

# 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 Science

University of California, Berkeley

{puggelli, alberto, ssesha}@eecs.berkeley.edu

## ABSTRACT

We address the problem of synthesizing control strategies for Ellipsoidal Markov Decision Processes (EMDP), i.e., MDPs whose transition probabilities are expressed using ellipsoidal uncertainty sets. The synthesized strategy aims to maximize the total expected reward of the EMDP, constrained to a specification expressed in Probabilistic Computation Tree Logic (PCTL). We prove that the EMDP strategy synthesis problem for the fragment of PCTL disabling operators with a finite time bound is NP-complete and propose a novel sound and complete algorithm to solve it.

We apply these results to the problem of synthesizing optimal energy pricing and dispatch strategies in smart grids that integrate renewable sources of energy. We use rewards to maximize the profit of the network operator and a PCTL specification to constrain the risk of power unbalance and guarantee quality-of-service for the users. The EMDP model used to represent the decision-making scenario was trained with measured data and quantitatively captures the uncertainty in the prediction of energy generation. 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) [1]. 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 [4]. 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 state-transition 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 and dispatch.

**Main Motivating Application.** Electricity consumption worldwide is projected to grow from 18 trillion kWh in 2006 to 32 trillion kWh in 2030, a 77% increase [5]. To avoid catastrophic pollution damage to the planet, it is necessary to employ energy sources alter-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ESWEEK'14, October 12 - 17 2014, New Delhi, India  
Copyright 2014 ACM 978-1-4503-3052-7/14/10 \$15.00.  
<http://dx.doi.org/10.1145/2656045.2656069>

native to fossil fuels [6]. In fact, an acceleration in the deployment of renewables is already taking place in Europe, Asia and North America. In this paper, we focus on wind energy, which nowadays 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 [6].

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 [7]. 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 profit (or some other form of *social welfare* cost function) while guaranteeing energy balance also in the presence of one fault in the network, the so-called *N-1* worst-case dispatch [7].

When considering the need for limiting operation risk, it is apparent that a high penetration of wind generation puts forth big operational challenges [7]. 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 [8]).

To compensate for supply uncertainty, researchers have proposed the concept of *demand response*, i.e., the management of customer energy consumption in response to supply conditions. Indeed, a large fraction of the total daily electricity consumption in developed countries is from residential and small commercial energy users, e.g., water heaters and dish washers, which do not need to operate at a specific moment but only within a time interval, e.g., some time overnight [8]. Traditionally, these energy users pay a fixed price per unit of electricity, which represents an average cost of power generation over a given time frame (e.g., a season). In smart grids with two-way communications, *realtime* pricing protocols can be implemented so that price can vary according to the availability of energy supply, to incentivize (or disincentivize) energy demand.

The first *stochastic* optimization frameworks have thus been proposed [9, 10]. These works aim to determine both optimal energy pricing and dispatch of non-renewable baseline energy (wind penetration usually accounts only up to 20-30% of the total energy generation, so fossil generation is still required). On the other hand, these techniques 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 assess the quality of the proposed solution. Unfortunately, these results offer little insight to the operator when the risk is too high. Indeed, Varaiya *et al.* [7] advocated the need for an optimization frame-

work 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 pricing and dispatch strategies for smart grids with wind energy sources (other renewables can be considered analogously). The model is an Ellipsoidal Markov Decision Process  $\mathcal{M}_C$  (EMDP), a special case of the Convex-MDP (CMDP) model first introduced by Puggelli *et al.* [11, 12], i.e., an MDP where transition probabilities are only known to lie in ellipsoidal sets. While previous works used analytical distributions, e.g., Gaussian [10], 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 finite-horizon History-dependent Deterministic (HD) strategies, i.e., for each state an optimal action to take is chosen deterministically, based on the entire (finite) execution history of the decision process. The limitation to finite-horizon strategies is not restrictive, since energy pricing and dispatch decisions are taken on a daily basis. As explained in Section 3, History-dependent Random (HR) strategies are in general more powerful, i.e., they can produce a higher expected reward. Nevertheless, we focus on deterministic strategies because we believe that deterministic pricing strategies are easier to adopt in a real-world scenario, since they can be better understood by the system agents (e.g., household users). Using a classical construction [4], we then unfold within the model the finite sequence of decision epochs over the day, by constructing a second EMDP  $\mathcal{M}'_C$  in which we replicate the states of the original EMDP  $\mathcal{M}_C$  at each decision epoch. In the strategy synthesis for  $\mathcal{M}'_C$ , we focus on Markov (for each state an optimal action is taken based only on the current state) Deterministic (MD) strategies. It can be proven [4], that the desired HD strategy for each state  $s$  of  $\mathcal{M}_C$  can be reconstructed by collecting the sequence of optimal MD actions for each replica  $s'$  of  $s$  along the decision epochs of  $\mathcal{M}'_C$ . While the two formulations produce the same result, we consider the synthesis problem of MD strategies, since this formulation allows us to leverage the results on model-checking of CMDPs described in [11].

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. This result shows that the problem complexity does *not* increase when introducing uncertainties in the state-transition probabilities with respect to the strategy synthesis problem for MDPs, which is also NP-complete [13]. We then develop an algorithm to solve the optimization version of the problem, i.e., maximize the reward. The algorithm is the first sound and complete synthesis algorithm for EMDPs that can process arbitrary formulas (also with multiple and nested quantitative oper-

ators) for the fragment of PCTL that disallows operators with a finite time bound. This technique thus represents an improvement to previously proposed approaches, which were not complete [14] or could only process one PCTL operator per time [15]. Its key advantage is the capability of ranking candidate strategies by the value of their reward. The first proposed strategy that satisfies the PCTL specification 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 (IMDPs) and for the dual problem of cost minimization (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 of research in Section 7.

An extended version of this paper is available at [12].

## 2. PRELIMINARIES

For a given vector  $\mathbf{v}$ , we will use:

$$\mathbf{1}^T \mathbf{v} = \sum_i \mathbf{v}[i] = \sum_i v_i$$

to represent the sum of all the elements of the vector.

**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.** **CMDP.** A labeled finite CMDP,  $\mathcal{M}_C$  is a tuple  $\mathcal{M}_C = (S, S_0, A, \Omega, \mathcal{F}, \mathcal{A}, \mathcal{X}, L)$ , 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 ( $M = |A|$ ),  $\Omega$  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  $\mathcal{M}_C$  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  $s_i \neq s_j$ . We will use this property in the case study presented in Section 5.

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

**Assumption 2.1. Rectangular Uncertainty.**  $\mathcal{F}^a$  can be factored as the Cartesian product of its rows, i.e., its rows are uncorrelated. Formally, for every  $a \in A$ ,  $\mathcal{F}^a = \mathcal{F}_{s_0}^a \times \dots \times \mathcal{F}_{s_{N-1}}^a$ .

This assumption will allow us to consider one state transition at a time and it eases the development of the synthesis algorithm.

**Assumption 2.2. CMDP Semantics.** CMDPs model nondeterministic 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 [16].)

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

The algorithm presented in Section 4 can be applied both to the interval and ellipsoidal models of uncertainty [11] (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 [17]. 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  $g_s^a$ :

$$g_s^a = \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \sum_{s'} h_{ss'}^a \log(f_{ss'}^a) \geq \beta_s^a\} \quad (1)$$

where  $\beta_s^a < \beta_{s,max}^a = \sum_{s'} h_{ss'}^a \log(h_{ss'}^a)$  represents the uncertainty level. The second-order approximation of  $g_s^a$  is [17]:

$$g_s^a \approx \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \sum_{s'} \frac{(f_{ss'}^a - h_{ss'}^a)^2}{h_{ss'}^a} \leq (\mathcal{K}_s^a)^2\} \quad (2)$$

with  $\mathcal{K}_s^a = 2(\beta_{s,max}^a - \beta_s^a) \geq 0$  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:

$$\begin{aligned} \mathcal{F}_s^a &= \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \mathbf{f}_s^a \geq \mathbf{0}, \mathbf{1}^T \mathbf{f}_s^a = 1, \\ &\quad \|E_s^a(\mathbf{f}_s^a - \mathbf{h}_s^a)\|_2 \leq 1, E_s^a \succ 0\} \end{aligned} \quad (3)$$

with  $E_s^a = (\mathcal{K}_s^a)^{-1} \times diag((h_{ss_0}^a)^{-0.5}, \dots, (h_{ss_N}^a)^{-0.5}) \succ 0$  positive definite. Note that the conic representation is crucial to enhance the scalability of the algorithmic approach presented in Section 4, because convex solvers process conic constraints more efficiently than generic quadratic constraints.

**Definition 2.3. Ellipsoidal-MDP (EMDP).** An EMDP is a CMDP where every uncertainty set  $\mathcal{F}_s^a \in \mathcal{F}$  is expressed using Set (3).

We determine the size  $\mathcal{R}$  of the CMDP  $\mathcal{M}_C$  as follows.  $\mathcal{M}_C$  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  $D = \max_{s \in S, a \in A} D_s^a$ .

The overall size of  $\mathcal{M}_C$  is thus  $\mathcal{R} = O(N^2 M + NMD)$ .

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

### 2.1.1 Strategies and Nature

To analyze quantitative properties, we need a probability space over infinite paths [18]. 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  $\mathcal{M}_C$ .

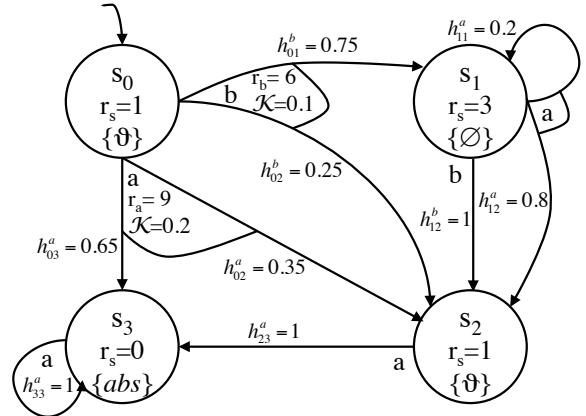


Fig. 1: Example EMDP.

**Definition 2.4. Strategy.** A randomized strategy for  $\mathcal{M}_C$  is a function  $\sigma = \Pi_{fin} \times A \rightarrow [0, 1]$ , with  $\sum_{A(last(\pi))} \sigma(\pi, a) = 1$ , and  $a \in \mathcal{A}(last(\pi))$  if  $\sigma(\pi, a) > 0$ . We call  $\Sigma$  the set of all strategies  $\sigma$  of  $\mathcal{M}_C$ .

Conversely, we call a *nature* [17] each possible resolution of uncertainty, i.e., a nature chooses a transition PD for each state and action of  $\mathcal{M}_C$ .

**Definition 2.5. Nature.** Given action  $a \in A$ , a randomized nature is the function  $\eta^a : \Pi_{fin} \times \text{Dist}(S) \rightarrow [0, 1]$  with  $\int_{\mathcal{F}_{last(\pi)}^a} \eta^a(\pi, \mathbf{f}_s^a) = 1$ , and  $\mathbf{f}_s^a \in \mathcal{F}_{last(\pi)}^a$  if  $\eta^a(\pi, \mathbf{f}_s^a) > 0$ . We call  $\text{Nat}$  the set of all natures  $\eta^a$  of  $\mathcal{M}_C$ .

A strategy  $\sigma$  (nature  $\eta^a$ ) is Markovian (M) if it depends only on  $last(\pi)$ . Also,  $\sigma(\eta^a)$  is deterministic (D) if  $\sigma(\pi, a) = 1$  for some  $a \in \mathcal{A}(last(\pi))$  ( $\eta^a(\pi, f_s^a) = 1$  for some  $f_s^a \in \mathcal{F}_{last(\pi)}^a$ ). As explained in Section 1, we developed a synthesis algorithm for MD strategies. There are in total  $I = |\mathcal{A}_{s_0}| \times \dots \times |\mathcal{A}_{s_N}| = O(M^N)$  MD strategies, collected in the set  $\Sigma^{MD}$ .

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

### 2.1.2 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 2.6. Reward Structure and Path Reward.** A reward structure for a CMDP  $\mathcal{M}_C$  is a tuple  $r = (r_s, r_a)$  comprising a state (action) reward function  $r_s : S \rightarrow \mathbb{R}_{\geq 0}$  ( $r_a : S \times A \rightarrow \mathbb{R}_{\geq 0}$ ). Given a (possibly infinite) path  $\pi$  with horizon  $T \in \mathbb{N} \cup +\infty$ , the path reward for  $\pi$  is  $\text{rew}_r(\pi, T) = \sum_{t=0}^T r_s(\pi_s[t]) + r_a(\pi_s[t], \pi_a[t])$ .

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

**Definition 2.7. Total Expected Reward.** The total expected reward for state  $s \in S$  under strategy  $\sigma \in \Sigma^{MD}$  is defined as:

$$\mathbb{W}_s^\sigma := \min_{\eta^a \in \text{Nat}} \mathbb{E}^{\sigma, \eta^a} [\text{rew}_r(\pi, T)] \quad (4)$$

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

We will only consider CMDPs such that  $\mathbb{W}_s^\sigma$  exists and it is finite  $\forall s \in S, \forall \sigma \in \Sigma^{MD}$ . These include finite and infinite-horizon CMDPs ( $T \in \mathbb{N} \cup \{+\infty\}$ ) with zero-reward absorbing states ( $\omega = \text{abs}$ ), as the ones used in Section 5. An extension to other classes of CMDPs, e.g., with negative rewards, is possible in some specific cases, but outside the scope of the paper. For more details, see [4]. Finally, according to Assumption 2.2, we can substitute  $\eta^a \in Nat$  with  $\mathbf{f}_s^a \in \mathcal{F}_s^a$ , and only consider MD natures.

## 2.2 Probabilistic Computation Tree Logic

We use Probabilistic Computation Tree Logic (PCTL), a probabilistic logic derived from CTL which includes a probabilistic operator  $P$  [2] and a reward operator  $R$  [19], to express properties of CMDPs [3]. The syntax of the fragment of the logic in which operators with a finite time bound are disallowed is defined as follows:

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

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

Path formulas use the *Next* ( $\mathbf{X}$ ) and *Unbounded Until* ( $\mathbf{U}$ ) operators. They are evaluated over paths and only allowed as parameters to the  $P$  operator. Reward formulas use the *Cumulative* ( $\mathbf{C}$ ) operator. The size  $Q$  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  $P_s^{\sigma, \eta^a}[\psi] \stackrel{\text{def}}{=} \text{Prob}(\{\pi \in \Pi_s^{\sigma, \eta^a} \mid \pi \models \psi\})$  the probability of taking a path  $\pi \in \Pi_s$  that satisfies  $\psi$  under strategy  $\sigma$  and nature  $\eta^a$ .  $P_s^{\sigma, \max}[\psi]$  ( $P_s^{\sigma, \min}[\psi]$ ) denotes the maximum (minimum) probability  $P_s^{\sigma, \eta^a}[\psi]$  across all natures  $\eta^a \in Nat$ , for a fixed strategy  $\sigma$ . An analogous definition holds also for reward properties, which can be expressed also using (multiple) reward structures different from the one used to maximize the total expected reward. For a CMDP  $\mathcal{M}_C$ , strategy  $\sigma$ , and property  $\phi$ , we will use  $\mathcal{M}_C, \sigma \models_{Nat} \phi$  to denote that, when starting from any initial state  $s \in S_0$ , and operating under  $\sigma$ ,  $\mathcal{M}_C$  satisfies  $\phi$  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 brevity.

## 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 [20] 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 [9], and a stochastic framework to optimize operator profits was presented in [10]. 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 constrain the operator risk in [7]. 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 [13]. 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.* [21] show how to synthesize MR controllers that are robust to linear perturbations via a reduction to a formula in the first-order logic of the 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 [14] and [15] routines for the model-checking 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 (i.e., there might exist a solution even if none is reported) [14] or can handle properties with only one quantitative operator [15]. In [22], the synthesis of multi-strategies for MDPs is studied. This approach can handle only a subset of PCTL properties and it only considers MDPs with no uncertainties. Finally, [23] studies two-player games with winning objectives expressed in PCTL. Also our formulation can be interpreted as a game, where the controller plays against nature. On the other hand, following the CMDP semantics defined in Assumption 2.2, we give nature the power of selecting a different strategy at each execution step, while [23] aims to analyze the more complex problem of finding the single optimal strategy for nature. Our formulation is useful to model time-varying processes (e.g., the power generated by the wind), and an optimal strategy for the controller can be synthesized algorithmically. The existence of an optimal controller and nature in the formulation of [23] is instead undecidable in general.

## 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  $\mathcal{M}_C$ , a reward structure  $r$ , and a PCTL formula  $\phi$ , determine strategy  $\sigma^*$  for  $\mathcal{M}_C$  such that:

$$\sigma^* = \underset{\sigma \in \Sigma_\phi^{MD}}{\operatorname{argmax}} \mathbb{W}_{S_0}^\sigma \quad (5)$$

where  $\mathbb{W}_{S_0}^\sigma$  is the sum of the expected rewards over all the initial states  $s \in S_0$ , and  $\Sigma_\phi^{MD}$  is the set of Markov-Deterministic strategies for which  $\mathcal{M}_C$  satisfies  $\phi$  for any  $\eta^a \in Nat$ , starting from any  $s \in S_0$  and operating under  $\sigma \in \Sigma_\phi^{MD}$ .

The same results hold for the dual problem of minimizing the EMDP total expected cost, after substituting all “max” (“min”) operators with “min” (“max”) in the derivations below.

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

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

$s \models$	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 $s \models \phi_1 \wedge s \models \phi_2$
$s \models P_{\bowtie p}(\triangleright p)[\psi]$		iff $P_s^{\sigma, \max(\min)}[\psi] \bowtie p(\triangleright p)$
$s \models R_{\bowtie v}^r[\mathbf{C} \phi]$		iff $\mathbb{E}_s^{\sigma, \max(\min)}[rew_r(\pi, t_\phi)] \bowtie v(\triangleright v)$
$\pi \models \mathbf{X} \phi$		iff $\pi[1] \models \phi$
$\pi \models \phi_1 \mathbf{U} \phi_2$		iff $\exists i \geq 0 \mid \pi[i] \models \phi_2 \wedge \forall j < i \pi[j] \models \phi_1$
$rew_r(\pi, t_\phi) := \sum_{t=0}^{t_\phi} r_s(\pi_s[t]) + r_a(\pi_s[t], \pi_a[t]),$		
$t_\phi = \min\{t \mid \pi_s[t] \models \phi\}$		

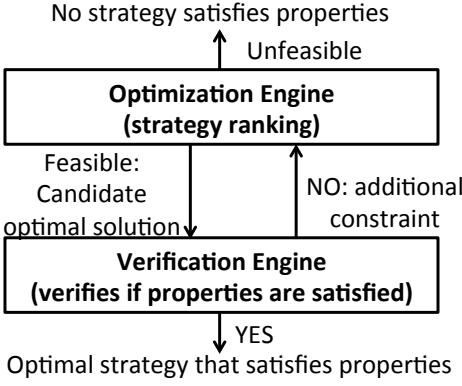


Fig. 2: *Lazy* algorithm for the constrained optimization.

**Lemma 4.1. Complexity of PCTL model-checking for CMDPs [11].** *Verifying that a CMDP  $\mathcal{M}_c$  satisfies a PCTL formula  $\phi$  is solvable in polynomial time.*

**Lemma 4.2. Computation of the total expected reward for Convex Markov Chains (CMCs) [17].** *Given a CMC, its total expected reward is computable in polynomial time.*

**Lemma 4.3. Complexity of PCTL strategy synthesis for MDPs [13].** *The problem of determining the existence of an MD strategy  $\sigma$  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 (5) is NP-complete.

**Theorem 4.1.** *The problem of determining the existence of an MD strategy  $\sigma$  for an EMDP  $\mathcal{M}_c$ , with total expected reward  $\mathbb{W}_{S_0}^{\sigma_c}$  larger or equal to  $\mathbb{W}_T$  and satisfying a specification  $\phi$  in PCTL, is NP-complete.*

*Proof.* Given a candidate solution  $\sigma_c$ , we can in polynomial time: 1) check whether  $\mathcal{M}_c, \sigma_c \models_{Nat} \phi$  by Lemma 4.1; 2) compute  $\mathbb{W}_{S_0}^{\sigma_c}$  on the induced CMC  $\mathcal{M}_c^{\sigma_c}$  by Lemma 4.2.  $\sigma_c$  is a solution if and only if check 1) passes and  $\mathbb{W}_{S_0}^{\sigma_c} \geq \mathbb{W}_T$ . 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 ( $\mathcal{K}_s^a \rightarrow +\infty, \forall s \in S, \forall a \in \mathcal{A}(s)$ ) in Equation (2).  $\square$

In the rest of the section, we describe an algorithm to solve Problem (5). We use a *lazy* approach based on strategy-iteration, conceptually similar to the ones proposed in [24] and, for non-linear constraints, [25]. As shown in Fig. 2, the algorithm is split into two main routines communicating in a loop. At each iteration, the *optimization engine* (OE) is responsible to generate a candidate strategy  $\sigma_c$ .  $\sigma_c$  maximizes the total expected reward, but it does not necessarily satisfy PCTL property  $\phi$ , since the OE formulation does not contain any information about  $\phi$ . The candidate solution is then passed to the *verification engine* (VE) which checks whether  $\mathcal{M}_c$  satisfies  $\phi$  for all resolutions of uncertainty, i.e.,  $\forall \eta^a \in Nat$ , when operating under  $\sigma_c$ . 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  $\sigma_c$  to be selected again, and the loop repeats. The novelty of our approach is devising a mathematical formulation for the OE capable of generating at each iteration the candidate strategy that maximizes the total expected reward among those still available (at each iteration in which the VE reports a failure, the candidate strategy is discarded). The first candidate strategy that satisfies the PCTL specification becomes the

solution of the synthesis problem. Such a strategy is feasible (satisfies  $\phi$ ) and it has the maximum total reward among strategies that satisfy  $\phi$ , so it solves Problem (5) exactly.

In general, there is not guarantee on the uniqueness of the optimal strategy  $\sigma^*$ , i.e., multiple strategies with the same expected reward might exist, each satisfying  $\phi$ . Since every such strategy is equivalent from the user perspective, the algorithm just reports the first one found. Alternatively, all equivalent strategies could be generated by continuing the iteration between the OE and the VE until finally a reduction in the expected reward gets detected, and by reporting all synthesized strategies satisfying  $\phi$ .

The next subsections give details on the OE and VE and analyze the algorithm properties.

## 4.1 Optimization Engine

We start with the classical linear-programming (LP) formulation to maximize the total expected reward for MDPs [4].

$$\begin{aligned} & \min_{x, l} \sum_{s \in S_0} x_s \\ & \text{s.t. } x_s - l_s^a = r_s^a + \mathbf{x}^T \mathbf{f}_s^a; \quad \forall s \in S, \forall a \in \mathcal{A}(s) \\ & \quad x_s, l_s^a \geq 0 \quad \forall s \in S, \forall a \in \mathcal{A}(s) \end{aligned} \quad (6)$$

Vector  $\mathbf{x}$  collects the total expected reward for each state  $s \in S$  (at the end of the optimization  $\mathbb{W}_s^{\sigma_c} = x_s, \forall s \in S$ ), and the cost function sums the total expected rewards for all the initial states  $s \in S_0$ . We then set  $\mathbb{W}_{S_0}^{\sigma_c} = \sum_{s \in S_0} x_s$ . 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 to modify such a formulation to allow a sub-optimal solution to be selected, in case the optimal solution does not satisfy the PCTL specification. 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 select sub-optimal solutions in order of optimality. We refer to Problem (7):

$$\begin{aligned} & \max_{x, z, l, n} \sum_{s \in S_0} x_s \\ & \text{s.t. } x_s - l_s^a + n_s^a = r_s^a + \mathbf{x}^T \mathbf{f}_s^a; \quad \forall s \in S, \forall a \in \mathcal{A}(s) \quad (7a) \\ & \quad l_s^a \leq B z_s^a, \quad n_s^a \leq B z_s^a; \quad \forall s \in S, \forall a \in \mathcal{A}(s) \quad (7b) \\ & \quad \mathbf{z}_s^T \mathbf{1} = M_s - 1; \quad \forall s \in S \quad (7c) \\ & \quad x_s, l_s^a, n_s^a \geq 0, \quad z_s^a \in \{0, 1\} \quad \forall s \in S, \forall a \in \mathcal{A}(s) \end{aligned}$$

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 (7c) guarantees that only one action can be selected for each state, where  $M_s = |\mathcal{A}(s)|$ . 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,  $z_{s_i}^{a_j} = 0$ , Constraint (7b) 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 (7a) sets the value of  $x_s$ .<sup>1</sup> 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

<sup>1</sup>  $B$  is a *big* number with respect to the problem data. It needs to be higher than the reward computed for any state, i.e.,  $B \geq x_s, \forall s \in S$ , but not excessively high to avoid convergence problems in the optimizer when solving Problem (7). A suitable value of  $B$  can be found by trial and error.

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 (7a) gets updated to Constraint (8), 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. We solve this difficulty for the ellipsoidal uncertainty model using duality. In Constraint (9), we replace the primal inner problem with its dual,  $\forall s \in S, a \in \mathcal{A}(s)$ :

$$x_s - l_s^a + n_s^a = r_s^a + \min_{\mathbf{f}_s^a \in \mathcal{F}_s^a} \mathbf{x}^T \mathbf{f}_s^a \quad (8)$$

↓

$$x_s - l_s^a + n_s^a = r_s^a + \max_{\lambda_s^a \in \mathcal{D}_s^a(\mathbf{x})} g(\lambda_s^a) \quad (9)$$

Vector Lagrange multiplier $\lambda_s^a = [\lambda_{1,s}^a, \lambda_{2,s}^a, \lambda_{3,s}^a]$
Dual cost function $g(\lambda_s^a) = \lambda_{1,s}^a - \lambda_{2,s}^a - \mathbf{h}_s^a E_s^a \lambda_{3,s}^a$
Dual feasibility set $\mathcal{D}_s^a(\mathbf{x}) = \{\lambda_s^a \mid \ \lambda_{3,s}^a\ _2 \leq \lambda_{2,s}^a, \mathbf{x} - \lambda_{1,s}^a \mathbf{1} + E_s^{aT} \lambda_{3,s}^a = \mathbf{0}\}$

The dual problem is convex by construction [26] and has size polynomial in the size  $\mathcal{R}$  of the EMDP [11]. 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 [26] 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 (9), 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 (7)):

$$\begin{aligned} & \max_{x, \lambda, l, n, z} \sum_{s \in S_0} x_s \\ \text{s.t. } & x_s - l_s^a + n_s^a = r_s^a + g(\lambda_s^a); \\ & l_s^a \leq Bz_s^a, \quad n_s^a \leq Bz_s^a; \\ & \mathbf{z}_s^T \mathbf{1} = M_s - 1; \\ & x_s, l_s^a, n_s^a \geq 0, \lambda_s^a \in \mathcal{D}_s^a(\mathbf{x}), z_s^a \in \{0, 1\} \end{aligned} \quad (10)$$

For the ellipsoidal model, Problem (10) can be written as Problem (11), which is a Mixed-Integer Quadratic-Constrained Program (MIQCP) that we solve using the back-end optimizer Gurobi [27].

$$\begin{aligned} & \max_{x, \lambda, z, l, n} \sum_{s \in S_0} x_s \\ \text{s.t. } & x_s - l_s^a + n_s^a = r_s^a + \lambda_{1,s}^a - \lambda_{2,s}^a - \mathbf{h}_s^a E_s^a \lambda_{3,s}^a; \\ & l_s^a \leq Bz_s^a, \quad n_s^a \leq Bz_s^a; \\ & \mathbf{z}_s^T \mathbf{1} = M_s - 1; \\ & x_s, l_s^a, n_s^a, \lambda_{2,s}^a \geq 0, \lambda_{3,s}^a \geq 0, z_s^a \in \{0, 1\}; \\ & \|\lambda_{3,s}^a\|_2 \leq \lambda_{2,s}^a, \mathbf{x} - \lambda_{1,s}^a \mathbf{1} + E_s^{aT} \lambda_{3,s}^a = \mathbf{0} \end{aligned} \quad (11)$$

## 4.2 Verification Engine

After fixing a candidate strategy  $\sigma_c$ , all the non-determinism in  $\mathcal{M}_C$  is resolved. The VE has thus the task of checking whether the *induced* Ellipsoidal-MC (EMC)  $\mathcal{M}_C^{\sigma_c} = (S, S_0, \Omega, \mathcal{F}, \mathcal{X}, L)$  satisfies the PCTL formula  $\phi$  for all resolutions of uncertainty, i.e.,  $\forall \eta^a \in Nat$ . We use the sound and complete verification algorithm presented in [11, 12] (we remember that EMCs are a special

case of CMDPs). If the EMC satisfies  $\phi$ , 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  $\mathbf{z}_c = [z_{s_0}^a \dots z_{s_N}^a]$  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:

$$\mathbf{z}_c^T \mathbf{1} \geq 1 \quad (12)$$

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  $\sigma_{c1}$ , which selects actions  $[b, a, a, a]$  for states  $[s_0 \dots s_3]$ , with  $\mathbb{W}_{s_0}^{\sigma_{c1}} = 10.625$ . The VE, though, reports  $P_{s_0}^{\sigma_{c1}, min}[\vartheta \mathbf{U} abs] = 0.207$ , so the strategy is rejected. The VE adds the constraint  $z_{s_0}^b + z_{s_1}^a + z_{s_2}^a + z_{s_3}^a = 1$  to the OE formulation, which generates at the second generation  $\sigma_{c2}([a, a, a, a])$ , with  $\mathbb{W}_{s_0}^{\sigma_{c2}} = 10.188$ . The VE computes the value  $P_{s_0}^{\sigma_{c2}, min}[\vartheta \mathbf{U} abs] = 1$ , and the algorithm terminates reporting  $\sigma^* = \sigma_{c2}$ .

## 4.3 Algorithm Analysis

We prove *soundness* (if a strategy  $\sigma^*$  is returned, it indeed optimally solves Problem (5)), *completeness* (if no solution is returned, no strategy  $\sigma \in \Sigma^{MD}$  exists such that  $\mathcal{M}_C, \sigma \models_{Nat} \phi$ ) and analyze the runtime performance of the proposed algorithm.

**Theorem 4.2.** *The algorithm presented in this section to solve Problem (5) 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  $\phi$ .*

*Proof.* Problem (10) returns the MD strategy  $\sigma_c$  that maximizes  $\mathbb{W}_{S_0}^{\sigma_c}$  among those still available. By Lemma 4.1, the VE is sound, so if it returns  $\mathcal{M}_C, \sigma_c \models_{Nat} \phi$ , indeed  $\sigma^* = \sigma_c$  (exit arrow at the bottom of Fig. 2). The VE is also complete, so if it returns  $\mathcal{M}_C, \sigma_c \not\models_{Nat} \phi$ , the current  $\sigma_c$  can be discarded. This is done by generating a constraint of the form of Constraint (12), which removes only the current  $\sigma_c$  from the strategies to be explored by the OE. This proves soundness of the overall algorithm.

Failure of finding a solution is declared only by the OE, when Problem (10) becomes unfeasible because all available strategies have previously been discarded by the VE (exit arrow at the top of Figure 2). This proves completeness.

Finally, the algorithm goes at most through  $I = O(M^N)$  iterations (from Section 2,  $I$  is the total number of MD strategies,  $M$  is the number of actions, and  $N$  the number of states of the EMDP). Each requires solving an instance of Problem (10) and a verification check (done in time polynomial in  $\mathcal{R}$  and  $\mathcal{Q}$  by Lemma 4.1). Problem (10) can be solved by branch-and-bound algorithms in time exponential in the number of binary variables (whose number is  $MN$  and remains constant across iterations) 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  $O(M^N \times (2^{MN} \times poly(M^N) + poly(\mathcal{Q})))$ , exponential in  $\mathcal{R}$  and polynomial in  $\mathcal{Q}$ .  $\square$

The algorithm performs better on problems that 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,  $\sigma^*$  could be determined by testing all  $I$  available MD strategies, and selecting the one with the highest reward among those satisfying  $\phi$  [13]. 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. Fi-

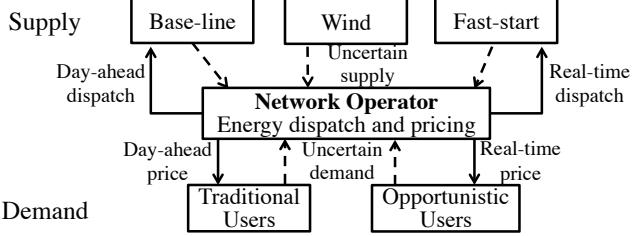


Fig. 3: Network operator inputs (dashed) and outputs (solid).

nally, speed-ups can be obtained by implementing online routines for integer-constraint simplification and to produce more succinct certificates of unfeasibility from the VE [24, 25]. 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. 3 [7, 10, 28]. More details are available in [12].

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, baseline and fast-start generators. We will use stochastic variables  $W, D_t, D_o$  to represent wind energy supply and traditional and opportunistic user demand, respectively.

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.,  $T_1 = 1\text{h}$ ), and each  $T_1$ -slot into  $K$   $T_2$ -slots (e.g.,  $T_2 = 30\text{min}$ ,  $K = 2$ ) [10]. The operator maximizes its economic profit by taking decisions on the two time scales. On day-ahead, for each  $T_1$ -slot, it dispatches  $Q$  units of baseline 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 one day ahead when to schedule their demand in the following day. The choice of  $u$  sets the expected demand of traditional users ( $\mathbb{E}[D_t]$ ). In real-time, for each  $T_2$ -slot, the operator first *observes* the values of traditional-user demand  $d_t$  and wind availability  $w$ . It then sets the price for opportunistic users ( $v$ ) to determine the expected demand of opportunistic users ( $\mathbb{E}[D_o]$ ). Third, it dispatches the production of more fast-start energy (unit cost,  $c_2$ ) or cancels part of the already dispatched baseline energy (paying only unit cost  $c_p$  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 [10]. The operator thus tries to use as much wind energy as possible and to dispatch on day-ahead the exact amount of baseline 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: (i) wind-energy supply ( $W$ ), (ii) traditional ( $D_t$ ) user demand, (iii) opportunistic ( $D_o$ ) user demand. We thus use a stochastic optimization framework. The result of the optimization should return optimal strate-

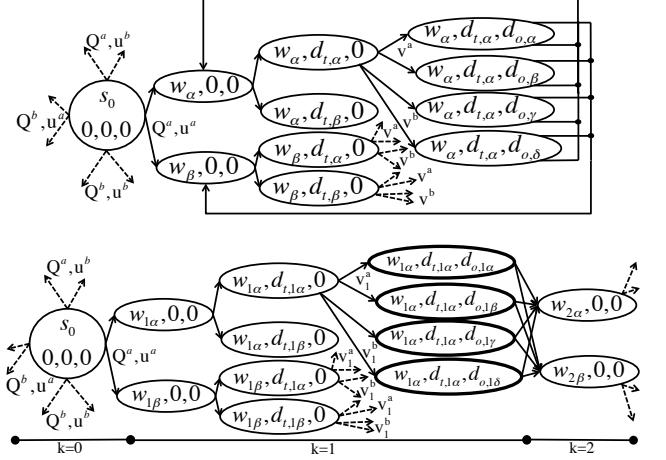


Fig. 4: Sketch of  $\mathcal{M}_E$  (top) and  $\mathcal{M}'_E$  (bottom). The latter is  $\mathcal{M}_E$  unfolded across decision epochs. For each EMDP, the initial state  $s_0$  is shown on the left. Each state is represented by the tuple  $(w, d_t, d_o)$  of observations of  $W, D_t, D_o$ . The pairs  $(Q, u)$  represent the day-ahead decisions about dispatch of baseline energy  $Q$  and energy pricing for traditional users  $u$ . Two arrows per decision depart from each state because in this figure we assumed only two *discretization levels* for each quantity (labeled with greek letters  $\alpha, \beta, \dots$ ). We only show the state graph in the EMDPs related to decision  $(Q^a, u^a)$ , but similar state graphs are present also for the other decision pairs  $(Q^a, u^b), (Q^b, u^a), (Q^b, u^b)$ , as hinted by the dashed arrows departing from  $s_0$ .

gies about 1) the day-ahead decisions ( $Q$  and  $u$ ) and 2) the real-time decisions ( $v$ ) for *each possible observation* of  $W$  and  $D_t$ . In real-time, the actual decision ( $v_{obs}$ ) will be taken deterministically among the synthesized ones based on the *observed* values  $w$  and  $d_t$ , i.e., the actual wind availability and traditional-user demand. In summary, 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  $v$ .

We use the Ellipsoidal-MDP  $\mathcal{M}_E = (S, S_0, A, \Omega, \mathcal{F}, \mathcal{A}, \mathcal{X}, L)$  sketched in Fig. 4 (top). All quantities are bounded and uniformly discretized to keep the state and action spaces finite. States  $s \in S$  are a tuple  $s = (w, d_t, d_o)$ , where  $w, d_t, d_o$  refer to the observed values of available wind energy and user demand in that state. We consider only one  $T_1$ -slot per time, so we model only one choice of day-ahead energy dispatch  $Q$  and pricing for traditional users  $u$  (at the initial state  $s_0$ ,  $(Q, u) \in A$ ). The process then transitions through  $K$  decision epochs, as follows. First, values of wind energy ( $w$ ) and traditional user demand ( $d_t$ ) are stochastically chosen according to the corresponding distributions (described below). Second, for each observation of  $W$  and  $D_t$ , a decision on  $v \in A$  is made and the opportunistic user demand ( $d_o$ ) is stochastically chosen. To transition between epochs a new value of wind energy  $w$  (the backward arrow in Fig. 4 (top)) is chosen and the steps repeat.

In general, the optimal decision  $v$  at each epoch depends on previous observations of wind availability ( $w$ ) and user demand ( $d_t, d_o$ ), so  $v$  is history-dependent. To synthesize control strategies using the algorithm in Section 4, we unfold the sequence of decision epochs in the EMDP  $\mathcal{M}'_E = (S', S_0, A', \Omega, \mathcal{F}', \mathcal{A}', \mathcal{X}', L')$  in Fig. 4 (bottom). In  $\mathcal{M}'_E$ , we have explicitly marked each quantity with an additional subscript  $k = 1, \dots, K$  to refer to the corresponding  $T_2$ -slot. Each state  $s \in S$  of  $\mathcal{M}_E$  (apart from the initial state) has been replicated  $K$  times in  $\mathcal{M}'_E$ ,  $s \rightarrow s_1, s_2, \dots, s_K$ . After  $K$  decision epochs, the states transition to an absorbing state (not shown in Fig. 4). Since now all decision epochs are explicitly

codified in the model, Markov strategies are optimal for  $\mathcal{M}'_E$ . The corresponding optimal history-dependent strategy for  $\mathcal{M}_E$  can be reconstructed for each  $s \in S$  by collecting the sequence of optimal decisions returned by the algorithm in the replicas  $s_1, s_2, \dots, s_K$ . In the following we will only consider  $\mathcal{M}'_E$ . As an additional advantage, different wind distributions can be used to transition between different epochs in  $\mathcal{M}'_E$  (only one distribution could be used in  $\mathcal{M}_E$ ), to account for the time-varying nature of wind availability. The modeling expressivity is thus increased.

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 [10], 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  $\gamma_t < 0$  ( $\gamma_o < 0$ ) 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 defined as:  $\gamma_t = u/\mathbb{E}[D_t] \cdot \partial \mathbb{E}[D_t]/\partial u$ . 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.<sup>2</sup>

*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 [29]. 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  $\mu_W$  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  $\mu_W$  to have such expected value  $\mathbb{E}[W_k]$ , thus obtaining  $\mu_{W_k}$ . Finally, we compute ellipsoidal Sets (3)  $\mathcal{F}'^{fa}_s$  (collected in  $\mathcal{F}'$ ) 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. Using classical results from statistics [17], we then compute the value of parameter  $\beta$  from Set (1) corresponding to the confidence level  $C_L$  in the measurements. In particular,  $0 \leq C_L \leq 1$  and  $C_L = 1 - cdf_{\chi_d^2}(2 * (\beta_{max} - \beta))$ , where  $cdf_{\chi_d^2}$  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  $W$ ). Moreover, an increasing value of parameter  $\beta$  can be used in the sequence of decision epochs to model the fact that forecast farther-away in time are less accurate.

We provide the states with thick circles in Fig. 4 (bottom) 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:

$$r_{s,k}^{Profit}[\$] = ud_{t,k} + vd_{o,k} - (c_p \Delta_k + c_1(q - \Delta_k)) \mathbf{1}_{\Delta_k \geq 0} - (c_1 q - c_2 \Delta_k) \mathbf{1}_{\Delta_k < 0} \quad (13a)$$

$$r_{s,k}^{LoL}[\text{MWh}] = \max(0, X\mathbb{E}[W_k] + Y\mathbb{E}[D_{t,k} + D_{o,k}] - \Delta_k) \quad (13b)$$

$$r_{s,k}^{Quality}[\text{MWh}] = d_{t,k} + d_{o,k} \quad (13c)$$

with  $\Delta_k = w_k + q - d_{t,k} - d_{o,k}$  representing the surplus of supply on demand,  $X$  and  $Y$  defined in the following, and  $\mathbf{1}$  the indicator function. Reward (13a) subtracts operating costs to the

<sup>2</sup>For simplicity, we assume that the distributions of  $D_t$  and  $D_o$  are known with certainty. In fact, these distributions are usually estimated more accurately than the wind forecast, by analyzing the history of measured data [10]. Adding uncertainty models also for these distributions is left as future work.

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 (13b) 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.  $X = 3\%$ ,  $Y = 10\%$ ) [30]. If  $\Delta + FS < 0$  the network incurs in a LoL, with potentially risky consequences. Reward (13c) 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  $\Omega = \{\textit{risk}, \textit{abs}\}$ .

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

$$\begin{aligned} \mathbb{W}_{s_0}^* &= \max_{Q,u} \min_{\mathbf{f}_s^a \in \mathcal{F}'^{fa}} \mathbb{E}_W^{\sigma, \mathbf{f}_s^a} \max_{v_1 \dots v_K} \mathbb{E}_{D_o}^\sigma r_{s,k}^{Profit}(\pi, K) \\ &\text{s.t. } \mathcal{M}'_E, \sigma^* \models_{Nat} \phi \quad \text{where:} \\ \phi &= R_{\leq EENS_M}^{LoL}[\mathbf{C} \textit{ abs}] \wedge R_{\geq QoS_m}^{Quality}[\mathbf{C} \textit{ abs}] \\ &\quad \wedge P_{\geq 1 - LoLP_M}[\neg \textit{risk} \mathbf{U} \textit{ abs}] \end{aligned} \quad (14)$$

In Problem (14), we maximize the expected value of the operator profit  $\mathbb{W}_{s_0}^*$  under the worst-case resolution of uncertainty in the wind-energy forecast by summing the instantaneous state rewards  $r_s^{Profit}$  along the paths  $\pi \in \Pi_{fin}$  of  $K$  steps of  $\mathcal{M}'_E$ , corresponding to the  $K T_2$ -slots. By replicating the decision process for each of the  $K T_2$ -slots while building model  $\mathcal{M}'_E$ , every  $K$ -step execution path traverses all decision epochs, so the computed reward, as introduced in Definition 2.6 and 2.7, will account for the sum of the contributions of each decision epoch.

According to the semantics defined in Table 1, the PCTL specification  $\phi$  constrains 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 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 [31] as a front-end for entering models and with Gurobi [27] 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 (14). 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 and dispatch strategies can achieve better performances than other solutions presented in the literature.

We define the following quantities:

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

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 get then compared to the corresponding thresholds ( $EENS_M$ ,  $QoS_m$ ,  $LoLP_M$ ) in Problem (14) to determine the satisfaction of  $\phi$ . We will then normalize the *Profit* to  $Profit_M$ , the maximum computed profit value for each set of experiments. We set  $T_1 = 1\text{h}$ ,  $T_2 = 30\text{min}$  so  $K = 2$ , and consider two pricing options both for traditional and opportunistic users. If not otherwise stated,

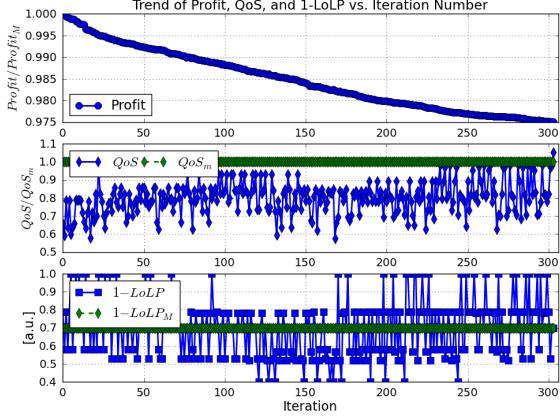


Fig. 5: Performance vs. iteration.

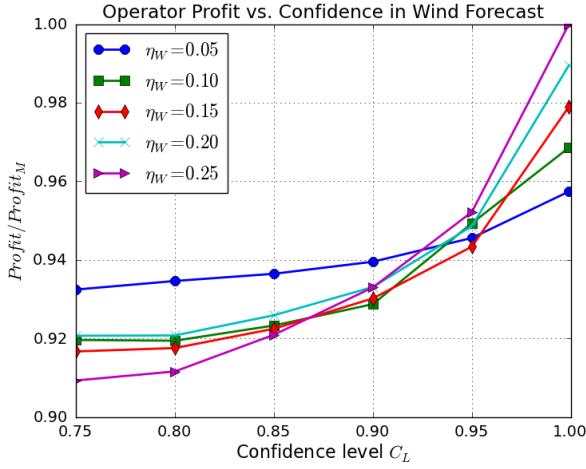


Fig. 6: Profit vs. confidence in forecast.

we will use  $C_L = 90\%$ , and discretize the wind energy  $W$  in 5 bins, and traditional  $D_t$ , opportunistic  $D_o$  demands and baseline 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 [10].

In Fig. 5, 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  $\sigma_c$  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 baseline energy  $Q$ , to reduce cancellation costs  $c_p$  but incurring in a higher risk, or by increasing the

Table 2: Performance Analysis

$W$ bins	5	10	15	20
<i>Profit</i>	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

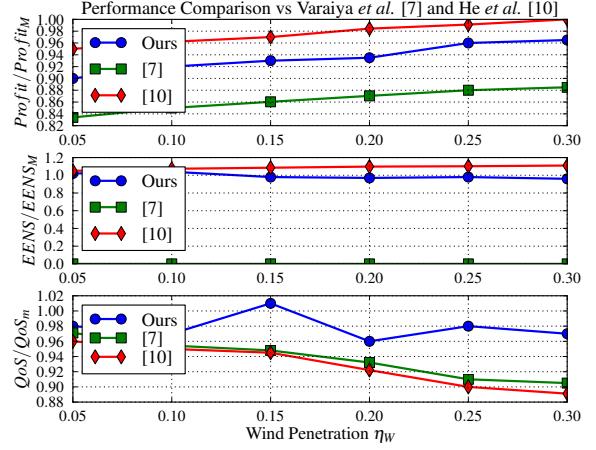


Fig. 7: Comparison to alternative approaches via Monte Carlo simulation.

energy price  $v$  for opportunistic users, with consequent reduction of  $QoS$  (the peaks of  $1 - LoLP$  are aligned to the valleys 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 specification). First, we note that the expected system performances do not substantially vary by changing the number of bins, supporting our choice of 5 bins in the other experiments. Second, runtime results show that the algorithm can handle in reasonable time problems more than  $10\times$  larger than the ones analyzed in [14], the only other algorithm in the literature capable of processing arbitrary PCTL formulas (we use  $N + Tr$ , the sum of states and transitions in the EMDP, as a proxy of the model size). Third, we note that the alternative approach of verifying all the  $I$  MD strategies  $\sigma \in \Sigma^{MD}$  is not practical, due to the exponential increase of  $I$  (defined in Section 2.1.1) with the problem size.

In Fig. 6, we study the effect of different confidence levels  $C_L$  in the wind-energy forecast on the expected operator *Profit*, 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  $\eta_W$  (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. 7 we compare results with two other energy-pricing formulations proposed in the literature. He *et al.* [10] solve the optimization problem without enforcing any constraint. Varaiya *et al.* [7] 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 [10] has higher *Profit* (up to 5%), but also up to 12% more  $EENS$  and 10% less  $QoS$ , compared to our approach. The strategy from [7] 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 more 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 and dispatch strategies proposed in the literature, while maintaining the runtime acceptable for this application.

## 7. CONCLUSIONS AND FUTURE WORK

We proposed the first sound and complete 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 energy pricing and dispatch in smart grids with renewables. Results 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 account.

As future work, we plan to investigate techniques to generate more concise constraints to prove failure of the verification step, 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.

**Acknowledgments.** The authors thank J. Venkatesh and prof. T. Simunic Rosing for providing the measurement data used in the case study. The research was partially funded by DARPA Award Number HR0011-12-2-0016 and by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. Approved for public release; distribution is unlimited. The content of this paper does not necessarily reflect the position or the policy of the US government and no official endorsement should be inferred.

## 8. REFERENCES

- [1] Courcoubetis, C. and Yannakakis, M., “The Complexity of Probabilistic Verification,” *Journal of ACM*, vol. 42(4), pp. 857–907, 1995.
- [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 *Proc. of FSTTCS*, ser. LNCS. Springer Berlin Heidelberg, 1995, vol. 1026, pp. 499–513.
- [4] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.
- [5] “International Energy Agency, World Energy Outlook 2009.” [Online]: [http://www.worldenergyoutlook.org/docs/weo2009/WEO2009\\_es\\_english.pdf](http://www.worldenergyoutlook.org/docs/weo2009/WEO2009_es_english.pdf).
- [6] A. Gore, *An Inconvenient Truth*. New York: Rodale, 2006.
- [7] 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.
- [8] K. Porter and J. Rogers, “Survey of Variable Generation Forecasting in the West,” in *NREL Subcontract Report SR-5500-54457*, April 2012, p. 56.
- [9] S. Koch *et al.*, “Modeling and Control of Aggregated Heterogeneous Thermostatically Controlled Loads for Ancillary Services,” *Proc. of PSCC*, pp. 1–7, 2011.
- [10] M. He, S. Murugesan, and J. Zhang, “Multiple Timescale Dispatch and Scheduling for Stochastic Reliability in Smart Grids with Wind Generation Integration,” in *Proc. of IEEE INFOCOM*, 2011, pp. 461–465.
- [11] A. Puggelli *et al.*, “Polynomial-Time Verification of PCTL properties of MDPs with Convex Uncertainties,” in *Proc. of CAV*, ser. LNCS. Springer Berlin Heidelberg, 2013, vol. 8044, pp. 527–542.
- [12] A. Puggelli, “Formal Techniques for the Verification and Optimal Control of Probabilistic Systems in the Presence of Modeling Uncertainties,” Ph.D. dissertation, University of California, Berkeley, Berkeley, CA, 2014.
- [13] C. Baier *et al.*, “Controller Synthesis for Probabilistic Systems,” in *Exploring New Frontiers of Theoretical Informatics*. Springer US, 2004, vol. 155, pp. 493–506.
- [14] M. Lahijanian *et al.*, “Temporal Logic Motion Planning and Control with Probabilistic Satisfaction Guarantees,” *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 396–409, 2012.
- [15] M. Kwiatkowska and D. Parker, “Automated Verification and Strategy Synthesis for Probabilistic Systems,” in *Proc. of ATVA*, ser. LNCS, vol. 8172. Springer, 2013, pp. 5–22.
- [16] K. Sen *et al.*, “Model-Checking Markov Chains in the Presence of Uncertainties,” in *Proc. of TACAS*, ser. LNCS. Springer Berlin Heidelberg, 2006, vol. 3920, pp. 394–410.
- [17] A. Nilim and L. El Ghaoui, “Robust Control of Markov Decision Processes with Uncertain Transition Matrices,” *Journal of Operations Research*, pp. 780–798, 2005.
- [18] M. Y. Vardi, “Automatic Verification of Probabilistic Concurrent Finite State Programs,” in *Proc. of SFCS*, 1985, pp. 327–338.
- [19] V. Forejt *et al.*, “Automated Verification Techniques for Probabilistic Systems,” *Formal Methods for Eternal Networked Software Systems*, vol. 6659, pp. 53–113, 2011.
- [20] F. Bouffard and F. Galina, “Stochastic Security for Operations Planning with Significant Wind Power Generation,” in *IEEE Transaction on Power Systems*, 2008, vol. 23, no. 2, pp. 306–316.
- [21] A. Kučera and O. Stražovský, “On the Controller Synthesis for Finite-State Markov Decision Processes,” in *Proc. of FSTTCS*, ser. LNCS. Springer Berlin Heidelberg, 2005, vol. 3821, pp. 541–552.
- [22] K. Draeger *et al.*, “Permissive Controller Synthesis for Probabilistic Systems,” in *Proc. of TACAS*, ser. LNCS. Springer, 2014, pp. 531–546.
- [23] T. Brazdil *et al.*, “Stochastic Games with Branching-Time Winning Objectives,” in *Proc. of IEEE SLCS*, Aug. 2006, pp. 349–358.
- [24] C. Hang *et al.*, “Synthesizing Cyber-Physical Architectural Models with Real-Time Constraints,” in *Proc. of CAV*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 441–456.
- [25] P. Nuzzo *et al.*, “CalCS: SMT Solving for Non-linear Convex Constraints,” in *Proc. of FMCAD*, 2010, pp. 71–80.
- [26] S. Boyd and L. Vandenberghe, “Convex Optimization,” *Cambridge University Press*, 2004.
- [27] “Gurobi Optimizer.” [Online]: <http://www.gurobi.com/>.
- [28] S. Stoft, *Power System Economics: Designing Markets for Electricity*. New York: Wiley, 2002.
- [29] Y. Wan, “Long-term Wind Power Variability,” NREL Technical Report TP-5500-53637. [Online]: <http://www.nrel.gov/docs/fy12osti/53637.pdf>, January 2012.
- [30] E. Ela *et al.*, “Operating Reserves and Variable Generation,” NREL Technical Report. [Online]: <http://www.nrel.gov/docs/fy11osti/51978.pdf>, August 2011.
- [31] M. Kwiatkowska *et al.*, “PRISM 4.0: Verification of Probabilistic Real-Time Systems,” *Proc. of CAV*, pp. 585–591, 2011.