Robust Strategy Synthesis for Probabilistic Systems Applied to Risk-Limiting Renewable-Energy Pricing

Alberto Alessandro Angelo Puggelli
Alberto L. Sangiovanni-Vincentelli
Sanjit A. Seshia

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2014-16
http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-16.html

February 10, 2014

Acknowledgement

The authors would like to thank J. Venkatesh and prof. T. S. Rosing for providing the raw measurement data of wind energy from the wind farm at Lake Benton, Minnesota, USA.

Robust Strategy Synthesis for Probabilistic Systems Applied to Risk-Limiting Renewable-Energy Pricing

Alberto Puggelli, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
{puggelli, alberto, sseshia}@eecs.berkeley.edu

Abstract. The use of economic incentives has been proposed to manage user demand in smart grids that integrate renewable sources of energy to compensate for the intrinsic uncertainty in the prediction of the supply generation. We address the problem of synthesizing optimal energy pricing strategies, while quantitatively constraining the risk due to uncertainty for the network operator and guaranteeing quality-of-service for the users. We use Ellipsoidal Markov Decision Processes (EMDP) to model the decision-making scenario. These models are trained with measured data and allow to quantitatively capture the uncertainty in the prediction of energy generation. We then cast the constrained optimization problem as the strategy synthesis problem for EMDPs, with the goal to maximize the total expected reward constrained to properties expressed using the Probabilistic Computation Tree Logic (PCTL), and propose a novel sound and complete synthesis algorithm. An experimental comparison shows the effectiveness of our method with respect to previous approaches presented in the literature.

1 Introduction

Several real-world multi-agent systems exhibit stochastic behavior, and can be modeled using formalisms such as Markov Decision Processes (MDPs) [29]. Desired properties of such systems can be both Boolean, e.g., expressible in logics such as probabilistic computation tree logic (PCTL) [2,3], or quantitative, such as maximizing a reward function [1]. The synthesis of strategies to satisfy Boolean properties and optimize quantitative measures is naturally a topic of much relevance. Moreover, for probabilistic models that are inferred from empirical data, it is necessary to design strategies that are robust to uncertainties in estimated probabilities. In this paper, we present a new approach to robust strategy synthesis for a class of MDPs with uncertainties, with application to risk-limiting renewable energy pricing.

Main Motivating Application. Electricity consumption is projected to grow from 18 trillion kWh in 2006 to 32 trillion kWh in 2030, a 77% increase [4]. To avoid catastrophic pollution damage to the planet, it is necessary to employ energy sources alternative to fossil fuels [5]. In this paper, we focus on wind energy, which currently has higher capacity than solar energy, and is expected to constitute a significant portion of renewable generation integrated to the power grids of North America [5].

The correct operation of power systems requires the balance between energy supply and demand at all times. The *risk* for the system operator can be quantified both by the probability of not meeting such a balance constraint, and by the (positive) gap between demand and supply. High values of either indicator make the occurrence of disruptions, faults and ultimately blackouts more likely [6]. In grids that only integrate fossil energy sources, the task for the system operator amounts to *dispatch* the production of energy during the day, based on averaged demand profiles. A wealth of *deterministic* optimization frameworks have been developed to solve the energy dispatch problem, aiming to maximize the operator profits while guaranteeing resiliency also in the presence of one fault in the network, the so-called *N-1* worst-case dispatch [6]. Heavy reliance on wind generation puts forth big operational challenges [6]. Unlike fossil energy resources, wind generation is non-dispatchable, i.e., it cannot be harvested by request. Further, wind availability exhibits high variability across all timescales, which makes it challenging to forecast (errors can be up to 20% of the forecast value [7]). To compensate for this supply uncertainty, researchers have proposed the concept of *demand response*, i.e., adapting customer energy consumption in response to supply conditions. In smart grids with two-way communications, *real-time* pricing protocols can be implemented so that, unlike the fixed price per unit that is traditionally employed, the price of electricity can vary according to the supply, to incentivize demand appropriately.

Stochastic modeling and optimization frameworks have been proposed for the problem of energy pricing and dispatch [8, 9]. These works aim to determine both optimal pricing and dispatch of non-renewable baseline energy (wind penetration usually accounts only up to 20-30% of the total energy generation, so fossil fuels are still required). These works though do not explicitly consider the risk of power unbalance during optimization, and only evaluate the probability of loss of load after synthesis via *Monte Carlo simulation*, to evaluate the quality of the proposed solution. Unfortunately, the resulting evaluation offers little insight to the operator when the risk is too high. Indeed, Varayia *et al.* [6] advocate the need for an optimization framework capable of bounding the risk, interpreted stochastically, at optimization time. Moreover, a minimum amount of delivered energy needs to be guaranteed to the users, since otherwise operators could potentially increase the energy price to force users out of the system to obtain power balance at times of little wind generation. We refer to this guaranteed delivered energy as Quality of Service (QoS) for the users. As summarized in the next section, our contribution targets these needs.

Paper Contributions. Our first contribution is a novel stochastic model to capture energy-dispatch and pricing strategies for smart grids with wind energy sources. The model is an Ellipsoidal Markov Decision Process (EMDP), a special case of the Convex-MDP (CMDP) model first introduced in [10], i.e., an MDP where transition probabilities are only known to lie in ellipsoidal sets. While previous works used analytical distributions, e.g., Gaussian [9], to model uncertainty in wind availability, we use measured data (from the wind farm at Lake Benton, Minnesota, USA), to train a likelihood model of the wind generation, and give quantitative means to represent the confidence in the forecast values. We then approximate the likelihood region with an ellipsoidal model, which is more accurate than the linear ones often used in the literature, while remaining computationally tractable. Our empirical approach has the promise of more faithfully representing the probability distribution of the generated energy because it is tailored to the specific wind farm under analysis, and it is robust to forecast errors.

As a second contribution, we cast the constrained optimization problem as the strategy synthesis problem for EMDPs, to maximize the total expected reward constrained to PCTL properties. The optimization aims to maximize the profits for the system operator, while constraints limit the risk of power unbalance and guarantee the desired QoS for the users. We focus on Markov Deterministic (MD) strategies, i.e, for each state an optimal action to take is chosen deterministically, based only on the current state and not on the entire execution history. Deterministic pricing and dispatch strategies are easier to adopt in a real-world scenario. Moreover, we encode the sequence of decision epochs over the day within the model itself, to guarantee the Markov property.

Finally, as our third contribution, we prove that the problem of determining the existence of an MD strategy for EMDPs, with total expected reward higher than a given threshold and constrained to specifications in PCTL, is NP-complete, and develop an algorithm to solve the optimization version of the problem, i.e, maximize the reward. The algorithm can process formulae with multiple and nested quantitative operators, and it is sound and complete. Its key advantage is the capability of ranking candidate strategies by the value of their reward. The first proposed strategy that satisfies all PCTL properties for any resolution of uncertainty is the solution of the synthesis problem. Although the algorithm worst-case running time is exponential in the size of the EMDP, this capability may allow considerable speed-ups in practical scenarios. These results hold also for Interval-MDPs (not considered in the paper). Further, the proposed algorithm can be applied to a wider range of applications, e.g., semi-autonomous car driving.

The rest of the paper is organized as follows. Section 2 gives background on CMDPs and PCTL. Section 3 presents related work. In Section 4, we describe the proposed algorithm for the synthesis of constrained optimal strategies for EMDPs. We then give details of the EMDP model used to synthesize energy-pricing strategies in smart grids with renewable sources in Section 5, and present experimental results in Section 6. Lastly, we conclude and discuss future directions in Section 7.

2 Preliminaries

Definition 2.1. *A Probability Distribution (PD) over a finite set Z of cardinality n is a vector $\mu \in \mathbb{R}^n$ satisfying $\mathbf{0} \leq \mu \leq \mathbf{1}$ and $\mathbf{1}^T\mu = 1$. The element $\mu[i]$ represents the probability of realization of event z_i. We call $Dist(Z)$ the set of distributions over Z.*

2.1 Convex Markov Decision Process (CMDP)

Definition 2.2. *A labeled finite CMDP, $\mathcal{M}_\mathcal{C}$ is a tuple $\mathcal{M}_\mathcal{C} = (S, S_0, A, \Omega, \mathcal{F}, \mathcal{A}, \mathcal{X}, L)$, where S is a finite set of states of cardinality $N = |S|$, S_0 is the set of initial states, A is a finite set of actions ($M = |A|$), Ω is a finite set of atomic propositions, $\mathcal{F}$ is a finite set of convex sets of transition PDs, $\mathcal{A} : S \rightarrow 2^A$ is a function that maps each state to the set of actions available at that state, $\mathcal{X} = S \times A \rightarrow \mathcal{F}$ is a function that associates to state s and action a the corresponding convex set $\mathcal{F}_s^a \in \mathcal{F}$ of transition PDs, and $L : S \rightarrow 2^\Omega$ is a labeling function.*

The set $\mathcal{F}_s^a = Dist_s^a(S)$ represents the *uncertainty* in defining a transition distribution for $\mathcal{M}_\mathcal{C}$ given state s and action a. We call $\mathbf{f}_s^a \in \mathcal{F}_s^a$ an observation of this uncertainty. Also, $\mathbf{f}_s^a \in \mathbb{R}^N$ and we collect the vectors $\mathbf{f}_s^a, \forall s \in S$ into an observed transition matrix $F^a \in \mathbb{R}^{N \times N}$. Abusing terminology, we call $\mathcal{F}^a$ the uncertainty set of the transition matrices, and $F^a \in \mathcal{F}^a$. $\mathcal{F}_s^a$ is interpreted as the row of $\mathcal{F}^a$ corresponding to state s. Finally, $f_{s_i s_j}^a = \mathbf{f}_{s_i}^a[j]$ is the observed transition probability from s_i to s_j under action a. The data-type of $a \in \mathcal{A}(s_i)$ can be different from the one of $b \in \mathcal{A}(s_j)$, if $s_i \neq s_j$.

To model uncertainty in state transitions, we make the following assumptions:

Assumption 2.1. *$\mathcal{F}^a$ can be factored as the Cartesian product of its rows, i.e., its rows are uncorrelated. Formally, for every $a \in A$, $\mathcal{F}^a = \mathcal{F}_{s_0}^a \times \cdots \times \mathcal{F}_{s_{N-1}}^a$. In [11] this assumption is referred to as* rectangular uncertainty.

Assumption 2.2. *[CMDP Semantics] CMDPs model non-deterministic choices made from a convex set of uncountably many choices. Each time a state is visited, a transition distribution within the set is adversarially picked, and a probabilistic step is taken accordingly. (The same semantics is used for IMDPs in [12].)*

A transition between state s to state s' in a CMDP occurs in three steps. First, an action $a \in \mathcal{A}(s)$ is chosen. The selection of a is *nondeterministic*. Secondly, an observed PD $\mathbf{f}_s^a \in \mathcal{F}_s^a$ is chosen. The selection of $\mathbf{f}_s^a$ models uncertainty in the transition. Lastly, a successor state s' is chosen randomly, according to the transition PD $\mathbf{f}_s^a$.

A path π in $\mathcal{M}_\mathcal{C}$ is a finite or infinite sequence of the form $s_0 \xrightarrow{f^{a_0}_{s_0 s_1}} s_1 \xrightarrow{f^{a_1}_{s_1 s_2}}, \cdots,$ where $s_i \in S$, $a_i \in \mathcal{A}(s_i)$ and $f^{a_i}_{s_i, s_{i+1}} > 0 \ \forall i \geq 0$. We indicate with $\mathit{\Pi}_{fin}$ ($\mathit{\Pi}_{inf}$) the set of all finite (infinite) paths of $\mathcal{M}_\mathcal{C}$. $\pi_s[i]$ ($\pi_a[i]$) is the i^{th} state (selected action) along the path and, for finite paths, $last(\pi)$ is the last state visited in $\pi \in \mathit{\Pi}_{fin}$. $\mathit{\Pi}_s = \{\pi \mid \pi[0] = s\}$ is the set of paths starting in state s.

The algorithm presented in Section 4 can be applied both to the interval and ellipsoidal models of uncertainty [10] (and to a mix of them), but in this paper we will focus on the latter one since it is more suitable for the application analyzed in Section 5. This model is a second-order approximation of the likelihood model of uncertainty (likelihood models are often used when transition probabilities are determined experimentally), and it is more accurate than a linear one [11]. The transition frequencies associated to action $a \in A$ are collected in matrix H^a. Uncertainty in each row of H^a can be described by the likelihood region g_s^a:

$$g_s^a = \{\mathbf{f}_s^a \in \mathbb{R}^N | \textstyle\sum_{s'} h^a_{ss'} \log(f^a_{ss'}) \geq \beta_s^a\} \approx \{\mathbf{f}_s^a \in \mathbb{R}^N | \textstyle\sum_{s'} \frac{\left(f^a_{ss'} - h^a_{ss'}\right)^2}{h^a_{ss'}} \leq (\mathcal{K}_s^a)^2\} \quad (1)$$

where $\beta_s^a < \beta^a_{s,max} = \sum_{s'} h^a_{ss'} \log(h^a_{ss'})$ represents the uncertainty level. Equation (1) also shows on the right the second-order approximation of g_s^a, with $\mathcal{K}_s^a = 2(\beta^a_{s,max} - \beta_s^a) \geq 0$ representing the uncertainty level. We then write the approximation of g_s^a in conic form, and intersect it with the probability simplex, to obtain the uncertainty set:

$$\mathcal{F}_s^a = \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \mathbf{f}_s^a \geq \mathbf{0}, \mathbf{1}^T \mathbf{f}_s^a = 1, \|E_s^a \left(\mathbf{f}_s^a - \mathbf{h}_s^a\right)\|_2 \leq 1, E_s^a \succ 0\} \quad (2)$$

where matrix $E_s^a = (\mathcal{K}_s^a)^{-1} \times diag\left((h^a_{ss_0})^{-0.5}, \cdots, (h^a_{ss_N})^{-0.5}\right) \succ 0$.

We determine the size $\mathcal{R}$ of the CMDP $\mathcal{M}_\mathcal{C}$ as follows. $\mathcal{M}_\mathcal{C}$ has N states, $O(M)$ actions per state and $O(N^2)$ transitions for each action. Let D_s^a denote the number of constraints required to express the rectangular uncertainty set $\mathcal{F}_s^a$, and $D = \underset{s \in S, a \in A}{max} D_s^a$. The overall size of $\mathcal{M}_\mathcal{C}$ is thus $\mathcal{R} = O(N^2 M + NMD)$.

Strategies and Nature To analyze *quantitative* properties, we need a probability space over infinite paths [13]. However, a probability space can only be constructed once *nondeterminism* and *uncertainty* have been resolved. We call each possible resolution of nondeterminism a *strategy*, which chooses an action in each state of $\mathcal{M}_\mathcal{C}$.

Definition 2.3. Strategy. *A randomized strategy for $\mathcal{M}_\mathcal{C}$ is a function $\sigma = \mathit{\Pi}_{fin} \times A \rightarrow [0, 1]$, with $\sum_{\mathcal{A}(last(\pi))} \sigma(\pi, a) = 1$, and $a \in \mathcal{A}(last(\pi))$ if $\sigma(\pi, a) > 0$. We call $\Sigma_{\mathcal{M}_\mathcal{C}}$ the set of all strategies σ of $\mathcal{M}_\mathcal{C}$.*

Conversely, we call a *nature* [11] each possible resolution of uncertainty, i.e., a nature chooses a transition PD for each state and action of $\mathcal{M}_\mathcal{C}$.

Definition 2.4. Nature. *Given action $a \in A$, a randomized nature is the function η^a : $\Pi_{fin} \times Dist(S) \to [0,1]$ with $\int_{\mathcal{F}^a_{last(\pi)}} \eta^a(\pi, \mathbf{f}^a_s) = 1$, and $\mathbf{f}^a_s \in \mathcal{F}^a_{last(\pi)}$ if $\eta^a(\pi, \mathbf{f}^a_s) > 0$. We call Nat the set of all natures η^a of $\mathcal{M}_\mathcal{C}$.*

A strategy σ (nature η^a) is Markovian (M) if it depends only on $last(\pi)$. Also, σ (η^a) is deterministic (D) if $\sigma(\pi, a) = 1$ for some $a \in \mathcal{A}(last(\pi))$ ($\eta^a(\pi, f^a_s) = 1$ for some $\mathbf{f}^a_s \in \mathcal{F}^a_{last(\pi)}$). As explained later, we will consider only MD strategies. There are in total $I = |\mathcal{A}_{s_0}| \times \cdots \times |\mathcal{A}_{s_N}| = O(M^N)$ MD strategies, collected in the set $\Sigma^{MD}_{\mathcal{M}_\mathcal{C}}$.

After fixing a strategy σ, all the non-determinism in $\mathcal{M}_\mathcal{C}$ is resolved. For MD strategies, we obtain the *induced* Convex Markov Chain (CMC) $\mathcal{M}^\sigma_C = (S, S_0, \Omega, \mathcal{F}, \mathcal{X}, L)$. $\mathcal{M}^\sigma_C$ has still size $\mathcal{R}$ since the state space S does not change for MD strategies. Further, the only convex set $\mathcal{F}^a_s \in \mathcal{F}$ of transition PDs available at each state $s \in S$ is the one corresponding to the action $a \in \mathcal{A}(s)$ such that $\sigma(s, a) = 1$.

Rewards. Rewards allow modeling additional quantitative measures of a CMDP, e.g., profit. We associate rewards to states and to actions available in each state.

Definition 2.5. *A reward structure for a CMDP $\mathcal{M}_\mathcal{C}$ is a tuple $r = (r_s, r_a)$ comprising a state (action) reward function $r_s : S \to \mathbb{R}_{\geq 0}$ ($r_a : S \times A \to \mathbb{R}_{\geq 0}$). Given a (possibly infinite) path π with horizon $T \in \mathbb{N} \cup +\infty$, the path reward for π is $rew(r)(\pi) = \Sigma^T_{t=0} r_s(\pi_s[t]) + r_a(\pi_s[t], \pi_a[t])$.*

In this paper, we will rank the available MD strategies for the CMDP based on their *total expected reward.*

Definition 2.6. *The total expected reward for state $s \in S$ under strategy $\sigma \in \Sigma^{MD}_{\mathcal{M}_\mathcal{C}}$ is defined as:*

$$\mathbb{W}^\sigma_{\mathcal{M}_\mathcal{C}, s} := \underset{\eta^a \in Nat}{min} \, \mathbb{E}^{\sigma, \eta^a}_{\mathcal{M}_\mathcal{C}}(rew(r)) \tag{3}$$

where we minimize the expected reward over all paths starting from s with horizon T visited under strategy σ across the action range $\eta^a \in Nat$ of the adversarial nature.

For simplicity, we will only consider CMDPs such that $\mathbb{W}^\sigma_{\mathcal{M}_\mathcal{C}, s}$ exists and it is finite $\forall s \in S, \forall \sigma \in \Sigma^{MD}_{\mathcal{M}_\mathcal{C}}$. These include infinite-horizon CMDPs ($T = +\infty$) with zero-reward absorbing states ($\omega = abs$), as the ones used in Section 5. For more details, see [1]. Further, according to Assumption 2.2, we can substitute $\eta^a \in Nat$ with $\mathbf{f^a_s} \in \mathcal{F}^a_s$, and only consider MD natures.

2.2 Probabilistic Computation Tree Logic (PCTL)

We use PCTL, a probabilistic logic derived from CTL which includes a probabilistic operator P [2] and a reward operator R [14], to express properties of CMDPs [3]. The syntax of this logic is defined as follows:

$$\begin{aligned} &\phi ::= True \mid \omega \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid P_{\bowtie p}[\psi] \mid R^r_{\bowtie v}[\rho] && \text{state formulas} \\ &\psi ::= \mathbf{X}\,\phi \mid \phi_1\, \mathbf{U}^{\leq k} \phi_2 \mid \phi_1\, \mathbf{U}\, \phi_2 && \text{path formulas} \\ &\rho ::= \mathbf{I}^{=k} \mid \mathbf{C}^{\leq k} \mid \mathbf{F}\,\phi && \text{rewards} \end{aligned}$$

with $\omega \in \Omega$ an atomic proposition, $\bowtie \in \{\leq, <, \geq, >\}$, $p \in [0,1]$, $v \in \mathbb{R}_{\geq 0}$ and $k \in \mathbb{N}$.

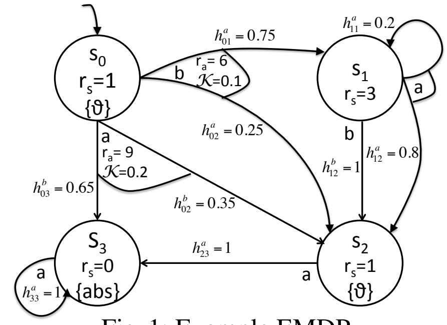

Fig. 1: Example EMDP.

Table 1: PCTL semantics for $\mathcal{M}_\mathcal{C}$.

$s \models$ True	
$s \models \omega$	iff $\omega \in L(s)$
$s \models \neg\phi$	iff $s \not\models \phi$
$s \models \phi_1 \wedge \phi_2$	iff $s \models \phi_1 \wedge s \models \phi_2$
$s \models P_{\lhd p(\rhd p)}\,[\psi]$	iff $P_s^{\sigma,max(min)}(\{\pi \in \Pi_s \mid \pi \models \psi\}) \lhd p(\rhd p)$
$s \models R_{\lhd v(\rhd v)}\,[\rho]$	iff $\mathbb{E}_s^{\sigma,max(min)}(rew(r,\rho)) \lhd v(\rhd v)$
$\pi \models \mathbf{X}\,\phi$	iff $\pi[1] \models \phi$
$\pi \models \phi_1\ \mathbf{U}^{\,\leq k}\phi_2$	iff $\exists i \leq k \mid \pi[i] \models \phi_2 \wedge \forall j < i\ \pi[j] \models \phi_1$
$\pi \models \phi_1\ \mathbf{U}\,\phi_2$	iff $\exists k \geq 0 \mid \pi \models \phi_1\, \mathcal{U}^{\leq k}\phi_2$
$rew(r, \mathbf{F}\,\phi)(\pi) := \Sigma_{t=0}^{t_\phi} r_s(\pi_s[t]) + r_a(\pi_s[t], \pi_a[t])$,	
$t_\phi = min\{t \mid \pi_s[t] \models \phi\}$	

Path formulas use the *Next* $(\mathbf{X}\,)$, *Bounded Until* $\left(\mathbf{U}^{\,\leq k}\right)$ and *Unbounded Until* $(\mathbf{U}\,)$ operators. They are evaluated over paths and only allowed as parameters to the P operator. Reward formulas use the *Instanteneous* $\left(\mathbf{I}^{\,=k}\right)$, *Bounded Cumulative* $\left(\mathbf{C}^{\,\leq k}\right)$ and *Cumulative* $(\mathbf{F}\,)$ operators. The size $\mathcal{Q}$ of a PCTL formula is defined as the number of Boolean connectives plus the number of P and R operators in the formula. We define $P_s^{\sigma,\eta^a}[\psi] \overset{def}{=} Prob\left(\{\pi \in \Pi_s^{\sigma,\eta^a} \mid \pi \models \psi\}\right)$ the probability of taking a path $\pi \in \Pi_s$ that satisfies ψ under strategy σ and nature η^a. $P_s^{\sigma,max}[\psi]$ ($P_s^{\sigma,min}[\psi]$) denote the maximum (minimum) probability $P_s^{\sigma,\eta^a}[\psi]$ across all natures $\eta^a \in Nat$, for a fixed strategy σ. An analogous definition holds also for reward properties, which can be expressed also using (multiple) reward structures different from the one use to maximize the total expected reward. For a CMDP $\mathcal{M}_\mathcal{C}$, strategy σ, and property ϕ, we will also use $\mathcal{M}_\mathcal{C}, \sigma \models_{Nat} \phi$ to denote that, when starting from any initial state $s \in S_0$, and operating under σ, $\mathcal{M}_\mathcal{C}$ satisfies ϕ for any $\eta^a \in Nat$. The semantics of the logic is reported in Table 1, where we write $\models$ instead of $\sigma \models_{Nat}$ for simplicity.

To illustrate our results, we will use the EMDP $\mathcal{M}_\mathcal{C}$ in Fig. 1, with $S = \{s_0 \cdots s_3\}$, $S_0 = \{s_0\}$, $A = \{a, b\}$, $\Omega = \{\vartheta, abs\}$, $\mathcal{A} : \{s_0, s_1\} \to \{a, b\}$; $\{s_2, s_3\} \to \{a\}$, $L : \{s_0, s_2\} \to \vartheta$; $\{s_3\} \to abs$. The parameters of the uncertainty ellipsoids are shown next to each transition.

3 Related Work

Related work falls into two main categories: renewable-energy pricing in smart grids, and strategy synthesis from PCTL specifications for probabilistic systems.

The integration of renewable energy sources in power grids has motivated the development of stochastic frameworks to solve the energy-pricing problem. The work in [15] presents a risk-limiting optimization framework. That effort focuses on modeling the uncertainty in energy availability on the supply side, but it does not consider the problem of controlling user demand through economic incentives. The effectiveness of demand response in balancing supply and demand in power grids was studied in [8], and a stochastic framework to optimize operator profits was presented in [9]. We closely follow the optimization setup presented in these works, but we also constrain the operator risk and the user QoS at synthesis time. Finally, Varaiya *et al.* argued the need to quantitatively constraint the operator risk in [6]. The risk-limiting dispatch approach proposed in that work is optimal for properties of the form $P_{\geq 1}$ or $P_{\leq 0}$, but sub-optimal for properties with satisfaction threshold $p \in (0, 1)$. Further, QoS is not considered.

The problem of strategy synthesis for MDPs from PCTL specifications was first studied in [16]. Strategies are divided into four categories depending on: 1) whether the transition is chosen deterministically (D) or randomly (R); 2) the choice does (does not) depend on the sequence of previously visited states (Markovian (M) and history-dependent (H)). Also, it is proven that the four types of strategies form a strict hierarchy (MD $\prec$ MR $\prec$ HD $\prec$ HR), and that determining whether it exists an MD/MR (HD/HR) strategy that meets all specifications is NP-complete (elementary). Kučera *et al.* [17] show how to synthesize MR controllers that are robust to linear perturbations via a reduction to a formula in the first-order logic of reals. This work is the closest to ours, albeit we consider also non-linear models of uncertainties. Also, to the best of our knowledge, the algorithm has not been applied to any case study. In [18] and [19] routines for the verification of PCTL properties of MDPs are adapted to the strategy synthesis problem. These algorithms are polynomial in the model size, but they are not complete [18] or can handle properties with only one quantitative operator [19]. Finally, [20] studies the synthesis of multi-strategies for MDPs. This approach can handle only a subset of PCTL properties and it only considers MDPs with no uncertainties.

4 Constrained Total Expected Reward Maximization for CMDPs

We formally define the optimization problem under analysis, prove that its decision problem version is NP-complete, and present an algorithm to solve the former.

Constrained Total Expected Reward Maximization for EMDPs. **Given** an EMDP $\mathcal{M}_{\mathcal{C}}$, a reward structure r, and a PCTL formula ϕ, **determine** strategy σ^* for $\mathcal{M}_{\mathcal{C}}$ such that:

$$\sigma^* = \underset{\sigma \in \Sigma^{MD}_{\mathcal{M}_{\mathcal{C}},\phi}}{\text{argmax}} \; \mathbb{W}^{\sigma}_{\mathcal{M}_{\mathcal{C}},s} \tag{4}$$

where $\Sigma^{MD}_{\mathcal{M}_{\mathcal{C}},\phi}$ is the set of Markov-Deterministic strategies for which $\mathcal{M}_{\mathcal{C}}$ satisfies ϕ for any $\eta^a \in Nat$, starting from any $s \in S_0$ and operating under $\sigma \in \Sigma_\phi$. The same results hold for the dual problem of minimizing the EMDP total expected cost.

We will use the following lemmas (the proofs are available in the references):

Lemma 4.1. Complexity of PCTL model-checking for CMDPs [10]. *Verifying that a CMDP $\mathcal{M}_{\mathcal{C}}$ satisfies a PCTL formula ϕ is solvable in polynomial time.*

Lemma 4.2. Computation of the total expected reward for CMCs [11]. *Given a CMC, its total expected reward is computable in polynomial time.*

Lemma 4.3. Complexity of PCTL strategy synthesis for MDPs [16]. *The problem of determining the existence of an MD strategy σ for an MDP $\mathcal{M}$ such that $\mathcal{M}, \sigma \models \phi$ is NP-complete.*

We now prove that the decision problem version of Problem (4) is NP-complete.

Theorem 4.1. *The problem of determining the existence of an MD strategy σ for an EMDP $\mathcal{M}_{\mathcal{C}}$, with total expected reward $\mathbb{W}^{\sigma}_{\mathcal{M}_{\mathcal{C}},s}$ larger or equal to $\mathbb{W}_T$ and satisfying specifications ϕ in PCTL, is NP-complete.*

Proof. Given a candidate solution σ_c, we can in polynomial time: 1) check whether $\mathcal{M}_{\mathcal{C}}, \sigma_c \models_{Nat} \phi$ by Lemma 4.1; 2) compute $\mathbb{W}^{\sigma_c}_{\mathcal{M}_{\mathcal{C}}}$ on the induced CMC $\mathcal{M}^{\sigma_c}_{C}$ by

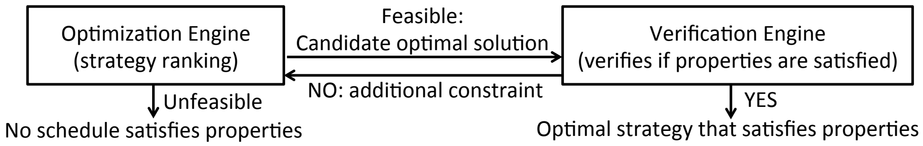
Optimization Engine
(strategy ranking)
Feasible:
Candidate optimal solution
Verification Engine
(verifies if properties are satisfied)
NO: additional constraint
Unfeasible
No schedule satisfies properties
YES
Optimal strategy that satisfies properties

Fig. 2: *Lazy* algorithm for constrained optimization.

Lemma 4.2. σ_c is a solution if and only if check 1) passes and $\mathbb{W}^{\sigma_c}_{\mathcal{M}_\mathcal{C}} \geq \mathbb{W}_T$. This proves that the problem is in NP. To prove NP-hardness, we reduce the problem in Lemma 4.3 to the one under analysis. We set $\mathbb{W}_T = 0$, and describe transition probabilities with point ellipses, i.e., ellipses with null axes ($\mathcal{K} \to +\infty$ in Equation (1)). □

In the rest of the section, we describe an algorithm to solve Problem (4). We use a *lazy* approach, conceptually similar to the ones proposed in [21] and, for non-linear constraints, [22]. As shown in Fig. 2, the algorithm is split into two main routines communicating in a loop. The *optimization* engine (OE) is responsible to generate a candidate strategy σ_c. The candidate solution is then passed to the *verification* engine (VE) which checks whether $\mathcal{M}_\mathcal{C}$ satisfies ϕ for all resolutions of uncertainty, i.e., $\forall \eta^a \in Nat$, when operating under σ_c. If the check passes, $\sigma^* = \sigma_c$ and the algorithm terminates. Otherwise, the VE generates an additional constraint for the OE to prevent the previous σ_c to be selected again, and the loop repeats. The novelty of our approach is devising a mathematical formulation for the OE capable of generating candidate strategies *in order of optimality* with respect to the total expected reward. The first candidate strategy that also satisfies all PCTL properties becomes the solution of the synthesis problem. The next subsections give details on the OE and VE and analyze the algorithm properties.

4.1 Optimization Engine

In Problem (5) on the left, we start with the classical linear-programming (LP) formulation to maximize the total expected reward for MDPs [1].

$$
\begin{array}{llll}
\min\limits_{x,l} \mathbf{x^T 1} & \max\limits_{x,z,l,n} \mathbf{x^T 1} & & \\
\text{s.t. } x_s - l_s^a = r_s^a + \mathbf{x^T f_s^a}; \Leftrightarrow & \text{s.t. } x_s - l_s^a + n_s^a = r_s^a + \mathbf{x^T f_s^a}; & \forall s \in S, \forall a \in \mathcal{A}(s) & \text{(5a)} \\
 & l_s^a \leq B z_s^a, \quad n_s^a \leq B z_s^a; & \forall s \in S, \forall a \in \mathcal{A}(s) & \text{(5b)} \\
 & \mathbf{z_s^T 1} = M_s - 1; & \forall s \in S & \text{(5c)} \\
x_s, l_s^a \geq 0 & x_s, l_s^a, n_s^a \geq 0, \quad z_s^a \in \{0,1\} \forall s \in S, \forall a \in \mathcal{A}(s) & &
\end{array}
$$

Vector $\mathbf{x}$ collects the total expected reward for each state $s \in S$ (at the end of the optimization $\mathbb{W}^{\sigma_c}_{\mathcal{M}_\mathcal{C},s} = x_s, \forall s \in S$), and variables l_s^a are slack variables for each constraint. Also, $r_s^a = r_s(s) + r_a(s,a), \forall s \in S, \forall a \in A$. Since the slack variables have negative sign, the slack can only be negative, i.e., the left-hand side (LHS) can only be larger or equal than the right-HS (RHS). The "min" operator makes sure that, for each state, the constraint with the highest RHS has null slack, i.e., $l_s^a = 0$. The optimal strategy can then be reconstructed by selecting the action $a \in \mathcal{A}(s), \forall s \in S$ corresponding to the constraint with null slack, e.g., $\sigma_c(s_0, a) = 1$ if $l_{s_0}^a = 0$. Our goal is modifying such a formulation to allow a sub-optimal solution to be selected, in case the optimal solution does not satisfy the PCTL specifications. We now describe an *equivalent* formulation to the original problem that is more suitable to achieve this goal. We will describe in Section 4.2 how to *add constraints* to this formulation to actually select sub-optimal solutions in order of optimality. We refer to Problem (5) on the right. We associate a binary variable z_s^a to each action for every state, so the problem becomes a Mixed-Integer Linear Program (MILP). $z_{s_i}^{a_j} = 0$ if action a_j is chosen for state s_i, and Constraint (5c) guarantees that only one action can be selected for each state ($M_s{=}|\mathcal{A}(s)|$). For example, $\sigma_c(s_0, a) = 1$ if $z_{s_0}^{a} = 0$. We then associate to each constraint a second slack variable n_s^a, with sign opposite to l_s^a. For selected actions, $z_{s_i}^{a_i} = 0$, Constraint (5b) makes sure that $z_{s_i}^{a_j} = 0$ implies $l_{s_i}^{a_j} = 0 \wedge n_{s_i}^{a_j} = 0$, so that the corresponding Constraint (5a) sets the value of x_s (B is a *big* number with respect to the problem data). For unselected actions, $z_{s_i}^{a_k} = 1$, variable $l_{s_i}^{a_k} > 0$ ($n_{s_i}^{a_k} > 0$) if selecting action a_k had resulted in a lower (higher) value of x_{s_i}. With these constraints, *any* action can be selected. We, finally, change the optimization operator to "max", so that, at the first iteration of the algorithm, the total expected reward gets maximized.

We now proceed to consider uncertainties in the transition probabilities. Constraint (5a) gets updated to Constraint (6) on the left, since the adversarial nature tries to minimize the expected reward. The new constraint can be made linear again for an arbitrary uncertainty model by replacing it with a set of constraints, one for each point in $\mathcal{F}_s^a$. However, this approach results in infinite constraints if the set $\mathcal{F}_s^a$ contains infinitely many points, as in the cases considered in the paper, thus making the problem not solvable. Using a construction similar to the one presented in [10], we solve this difficulty for the ellipsoidal uncertainty model using duality. In Constraint (6) on the left, we replace the primal inner problem with its dual, $\forall s \in S, a \in \mathcal{A}(s)$:

$$x_s - l_s^a + n_s^a = r_s^a + \min_{\mathbf{f_s^a} \in \mathcal{F}_s^a} \mathbf{x}^{\mathbf{T}} \mathbf{f_s^a} \qquad \Rightarrow \qquad x_s - l_s^a + n_s^a = r_s^a + \max_{\lambda_s^a \in \mathcal{D}_s^a} g\left(\lambda_s^a, \mathbf{x}\right) \tag{6}$$

$\lambda_\mathbf{s}^\mathbf{a} = \boxed{\lambda_{1,s}^a, \lambda_{2,s}^a, \lambda_{\mathbf{3,s}}^\mathbf{a}}$	Vector Lagrange multiplier
$g\left(\lambda_s^a, \mathbf{x}\right) = \lambda_{1,s}^a - \lambda_{2,s}^a - \mathbf{h_s^a}^T E_s^a \lambda_{\mathbf{3,s}}^\mathbf{a}$	Dual cost function
$\mathcal{D}_s^a = \{\lambda_s^a \mid \lVert\lambda_{3,s}^a\rVert_2 \leq \lambda_{2,s}^a,\ \mathbf{x} - \lambda_{1,s}^a \mathbf{1} + E_s^{aT} \lambda_{\mathbf{3,s}}^\mathbf{a} = \mathbf{0}\}$	Dual feasibility set

The dual problem is convex by construction [23] and has size polynomial in $\mathcal{R}$ [10]. Since also the primal problem is convex, strong duality holds, i.e., the primal and dual optimal solutions coincide, because the primal problem satisfies Slater's condition [23] for any non-trivial uncertainty set $\mathcal{F}_s^a$. Any dual solution underestimates the primal solution. When substituting the primal problem with the dual in Constraint (6), we can drop the inner optimization operator because the outer optimization operator will nevertheless aim to find the least underestimate to maximize its cost function. We get the full formulation for the OE (the quantifiers $\forall s \in S, \forall a \in \mathcal{A}(s)$ are equal to Problem (5)):

$$\begin{array}{lcl}
\max\limits_{x,\lambda,l,n,z} \ \mathbf{x_s^T 1} & & \max\limits_{x,\lambda,z,l,n} \ \mathbf{x^T 1} \\
\text{s.t. } x_s - l_s^a + n_s^a = r_s^a + g\left(\lambda_s^a, \mathbf{x}\right); & \Rightarrow & \text{s.t. } x_s - l_s^a + n_s^a = r_s^a + \lambda_{1,s}^a - \lambda_{2,s}^a - \mathbf{h_s^a}^T E_s^a \lambda_{\mathbf{3,s}}^\mathbf{a}; \\
\quad l_s^a \leq B z_s^a, \quad n_s^a \leq B z_s^a; & & \quad l_s^a \leq B z_s^a, \quad n_s^a \leq B z_s^a; \\
\quad \mathbf{z_s^T 1} = M_s - 1; & & \quad \mathbf{z_s^T 1} = M_s - 1; \\
\quad x_s, l_s^a, n_s^a \geq 0, \lambda_s^a \in \mathcal{D}_s^a, z_s^a \in \{0,1\} & & \quad x_s, l_s^a, n_s^a, \lambda_{2,s}^a \geq 0,\ \lambda_{\mathbf{3,s}}^\mathbf{a} \geq \mathbf{0},\ z_s^a \in \{0,1\}; \\
 & & \quad \lVert\lambda_{3,s}^a\rVert_2 \leq \lambda_{2,s}^a,\ \mathbf{x} - \lambda_{1,s}^a \mathbf{1} + E_s^{aT} \lambda_{\mathbf{3,s}}^\mathbf{a} = \mathbf{0}
\end{array} \tag{7}$$

For the ellipsoidal model, Problem (7) is a Mixed-Integer Quadratic-Constrained Program (MIQCP), which we solve using the back-end optimizer Gurobi [24].

4.2 Verification Engine

After fixing a candidate strategy σ_c, all the non-determinism in $\mathcal{M}_\mathcal{C}$ is resolved. The VE has thus the task of checking whether the *induced* Ellipsoidal-MC (EMC) $\mathcal{M}_{\sigma_c} =$

$(S, S_0, \Omega, \mathcal{F}, \mathcal{X}, L)$ satisfies the PCTL formula ϕ for all resolutions of uncertainty, i.e., $\forall \eta^a \in Nat$. We use the sound and complete verification algorithm presented in [10]. the main result presented there (EMCs are a special case of CMDPs). If the EMC satisfies ϕ, then the optimal strategy has been found and $\sigma^* = \sigma_c$. Otherwise, the VE needs to generate an additional constraint to be passed to the OE, so that the same candidate solution does not get selected anymore. If vector $\mathbf{z_c} = [z^a_{s_0} \cdots z^a_{s_N}]$ collects all the binary decision variables that were set to zero in the previous round of optimization, i.e., the variables corresponding to the previously selected actions, we just need to add constraint:

$$\mathbf{z_c^T 1} = 1 \tag{8}$$

so that it is not possible to select the same set of actions again.

As an example, we optimize the total expected reward of the EMDP in Fig. 1 subject to property $\phi = P_{\geq 0.8}[\vartheta \mathbf{U}\ abs]$. The first iteration of the OE generates strategy σ_{c1}, which selects actions $[b, a, a, a]$ for states $[s_0 \cdots s_3]$, with $\mathbb{W}^{\sigma_{c1}}_{s_0} = 10.625$, but the VE reports $P^{\sigma_{c1},min}_{s_0}[\vartheta \mathbf{U}\ abs] = 0.207$, so the strategy is rejected. The VE adds the constraint $z^b_{s_0} + z^a_{s_1} + z^a_{s_2} + z^a_{s_3} = 1$ to the OE formulation, which generates σ_{c2} ($[a, a, a, a]$), with $\mathbb{W}^{\sigma_{c1}}_{s_0} = 10.188$, at the second generation. The VE computes $P^{\sigma_{c2},min}_{s_0}[\vartheta \mathbf{U}\ abs] = 1$, and the algorithm terminates reporting $\sigma^* = \sigma_{c2}$.

4.3 Algorithm Analysis

We prove *soundness* (if a strategy σ^* is returned, it indeed optimally solves Problem 4), *completeness* (if no solution is returned, no strategy $\sigma \in \Sigma^{MD}_{\mathcal{M}_C}$ exists such that $\mathcal{M}_\mathcal{C}, \sigma \models_{Nat} \phi$) and analyze the runtime performance of the proposed algorithm.

Theorem 4.2. *The algorithm presented in this section to solve Problem 4 is sound, complete and has runtime exponential in the size $\mathcal{R}$ of the EMDP and polynomial in the size $\mathcal{Q}$ of the PCTL specification.*

Proof. Problem (7) returns the MD strategy σ_c that maximizes $\mathbb{W}^{\sigma}_{\mathcal{M}_C}$ among those still available. By Lemma 4.1, the VE is sound, so if it returns $\mathcal{M}_\mathcal{C}, \sigma_c \models_{Nat} \phi$, indeed $\sigma^* = \sigma_c$ (exit arrow at the bottom-right corner of Fig. 2). The VE is also complete, so if it returns $\mathcal{M}_\mathcal{C}, \sigma_c \not\models_{Nat} \phi$, the current σ_c can be discarded. This is done by generating a constraint of the form of Constraint (8), which removes only the current σ_c from the strategies to be explored by the OE. This proves the soundness of the overall algorithm.

Failure of finding a solution is declared only by the OE, when Problem (7) becomes unfeasible because all available strategies have previously been discarded by the VE (exit arrow at the bottom-left corner of Figure 2). This proves completeness.

Finally, the algorithm goes at most through $I = O(M^N)$ iterations. (Recall that M is the number of actions, and N the number of states of the EMDP.) Each requires solving an instance of Problem (7) and a verification check (done in time polynomial in $\mathcal{R}$ and $\mathcal{Q}$ by Lemma 4.1). Problem (7) can be solved by branch-and-bound algorithms in time exponential in the number of binary variables (whose number is constant) and polynomial in the number of constraints (whose number is polynomial in $\mathcal{R}$ at the first iteration, and it grows at each iteration limited by $I = O(M^N)$). The total complexity is $O(M^N \times \big(2^{MN} \times poly(M^N) + poly(\mathcal{Q})\big))$, exponential in $\mathcal{R}$ and polynomial in $\mathcal{Q}$. □

The algorithm performs better on problems which do have a feasible solution, arguably the most interesting ones, while the optimization step could be removed if the goal is to prove unfeasibility. As an alternative to our approach, σ^* could be determined by testing all I available MD strategies, and selecting the one with the highest reward among those satisfying ϕ [16]. We believe (and experimentally show in Section 6) that our approach can achieve better running time by decoupling the problem into an optimization and a verification part and by testing strategies in order of optimality. Finally, speed-ups can be obtained by implementing online routines for integer-constraint simplification and to produce more succinct certificates of unfeasibility from the VE [21, 22]. These routines are outside the scope of this paper and will be covered in future work.

5 Supply Scheduling and Energy Pricing with Renewable Sources

We model the pricing and dispatch problem following the scenario sketched in Fig. 3a [6, 9, 25]. An extended description is in Appendix A. Three agents operate in the system: the network operator, and two types of users, traditional and opportunistic. Further, three energy sources are available: non-dispatchable wind, and two dispatchable sources, base-line and fast-start generators. The network operator takes two kinds of decisions: 1) dispatch of non-renewable sources, to guarantee that the aggregate energy supply matches the demand; 2) pricing of energy, to maximize profits and incentivize users to join or leave the network depending on energy availability. Traditional and opportunistic users react to pricing decisions on different timescales. Traditional users only react to *day-ahead* pricing to decide how much energy they are willing to purchase, while opportunistic users are capable of rescheduling in *real-time* their energy demand, depending on the energy price, in exchange of lower *expected* prices. A 24-hour period gets divided into T_1-slots of equal length (e.g. $T_1 = 1h$), and each T_1-slot into K T_2-slots (e.g. $T_2 = 30min, K = 2$) [9]. The operator maximizes its economic profit by taking decisions on the two time-scales: 1) on day-ahead, for each T_1-slot, it dispatches Q units of base-line energy (unit cost, c_1), with $q = Q/K$ units per T_2-slot, and sets the price for traditional users (u), so that they can decide when to schedule their demand; 2) in real-time, for each T_2-slot, it sets the price for opportunistic users ($v_k, 1 \leq k \leq K$), and dispatches the production of more fast-start energy (c_2) or cancels part of the already dispatched base-line energy (paying c_p instead of c_1), depending on wind availability and user demand, to balance supply and demand. In real scenarios, $c_p < c_1 < c_2$ and wind-energy is assumed to be free for brevity [9]. The operator thus tries to use as much wind energy as possible and to dispatch on day-ahead the exact amount of base-line energy, not to incur in real-time in cancellation costs or in the extra cost for fast-start supplies. Since more profitable strategies might imply a higher reliance on the uncertain wind energy or an increase in energy prices, correct system functionality needs limits on energy-unbalance risks and QoS guarantees for users.

There are three sources of stochastic behavior: traditional (D_t) and opportunistic (D_o) user demand and wind-energy supply (W). We thus use a stochastic optimization framework. We focus on day-ahead decisions (Q and u), which are taken based on the *expected* value of D_t, D_o, W. Real-time decisions (v_k) are instead taken deterministically based on the *observed* values. Nevertheless, in day-ahead, the operator needs to predict the optimal value of v_k for *each possible observation* of W_k and $D_{t,k}$, since $D_{o,k}$ depends on that decision, so also v_k is a decision variable. We will optimize over one T_1-slot (the decision problem is periodic, so we can run one optimization for each T_1-slot stand-alone), and aim to determine optimal values for Q, u and $v_k, 1 \leq k \leq K$.

We use the Ellipsoidal-MDP $\mathcal{M}_E = (S, S_0, A, \Omega, \mathcal{F}, \mathcal{A}, \mathcal{X}, L)$ sketched in Fig. 3b. All quantities are bounded and uniformly discretized to keep the state and action spaces

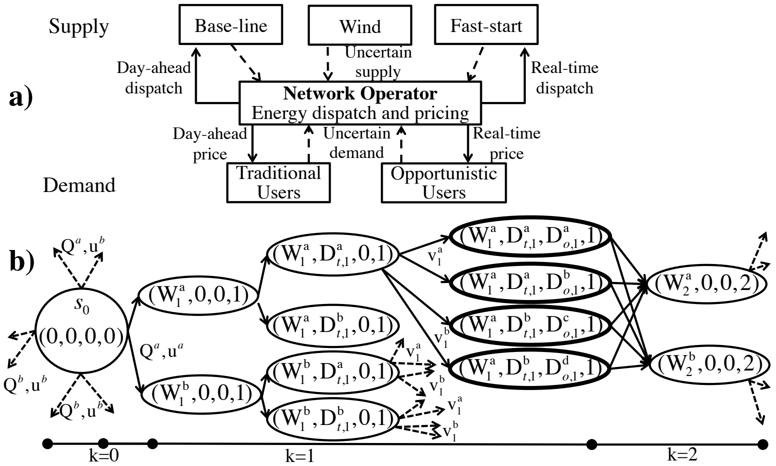

Fig. 3: a) Network operator inputs (dashed) and outputs (solid). b) Sketch of $\mathcal{M}_E$ assuming two *discretization levels* for each quantity (labeled with superscripts).

finite. States $s \in S$ are a tuple $s = (D_{t,k}, D_{o,k}, W_k, t_k)$, where $1 \leq k \leq K$ indexes one of the K T_2-slots and $D_{t,k}, D_{o,k}, W_k$ refer to the observed values of user demand and wind energy in that state. At the initial state s_0, the operator makes the day-ahead decisions on energy dispatch Q and pricing for traditional users u (pair $(Q, u) \in A$). The process then transitions through K decision epochs, as follows. First, values of wind energy (W_k) and traditional user demand ($D_{t,k}$) are stochastically chosen according to the corresponding distributions (described below). Second, for each observation of W_k and $D_{t,k}$, a decision on $v_k \in A$ is made and the opportunistic user demand ($D_{o,k}$) is stochastically chosen. To transition between epochs a new value of wind energy W_{k+1} is chosen and the steps repeat. At the end, the states transition to an absorbing state. State transition probabilities are computed using the following stochastic models.

User demand. The demand of both traditional (D_t) and opportunistic (D_o) users is modeled using Gaussian distributions [9], with $D_t \sim \mathcal{N}(\alpha_t u^{\gamma_t}, \beta_t \mathbb{E}[D_t])$, and $D_o \sim \mathcal{N}(\alpha_o v^{\gamma_o}, \beta_o \mathbb{E}[D_o])$. Parameter γ_t (γ_o) is the *elasticity* of the traditional (opportunistic) users, i.e., the ratio of the percentage change of the expected demand to that of price variations, formally: $\gamma_t = {}^{u}\!/_{\mathbb{E}[D_t]} \cdot {}^{\partial \mathbb{E}[D_t]}\!/_{\partial u}$. Parameters $\alpha_t, \alpha_o, \beta_t, \beta_o$ are fitting parameters. To compute transition probabilities, we truncate and discretize the continuous PDs in equally-sized intervals, pick the middle point of each interval as the discretization value and then integrate the PD across the interval to determine how likely the system transitions to that discretized value. In principle, also user demand can be expressed using data-driven uncertainty models, but we did not consider this in this paper.

Wind-energy availability. We created a stochastic model of the available wind energy starting from measured data collected from the wind farm at Lake Benton, Minnesota, USA. The goal is to take forecast values into account, while also considering the intrinsic inaccuracies of these predictions. First, we compute the (discrete) empirical PD μ_W of a training set of collected wind-energy data. Second, we divide a new set of data in T_2-slots, and consider the average value for each new T_2-slot as the forecast energy value. We then scale μ_W to have such expected value $\mathbb{E}[W_k]$, thus obtaining μ_{W_k}. Finally, we compute ellipsoidal Sets (2) $\mathcal{F}_s^a$ (collected in $\mathcal{X}$) to represent uncertainty in the transition probability between two discretized energy levels in two consecutive

T_2-slots. Transition frequencies are computed by counting observed transitions in a training set of data. Further, using classical results from statistics [11], we can compute the value of parameter β from Set (1) corresponding to a desired confidence level C_L in the measurements. In particular, $0 \leq C_L \leq 1$ and $C_L = 1 - cdf_{\chi^2_d}(2 * (\beta_{max} - \beta))$, where $cdf_{\chi^2_d}$ is the cumulative density function of the Chi-squared distribution with d degrees of freedom (d is equal to the number of bins used to discretize W).

We provide the states with thick circles in Fig. 3b with three reward structures to express the profit and risk of the operator and the QoS for the users. We choose those states because the quantities $D_{t,k}, D_{o,k}, W_k$ are all fully observable in them, thus allowing the computation of the rewards. We set:

$$r^{Profit}_{s,k}[\$] = uD_{t,k} + v_k D_{o,k} - (c_p \Delta_k + c_1(q - \Delta_k))\mathbf{1}^A_{\Delta_k \geq 0} - (c_1 q - c_2 \Delta_k)\mathbf{1}^B_{\Delta_k < 0} \quad (9a)$$

$$r^{LoL}_{s,k}[MWh] = max(0, X\%\mathbb{E}[W_k] + Y\%\mathbb{E}[D_{t,k} + D_{o,k}] - \Delta_k) \quad (9b)$$

$$r^{Quality}_{s,k}[MWh] = D_{t,k} + D_{o,k} \quad (9c)$$

with $\Delta_k = W_k + q - D_{t,k} - D_{o,k}$ representing the surplus of supply on demand, and $\mathbf{1}$ the indicator function. Reward (9a) subtracts operating costs to the operator revenue to compute the net profit. Indicator $\mathbf{1}^A$ ($\mathbf{1}^B$) corresponds to the scenario when the sum of day-ahead dispatched and wind energy is sufficient (insufficient) to cover the demand. In the latter case, fast-start energy needs to be dispatched in real-time. Reward (9b) computes the Loss of Load (LoL). In practical scenarios, the amount of fast-start energy available in real-time is limited. Often this limit is computed with the formula $FS \leq X\%\mathbb{E}[W] + Y\%\mathbb{E}[D_t + D_o]$ (e.g. $X = 3, Y = 10$) [26]. If $\Delta + FS < 0$ the network incurs in a LoL, with potentially risky consequences. Reward (9c) accounts for user demand incentivized by energy pricing. Finally, we mark all states with $\Delta + FS < 0$ with the label *risk*, and use label *abs* for the absorbing state, so $\Omega = \{risk, abs\}$.

The optimal strategy $\sigma^* = (u^*, Q^*, v^*_k), 1 \leq k \leq K$ is the solution of problem:

$$\begin{aligned} \mathbb{W}^*_{\mathcal{M}_E, s_0} = \max_{Q,u} \min_{\mathbf{f^a_s} \in \mathcal{F}^a_s} \mathbb{E}^{\sigma, \mathbf{f^a_s}}_W \mathbb{E}^{\sigma}_{D_t} \textstyle\sum_{k=1}^K \max_{v_k} \mathbb{E}^{\sigma}_{D_o} rew(r^{Profit}_s) \\ s.t.\ \mathcal{M}_E, \sigma^* \models_{Nat} \phi \quad \text{where:} \qquad (10) \\ \phi = R^{r^{LoL}}_{\leq EENS_M}[\mathbf{F}\, abs] \wedge R^{r^{Quality}}_{\geq QoS_m}[\mathbf{F}\, abs] \wedge P_{\geq 1 - LoLP_M}[\neg risk \mathbf{U}\, abs] \end{aligned}$$

which can be solved with the algorithm presented in Section 4. In Problem (10), we maximize the sum of the expected operator profit across the K T_2-slots for the worst-case resolution of uncertainty in the wind-energy forecast. According to the semantics defined in Table 1, the PCTL specification ϕ constraints the expected operator risk and user QoS across the decision horizon. $EENS_M$ is the desired maximum value of Expected Energy Not Served, $LoLP_M$ is the maximum allowed value of Loss of Load Probability (these two properties limit the risk for the operator), and QoS_m is the the minimum value of QoS that needs to be guaranteed to the users.

6 Experimental Results

We implemented the algorithm in Python, and interfaced it with PRISM [28] as a front-end for entering models and with Gurobi [24] as the back-end optimizer. Experiments were run on a 2.4 GHz Intel Xeon with 32GB of RAM.

In this section, we present experimental results obtained by solving Problem (10) using the proposed algorithm. Our goals are to give insight about the algorithm functionality, compare its scalability to other strategy-synthesis approaches, and show that the synthesized energy-pricing strategies can achieve better performances than other solutions presented in the literature. We define the following quantities:

$Profit := \mathbb{W}^{*}_{\mathcal{M}_E,s_0}$	$EENS := R_{s_0}^{r^{LoL},\sigma^*,max}[\mathbf{F}\,abs]$
$QoS := R_{s_0}^{r^{Quality},\sigma^*,min}[\mathbf{F}\,abs]$	$1 - LoLP := P_{s_0}^{\sigma^*,min}[\neg risk\mathbf{U}\,abs]$

where the min and max operators refer to the action range of nature Nat. As defined in Section 2.2, these quantities represent the *quantitative* values of rewards and satisfaction probability that then get compared to the corresponding thresholds ($EENS_M,\ QoS_m,\ LoLP_M$) in Problem (10) to determine the satisfaction of ϕ. We will then normalize the $Profit$ to $Profit_M$, the maximum computed profit value for each set of experiments. We set $T_1 = 1h$, $T_2 = 30min$ so $K = 2$, and consider two pricing options both for traditional and opportunistic users. If not otherwise stated, we will use $C_L = 90\%$, and discretize the wind energy W in 5 bins, and traditional D_t, opportunistic D_o demands and base-line supply Q in 2 bins. We set $QoS_m = 80\% \sum_k \mathbb{E}[D_{t,k} + D_{o,k}]$, $LoLP_M = 10\%$, $EENS_M = 5\% \sum_k \mathbb{E}[W_k + q]$. The other parameter values were taken from [9].

In Fig. 4, we show the trend of the expected system performance as a function of the synthesis algorithm iteration (enlarged figures are available in App. B). The $Profit$ monotonically decreases until the proposed candidate strategy σ_c meets all specifications. We note that $1 - LoLP$ and QoS ($EENS$, not shown, has a trend similar to $1 - LoLP$) instead vary non-monotonically. Intuitively, this is because the $Profit$ can be increased either by scheduling less base-line energy Q, to reduce cancellation costs c_p but incurring in a higher risk, or by increasing the energy price v for opportunistic users, with consequent reduction of QoS. By ranking strategies by expected $Profit$, our algorithm is capable of selecting the optimal strategy despite the complex parameter interdependences in the model under analysis. In Table 2, we compare synthesis results while varying the number of discretization bins for W (all values are normalized to the corresponding target specification). First, we note that the expected system performances do not substantially vary by changing the number of bins, thus supporting our choice of 5 bins in the other experiments. Second, runtime results show that the algorithm can handle in reasonable time problems of size more than $10\times$ larger than the ones analyzed by [18], the only other algorithm proposed in the literature capable of accepting arbitrary PCTL formulas (we use $N + Tr$, the sum of states and transitions in the EMDP, to represent the model size). Third, results show that the alternative approach of verifying all the I MD strategies $\sigma \in \Sigma^{MD}$ is not practical, due to the exponential increase of I (as defined in Section 2.1) with the problem size. In Fig. 5, we study the effect of different confidence levels C_L in the wind-energy forecast on the expected $Profit$ for the operator, while varying the value of wind penetration $\eta_W = \sum_{k=1}^{K} \mathbb{E}[W_k] / \mathbb{E}[W_k + q]$ and keeping all constraints constant. At high C_L, higher profits can be expected for increasing η_W (wind energy is assumed free). On the other hand, for low C_L, higher wind penetration creates more uncertainty thus lowering the expected $Profit$. The network operator can use these curves to assess the return of investment in employing more accurate (and expensive) forecast techniques. Finally, in Fig. 6 we compare results with two other energy-pricing formulations proposed in the literature. He *et al.* [9] solve the optimization problem without enforcing any constraint. Varaiya *et al.* [6] only put limits on the acceptable LoLP (their approach is not trivially extendable to reward properties expressed using the R operator) and solve optimally only for $LoLP = 0$. Comparison is done by solving the different optimization problems and then running Monte Carlo simulations (1000 runs) of the controlled system on test data (different from the training ones) to evaluate its performance ($Profit$, $EENS$ and QoS). As expected, the unconstrained strategy from [9] has higher $Profit$ (up to 5%), but also up to 12% more $EENS$ and 10% less QoS, compared to our approach. The strategy from [6] guarantees null $EENS$, but it has up to 6% less $Profit$ (due to over constraining $EENS$) and 10% less QoS (which is left unconstrained). As a final remark, we note that runtime may increase exponentially as we tighten the specification thresholds ($QoS_m,\ LoLP_M,\ EENS_M$), since it becomes increasingly difficult to find a solution within the exponentially-sized search space. Nevertheless, the chosen values were tight enough to improve the quality of alternative energy-pricing strategies proposed in the literature, while maintaining the runtime acceptable for this application.

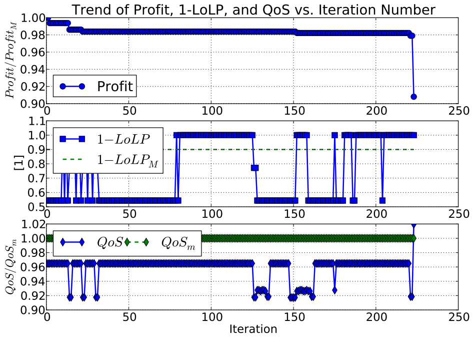

Fig. 4: Performance vs. iteration.

Table 2: Performance Analysis

W bins	5	10	15	20
$Profit$	1	0.98	0.97	0.965
$1 - LoLP$	0.99	0.99	0.99	0.99
$EENS$	0.98	0.98	0.98	0.98
QoS	1.01	1.01	1.01	1.01
Runtime	144s	400s	1368s	3289s
#Iter.	223	53	547	332
$N + Tr$	1343	2719	4115	5591
#MD Strat. (I)	4096	4.2e6	4.3e9	4.4e12

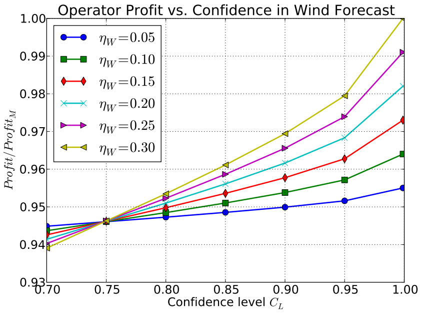

Fig. 5: Profit vs. confidence in forecast.

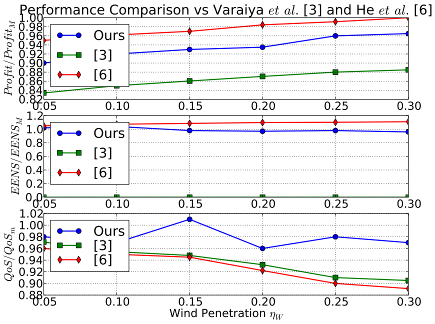

Fig. 6: Comparison to alternative approaches via Monte Carlo simulation.

7 Conclusions and Future Work

We first proposed a novel algorithm for the synthesis of control strategies for MDPs, satisfying properties expressed in PCTL and robust to uncertainties in the transition probabilities. We then applied the algorithm to the problem of renewable-energy pricing in smart grids and showed that network-operator risks can be effectively constrained at design time and that more accurate predictions of the expected profit can be obtained by taking the uncertainty of wind availability into consideration.

As future work, we plan to investigate techniques to generate more concise constraints to prove failure of the verification, in order to prune more effectively the search space for the optimization engine, and to apply the proposed strategy synthesis approach to further case studies, e.g., semi-autonomous car driving.

References

1. M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.
2. H. Hansson and B. Jonsson, “A Logic for Reasoning About Time and Reliability,” *Formal Aspects of Computing*, vol. 6(5), pp. 512–535, 1994.
3. A. Bianco and L. de Alfaro, “Model Checking of Probabilistic and Nondeterministic Systems,” in *Foundations of Software Technology and Theoretical Computer Science*, ser. Lecture Notes in Computer Science, P. Thiagarajan, Ed. Springer Berlin Heidelberg, 1995, vol. 1026, pp. 499–513. [Online]. Available: http://dx.doi.org/10.1007/3-540-60692-0_70
4. “International Energy Agency, World Energy Outlook 2009.” [Online]. Available: http://www.worldenergyoutlook.org/docs/weo2009/WEO2009_es_english.pdf.
5. A. Gore, *An Inconvenient Truth*. New York: Rodale, 2006.
6. P. Varaiya, F. Wu, and J. Bialek, “Smart Operation of Smart Grid: Risk-Limiting Dispatch,” *Proceedings of the IEEE*, vol. 99, no. 1, pp. 40–57, 2011.
7. K. Porter and J. Rogers, “Survey of Variable Generation Forecasting in the West,” in *NREL Subcontract Report SR-5500-54457*, April 2012, p. 56pp.
8. S. Koch, J. Mathieu, and D. Callaway, “Modeling and Control of Aggregated Heterogeneous Thermostatically Controlled Loads for Ancillary Services,” *Proceedings of the* 17^{th} *Power Systems Computation Conference*, 2011.
9. M. He, S. Murugesan, and J. Zhang, “Multiple Timescale Dispatch and Scheduling for Stochastic Reliability in Smart Grids with Wind Generation Integration,” in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 461–465.
10. A. Puggelli, W. Li, A. Sangiovanni-Vincentelli, and S. Seshia, “Polynomial-Time Verification of PCTL properties of MDPs with Convex Uncertainties,” in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, N. Sharygina and H. Veith, Eds. Springer Berlin Heidelberg, 2013, vol. 8044, pp. 527–542.
11. A. Nilim and L. El Ghaoui, “Robust Control of Markov Decision Processes with Uncertain Transition Matrices,” *Journal of Operations Research*, pp. 780–798, 2005.
12. K. Sen, M. Viswanathan, and G. Agha, “Model-Checking Markov Chains in the Presence of Uncertainties,” in *Tools and Algorithms for the Construction and Analysis of Systems*, ser. Lecture Notes in Computer Science, H. Hermanns and J. Palsberg, Eds. Springer Berlin Heidelberg, 2006, vol. 3920, pp. 394–410.
13. M. Y. Vardi, “Automatic Verification of Probabilistic Concurrent Finite State Programs,” in *Proc. of the 26th Symp. on Foundations of Computer Science*, ser. SFCS, 1985, pp. 327–338.
14. V. Forejt *et al.*, “Automated Verification Techniques for Probabilistic Systems,” *Formal Methods for Eternal Networked Software Systems (SFM)*, vol. 6659, pp. 53–113, 2011.
15. F. Bouffard and F. Galiana, “Stochastic Security for Operations Planning with Significant Wind Power Generation,” in *IEEE Transaction on Power Systems*, May 2008, vol. 23, no. 2, pp. 306–316.
16. C. Baier, M. Grer, M. Leucker, B. Bollig, and F. Ciesinski, “Controller synthesis for probabilistic systems,” in *Exploring New Frontiers of Theoretical Informatics*, ser. IFIP International Federation for Information Processing, J.-J. Levy, E. Mayr, and J. Mitchell, Eds. Springer US, 2004, vol. 155, pp. 493–506.
17. A. Kučera and O. Stražovský, “On the Controller Synthesis for Finite-State Markov Decision Processes,” in *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science*, ser. Lecture Notes in Computer Science, S. Sarukkai and S. Sen, Eds. Springer Berlin Heidelberg, 2005, vol. 3821, pp. 541–552.
18. M. Lahijanian, S. Andersson, and C. Belta, “Temporal logic motion planning and control with probabilistic satisfaction guarantees,” *Robotics, IEEE Transactions on*, vol. 28, no. 2, pp. 396–409, 2012.

19. M. Kwiatkowska and D. Parker, “Automated Verification and Strategy Synthesis for Probabilistic Systems,” in *Proc. 11th International Symposium on Automated Technology for Verification and Analysis (ATVA’13)*, ser. LNCS, D. V. Hung and M. Ogawa, Eds., vol. 8172. Springer, 2013, pp. 5–22.
20. K. Draeger, V. Forejt, M. Kwiatkowska, D. Parker, and M. Ujma, “Permissive Controller Synthesis for Probabilistic Systems,” in *Proc. 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS’14)*, ser. LNCS. Springer, 2014, to appear.
21. C. Hang, P. Manolios, and V. Papavasileiou, “Synthesizing Cyber-Physical Architectural Models with Real-Time Constraints,” in *Proceedings of the 23rd International Conference on Computer Aided Verification*, ser. CAV’11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 441–456.
22. P. Nuzzo, A. Puggelli, S. Seshia, and A. Sangiovanni-Vincentelli, “CalCS: SMT Solving for Non-linear Convex Constraints,” in *Proceedings of the 2010 Conference on Formal Methods in Computer-Aided Design*, ser. FMCAD ’10, 2010, pp. 71–80.
23. S. Boyd and L. Vandenberghe, “Convex Optimization,” *Cambridge University Press*, 2004.
24. “Gurobi Optimizer.” [Online]: http://www.gurobi.com/.
25. S. Stoft, *Power System Economics: Designing Markets for Electricity*. NewYork: Wiley, 2002.
26. E. Ela, M. Milligan, and B. Kirby, “Operating Reserves and Variable Generation,” in *NREL Technical Report TP-5500-51978*, August 2011.
27. Online: http://www.eecs.berkeley.edu/~puggelli/.
28. M. Kwiatkowska *et al.*, “PRISM 4.0: Verification of Probabilistic Real-Time Systems,” *Proc. of* 23^{rd} *Intl. Conf. on Computed Aided Verification*, pp. 585–591, 2011.
29. C. Courcoubetis and M. Yannakakis, “The Complexity of Probabilistic Verification,” *Journal of ACM*, vol. 42(4), pp. 857–907, 1995.

Appendices

A Supply Scheduling and Energy Pricing with Renewable Sources

In this appendix, we provide extra material to more extensively describe the case study analyzed in the paper, i.e., supply scheduling and energy pricing in smart-grids that integrate renewable sources, to help the reader better appreciate the presented results. For further details, we refer the reader to the work presented in [6, 9, 25].

We model the energy pricing and dispatch problem following the scenario sketched in Fig. 7a. Three agents operate in the system: the network operator, and two types of users, traditional and opportunistic. Further, three energy sources are available: non-dispatchable wind, and two dispatchable sources, base-line (e.g., thermal units) and fast-start generators (e.g., gas turbines).

The network operator aims to maximize its economic profit and to guarantee the correct operation of the network, by enforcing the supply and demand to be balanced at all times. The operator is provided with two sources of energy to guarantee power balance. First, classical fossil-based energy, which still accounts for the majority of the energy supply (nowadays, the penetration of renewable sources is rarely beyond 30% of the total energy supplied by the network). Fossil-energy supplies gets further divided into base-line (e.g., thermal units) which need to be dispatched with longer notice (e.g., a day) but have lower operational cost, and into fast-start (e.g., gas turbines) which can be dispatched on a short notice (e.g., half an hour) but have higher costs. As a second source of energy, the operator can count on renewable sources (we focused on wind-based energy generation, but a similar reasoning applies to other kinds of renewable sources, e.g., solar). As a further degree of freedom for the network operator to guarantee power balance in the network, we assume a scenario in which network users dynamically respond to changes in the price of energy by adapting their demand. This scenario is referred to in the literature with the concept of *demand response*. When the price of energy increases, users decrease their energy consumption and vice-versa. The network operator can thus compensate for renewable supply uncertainty, by dynamically adapting the price of energy. Decisions on price energy are taken based on the information about wind-energy availability. This information is provided on two time-scales. On *day-ahead*, the operator is provided with *forecast* values of energy availability for the following day. Although forecasting techniques have substantially improved in the last decades, this information is still affected by high uncertainty. In *real-time*, the operator has instead full observability of the wind-energy availability and it can finely adjust the energy price taking the additional information into account. Overall the operator needs to take two kinds of decisions: 1) dispatch of non-renewable sources, to guarantee that the aggregate energy supply (renewable plus fossil) matches the demand; 2) pricing of energy, to maximize profits and incentivize users to join or leave the network depending on energy availability.

On the demand side, researchers have foreseen a scenario in which users will be able to react to changes in energy-pricing on two time-scales. Most users (referred to as *traditional*) will require some notice before being able to react. We use variable D_t to represent the amount of energy (in MWh) demanded by traditional users. A fraction of the users (referred to as *opportunistic*) will instead try to adapt their demand (D_o) on a shorter notice in exchange for example of lower expected energy prices. For simplicity, we will use the assumption made in the literature that traditional users respond to *day-ahead* energy-pricing decisions made by the operator, while opportunistic users can adjust their demands in *real-time*.

A 24-hour period gets divided into T_1-slots of equal length (e.g. $T_1 = 1h$), and each T_1-slot into K T_2-slots (e.g. $T_2 = 30min, K = 2$) [9]. On day-ahead, the operator: 1) posts the price of energy for each T_1-slot, based on the available forecast value of wind-energy, so that traditional users can decide when to schedule their activities (which require energy consumption D_t); 2) dispatches Q units of base-line energy for each T_1-slot (unit cost, c_1), with $q = Q/K$ units per T_2-slot. In real-time, after *observing* the available amount of wind energy and the actual level of demand D_t of traditional users, the operator: 1) sets the price for opportunistic users ($v_k, 1 \leq k \leq K$) for each T_2-slot; 2) dispatches the production of more fast-start energy (unit cost, c_2) or cancels part of the already dispatch base-line energy (paying c_p instead of c_1), depending on wind availability and user demand, to guarantee the power balance at all times. In real scenarios, $c_p < c_1 < c_2$ and wind-energy is supposed to be free for brevity [9]. The operator thus tries to use as much wind energy as possible and to dispatch on day-ahead the exact amount of base-line energy, not to incur in real-time in cancellation costs or in the extra cost for fast-start supplies. Since more profitable strategies might imply a higher reliance on the uncertain wind energy or an increase in energy prices to incentivize users to leave the network[1], correct system functionality needs limits on energy-unbalance risks and QoS guarantees for users.

There are three sources of stochastic behavior: traditional (D_t) and opportunistic (D_o) user demand and wind-energy supply (W). We thus use a stochastic framework to solve the decision-making problem. We focus on day-ahead decisions (Q and u), which are taken based on the *expected* value of D_t, D_o, W. Real-time decisions (v_k) are instead taken deterministically based on the *observed* values. Nevertheless, in day-ahead, the operator still needs to predict which will be the optimal value of v_k for each possible observation of W and D_t, since $D_{o,k}$ depends on that decision, so also v_k is a decision variable in the day-ahead decision problem. We will optimize over one T_1-slot (the decision problem is periodic, so we can run one optimization for each T_1-slot stand-alone), and aim to determine optimal values for Q, u and $v_k, 1 \leq k \leq K$.

We use the Ellipsoidal-MDP $\mathcal{M}_E = (S, S_0, A, \Omega, \mathcal{F}, \mathcal{A}, \mathcal{X}, L)$ sketched in Fig. 7b. All quantities are bounded and uniformly discretized to keep the state and action spaces finite. States $s \in S$ are a tuple $s = (D_{t,k}, D_{o,k}, W_k, t_k)$, where $1 \leq k \leq K$ indexes one of the K T_2-slots and $D_{t,k}, D_{o,k}, W_k$ refer to the observed values of user demand and wind energy in that state. At the initial state s_0, the operator makes the day-ahead decisions on energy dispatch Q and pricing for traditional users u (pair $(Q, u) \in A$). The process then transitions through K decision epochs, as follows. First, a value of wind energy (W_k) and traditional user demand ($D_{t,k}$) is stochastically chosen according to the corresponding distributions (described below). Second, for each observation of W_k

[1] Even if there are fewer users in the network, the ones that remain in the network need to pay more, thus maintaining the operator profit high while reducing the risk of power unbalance. This solution may be acceptable for the operator but it is obviously disadvantageous for the users.

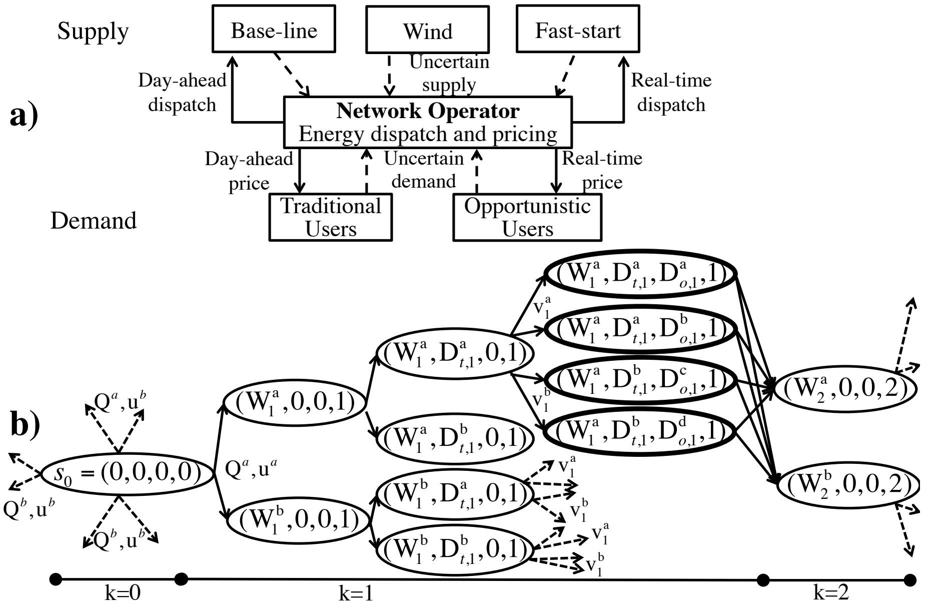

Fig. 7: a) Network operator inputs (dashed) and outputs (solid). b) Sketch of $\mathcal{M}_E$ assuming two *discretization levels* for each quantity (labeled with superscripts).

and $D_{t,k}$, a decision on $v_k \in A$ is made and the opportunistic user demand ($D_{o,k}$) is stochastically chosen. To transition between epochs a new value of wind energy W_{k+1} is stochastically chosen and the steps then repeat. At the end, all states transition to an absorbing state. State transition probabilities are computed using the following stochastic models.

User demand. The demand of both traditional (D_t) and opportunistic (D_o) users is modeled using Gaussian distributions [9], with $D_t \sim \mathcal{N}(\alpha_t u^{\gamma_t}, \beta_t \mathbb{E}[D_t])$, and $D_o \sim \mathcal{N}(\alpha_o v^{\gamma_o}, \beta_o \mathbb{E}[D_o])$. Parameter γ_t (γ_o) is the *elasticity* of the traditional (opportunistic) users, i.e., the ratio of the percentage change of the expected demand to that of price variations, formally: $\gamma_t = {}^{u}\!/_{\mathbb{E}[D_t]} \cdot {}^{\partial \mathbb{E}[D_t]}\!/_{\partial u}$. Parameters $\alpha_t, \alpha_o, \beta_t, \beta_o$ are fitting parameters. To compute transition probabilities, we truncate and discretize the continuous PDs in equally-sized intervals, pick the middle point of each interval as the discretization value and then integrate the PD across the interval to determine how likely the system transitions to that discretized value. In principle, also user demand can be expressed using data-driven uncertainty models, but we did not consider this in this paper.

Wind-energy availability. We created a stochastic model of the available wind energy starting from measured data collected from the wind farm at Lake Benton, Minnesota, USA. The goal is to take forecast values into account, while also considering the intrinsic inaccuracies of these predictions. First, we compute the (discrete) empirical PD μ_W of a training set of collected wind-energy data. Second, we divide a new set of data in T_2-slots, and consider the average value for each new T_2-slot as the forecast energy value. We then scale μ_W to have such expected value $\mathbb{E}[W_k]$, thus obtaining μ_{W_k}. Finally, we compute ellipsoidal Sets (2) $\mathcal{F}_s^a$ (collected in $\mathcal{X}$) to represent uncertainty in the transition probability between two discretized energy levels in two consecutive T_2-slots. Transition frequencies are computed by counting observed transitions in a training set of data. Further, using classical results from statistics [11], we can compute the value of parameter β from Set (1) corresponding to a desired confidence level C_L in the measurements. In particular, $0 \leq C_L \leq 1$ and $C_L = 1 - cdf_{\chi^2_d}\,(2 * (\beta_{max} - \beta))$, where $cdf_{\chi^2_d}$ is the cumulative density function of the Chi-squared distribution with d degrees of freedom (d is equal to the number of bins used to discretize W).

We provide the states with thick circles in Fig. 7b with three reward structures to express the profit and risk of the operator and the QoS for the users. We choose those states because the quantities $D_{t,k}, D_{o,k}, W_k$ are all fully observable in them, thus allowing the computation of the rewards. We set:

$$r^{Profit}_{s,k}[\$] = uD_{t,k}+v_k D_{o,k}-(c_p\mathit{\Delta}_k+c_1(q-\mathit{\Delta}_k))\mathbf{1}^A_{\mathit{\Delta}_k\geq 0}-(c_1 q-c_2\mathit{\Delta}_k)\mathbf{1}^B_{\mathit{\Delta}_k<0} \quad (11a)$$

$$r^{LoL}_{s,k}[MWh] = max(0, X\%\mathbb{E}[W_k] + Y\%\mathbb{E}[D_{t,k} + D_{o,k}] - \mathit{\Delta}_k) \quad (11b)$$

$$r^{Quality}_{s,k}[MWh] = D_{t,k} + D_{o,k} \quad (11c)$$

with $\mathit{\Delta}_k = W_k + q - D_{t,k} - D_{o,k}$ representing the surplus of supply on demand, and $\mathbf{1}$ the indicator function. Reward (11a) subtracts operating costs to the operator revenue to compute the net profit. Indicator $\mathbf{1}^A$ ($\mathbf{1}^B$) corresponds to the scenario when the sum of day-ahead dispatched and wind energy is sufficient (insufficient) to cover the demand. In the latter case, fast-start energy needs to be dispatched in real-time. Reward (11b) computes the Loss of Load (LoL). In practical scenarios, the amount of fast-start energy available in real-time is limited. Often this limit is computed with the formula $FS \leq X\%\mathbb{E}[W] + Y\%\mathbb{E}[D_t + D_o]$ (e.g. $X = 3, Y = 10$) [26]. If $\mathit{\Delta} + FS < 0$ the network incurs in a LoL, with potentially risky consequences. Reward (11c) accounts for user demand incentivized by energy pricing. Finally, we mark all states with $\mathit{\Delta} + FS < 0$ with the label *risk*, and use label *abs* for the absorbing state, so $\mathit{\Omega} = \{risk, abs\}$.

The optimal strategy $\sigma^* = (u^*, Q^*, v^*_k), 1 \leq k \leq K$ is the solution of problem:

$$\begin{aligned} \mathbb{W}^*_{\mathcal{M}_E,s_0} = \max_{Q,u}\min_{\mathbf{f^a_s}\in\mathcal{F}^a_s} & \mathbb{E}^{\sigma,\mathbf{f^a_s}}_{W}\mathbb{E}^{\sigma}_{D_t}\textstyle\sum_{k=1}^{K}\max_{v_k}\mathbb{E}^{\sigma}_{D_o} rew(r^{Profit}_s) \\ & s.t.\ \mathcal{M}_E, \sigma^* \models_{Nat} \phi \quad \text{where:} \qquad (12) \\ & \phi = R^{r^{LoL}}_{\leq EENS_M}[\mathbf{F}\, abs] \wedge R^{r^{Quality}}_{\geq QoS_m}[\mathbf{F}\, abs] \wedge P_{\geq 1-LoLP_M}[\neg risk\mathbf{U}\, abs] \end{aligned}$$

which can be solved with the algorithm presented in Section 4. In Problem (12), we maximize the sum of the operator profit across the K T_2-slots for the worst-case resolution of uncertainty in the wind-energy forecast. According to the semantics defined in Table 1, the PCTL specification ϕ constraints the expected operator risk and user QoS across the decision horizon. $EENS_M$ is the desired maximum value of Expected Energy Not Served, $LoLP_M$ is the maximum allowed value of Loss of Load Probability (these two properties limit the risk for the operator), and QoS_m is the the minimum value of QoS that needs to be guaranteed to the users.

B Experimental Results

We implemented the algorithm in Python, and interfaced it with PRISM [28] as a front-end for entering models and with Gurobi [24] as the back-end optimizer. Experiments were run on a 2.4 GHz Intel Xeon with 32GB of RAM.

In this section, we present experimental results obtained by solving Problem (10) using the proposed algorithm. Our goals are to give insight about the algorithm functionality, compare its scalability to other strategy-synthesis approaches, and show that the synthesized energy-pricing strategies can achieve better performances than other solutions presented in the literature. We define the following quantities:

$Profit := \mathbb{W}^*_{\mathcal{M}_E,s_0}$	$EENS := R^{r^{LoL},\sigma^*,max}_{s_0}[\mathbf{F}\, abs]$
$QoS := R^{r^{Quality},\sigma^*,min}_{s_0}[\mathbf{F}\, abs]$	$1 - LoLP := P^{\sigma^*,min}_{s_0}[\neg risk\mathbf{U}\, abs]$

where the min and max operators refer to the action range of nature Nat. As defined in Section 2.2, these quantities represent the *quantitative* values of rewards and satisfaction probability that then get compared to the corresponding thresholds ($EENS_M$, QoS_m, $LoLP_M$) in Problem (10) to determine the satisfaction of ϕ. We will then normalize the $Profit$ to $Profit_M$, the maximum computed profit value for each set of experiments. We set $T_1 = 1h$, $T_2 = 30min$ so $K = 2$, and consider two pricing options both for traditional and opportunistic users. If not otherwise stated, we will use $C_L = 90\%$, and discretize the wind energy W in 5 bins, and traditional D_t, opportunistic D_o demands and base-line supply Q in 2 bins. We set $QoS_m = 80\% \sum_k^K \mathbb{E}[D_{t,k} + D_{o,k}]$, $LoLP_M = 10\%$, $EENS_M = 5\% \sum_k^K \mathbb{E}[W_k + q]$. The other parameter values were taken from [9].

In Fig. 8, we show the trend of the expected system performance as a function of the synthesis algorithm iteration. The $Profit$ monotonically decreases until the proposed candidate strategy σ_c meets all specifications. We note that $1 - LoLP$ and QoS ($EENS$, not shown, has a trend similar to $1 - LoLP$) instead vary non-monotonically. Intuitively, this is because the $Profit$ can be increased either by scheduling less base-line energy Q, to reduce cancellation costs c_p but incurring in a higher risk, or by increasing the energy price v for opportunistic users, with consequent reduction of QoS. By ranking strategies by expected $Profit$, our algorithm is capable of selecting the optimal strategy despite the complex parameter interdependences in the model under analysis. In Table 3, we compare synthesis results while varying the number of discretization bins for W (all values are normalized to the corresponding target specification). First, we note that the expected system performances do not substantially vary by changing the number of bins, thus supporting our choice of 5 bins in the other experiments. Second, runtime results show that the algorithm can handle in reasonable time problems of size more than 10× larger than the ones analyzed by [18], the only other algorithm proposed in the literature capable of accepting arbitrary PCTL formulas (we use $N + Tr$, the sum of states and transitions in the EMDP, to represent the model size). This is mainly due to the effective decomposition of the constrained optimization problem into an unconstrained optimization step and a verification step, exploiting decades of improvements in mixed-integer and non-linear optimization engines [24] and in verification algorithms [3,10,28]. Third, results show that the alternative approach of verifying all the I MD strategies $\sigma \in \Sigma^{MD}$ is not practical, due to the exponential increase of I (as defined in Section 2.1) with the problem size. In Fig. 9, we study the effect of different confidence levels C_L in the wind-energy forecast on the expected $Profit$ for the operator, while varying the value of wind penetration $\eta_W = \sum_{k=1}^{K} {}^{\mathbb{E}[W_k]}\!/_{\mathbb{E}[W_k + q]}$ and keeping all constraints constant. At high C_L, the plot shows that higher profits can be expected for increasing η_W (wind energy is assumed free in the model). On the other hand, for low C_L, higher wind penetration creates more uncertainty thus lowering the expected $Profit$. The network operator can use these curves to assess the return of investment in employing more accurate (and expensive) forecast techniques. Finally, in Fig. 10 we compare results with two other energy-pricing formulations proposed in the literature. He *et al.* [9] solve the optimization problem without enforcing any constraint. Varaiya *et al.* [6] only put limits on the acceptable LoLP (their approach is not trivially extendable to properties expressed using the R operator) and solve optimally only for $LoLP = 0$. Comparison is done by solving the different optimization problems and then running Monte Carlo simulations (1000 runs) of the controlled system on test data (different from the training ones) to evaluate its performance ($Profit$, $EENS$ and QoS). As expected, the unconstrained strategy from [9] has higher $Profit$ (up to 5%), but also up to 12% more $EENS$ and 10% less QoS, compared to our approach. The strategy from [6] guarantees null $EENS$, but it has up to 6% less $Profit$ (due to over constraining $EENS$) and 10% less QoS (which is left unconstrained). As a final remark, we note that runtime may increase exponentially as we tighten the specification thresholds ($QoS_m,\ LoLP_M,\ EENS_M$), since it becomes increasingly difficult to find a solution within the exponentially-sized search space. Nevertheless, the chosen values were tight enough to improve the quality of alternative energy-pricing strategies proposed in the literature, while maintaining the runtime acceptable for this application.

Table 3: Performance Analysis

W bins	5	10	15	20
$Profit$	1	0.98	0.97	0.965
$1 - LoLP$	0.99	0.99	0.99	0.99
$EENS$	0.98	0.98	0.98	0.98
QoS	1.01	1.01	1.01	1.01
Runtime	144s	400s	1368s	3289s
#Iter.	223	53	547	332
$N + Tr$	1343	2719	4115	5591
#MD Strat. (I)	4096	4.2e6	4.3e9	4.4e12

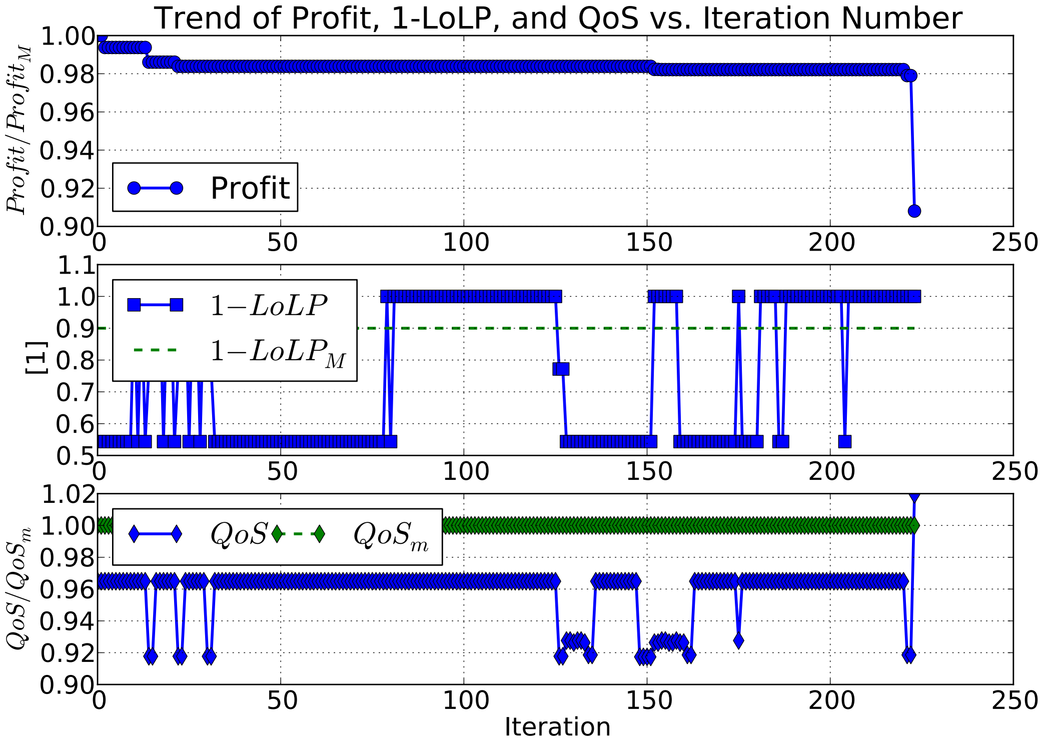

Fig. 8: The figure shows the trend of three quantitative performances of the system as a function of the iteration number. $LoLP_M$, QoS_m represent the numerical thresholds specified in property ϕ. $Profit_M$ is the maximum achievable profit, obtained at the first iteration which is unconstrained.

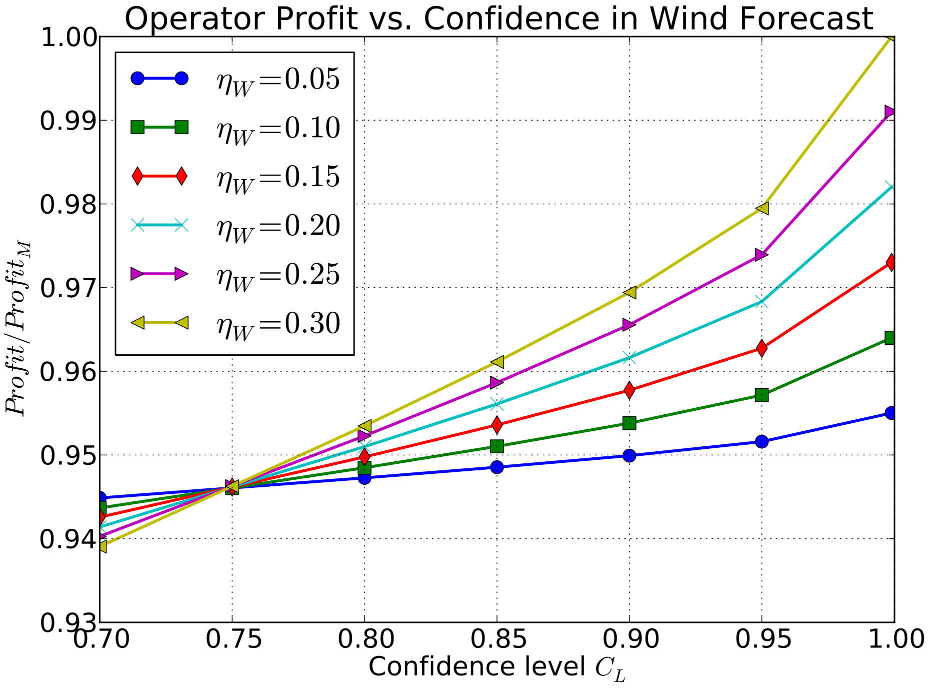

Fig. 9: The picture shows the expected profit as a function of the forecast confidence level C_L, parameterized by the wind penetration η_W. $Profit_M$ is the maximum achievable profit, corresponding to $\eta_W = 0.3, C_L = 1$.

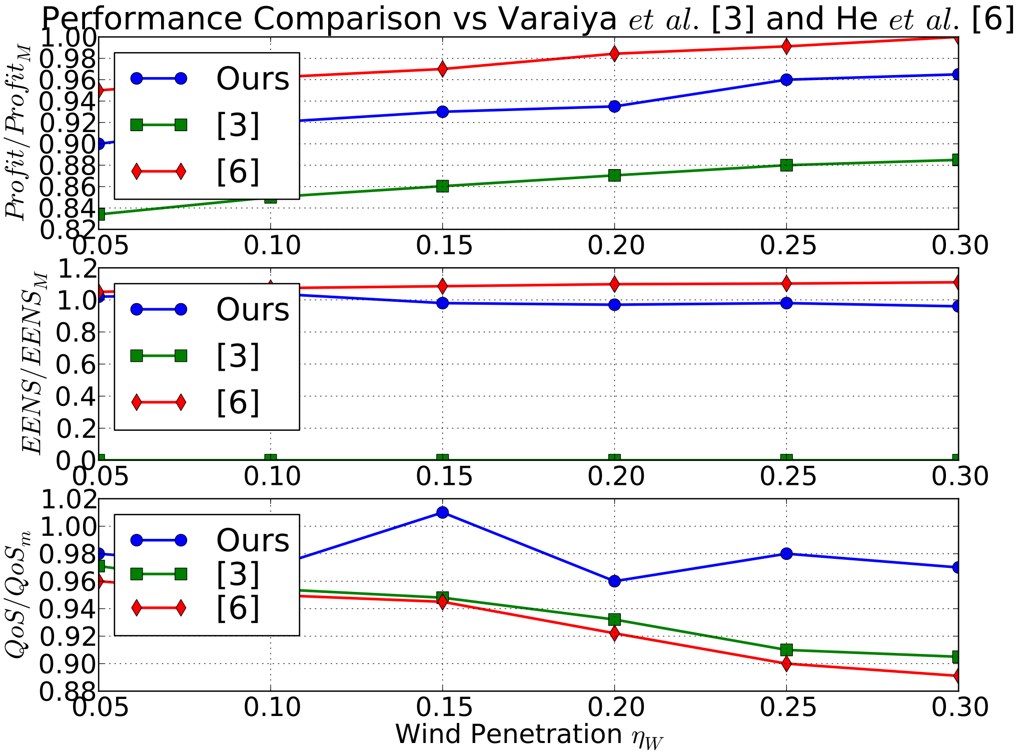

Fig. 10: Comparison to alternative approaches via Monte Carlo simulation. $QoS_m,\ EENS_M$ represent the numerical thresholds specified in property ϕ. $Profit_M$ is the maximum achievable profit, obtained by the unconstrained optimization approach in [9] for maximum wind penetration $\eta_W = 0.3$.