty-equivalent (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 (2.1), 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 to form an adequate approximation. Moreover, the algorithm only "knows" about the 300 scenarios, and this number may not be enough to adequately represent the uncertainty.

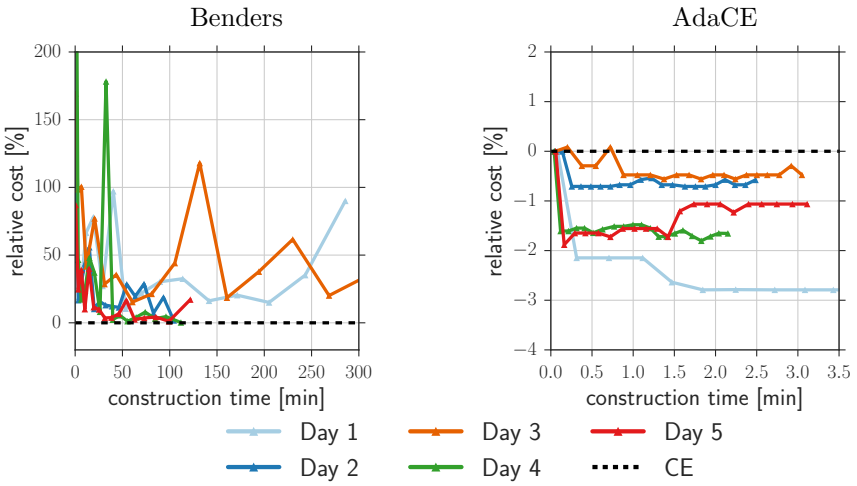

Figure 5.4: Performance of algorithms on Case A

Figure 5.5 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. 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 5.6.

5.5.2 Case B

Figure 5.7 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 is smaller 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,

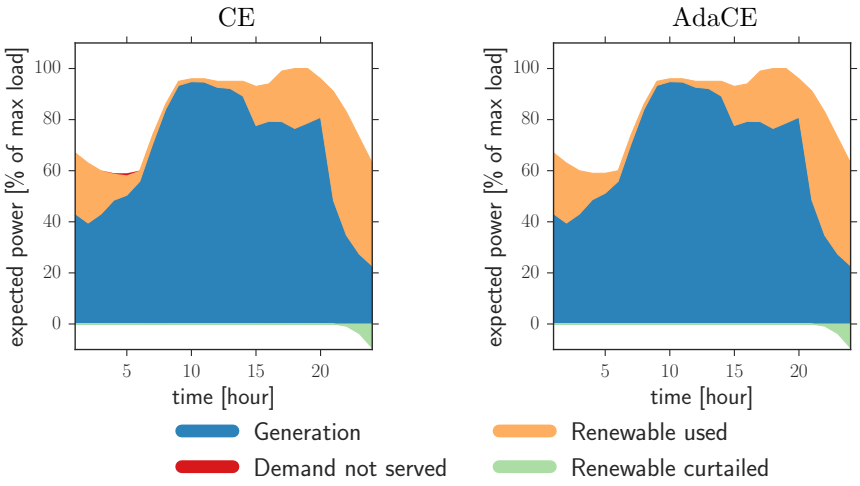

Figure 5.5: Expected generation profile for Case A, Day 5

and to the insufficiency of the 300 scenarios for adequately capturing the uncertainty.

Figure 5.8 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 becomes clear when the production by technology is considered. This is shown in Figure 5.9. As this and Figure 5.2 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.

5.5.3 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 noise reduction strategy in the AdaCE algorithm was analyzed on this case. Figure 5.10 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

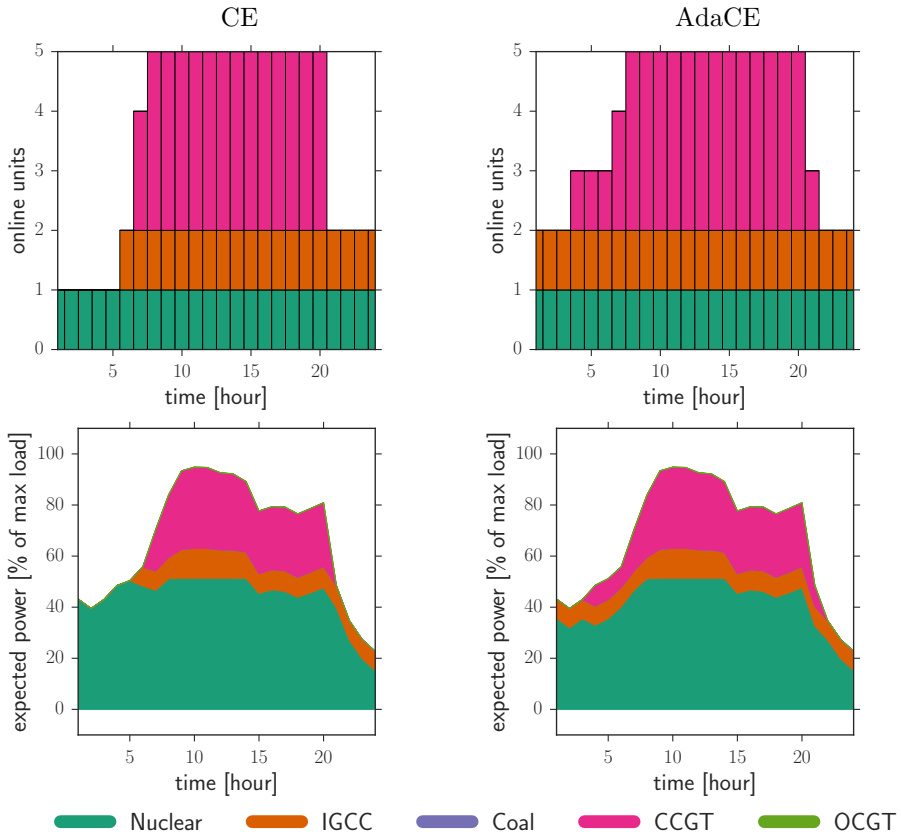

Figure 5.6: Generation mix for Case A, Day 5

cost compared with the CE solution, especially without noise reduction.

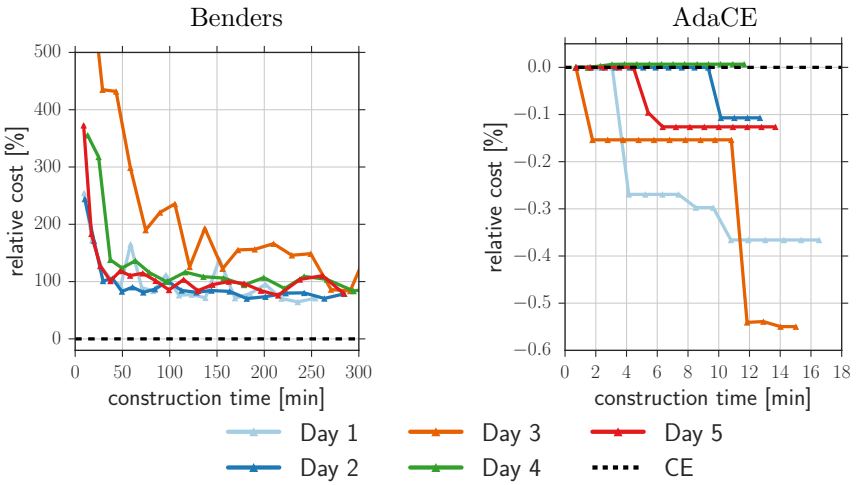

Figure 5.7: Performance of algorithms on Case B

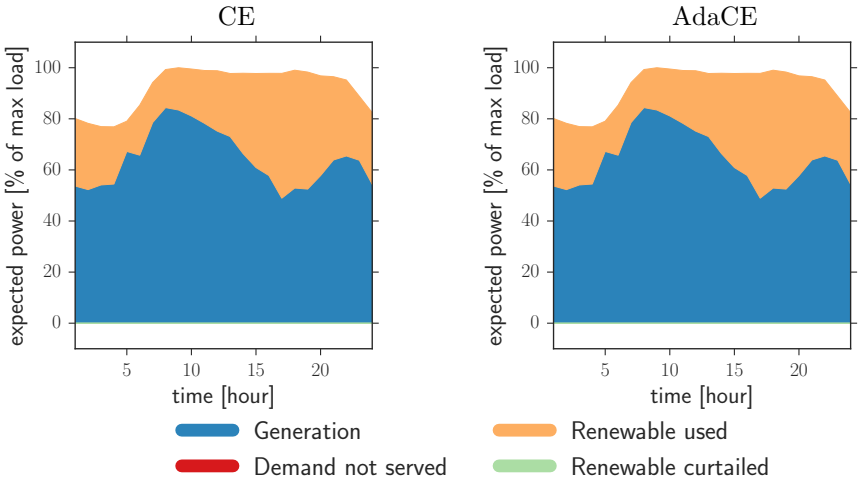

Figure 5.8: Expected generation profile for Case B, Day 1

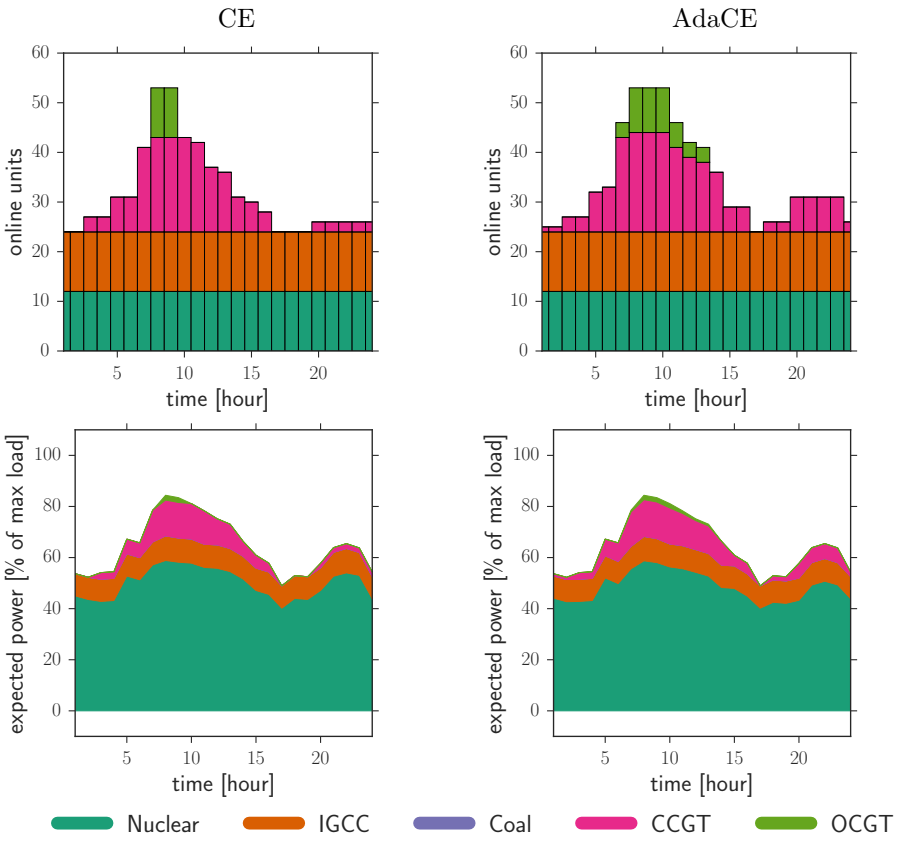

Figure 5.9: Generation mix for Case B, Day 1

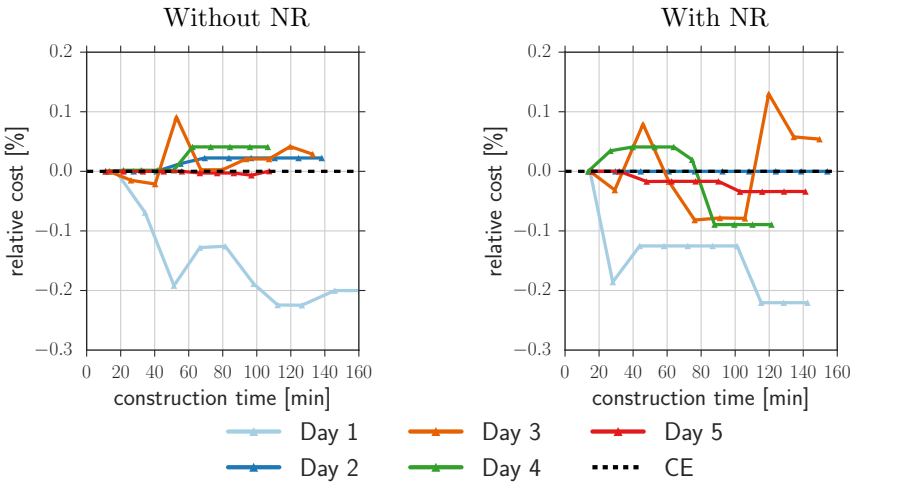

Figure 5.10: Performance of AdaCE on Case C

Chapter 6

Conclusions

In this work, 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 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 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.

Future research steps include a sensitivity analysis of the performance of the algorithm with respect to the number of segments of the piece-wise linear approximations of the generator cost functions. Also, an extension of the SHA framework that allows a richer type of updates beyond slope corrections needs to be investigated in order to determine whether theoreti-

cal guarantees can be obtained in problems with binary restrictions. Lastly, an interesting research direction consists of investigating the performance of the AdaCE algorithm on stochastic UC problems that include market-based elements, such as power production levels in the first-stage problem and different pricing schemes.

Bibliography

[1] 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.

[2] M. Tahanan, W. Van Ackooij, A. Frangioni, and F. Lacalandra. Large-scale unit commitment under uncertainty. *4OR*, 13(2):115–171, 2015.

[3] Y. Chen, A. Casto, X. Wang, J. Wan, and F. Wang. Day-ahead market clearing software performance improvement. `https://www.ferc.gov/june-tech-conf/2015/abstracts/m1-2.html`, 2015. [Online; accessed 11-April-2017].

[4] International Energy Agency. System integration of renewables - Implications for electricity security. Technical Report to the G7, February 2016. [Online; accessed 11-April-2017].

[5] 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.

[6] 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.

[7] A. Ott. Evolution of computing requirements in the pjm market: Past and future. In *IEEE PES General Meeting*, pages 1–4, July 2010.

[8] 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.

[9] 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.

[10] 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, Feb 2013.

[11] 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.

[12] J. Wang, J. Wang, C. Liu, and J. Ruiz. Stochastic unit commitment with sub-hourly dispatch constraints. *Applied Energy*, 105:418 – 422, 2013.

[13] L. Wu, M. Shahidehpour, and T. Li. Stochastic security-constrained unit commitment. *IEEE Transactions on Power Systems*, 22(2):800–811, May 2007.

[14] 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.

[15] R. Cheung and W. Powell. SHAPE - A stochastic hybrid approximation procedure for two-stage stochastic programs. *Operations. Research.*, 48(1):73–79, 2000.

[16] 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.

[17] T. Tinoco De Rubira and G. Hug. Primal-dual stochastic hybrid approximation algorithm. Computational Optimization and Applications (under review), 2016.

[18] T. Tinoco De Rubira, L. Roald, and G. Hug. Multi-stage stochastic optimization via parameterized stochastic hybrid approximation. (under review).

[19] 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.

[20] 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`, [Online; accessed 11-April-2017].

[21] J. Zhu. *Optimization of Power System Operation*. John Wiley & Sons, Inc., 2009.

[22] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering, 2011.

[23] 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.

[24] H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Stochastic Modelling and Applied Probability. Springer-Verlag, 2003.

[25] F. Louveaux. Piecewise convex programs. *Mathematical Programming*, 15(1):53–62, 1978.

[26] 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.

[27] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[28] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2016.

[29] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.

[30] 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.

[31] 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. A report prepared by the reliability test system task force of the application of probability methods subcommittee. *IEEE Transactions on Power Systems*, 14(3):1010–1020, Aug 1999.

[32] Black and Veatch Holding Company. Cost and performance data for power generation technologies. Prepared for the National Renewable Energy Laboratory. `https://www.bv.com/docs/reports-studies/nrel-cost-report.pdf`, [Online; accessed 11-April-2017].