Received: 9 August 2019 | Revised: 20 December 2019 | Accepted: 26 January 2020

DOI: 10.1002/we.2489

RESEARCH ARTICLE

WILEY

Optimal strategies for wind turbine environmental curtailment

Jonathan Rogers

Georgia Institute of Technology, Atlanta, GA

Correspondence
Jonathan Rogers, Lockheed Martin Associate Professor, School of Aerospace Engineering, 270 Ferst Dr., Atlanta, GA 30332.
Email: jrogers8@gmail.com

Abstract

Wind turbine curtailment is oftentimes required as a means to mitigate environmental impacts of a wind energy installation. For example, curtailment may be required to satisfy constraints on the occurrence of shadow flicker on nearby homes or to reduce wildlife fatalities below a certain limit. This paper introduces an optimal curtailment strategy, which seeks to maximize wind plant revenue while meeting imposed environmental constraints by intelligently selecting curtailment times. To formulate the problem, a discrete set of curtailment decision times is defined, and long-term forecast data are used to predict demand-weighted power production at each time. Dynamic programming is used in conjunction with in situ meteorological sensors to compute the probabilistically optimal curtailment decision in real time. By leveraging both forecast data and real-time measurements, the algorithm ensures that the constraint budget is expended strategically, curtailing when revenue is low and saving operating hours for times when revenue is likely to be higher. Through a series of simulation studies involving shadow flicker constraints, algorithm performance is characterized and compared with two simpler approaches: a greedy scheme and a threshold-based scheme. These studies highlight the benefit of the optimal algorithm as well as the performance tradeoffs inherent in the three different solution approaches. Overall, the dynamic programming algorithm is shown to exhibit significant benefits in cases where sufficient meteorological and demand data are available.

KEYWORDS

Curtailment, Optimal Control

1 | INTRODUCTION

Over the past several decades, wind energy development in the United States and other developed countries has primarily targeted sparsely populated areas with large wind resource, readily available transmission interconnects, and minimal presence of vulnerable wildlife. Many of these ideal locations for onshore wind have already been successfully developed, and thus, new activities are focusing on locations closer to populated areas. A major challenge that arises during these development activities is community acceptance and adherence to relevant wildlife regulations. Wind farms impact local residences and wildlife through sound emissions, shadow flicker, visual impact, and encroachment on the habitats of birds, bats, and other species. As a result, wind energy developers are spending increasing effort working with communities and permitting authorities to mitigate these impacts and gain community buy-in for development activities. These mitigation steps oftentimes involve curtailing turbine operation at times when they impose negative impacts on the community or wildlife population—this is sometimes deemed "environmental curtailment." For instance, curtailment may be enforced during certain hours of the night to avoid bat fatalities.[1-3] As another example, if a turbine imposes shadow flicker on a residence beyond a mutually agreed accepted limit, the operator may be required to curtail turbine operation when flicker occurs after reaching the maximum number of allowed flicker hours per year.[4] As a result, environmental curtailments reduce the return on investment for wind installations, increasing the levelized cost of energy and serving as a potential market barrier to entry for wind power.

In many scenarios, a wind farm operator may have the flexibility to decide when to curtail operation in order to meet environmental constraints, leading to the possibility that such decisions can be optimized with respect to power production and energy demand. If the operator curtails turbine operation during a period of low energy production and/or low demand, little cost may be incurred. Likewise, if the operator curtails during periods of high energy production and high demand, significant revenue may be lost. Thus, in cases where curtailment limits are enforced, there is a need for optimal control strategies that optimize curtailment decisions based on measurements of current meteorological conditions and predicted future performance and energy demand. For instance, assume that a yearly shadow flicker budget is enforced on a wind turbine such that the total yearly flicker hours must not exceed a certain limit. An optimal control strategy may be developed such that the flicker hour budget is expended strategically during portions of the year when power generation is likely to be high. This strategy may curtail turbine operation when flicker occurs but power production or demand is low, thereby saving flicker hours to expend later when power production and demand is likely to be higher.

While curtailment optimization has been investigated to some degree, prior studies have focused primarily on curtailment due to energy demand or transmission limitations rather than environmental curtailments. Fine et al[5] defines voluntary curtailment as arising from a mismatch between energy supply and demand, while involuntary curtailment arises from use of a transmission system that cannot accommodate the full power output of a wind facility. The factors affecting these market-driven or grid-driven curtailments are studied in depth by Burke and O'Malley,[6] and Wu *et al.*[7] propose a robust approach to optimize the use of wind turbine generators (the so-called dispatch problem) to account for uncertainty in wind resource over short timescales. It is important to note that generation-related curtailment is inherently a different problem than environmental curtailment. Environmental curtailments often involve some type of specified, quantifiable limit on turbine operational impact—for instance, a budget on allowable shadow flicker hours per year or a limit on the number of nighttime operating hours per year to minimize bat fatalities.[2] Generation-related curtailment involves curtailment decisions that support near-term or immediate demands of the energy grid or transmission infrastructure. Because environmental curtailment involves constraint budgets that are often known well in advance, curtailment decisions can be optimized with respect to them over a relatively long timescale.

This paper seeks to optimize environmental curtailment decisions through the use of stochastic optimization techniques. The curtailment problem is inherently probabilistic in nature since the wind resource, meteorological conditions, and power demand at future times are inherently uncertain. Several authors have applied stochastic optimization algorithms to wind energy problems in recent literature. Chen *et al*[8] optimized power factors across a series of wind turbines using sequential quadratic programming, accounting for uncertainty in future wind power generation and other factors. Cheng and Zhang[9] proposed a probabilistic optimization algorithm for the wind turbine dispatch problem in a chance-constrained formulation, where again wind energy output over a finite horizon is considered as the primary source of uncertainty. Byon *et al*[10] optimized the maintenance strategies for a wind farm in the framework of a partially observable Markov decision process.

To the author's knowledge, the work described here represents the first algorithm developed to date to optimize environmental curtailment decisions. The goal of the algorithm is to determine the optimal curtailment decision in real time that balances power production and energy demand with environmental impact constraints. The problem is formulated by first defining a set of discrete times over a finite horizon at which curtailment decisions must be made. For instance, in the case of shadow flicker, these times are the hours during the year in which the solar geometry will lead to flicker on a given residence in the absence of clouds. The optimization problem is then to solve for curtailment decisions at each of the discrete times that maximize the expected value of a score function while meeting the environmental constraints. To solve this problem, it is assumed that forecast data are available to estimate the expected power production and demand at each of the decision times, and in situ sensors are available to provide real-time measurements of wind speed, energy demand, and possibly other environmental data. Under these assumptions, a dynamic programming algorithm[11] is developed that yields the probabilistically optimal solution to the curtailment problem. Specifically, the dynamic programming algorithm yields an expected value table that predicts the expected total reward for the system existing in any portion of the state space, at any of the discrete decision times. This expected value table, when used in conjunction with real-time measurements, provides a straightforward means to make the optimal curtailment decision. It is important to note that this dynamic programming solution is *provably optimal*, meaning that no alternative decision-making scheme can generate more expected reward as long as the expected power production and demand values are accurate.

The paper proceeds as follows. The formal definition of the environmental curtailment optimization problem is presented as a discrete optimization problem under uncertainty. As an example of how the formulation can be applied to a particular environmental constraint, the problem is specialized to the case of shadow flicker optimization. The dynamic programming solution algorithm is derived, and two alternative solution approaches (greedy and threshold-based) are defined for comparison. Simulation results for turbines operating under shadow flicker constraints are presented, highlighting the various performance tradeoffs between each of the methods. Results show that curtailment decisions made using the dynamic programming algorithm may yield significantly more wind plant revenue on average (during the hours that are considered for optimization) compared with other solution approaches as long as predictions of average wind resource and average energy demand are sufficiently accurate. Closing remarks provide guidance on application of the proposed optimization technique to other environmental constraints of interest to the wind energy community.

2 | ENVIRONMENTAL CURTAILMENT PROBLEM DEFINITION

2.1 | General problem definition

Several assumptions are invoked in the general problem formulation defined in this paper. First, it is assumed that curtailment optimization will be performed over a finite time horizon. Many environmental constraints are defined over a 1-year period (for instance, maximum number of shadow flicker hours per year), and thus, a 1-year time horizon may be a natural choice. A discrete set of N curtailment decision times are defined over this finite horizon, labeled as $k = \{0, \ldots, N - 1\}$. It is assumed that at each discrete decision time, a binary decision u_k is made between either curtailing turbine operation ($u_k = 0$) or not curtailing the turbine ($u_k = 1$). While in some environmental constraint scenarios (for instance, noise curtailment) the control decision is not binary and may involve selection between various operational modes, for the purposes of this work, the curtailment decision is assumed to be binary. Expansion of the decision space beyond this binary decision is left to future investigations. A final assumption employed here is that the curtailment decision at time k is maintained throughout the time interval represented by time k. For instance, suppose that the time discretization is performed on hourly intervals. If the decision is made to curtail the turbine at hour $k = 3$, then the turbine is curtailed for the entire hour and the decision is not revisited until the next hour, $k = 4$. Note that faster control update rates are possible by discretizing the decision times into smaller intervals (for instance, from 1-h intervals to 1-min intervals). This leads to an increased number of decisions that must be made but does not otherwise affect the problem formulation. It should also be noted that although this work focuses on optimal control over a finite horizon, infinite-horizon optimal control formulations also exist and can be leveraged for curtailment decision-making. Extension to infinite horizon problems is likewise left to future investigations.

Given the time discretization described above, define the state of the system as $\vec{x}_k \in \mathbb{R}^n$. The evolution of the state as a function of the control decision is given by the difference equation,

$$\vec{x}_{k+1} = \vec{x}_k + \vec{a}_k u_k, \tag{1}$$

where $\vec{a}_k \in \mathbb{R}^n$ is the so-called state transition activation vector that determines the effect of the control input on the state vector at decision time k. Vector $\vec{a}_k$ is a random variable such that $\vec{a}_k \tilde{A}_k$, where A_k is a vector-valued probability distribution. The system begins from an initial state $\vec{x}_0$ and ends at a terminal state $\vec{x}_N$ after the final curtailment decision is made. The environmental constraint is enforced on the state vector according to

$$\vec{x}_k \leq \vec{X}_{\max} \quad \forall k \in \{0, \ldots, N\}. \tag{2}$$

The reward function J defined for the curtailment optimization problem is defined as follows:

$$J = \sum_{k=0}^{N-1} r_k \left(u_k, \vec{w}_k \right). \tag{3}$$

In Equation (3), $\vec{w}_k$ is a random vector called the *reward activation vector* and is defined such that $\vec{w}_k \tilde{W}_k$ where the probability distribution W_k is vector-valued. The vector $\vec{w}_k$ determines how the curtailment decision u_k affects the incremental score function r_k. For instance, $\vec{w}_k$ may account for the random wind speed at timestep k so that if the wind is below the cut-in speed at time k, a decision to operate the turbine will not provide any positive reward since no energy will be produced. Further clarity on the practical meaning of the state transition activation and reward activation vectors $\vec{a}_k$ and $\vec{w}_k$ will be provided in the next section when the problem is specialized to the shadow flicker application.

With these definitions, the environmental curtailment optimization problem can be formally stated as follows:

$$\begin{gathered} \underset{u_k \in \{0,1\}}{\operatorname{argmax}} \mathbb{E}[J] \\ \text{subject to } \vec{x}_{k+1} = \vec{x}_k + \vec{a}_k u_k \\ \vec{x}_k \leq \vec{X}\text{max } \forall \text{k} \in \{0, \ldots, \text{N}\}, \end{gathered} \tag{4}$$

where $\mathbb{E}[\cdot]$ denotes the expected value. Note that the reward function is explicitly and implicitly a function of random variables $\vec{a}_k$ and $\vec{w}_k$, and thus (4) is a *stochastic* optimization problem that seeks to maximize the *expected value* of the reward function through selection of the curtailment decisions u_k, $k = \{0, \ldots, N - 1\}$.

2.2 | Application to shadow flicker curtailment

The purpose of this section is to illustrate application of the general problem formulation in (4) to the shadow flicker problem. It is common practice in the wind industry to locate turbines at sufficient distances from residences such that the annual number of shadow flicker hours on nearby houses are less than the threshold specified in wind ordinances (often 30 hours per year,[12] but sometimes as low as 8 h per year[13]). However, in cases where this threshold cannot be met due to other siting restrictions, curtailment may be enforced to meet this threshold. To formalize the curtailment optimization problem for shadow flicker, consider the case of a single turbine and a single nearby receptor. The hours each year during which the turbine will impose shadow flicker on this receptor can be computed by the solar geometry and terrain information. Let N be the total number of shadow flicker hours per year imposed on the receptor, assuming that the rotor is always turning and that conditions are always sunny. This is typically referred to as the number of "worst case" shadow flicker hours, since cloudiness and/or wind below the cut-in speed reduces the actual number of flicker hours observed each year. Each decision time k represents one "worst case" flicker hour over the course of the year.

Let x_k represent the number of flicker hours incurred by the receptor since the beginning of the year ($k = 0$), and thus, the initial starting condition is $x_0 = 0$. The imposed yearly shadow flicker hour limit is therefore $X_{\max}$. The time $k = 0$ can represent any reference starting point for the year, but for this work will be considered as the first flicker hour past January 1 for a given year. Each decision time k therefore corresponds to an hour at which flicker is predicted, from January 1 to December 31. The state transition activation variable a_k is a scalar representing the presence of clouds and, for the purposes of this work, is defined as a Bernoulli-distributed random variable with parameter p_k. As such, $a_k \in \{0,1\}$, meaning that only two states of cloudiness are assumed to occur—clear or mild cloud conditions which produce flicker ($a_k = 1$), or moderate-to-heavy clouds which preclude flicker ($a_k = 0$). The Bernoulli parameter p_k is the probability that clear or mild cloudiness conditions occur at time k, while the parameter $q_k = 1 - p_k$ represents the probability of cloudy conditions at time k. Note that if $a_k = 1$, then a decision not to curtail at time k ($u_k = 1$) will increment the number of incurred flicker hours by 1 according to Equation (1). Likewise, if $a_k = 0$, then a decision to operate the turbine will not increment the flicker hours since clouds prevent a shadow. The assumption that a_k may only take on two discrete values is sufficiently valid for the shadow flicker problem, since the generation of a shadow is usually considered to be binary. In other environmental curtailment problems, the state transition activation variable may take on more general values, although this may complicate the solution process since the possible states that x_k can take on may grow significantly. This becomes a particular concern if a_k is a continuous random variable, which means that x_k exists in a continuous state space. Exploration of these more complex cases is beyond the scope of this initial study.

The reward activation vector is defined for the shadow flicker case as consisting of two components $\vec{w} = \{w_1, w_2\}$, where $w_{1,k}$ is a random variable quantifying power production at decision time k and $w_{2,k}$ is a random variable quantifying energy demand at decision time k ($w_{2,k}$ is assumed to be normalized between 0 and 1 throughout this work). The reward function at time k is then defined as

$$r_k = w_{1,k} w_{2,k} u_k. \tag{4}$$

The reward function in (4) provides positive reward only if the decision is made to operate the turbine ($u_k = 1$). The reward amount is scaled by the power produced and energy demand such that reward is highest when both energy production and energy demand are maximum. Reward decreases when there is low demand or insufficient wind to produce significant power.

3 | SOLUTION ALGORITHMS

Three solution algorithms are presented in this section. The first is an optimal solution algorithm derived through dynamic programming. The other algorithms are greedy and threshold-based approaches offered for the purpose of performance comparisons with respect to the optimal solution. Prior to introducing the solution methods, in situ sensing techniques and forecasting methods are described that provide necessary data for the candidate solution techniques with respect to the shadow flicker application.

3.1 | In situ sensing and production and demand forecasting

When applied to the shadow flicker problem, the solution algorithms described in this section will rely on in situ sensors located at the turbine that can (1) predict the presence of shadow flicker at the current time and (2) estimate the wind speed, and thus power production potential, at the current time. Sensing solutions have been developed that can determine whether the turbine is producing a shadow given current ambient lighting conditions. For instance, the Vestas Shadow Detection System (VSDS) is an integrated set of light sensors attached to the turbine nacelle oriented East and West of the turbine.[14] VSDS can detect the presence of a shadow and, "if the unit registers shadow flicker beyond a certain threshold, it pauses the blades to avoid the strobing effect on nearby properties."[14] Note that this system was supposedly installed in more than 50 wind energy sites as of 2012,[15] and thus can be considered a mature technology for the purposes of this work. Likewise, numerous wind site meteorological stations are commercially available that can be used to measure the current wind speed (and thereby estimate the current power production capability of the turbine). It is assumed that some measurement of energy demand at the current time is also available for the algorithm to use, although implementation details of how this demand estimate is obtained are beyond the scope of this initial work.

In addition to in situ sensors that provide measurements of current conditions, the optimal solution algorithm described in the next section requires forecasts of wind conditions, cloudiness conditions, and energy demand at each of the decision times. A variety meteorological and energy demand forecasts are available through both government[16] and industry[17] sources, many of which are already in use by wind plant operators. The cloudiness forecast is used to derive the Bernoulli parameters $p_k, \forall\, k \in \{0, \ldots, N-1\}$ used in construction of the Bernoulli probability distributions A_k. The wind and energy demand forecasts are used to compute the probability distributions for w_1 and w_2 at each decision time k (the vector-valued distribution W_k). Defining the *demand-weighted power production coefficient* at time k as $w_k^* \equiv w_{1,k} w_{2,k}$ (so that $r_k = w_k^* u_k$), the probability distribution of w_k^* can be computed from W_k at each decision time. Thus, for the remainder of this work, it is assumed that the cloudiness probabilities $q_k = 1 - p_k$ and the probability distribution of the demand-weighted power production w_k^* are available from forecast data at each of the decision times $k \in \{0, \ldots, N-1\}$.

3.2 | Dynamic programming solution algorithm

The curtailment optimization problem described by Equation (4) has unique characteristics that allow it to be solved through dynamic programming. The problem is structured in terms of multiple stages, where a control decision must be made at each stage. Most importantly, the problem obeys the so-called *principle of optimality*.[11] This states that the optimal control policy (which dictates how the optimal u_k should be selected at each time or stage) has the property that whatever state results from the control decision at stage k, the subsequent control decisions made from stages k+1 to N−1 will constitute an optimal control sequence.[11] In simpler terms, an optimal control problem that satisfies the principle of optimality can be solved recursively by finding the optimal control at the last stage, and then using this result to find the optimal control at the second-to-last stage, and so on. This is because the optimal control over a series of stages starting from a given state does not depend on the control decisions made previously. Optimal control sequences for problems that satisfy the principle of optimality can be found through dynamic programming.

To develop the dynamic programming solution, the concept of the expected value table must first be introduced. The expected value table specifies the total reward expected from the system existing at each of its possible states, at each possible stage. That is, element (i, j) of the expected value table specifies the total expected reward that the system is predicted to gather from stages j through N−1, starting from the ith discretized value of x at stage j. For a one-dimensional state, the table is arranged such that each row corresponds to a particular state value (0 through X_{MAX}), while each column corresponds to a particular stage (note that this arrangement can be generalized to systems with higher-dimensional state vectors.) The Bellman equation[18] can then be used in conjunction with the expected value table to compute the optimal control at each stage, given observations of the reward and the state transition model. The Bellman equation is given as follows:

$$V(x_k) = \max_{u_k \in \{0,1\}} \left(r_k\left(x_k, u_k, \vec{w}_k\right) + \mathbb{E}\left[V\left(x_{k+1} | x_k, u_k, \vec{a}_k\right)\right]\right), \tag{5}$$

where $\mathbb{E}[\cdot]$ denotes the expected value operator. This equation dictates that the value of being in state x_k at stage k is the total maximum of the reward at the current stage, plus the expected reward at all future stages starting from x_{k+1}, where x_{k+1} results from giving the optimal u_k at the current stage. In practice, since u_k is either 0 or 1 in the curtailment problem, the maximum argument in (5) must only be computed for these two different options, and the one that is higher is selected as the value $V(x_k)$. In the case of shadow flicker curtailment, Equation (5) can be written as

$$V(x_k) = \max_{u_k \in \{0,1\}} \left(w_k^* u_k + \mathbb{E}[V(x_{k+1} | x_k, u_k, a_k)]\right). \tag{6}$$

In Equation (6), note that $w_k^* u_k$ is the reward obtained for giving control u_k at stage k. Furthermore, the state transition activation vector is reduced to the scalar cloudiness realization a_k, which dictates whether $x_{k+1} = x_k$ or $x_{k+1} = x_k$+1 if u_k = 1. For the remainder of this section, the optimal control algorithm will be described in the context of flicker curtailment, although the algorithm can be easily generalized to other forms of environmental curtailment.

The fact that the curtailment optimization problem obeys the principle of optimality means that the expected value table can be derived through a *backward induction* process where the expected value of the state can be found for the final stage (N−1), then that result is used to find the expected values of the states at the previous stage N−2, and so on until the first stage in the problem is reached. This expected value table is stored and used by the curtailment decision-making algorithm to arrive at the optimal curtailment decision in real time using Equation (6).

To compute the expected value table for the shadow flicker curtailment problem using backward induction, the expected value of the total reward for each possible state at stage $N-1$ is computed. Table 1 shows the results, which represent the final column in the expected value table. In this table (and throughout this section), let the sum of the resulting rewards at stage k and all following stages be denoted as $J_k \equiv \sum_k^{N-1} r_k$. As shown in Table 1, if $x_{N-1} \leq X_{MAX} - 1$, then the turbine can be operated without violating the flicker hour budget since this is the final flicker hour of the year (and thus the final control decision). Thus, the expected reward for existing in any of these states at stage $N-1$ is the expected value of the demand-weighted power at this stage, w_{N-1}^*. If $x_{N-1} = X_{MAX}$, then the flicker hour budget has already been used, and the turbine can be operated only if conditions are cloudy at this stage, which occurs with probability q_{N-1}. Since this is the only condition in which nonzero reward is obtained, the expected reward for existing in state $x_{N-1} = X_{MAX}$ is $q_{N-1}\mathbb{E}\left[w_{N-1}^*\right]$.

Prior to deriving the expected value at previous stages, it is helpful to examine the special case in which the system has used the entire flicker hour budget at any stage k, ie, the case in which $x_k = X_{MAX} \quad \forall\, k \in \{0, \ldots, N-1\}$. If the system has reached its flicker hour budget at any stage k, then the decisions for that stage and all future stages are simple: the turbine must be curtailed if it is sunny and can be operated if it is cloudy. That means that the expected total reward for stage k and all future stages is given by

$$\mathbb{E}[J_k | x_k = X_{MAX}] = \sum_{i=k}^{N-1} q_i \mathbb{E}\left[w_i^*\right]. \tag{7}$$

In other words, the total reward expected if $x_k = X_{MAX}$ is simply the sum of the cloudiness probabilities multiplied by the expected demand-weighted power at each stage. Since the expected value table is arranged such that the state values correspond to the rows and the stages correspond to the columns, Equation 7 is used to populate all entries in the last row of the table.

To compute the rest of the table entries, consider entry (i, k), which means that $x_k = i$ at stage k. Let the expected total reward of state $x_{k+1} = i$ at stage k+1 be given by $\alpha_{i,k+1}$, which is entry (i, k+1) in the expected value table. Likewise, let the expected total reward of state $x_{k+1} = i+1$ at stage k+1 be given by $\alpha_{i+1,k+1}$, which is entry (i+1, k+1) in the expected value table. Table 2 visualizes just this portion of the expected value table, showing portions of columns k and k+1. Given $\alpha_{i,k+1}$ and $\alpha_{i+1,k+1}$, the goal is then to compute $\mathbb{E}[J_k | x_k = i]$ using backward induction. Since the indices (i, k) are general and can range from $i \in \{0, \ldots, X_{MAX} - 1\}$ and $k \in \{0, \ldots, N-2\}$, the process detailed below can be applied to find any entry in the expected value table (except the last row and column, which are computed as described above).

To compute $\mathbb{E}[J_k | x_k = i]$, assume that the system arrives at stage k in state $x_k = i$. There are four possible scenarios if this occurs. Conditions can be sunny, and the observed demand-weighted power $w_k^* \geq \alpha_{i,k+1} - \alpha_{i+1,k+1}$, or conditions can be sunny and $w_k^* < \alpha_{i,k+1} - \alpha_{i+1,k+1}$. Likewise, conditions can be cloudy, and the observed demand-weighted power $w_k^* \geq \alpha_{i,k+1} - \alpha_{i+1,k+1}$ or conditions can be cloudy and $w_k^* < \alpha_{i,k+1} - \alpha_{i+1,k+1}$. Table 3 shows each of these four possibilities along with their associated probabilities, where $\delta\alpha \equiv \alpha_{i,k+1} - \alpha_{i+1,k+1}$ and $F_k(\cdot)$ denotes the cumulative density function (CDF) of w_k^*. Note that these four possibilities cover all possible scenarios once a_k and w_k^* are measured upon arriving at stage k.

TABLE 1 Total expected reward, one stage to go

x_{N-1}	$\mathbb{E}[J_{N-1}]$
0	$\mathbb{E}\left[w_{N-1}^*\right]$
1	$\mathbb{E}\left[w_{N-1}^*\right]$
$\vdots$	$\mathbb{E}\left[w_{N-1}^*\right]$
$X_{MAX} - 1$	$\mathbb{E}\left[w_{N-1}^*\right]$
X_{MAX}	$q_{N-1}\mathbb{E}\left[w_{N-1}^*\right]$

TABLE 2 General entry in expected value table

x	$\mathbb{E}[J_k]$	$\mathbb{E}[J_{k+1}]$
i	$\mathbb{E}[J_k \vert x_k = i]$	$\alpha_{i,k+1}$
i+1	-	$\alpha_{i+1,k+1}$

Note. The right two columns are excerpts from expected value table.

TABLE 3 Expected total reward calculation for $x_k = i$ at stage k

	a_k	w_k^*	Probability	u_k	$J_k + \mathbb{E}[J_{k+1}]$	$\mathbb{E}[J_k]$
Sunny	1	$w_k^* \geq \delta\alpha$	$(1-q_k)(1-F_k(\delta\alpha))$	1	$w_k^* + \alpha_{i+1,k+1}$	$(1-q_k)\left[(1-F_k(\delta\alpha))\left(\tilde{w}_k^* + \alpha_{i+1,k+1}\right) + F_k(\delta\alpha)\alpha_{i,k+1}\right] + q_k\left(\mathbb{E}\left[w_k^*\right] + \alpha_{i,k+1}\right)$
	1	$w_k^* < \delta\alpha$	$(1-q_k)F_k(\delta\alpha)$	0	$\alpha_{i,k+1}$	
Cloudy	0	$w_k^* \geq \delta\alpha$	$q_k(1-F_k(\delta\alpha))$	1	$w_k^* + \alpha_{i,k+1}$	
	0	$w_k^* < \delta\alpha$	$q_k F_k(\delta\alpha)$	1	$w_k^* + \alpha_{i,k+1}$	

The fourth column in Table 3 shows the probability of each of these four scenarios occurring. The probability value depends on the cloudiness probability q_k as well as the CDF of w_k^*, denoted as $F_k(\cdot)$, which yields the probability that w_k^* is below the value specified in the argument. The probability values are thus computed directly from the conditions in columns 2 and 3 of Table 3. Column 5 of the table shows the control input that will be selected in each scenario, assuming that the decision-making algorithm acts to maximize the sum of current and expected future rewards. For instance, if conditions are sunny at stage k but the observed current reward w_k^* is less than the difference between the expected rewards for $x_{k+1} = i$ and $x_{k+1} = i+1$ (ie, row 3 in Table 3), then the use of a flicker hour at this stage is not justified and $u_k = 0$ will be selected. This is because, if $w_k^* < \delta\alpha$, then $w_k^* + \alpha_{i+1,k+1} < \alpha_{i,k+1}$, and thus, the sum of current and future expected rewards is higher if the turbine is curtailed at this stage. Likewise, if conditions are cloudy at stage k, then $u_k = 1$ is selected regardless of the reward because no flicker penalty will result.

Column 6 of Table 3 shows the sum of the stage k reward plus the expected reward of the state that results for each of the four possibilities. To find the total expected reward for existing in state $x_k = i$ at stage k, the expected value of the entries in column 5 are computed, then weighted by their associated probabilities of occurring and summed together. The result, shown in column 7 of Table 3, is entry (i, k) of the expected value table. Note the presence of the quantity $\tilde{w}_k^*$ in the total expected reward. This is defined as

$$\tilde{w}_k^* \equiv \mathbb{E}\left[w_k^* | w_k^* \geq \delta\alpha\right]. \tag{8}$$

Equation (8) shows that $\tilde{w}_k^*$ represents the expected value of the reward at stage w_k^*, assuming it is greater than or equal to δα. Applying the definition of the expected value to Equation (8) yields

$$\tilde{w}_k^* = \int_{\delta\alpha}^{\infty} x f_k(x) dx, \tag{9}$$

where $f_k(\cdot)$ is the probability density function (PDF) governing w_k^*. As a result, the total expected reward shown in the final column depends on the both the PDF and CDF of the uncertain reward w_k^*. These functions can be estimated from meteorological and demand data sets, with natural tradeoffs between availability of data and resulting estimation accuracy.

To compute the entire expected value table, the final row and final column are initialized first using the procedure described above. Then, starting from stage N−2, the remaining entries are computed at each previous stage using the formula in the last column of Table 3. A final exception is that for all elements of the table such that the number of flicker hours remaining in the budget is less than the number of stages remaining (ie, $X_{MAX} - x_k \geq N - k$), the expected total reward is just the sum of all remaining expected rewards ($\mathbb{E}[J_k] = \sum_{k}^{N-1} \mathbb{E}\left[w_k^*\right]$). This is because there are more hours remaining in the flicker budget than hours (or stages) remaining, and thus, they can be used regardless of the measured reward at any remaining stage.

The expected value table is computed offline and stored for use in real-time decision-making at each stage. When the operator arrives at stage k in state x_k, measurements of a_k and w_k^* are obtained. If $a_k = 0$, the turbine is operated. If $a_k = 1$ and the current reward is higher than the difference between the expected reward at the next stage for $x_{k+1} = i$ and $x_{k+1} = i+1$ as read from the expected value table, then the turbine is operated. Otherwise, it is curtailed. The control decision-making algorithm is essentially the direct application of Bellman's equation in Equation (6) using the computed expected rewards at each state.

Given that implementation of the optimal control algorithm relies on storage of the expected value table, it is important to quantify typical data storage requirements. A typical flicker curtailment scenario may involve 100 candidate flicker hours per year, with a budget of 30 allowable flicker hours per year. This results in 3000 total values that must be stored, which is extremely low given modern data storage capabilities. Wildlife curtailment problems may involve many more stages and much larger operational budgets ($X_{\max}$). However, given that the number of values to be stored in the expected value table is the number of stages multiplied by the number of discretized values of the state, both the state and time discretizations can be adjusted as necessary in larger problems to reduce data storage requirements if they become too large. A natural tradeoff exists between the resolution of time and state discretizations and the degree of optimality achieved by the algorithm.

A final discussion is in order regarding how the above optimal decision-making algorithm can be extended to other curtailment problems, particularly wildlife curtailment which is of specific interest to the wind energy community. The wildlife curtailment problem can be formulated by establishing an acceptable fatality budget X_{MAX} over a fixed time period (say, 1 year), a set of selectable operating modes in which the turbine can operate at each decision time ($u_k \in \{u^{(1)}, u^{(2)}, \ldots, u^{(m)}\}$), and a model $h\left(\vec{a}_k, u_k\right)$ for how many fatalities occur over each decision interval k that can be a function of the operating mode u_k and environmental parameters $\vec{a}_k$. Although the development of such predictive models is still an active area of research, new wildlife "smart curtailment" approaches[19] have begun to use such models for real-time decision-making. Unlike in Equation (1) where the state update is simply computed as $\vec{x}_{k+1} = \vec{x}_k + \vec{a}_k u_k$, for wildlife curtailment a more general state update equation of the form $\vec{x}_{k+1} = \vec{x}_k + h\left(\vec{a}_k, u_k\right)$ should be used. With these definitions, the optimal decision-making algorithm in this section can be applied without modification, although the number of possible states will be much larger than for flicker curtailment. This will add to the offline computation necessary to build the expected value table. While further investigation of this algorithm applied to wildlife curtailment is certainty warranted, it is beyond the scope of this initial study and is left to future work.

3.3 | Greedy solution algorithm

A greedy solution algorithm is also presented in order to compare performance with respect to the optimal controller. The greedy solution algorithm simply operates the turbine at the first available opportunities, regardless of the reward w_k^*, until the flicker budget is exhausted. This approach can be mathematically described as follows:

$$u_k = \begin{cases} 1 & \text{if } a_k = 0 \text{ or } x_k < X_{\max} \\ 0 & \text{else} \end{cases} \quad \forall k \in \{0, \ldots, N-1\}. \tag{10}$$

According to the description in Vestas Wind Systems A/S,[1] the Vestas VSDS shadow flicker detection system implements this greedy scheme, and thus, it represents an important algorithm for purposes of comparison against state-of-the-practice in the wind industry.

3.4 | Threshold-based solution algorithm

A third solution algorithm uses a fixed threshold on demand-weighted power production to determine whether or not to curtail. Let this fixed threshold be given as w_{th}^*. This control scheme is given as

$$u_k = \begin{cases} 1 \text{ if } a_k = 0 \text{ or } \left(x_k < X_{\max} \text{ and } w_k^* > w_{\text{th}}^*\right) \text{ or } (N - k \le X_{MAX} - x_k) \\ 0 \quad \text{else} \end{cases} \quad \forall k \in \{0, \ldots, N-1\}. \tag{11}$$

As shown in (11), the threshold-based scheme is a modified greedy scheme that avoids using operational hours when the demand-weighted power production is low, thereby saving operational hours for later when reward may be higher. However, unlike the dynamic programming approach, the threshold-based scheme does not leverage any predictions of future demand-weighted production, making it susceptible to poor decision-making in which operational hours are saved for times when the demand-weighted production is unlikely to exceed the threshold w_{th}^*. At the same time, though, because the threshold-based scheme does not use any forecasting, it is unaffected by inaccurate forecasts, unlike the dynamic programming scheme. These tradeoffs will be explored in Section 4. Note in Equation (11) that the condition on ($N - k \le X_{MAX} - x_k$) is included so that the turbine is always operated if the number of remaining hours in the year is less than the number of hours remaining in the flicker budget.

It is possible to create a threshold-based scheme in which specific thresholds are set for each stage, rather than a single threshold across all stages. However, this results in a potentially very large number of tuning parameters that must be set. If demand-weighted production forecasts are used to determine these thresholds, the resulting scheme resembles the dynamic programming approach and offers no benefit beyond the scheme described in Section 3.2. As a result, the threshold-based scheme explored in this work uses a single, fixed threshold applied over all stages.

4 | RESULTS

A series of simulation results is presented in this section to quantify the benefit of the optimal curtailment algorithm with respect to the alternative schemes, and to explore the tradeoffs associated with each approach. First, results are presented for an example scenario in which the expected demand-weighted power production and the cloudiness probabilities are defined as a simple sine wave. These simplified probability distributions, used for demonstration purposes, are employed in subsequent results to highlight various benefits associated with both the threshold-based and dynamic programming algorithms. A final set of results uses publicly available wind data and power demand data for Northern California, combined with a shadow flicker prediction simulation, to quantify the improvement in power generation using the optimal scheme compared with the threshold and greedy algorithms in a more realistic example. All results in this section use a power curve based on the General Electric 2.5 MW turbine with a 100-m rotor diameter, cut-in speed of 3 m/s, rated speed of 11 m/s, and cut-out speed of 26 m/s. This power curve is shown in Figure 1.

4.1 | Example case

The example case in this section considers flicker curtailment with N = 120 candidate flicker hours and a limit of X_{MAX} = 30 hours of allowable flicker per year. To demonstrate performance of the optimal algorithm in comparison to the other approaches, simple sine functions are used for

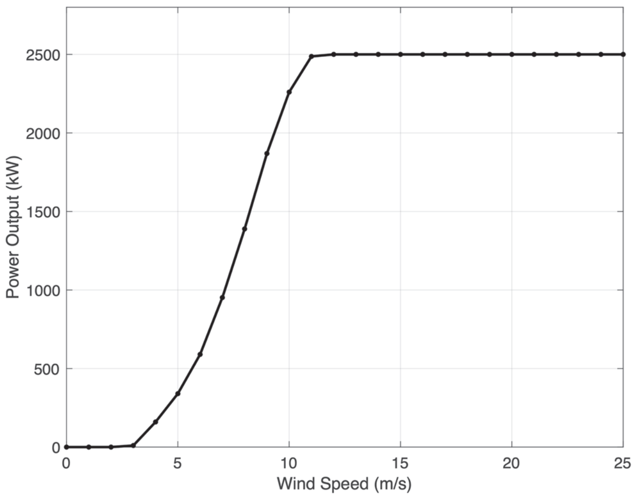

FIGURE 1 Example power curve for 2.5 MW turbine, based on GE 2.5-100

the expected demand-weighted power production and the cloudiness probabilities over the course of a 12-month period. The mean demand-weighted power at each stage for this example is given by

$$\mathbb{E}\left[w_k^*\right] = 1200 + 800\cos\left(\frac{2\pi k}{N} + \pi\right) \text{ kW}. \tag{12}$$

Likewise, the cloudiness probabilities are defined for this example as

$$q_k = 0.5 + 0.1\cos\left(\frac{2\pi k}{N} + \frac{\pi}{2}\right) \tag{13}$$

To build the expected value table, a Gaussian probability distribution is constructed at each stage using the mean value from Equation (12) and a standard deviation of 900 kW. These distributions are truncated between 0 and 2500 kW due to the power curve in Figure 1. This yields the PDFs f_k and CDFs F_k needed to compute the expected value table.

To perform a single-year simulation, demand-weighted power at each candidate hour w_k^* is sampled from the Gaussian distributions f_k. Similarly, at each candidate hour, a cloudiness value $a_k \in \{0,1\}$ is sampled from a Bernoulli distribution with parameter q_k taken from Equation (13). The resulting data for w_k^* and q_k for $k = 0,\ldots,N-1$ are then provided to each algorithm, yielding curtailment decisions for each different solution approach. The threshold algorithm uses $w_{\text{th}}^* = 400$ kW in this example.

The results for an example simulation are shown in Figure 2. The two sublots at the top of Figure 2 depict the functions for q_k (from Equation (13)) and $\mathbb{E}\left[w_k^*\right]$ (from Equation (12)), respectively. The third row shows the resulting samples for a_k at each candidate decision hour, while the fourth row depicts the samples for w_k^*. Note that the samples for w_k^* show a mild (but still noticeable) correlation with the expected trend of $\mathbb{E}\left[w_k^*\right]$ due to the rather large standard deviation of 900 kW imposed at each sampling time. Rows 5 to 7 show the curtailment decisions of the optimal, threshold-based, and greedy algorithms, respectively, where $u_k = 0$ indicates a decision to curtail operation. The final row in Figure 2 shows the total accumulated flicker hours x_k for each of the algorithms. Note that, as expected, the greedy algorithm uses its flicker budget at the first available opportunity, while the other two algorithms take longer to reach the 30-hour limit. The (demand-weighted) energy production results for this case are shown in Table 4. Note that the optimal algorithm yields the best performance, producing 31.2% more energy over the course of the candidate hours than the greedy algorithm and 11.8% more than the threshold-based approach.

The problem considered here is stochastic in nature, meaning that some random draws from the power production and cloudiness distributions will change the performance of one algorithm with respect to another on a per-trial basis. Thus, a Monte Carlo simulation was performed to quantify the average performance over a large number of trials. A total of 1000 simulations was performed using the same setup as above, with only the random sequences of w_k^* and q_k changing between each trial. Energy production was recorded in each case. The results of these simulations are shown in Figure 3 and Table 4. The trends in these results verify the better performance, on average, of the optimal algorithm compared with both the greedy and threshold-based approaches. On average, the optimal algorithm generates 20.8% more energy per year than the greedy approach and 7.5% more energy per year than the threshold-based approach.

It should be noted when considering these results (and subsequent examples below) that the increased energy captured by the optimal algorithm represents a significant increase over that obtained by the greedy approach only during the 120 candidate flicker hours each year. Viewed as a fraction of the total annualized energy production (AEP) for the turbine, the improvement obtained by the optimal algorithm is quite small

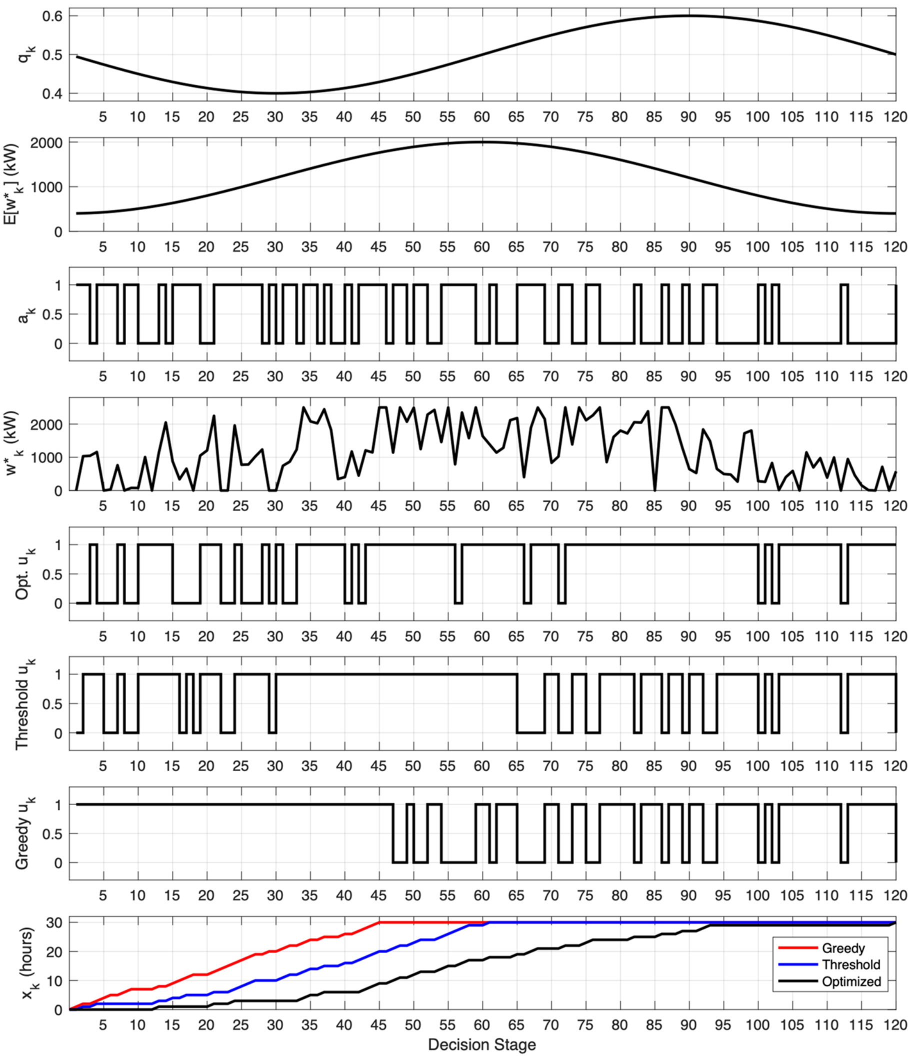

FIGURE 2 Curtailment results for example simulation [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Demand-weighted energy production for example cases

	Algorithm	Energy Production, MWh	% Improvement Over Greedy
Example case	Optimal	128.6	31.2
	Threshold	115.0	17.4
	Greedy	98.0	-
Monte Carlo	Optimal	129.3 (mean)	20.8
	Threshold	120.3 (mean)	12.4
	Greedy	107.0 (mean)	-

(since there are only 120 candidate flicker hours to optimize each year out of a total of approximately 8766 hours per year that the turbine may operate). If the algorithm were to be successfully applied to wildlife curtailment, the resulting AEP improvement may be far more significant since the number of candidate curtailment hours is usually much larger. As mentioned above, consideration of the wildlife curtailment problem is certainly warranted and will be the subject of future work.

The amount of energy produced per year by the threshold-based algorithm in a given scenario is directly dependent on the selected threshold w^*_{th}. To investigate this, Monte Carlo simulations were performed for an array of different w^*_{th} values ranging from 100 to 1700 kW, with 1000 simulations performed for each setting of w^*_{th}. The average demand-weighted energy over each of the 1000 runs was recorded in each case and is shown in Figure 4. Average results for the optimal and greedy algorithms are shown as horizontal lines, since they do not change with w^*_{th}. As shown in the figure, the average annual energy production using the threshold algorithm peaks at about w^*_{th} = 1100 kW at a value near that achieved by the optimal algorithm (but still slightly less). Note that this curve is highly dependent on the scenario—different values of N and X_{MAX} and different distributions for cloudiness probability and demand-weighted power production will change this curve significantly, yielding

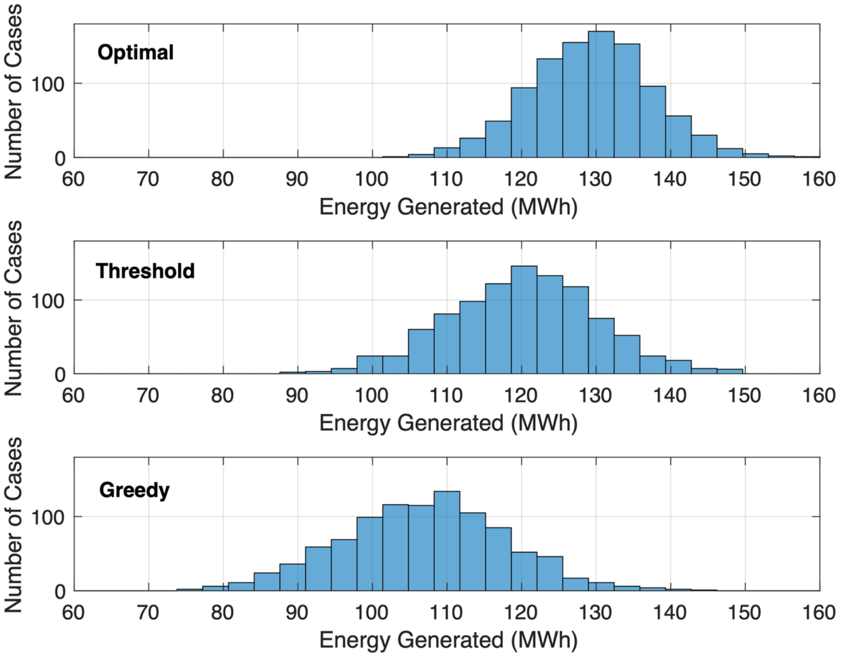

FIGURE 3 Histogram of Monte Carlo results for example scenario [Colour figure can be viewed at wileyonlinelibrary.com]

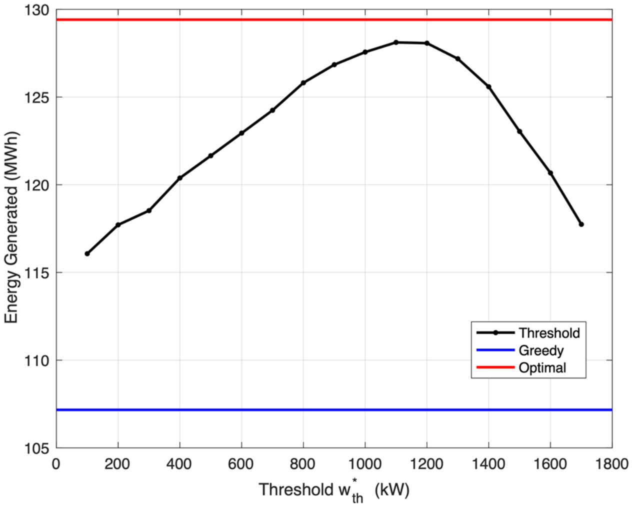

FIGURE 4 Average annual demand-weighted energy production for various w^*_{th} values [Colour figure can be viewed at wileyonlinelibrary.com]

different optimal thresholds. The results in Figure 4 highlight both the sensitivity of the algorithm's performance to this parameter, as well as the fact that the average energy captured by the threshold-based code is still less than that of the dynamic programming algorithm, even with optimal selection of w^*_{th}.

4.2 | Comparisons of threshold and optimal algorithm performance

Specific tradeoffs exist as to the benefits and drawbacks of the threshold-based and optimal algorithms. The results in this section highlight those tradeoffs through specific targeted examples. The first example considers the optimal algorithm's sensitivity to errors in the forecast of demand-weighted power production over the course of the year. To demonstrate this sensitivity, consider a scenario identical to the one in the previous section, except that the forecast for $\mathbb{E}[w^*_k]$ is given by Equation (14):

$$\mathbb{E}[w^*_k] = 1200 + 1000\cos\left(\frac{2\pi k}{N} + \frac{\pi}{2}\right) \text{ kW}. \tag{14}$$

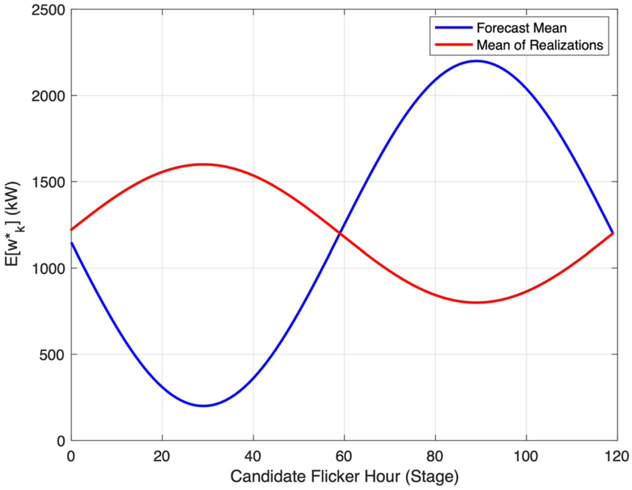

FIGURE 5 Forecast (blue) and actual (red) expected w_k^* for AEP [Colour figure can be viewed at wileyonlinelibrary.com]

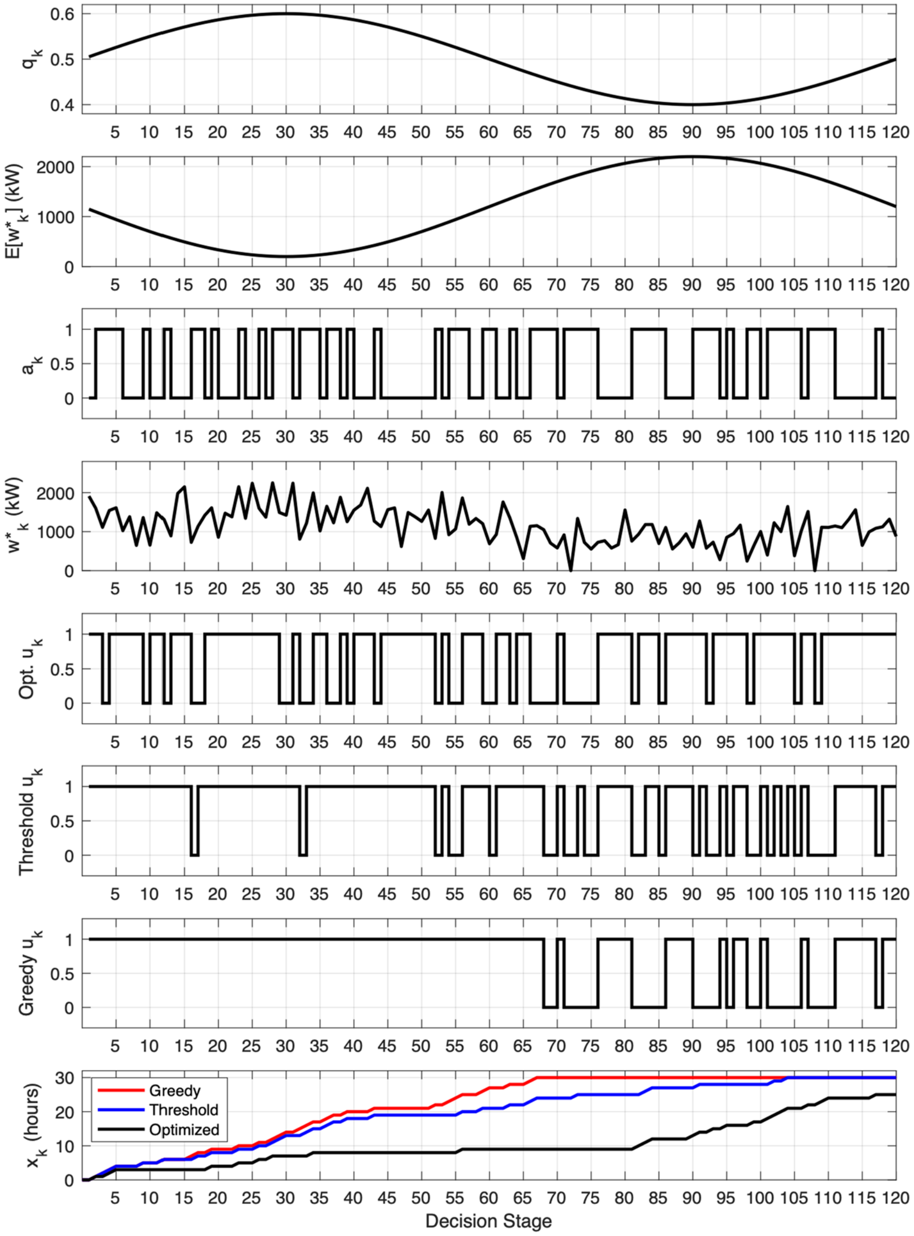

FIGURE 6 Curtailment results for inaccurate forecast example [Colour figure can be viewed at wileyonlinelibrary.com]

However, suppose that in a given year the forecast is significantly inaccurate and the actual mean demand-weighted power production is given by

$$\mathbb{E}\left[w_k^*\right] = 1200 + 400\cos\left(\frac{2\pi k}{N} + \frac{3\pi}{2}\right) \text{ kW}. \tag{15}$$

This means that the distributions f_k, which govern w_k^*, are different between those used to build the expected value table (Equation (14)), and those used in generation of the simulated realizations of w_k^* (Equation (15)). The forecast and actual curves for $\mathbb{E}\left[w_k^*\right]$ from (14) and (15) are shown in Figure 5. Note that the forecast $\mathbb{E}\left[w_k^*\right]$ exhibits a maximum that is both higher and later than the maximum value of the actual $\mathbb{E}\left[w_k^*\right]$. The effect of this is that the dynamic programming algorithm will curtail the turbine at early stages because it expects better demand-weighted power generation to be available later in the year. In actuality, the best power is available at early stages and thus performance suffers as a result.

Figure 6 shows curtailment results for an example scenario using the $\mathbb{E}\left[w_k^*\right]$ curves shown in Figure 5. In this case, the threshold-based algorithm uses its optimal value of $w_{\text{th}}^* = 1100$ kW. As anticipated, the optimal algorithm curtailed repeatedly during early stages, saving flicker hours for later stages when demand-weighted power turned out to be fairly low. As a result, the optimal algorithm did not even use the complete flicker hour budget of 30 hours. By comparison, the threshold-based algorithm performed the best since it is insensitive to errors in the forecast model. In this scenario, the threshold algorithm produced 117.8 MWh, the optimal algorithm produced 107.1 MWh, and the greedy algorithm produced 115.0 MWh. To verify these trends, a Monte Carlo simulation was performed with 1000 simulation runs, where random sequences of w_k^* and a_k were used in each case. The results of these Monte Carlo simulations are provided in Figure 7, where the histograms show that the optimal algorithm generates roughly 6% less (demand-weighted) energy per year compared with the threshold-based approach.

Overall, these results highlight the sensitivity of the optimal algorithm to errors in the forecast model for demand-weighted power. The fact that the maximum value of the actual $\mathbb{E}\left[w_k^*\right]$ is both lower and earlier than the forecast maximum causes the optimal algorithm to wait for better conditions, which never actually materialize (on average). However, it should be noted that the mismatch in the forecast and actual $\mathbb{E}\left[w_k^*\right]$ used in this example is somewhat of a contrived, worst-case scenario and thus not very representative of actual forecast errors. Numerous simulation studies (which are not shown here due to space limitations) showed that reasonable differences between the forecast $\mathbb{E}\left[w_k^*\right]$ and the actual $\mathbb{E}\left[w_k^*\right]$ have less of an effect on performance of the optimal algorithm than that shown in this example. In general, mild forecast errors tend to reduce performance of the optimal algorithm but rarely to the extent that it performs worse than the threshold algorithm. Nevertheless, this example highlights the important fact that the optimal algorithm is sensitive to errors in the forecast model, whereas the threshold-based algorithm is not. Thus, in cases where there is little confidence in the forecast for demand-weighted power, an operator may select the threshold-based rather than the optimal approach. Further study of the effects of forecast uncertainty on algorithm selection is warranted but is beyond the scope of this initial work.

A second example in this section highlights a possible drawback of the threshold-based algorithm. Because the algorithm uses a fixed, predetermined threshold for curtailment decisions, it is susceptible to off-nominal scenarios in which demand-weighted power rarely or never exceeds that threshold over the decision times. This may be particularly problematic when the threshold value w_{th}^* is set fairly high. In contrast,

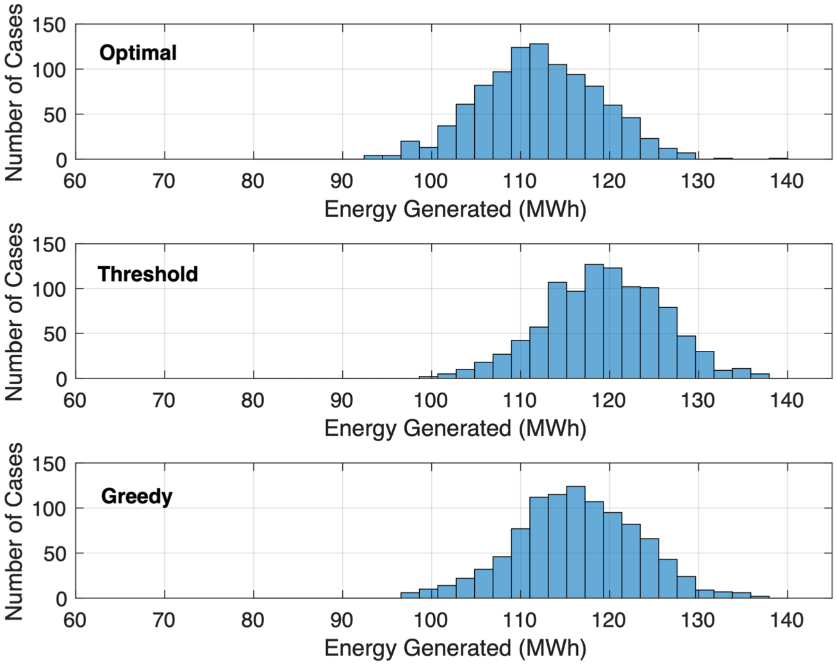

FIGURE 7 Histograms of Monte Carlo results for inaccurate forecast example [Colour figure can be viewed at wileyonlinelibrary.com]

the optimal algorithm essentially adjusts its threshold used for the curtailment decision based on the current stage and state of the system. For instance, at later stages, if x_k is still much lower than X_{MAX}, the optimal algorithm will naturally lower the value at which it will operate the turbine (and use a flicker hour) to ensure it extracts as much energy as possible in the remaining stages. In contrast, the threshold-based algorithm does not adapt to the current state and stage and is thus susceptible to making poor decisions when demand-weighted power does not exceed the threshold as often as expected.

To study this in more depth, consider a somewhat extreme example in which the actual demand-weighted power never reaches w^*_{th} at any of the decision stages. The scenario for this example is identical to the case studied in Section A, except that any time a sample for $w^*_k > w^*_{\text{th}}$, it is set to $w^*_k = w^*_{\text{th}} - 100$ kW (ie, slightly below the threshold). As a result, the threshold-based algorithm only operates the turbine when conditions happen to be cloudy, refusing to use an hour of the flicker budget at any stage since the demand-weighted power never exceeds its threshold. Figure 8 shows example simulation results for this case. As expected, the threshold-based algorithm uses only 13 hours out of the 30-hour budget, operating the turbine only when it is cloudy or during the last 30 hours of the year (since the algorithm always attempts to use the remaining budget in the final stages according to Equation (11)). This results in demand-weighted energy production that is 32.9% less than that produced by the optimal algorithm and 24.4% less than that produced by the greedy algorithm. Monte Carlo results for this same scenario are shown in Figure 9. In the Monte Carlo studies, the mean (demand-weighted) energy produced by the threshold algorithm is 49.4 MWh, compared with 63.8 MWh for the greedy and 72.7 MWh for the optimal algorithms.

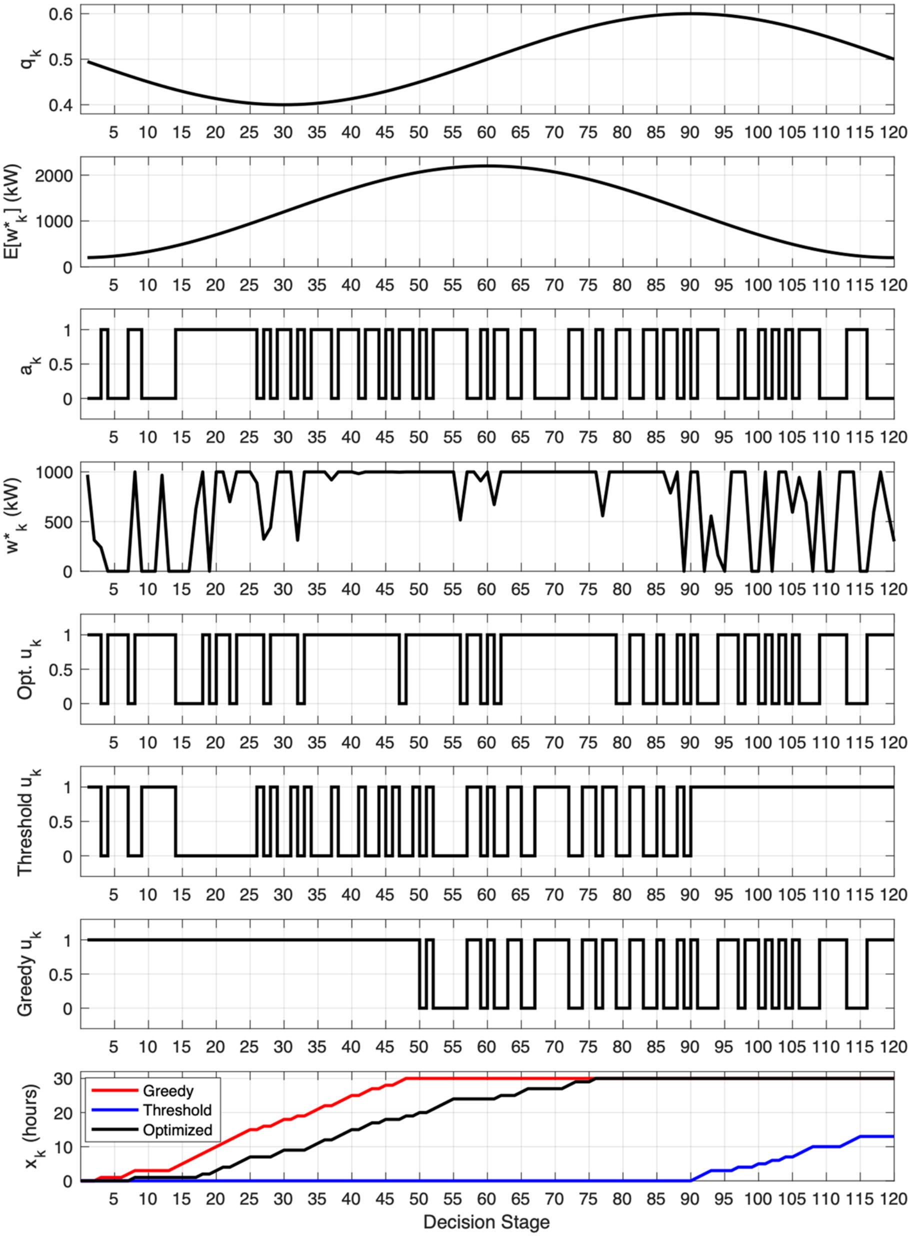

FIGURE 8 Curtailment results for threshold saturation example [Colour figure can be viewed at wileyonlinelibrary.com]

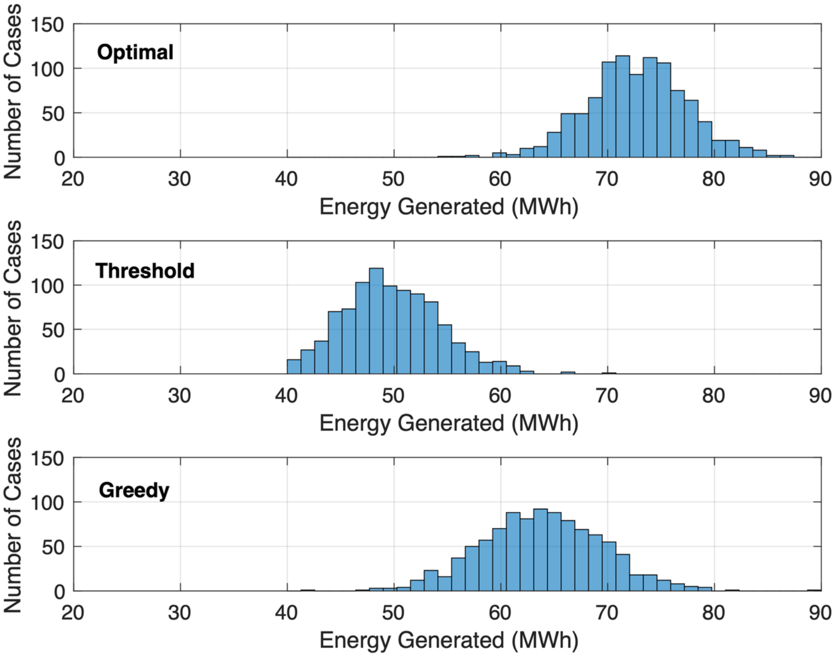

FIGURE 9 Monte Carlo simulation results for threshold saturation example [Colour figure can be viewed at wileyonlinelibrary.com]

This example highlights the fact that the threshold algorithm is sensitive to a particular type of uncertainty in demand-weighted power forecasting—namely, one in which the mean of the realized power production is biased lower than the forecasted mean. This sensitivity will impact the results more in cases where the threshold is higher. As shown in Figure 4, there is an optimal threshold value for each scenario (combination of N, X_{MAX}, and probability distributions for w_k^* and a_k). As X_{MAX} gets smaller with respect to N, the optimal threshold will get larger since the threshold algorithm should only use one of the few allowable flicker hours when w_k^* is high. This higher threshold setting increases the algorithm's sensitivity to lower-than-expected demand or wind speed conditions. Thus, this sensitivity will be important when the constraint X_{MAX} is relatively small compared with the number of stages N.

4.3 | Example using real-world data

A final example explores algorithm performance in a scenario constructed from real-world data. This example considers a single turbine and a nearby receptor, as pictured in Figure 10. The receptor is located approximately 258 m to the Northwest of the turbine, separated by flat ground. Shadow flicker modeling software from Persimia, LLC[20] was used to predict the number of worst-case flicker hours per year using the GE

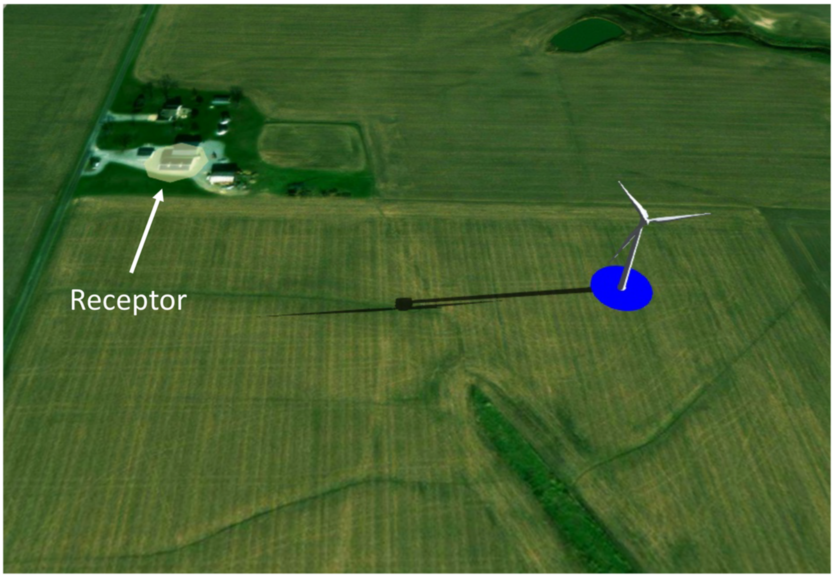

FIGURE 10 Turbine and receptor geometry for final example[20] (not to scale) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 5 Number of shadow flicker hours per month

Month	Number of Flicker Hours
January	8
February	35
March	36
April	0
May	0
June	0
July	0
August	0
September	20
October	27
November	0
December	0
Total	126

2.5 MW turbine described at the beginning of this section. The number of flicker hours per month predicted to occur at this receptor is given in Table 5. Note that the total number of predicted flicker hours per year is quite high at 126 hours, relative to what is usually acceptable under typical setbacks. The geometry used here is shown for illustrative purposes only and is not necessarily meant to be indicative of an actual scenario under typical current setbacks.

Meteorological and power demand data were gathered from publicly available sources for this example. Monthly average wind speed and monthly cloudiness data for San Francisco, CA, was obtained from the National Oceanic and Atmospheric Administration Comparative Climatic Data,[16] which provides an aggregation of weather data for San Francisco recorded between 1984 and 2018. The monthly average wind speeds for this site (averaged over this 32-year period) are shown in Figure 11 (top). To obtain the approximate average wind speed at the turbine hub height of 75 m, the average wind data were corrected using the wind profile power law[21] with an exponent of 1/7, assuming the meteorological station was located at an altitude of 4m (the mean sea level altitude of San Francisco International Airport). The monthly cloudiness data provided by National Centers for Environmental Information[2] report the average number of cloudy, partly cloudy, and sunny days each month. The cloudiness probabilities for each month were then computed by summing the average number of cloudy days and one half the average number of partly cloudy days, and dividing by the total number of days in each month. The monthly cloudiness probabilities are provided in Figure 11 (center). Finally, power demand data were obtained from average monthly electricity usage information for San Francisco County for the years 2015,

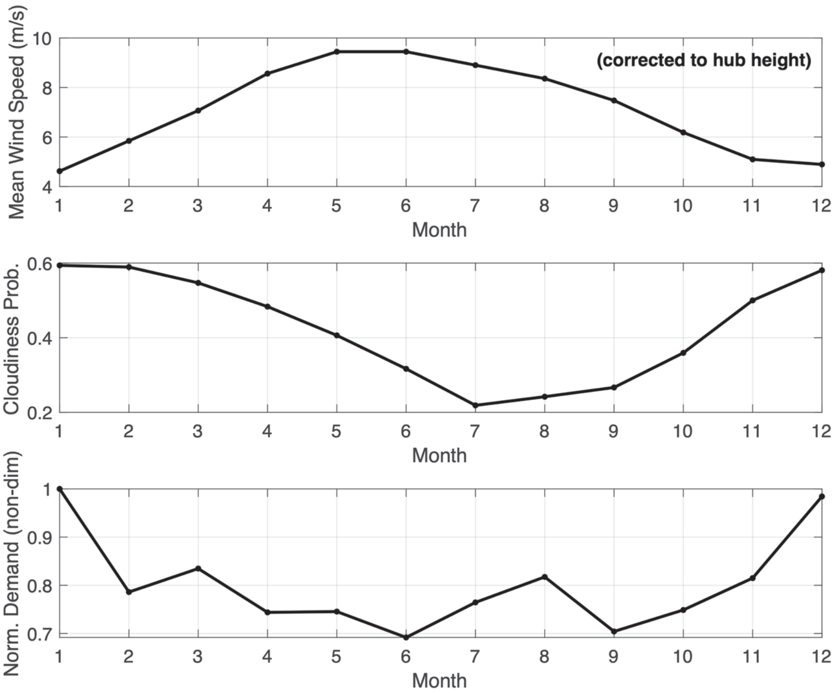

FIGURE 11 Average wind speed, average normalized demand, and cloudiness probabilities for example case [Colour figure can be viewed at wileyonlinelibrary.com]

2016, and 2017, provided by the Center for Sustainable Energy.[22] This data were averaged over each of the 3 years and normalized, with the results shown in Figure 11 (bottom). In this figure, month 1 corresponds to January and month 12 to December.

To obtain the data needed for the expected value table and to perform simulation studies, the data in Figure 11 have to be converted to PDFs, from which samples can be obtained. A Weibull distribution was created for each month with its mean at the value in Figure 11 (top), and a shape parameter of 2.5. A Gaussian distribution was used to model demand each month, with the mean given in Figure 11 and a standard deviation of 0.2 (in normalized units). This distribution was truncated at 0 and 1. To obtain samples for demand-weighted power at each stage, samples from the Weibull distribution were generated and converted to power using the power curve in Figure 1. These values were then multiplied by samples from the truncated Gaussian distribution representing power demand. These realizations of w_k^* were then used to estimate the PDF and CDF of w_k^*, which are required to construct the expected value table.

An example simulation is performed using this data to compare performance of the three algorithms. The threshold algorithm uses w_{th}^* = 800 kW, and the flicker hour budget is set to X_{MAX} = 30 hours. Figure 12 shows the results of the simulation, where it can be seen that the greedy and optimal algorithms use the entire 30-hour flicker budget over the 12-month period, while the threshold algorithm uses only 26 hours. The optimal algorithm generates 79.6 MWh, while the threshold-based algorithm generates 73.2 MWh and the greedy algorithm only 63.8 MWh. It is interesting to compare the approaches of the optimal and threshold algorithms in this case. As seen in the bottom of Figure 12, the algorithms make similar decisions through most of the stages, but during roughly the final third of the stages the optimal algorithm is willing to operate the

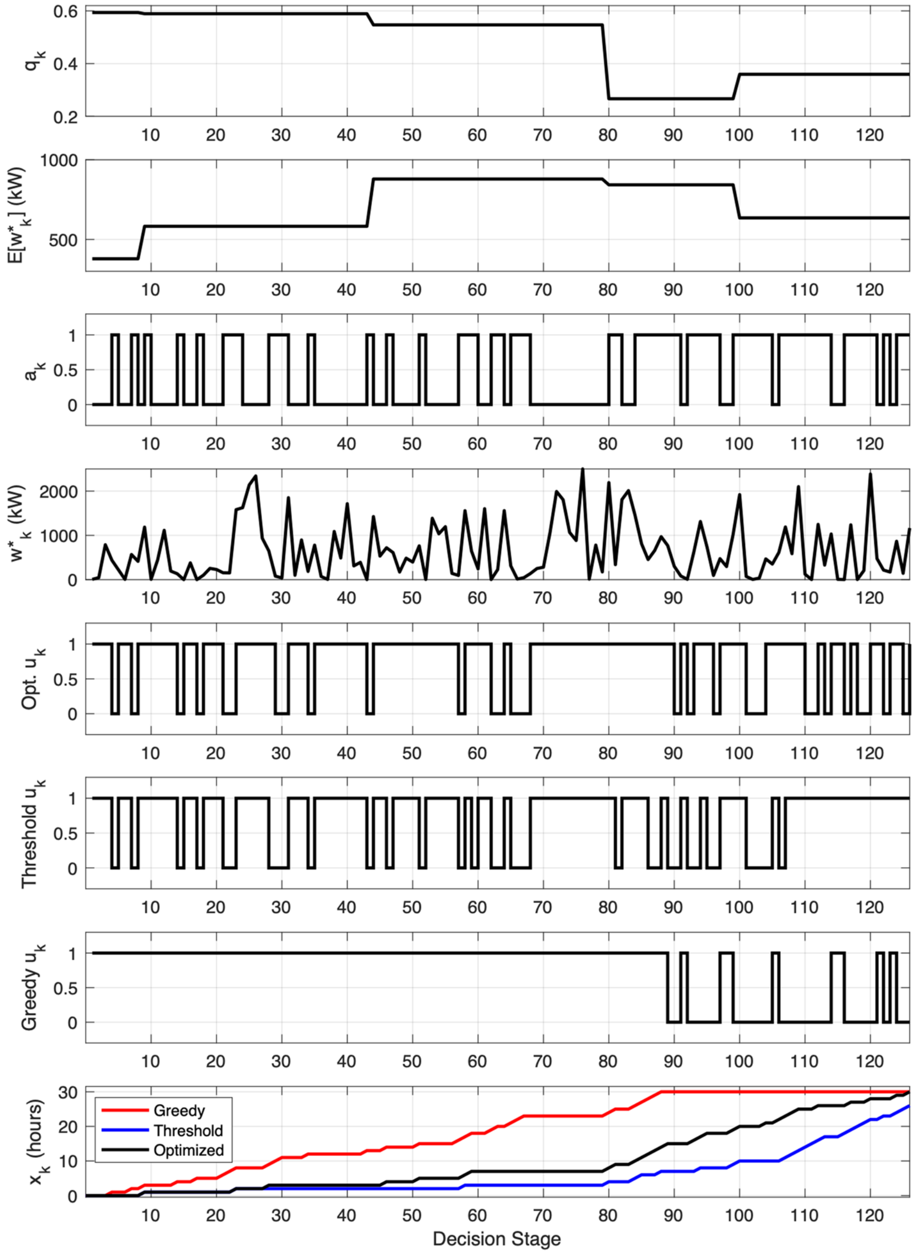

FIGURE 12 Curtailment results for real-world data example [Colour figure can be viewed at wileyonlinelibrary.com]

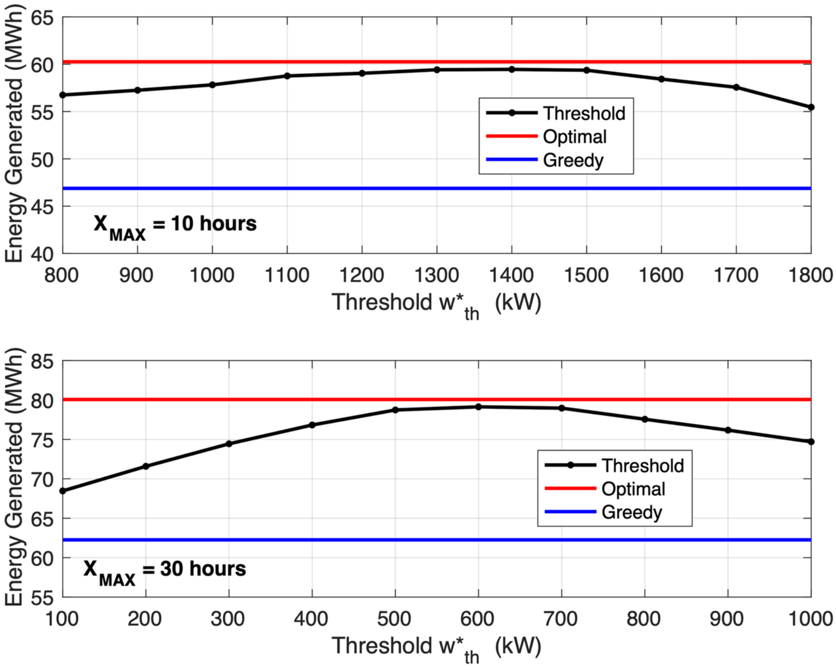

FIGURE 13 Average demand-weighted energy generated for monte carlo simulations at various w_{th}^* values [Colour figure can be viewed at wileyonlinelibrary.com]

turbine during periods of lower demand-weighted power production since there are few remaining stages left. This highlights one of the significant benefits of the optimal algorithm compared with the threshold scheme. Note that the threshold algorithm does attempt to use all the remaining hours during the final stages, but the presence of clouds during some of the final stages means that it does not use the entire budget.

Monte Carlo studies were performed to assess performance trends across numerous simulated cases. One Monte Carlo simulation was performed at each of an array of threshold values w_{th}^*, with 1000 simulations performed in each case. This was repeated for X_{MAX} = 30 hours and X_{MAX} = 10 hours. The average demand-weighted energy produced for each Monte Carlo simulation is shown in Figure 13. Because the greedy and optimal results are independent of the w_{th}^* used, the red and blue lines represent the results averaged over all the Monte Carlo runs for the particular value of X_{MAX}. Several aspects of Figure 13 are important to mention. First, regardless of the value of X_{MAX}, the optimal algorithm produces more energy on average compared with the other two schemes, even at the best value of w_{th}^*. The advantage over the greedy algorithm is significant, while the improvement over the threshold algorithm depends heavily on the value of w_{th}^*. Also, note that the best value of w_{th}^* is highly dependent on the flicker hour budget X_{MAX}, or more precisely, the ratio of X_{MAX} to the number of stages N. When the ratio X_{MAX}/N is very low, w_{th}^* should be set higher to make the threshold algorithm more conservative. As the ratio X_{MAX}/N gets larger, w_{th}^* should be increased to allow the threshold algorithm to operate the turbine more often.

Overall, the results in this section highlight the benefit of the optimal algorithm as well as the interesting tradeoffs that arise when considering the curtailment optimization problem. In all cases, the threshold and optimal algorithms are shown to generate significantly more energy than the greedy approach. The tradeoff between the threshold and optimal algorithms is more nuanced. While the optimal algorithm relies on forecast data, it essentially adapts its threshold for curtailing the turbine based on the number of stages remaining, the number of hours used, and predicted conditions at the remaining stages. Conversely, the threshold-based algorithm uses a fixed threshold and thus suffers in cases where demand-weighted power is unexpectedly low, but otherwise remains largely insensitive to forecast errors. Results show that as long as high-level forecasting of wind speed trends and energy demand remain somewhat accurate, the optimal algorithm provides better performance on average compared with the other two schemes.

5 | CONCLUSION

The problem of environmental curtailment optimization has been formulated, wherein wind turbine curtailment decisions must be made in order to satisfy particular environmental constraints. Recognizing that the problem satisfies the so-called principle of optimality, the optimal control solution was derived through the use of dynamic programming. This optimal algorithm uses forecast data for wind speed, energy demand, and other environmental factors to compute a table of expected rewards. This table is used online in conjunction with measurements of current conditions to arrive at an optimal curtailment decision. Two alternative control schemes were introduced for comparison—a threshold-based scheme, and a greedy scheme. Simulation results showed that the optimal algorithm provides better performance on average compared with the other two schemes but exhibits some sensitivity to forecast accuracy. While the examples in this work centered on shadow flicker curtailment, the algorithm can be easily extended to other types of environmental curtailment involving wildlife or damage mitigation to life-limited components. This can be accomplished through appropriate definition of the decision stages, reward activation and state transition activation vectors, and reward values for a particular problem. The algorithm presented here therefore offers a promising technique to address a wide range of wind turbine curtailment scenarios.

ACKNOWLEDGEMENTS

The author would like to thank Geoffrey Klise, Brian Naughton, David Maniaci, Josh Paquette, and others at the Sandia National Laboratories Wind Energy Technology Department for their insightful and interesting comments regarding this work.

NOMENCLATURE

$\vec{a}_k$	state transition activation vector at stage *k*
N	number of stages/curtailment decisions
p_k	probability of clear conditions at stage *k*
q_k	cloudiness probability at stage *k*
r_k	reward at stage *k*
u_k	curtailment control at stage *k*
w^*_{th}	threshold demand-weighted power for turbine operation
w^*_k	demand-weighted power produced at stage *k*
$\vec{w}_k$	reward activation vector at stage *k*
$\vec{x}_k$	state vector at stage *k*
$\vec{X}_{MAX}$	environmental constraint on system state

ORCID

Jonathan Rogers https://orcid.org/0000-0001-7196-1691

REFERENCES

1. McGlinchey D, Caparossi S. *A guide to drafting wind turbine regulations*. Plymouth, MA: Manomet Center for Conservation Sciences, September; 2013.
2. Arnett E, Huso M, Schirmacher M, Hayes J. Altering turbine speed reduces bat mortality at wind-energy facilities. *Front Ecol Environ*. 2011;9(4): 209-214.
3. Baerwald E, Edworthy J, Holder M, Barclary R. A large-scale mitigation experiment to reduce bat fatalities at wind energy facilities. *Journal of Wildlife Management*. 2010;73(7):1077-1081.
4. Anonymous, "Rhode Island land-based wind siting guidelines," Rhode Island Office of Energy Resources, Providence, RI, January 2017.
5. Fine S, D'Costa P, Kumaraswamy K. Policies for Accommodating Higher Penetration of Variable Energy Resources (VERs) – U.S. Outlook and Perspectives. In: Jones L, ed. *Renewable Energy Integration*. Amsterdam, The Netherlands: Elsevier; 2017.
6. Burke D, O'Malley M. Factors influencing wind energy curtailment. *IEEE Transactions on Sustainable Energy*. 2011;2(2):185-193.
7. Wu W, Chen J, Zhang B, Sun H. A robust wind power optimization method for look-ahead power dispatch. *IEEE Transactions on Sustainable Energy*. April 2014;5(2):507-515.
8. Chen P, Siano P, Bak-Jensen B, Chen Z. Stochastic optimization of wind turbine power factor using stochastic model of wind power. *IEEE Transactions on Sustainable Energy*. April 2010;1(1):19-29.
9. Cheng W, Zhang H. A dynamic economic dispatch model incorporating wind power based on chance constrained programming. *Energies*. 2015;8: 233-256.
10. Byon E, Ntaimo L, Ding Y. Optimal maintenance strategies for wind turbine systems under stochastic weather conditions. *IEEE Transactions on Reliability*. June 2010;59(2):393-404.
11. Bradley S, Hax A, Magnanti T. *Applied mathematical programming*. Boston, MA: Addison-Wesley; 1977, Ch:11.
12. Anonymous, "Planning Policy Statement 18: renewable energy," Northern Ireland Department of the Environment, Belfast, UK, 2007.
13. Anonymous, "Update of UK shadow flicker evidence base," Parsons Brinckerhoff, Newcastle Upon Tyne, United Kingdom.
14. Vestas Wind Systems A/S, "Vestas Shadow Detection System," Aarhus, Denmark, 2017.
15. Marks P. "Smart sensors stop flickering wind turbines," *New Scientist*, 12 October2011, https://www.newscientist.com/article/mg21228346-300-smart-sensors-stop-flickering-wind-turbines/.
16. National Centers for Environmental Information, "Comparative climatic data," National Oceanic and Atmospheric Administration," Accessed 6 August 2019, https://www.ncdc.noaa.gov/ghcn/comparative-climatic-data.
17. Aleasoft Energy Forecasting, "Demand and renewable energy forecasting," Accessed 6 August 2019, https://aleasoft.com/energy-demand-forecasting/.
18. Bellman R. On the theory of dynamic programming. *Proc Natl Acad Sci*. 1957;38:716-719.
19. Hayes M, Hooton L, Gilland K, Grandgent C. A smart curtailment approach for reducing bat fatalities and curtailment time at wind energy facilities. *Ecol Appl*. 2019;29(4):1-18.
20. Persimia, LLC., "Our services: Shadow flicker modeling," Accessed 6 August 2019, https://persimia.com/how-it-works.
21. Manwell J, McGowan J, Rogers A. *Wind energy explained*. Hoboken, NJ: John Wiley and Sons; 2010.

22. Equinox Project, “Energy use: Residential electricity consumption,” Center for Sustainable Energy, Accessed 6 August 2019, http://sites.energycenter.org/equinox/dashboard/residential-electricity-consumption.

How to cite this article: Rogers J. Optimal strategies for wind turbine environmental curtailment. *Wind Energy*. 2020;1–20. https://doi.org/10.1002/we.2489