MDP Formulation for Optimal Wind Turbine Placement

Lisa Fu
Department of Computer Science
Stanford University
Stanford, USA
lfu2@stanford.edu

Tara Iyer
Department of Computer Science
Stanford University
Stanford, USA
tiyer@stanford.edu

Sharman Tan
Department of Computer Science
Stanford University
Stanford, USA
sharmant@stanford.edu

***Abstract*—Wind turbines are vital to green power production, and strategically placing wind turbines in wind farms can significantly affect the output and performance of wind farms. For our project, we focus on optimizing wind turbine placement to yield the highest reward, where reward is defined in terms of the power generated by the wind farm. Specifically, we model this problem as a Markov Decision Process (MDP) and find the optimal placement of a set number of active wind turbines in grid representing a wind farm of a set size. We run our MDP on an array of different algorithms (Value Iteration, Value Iteration GS (Gauss Seidel), Finite Horizon, Relative Value Iteration, Policy Iteration, Policy Iteration Modified, and Q-Learning) under three different reward scenarios (Normal, Corrective Maintenance, Preventative Maintenance). Overall, we find that in general, our MDP solvers place active wind turbines at optimal locations corresponding to those with high wind speeds. However, this is not true when we take into account corrective maintenance costs, suggesting that how we account for variables such as turbine condition and wind speed in the reward function may significantly impact our MDP solvers' optimal turbine placements.**

I. Introduction

Wind turbines are an important method of green power production, and strategic organization of wind turbines can greatly impact the power productivity and output of wind farms. Therefore, optimization strategies are vital as we seek to decrease our reliance on coal-powered energy and increase the efficiency of green sources of energy production. [4]

A number of factors contribute to the efficiency of a wind farm, and the two we are examining are wind turbine placement (its physical location in a geographical space), and maintenance (induced wind turbine inactivity due to maintenance). Both factors stem from sources of uncertainty: wind turbine placement is affected by the unforeseeable variability of wind, or the presence of animals and natural habitats, or neighboring turbine and/or radar signature interference. Wind turbine maintenance may be caused by natural disasters, construction material integrity, or wear and tear over time.

In this project, we take an exploratory approach to modeling wind turbine placement and maintenance by formulating it as a Markov Decision Process (MDP) and running various algorithms to solve the MDP in various maintenance cost conditions. Our ultimate goal is to understand notable patterns or trends in wind turbine placement in order to better inform current and future development of wind farms.

II. Problem

A. Relevant Literature

Optimal Wind Turbine Placement via Randomized Optimization Techniques [8] tackles the same task of optimal wind turbine placement using a combination of genetic algorithms and Markov Chain Monte Carlo methods. A survey of the literature relevant to our task suggests that genetic algorithms have been common in wind turbine placement research in general. However, we tackle this task using without genetic algorithms, instead deciding to model our problem as an MDP.

Wind Turbine Placement Optimization by means of the Monte Carlo Simulation Method [2] models the problem of optimal wind farm turbine placement using the Monte Carlo simulation method. The inputs to the problem were wind data, dominant wind direction, and wind intensity, and the paper finds the optimal placement of 30 wind turbines in a wind park equally divided into 100 cells. Similarly, our project uses wind data to find the optimal placement of some set number of wind turbines in a wind park consisting of some set number of cells. Due to computational constraints from working with large sample spaces, we tackle this same problem in a much smaller scale, as we discuss later.

Season-Dependent Condition-Based Maintenance for a Wind Turbine Using a Partially Observed Markov Decision Process [3] addresses the problem of maintaining wind turbines in a season-dependent and condition-based manner by modeling the problem as a Partially Observed Markov Decision Process (POMDP). Although this paper specifically focuses on maintenance of wind turbines whereas we focus on optimal placement of wind turbines, this paper was important in helping us understand and model our problem. The paper introduces important variables that influence the cost and rewards of wind turbines (e.g. turbine condition, season) as well as different actions (minor or major preventative or corrective maintenance) that can optimize these costs in the long term. Although we do not model our problem in such detail to account for all these variables, this paper introduces the importance of changing seasonal conditions on wind turbines' performance and ways we can improve our MDP approach by considering more variables moving forward.

Optimal Planning and Learning in Uncertain Environments for the Management of Wind Farms [5] directly responds to the work of [3] by addressing the main limitations of POMDPs: fixed state transitions and observations. The paper notes that in reality, these state transitions and observations are subject to a significant amount of uncertainty. Therefore, [5] utilizes the framework of the Bayes-adaptive POMDP (BA-POMDP) to treat these transition probabilities as random variables that are updated using the data. This paper introduces planning and learning in uncertain dynamic systems (PLUS) within a BA-POMDP to incorporate the conditional probabilities of wind turbine degradation and damage as uncertain and flexible transition probabilities. While this paper specifically brings up these considerations in the context of POMDPs, its arguments also apply to our MDP and suggest important areas for improvement.

B. Problem Formulation

We formulate the problem of optimal wind turbine placement as an MDP defined with the following state space, action space, transition probabilities, and rewards.

1. State Space

Each state in our MDP is a set of turbine placements in a grid representing a section of a wind farm. Each cell of this grid represents a single possible turbine placement, and each turbine can be either active (1) or inactive (0). An active wind turbine is a turbine that is producing energy. An inactive turbine is a turbine that is not producing energy, possibly due to maintenance or power curtailment. We assume that every cell in our grid may contain a turbine, but only a set number m of these turbines may be active at once.

Due to computational constraints from our large state space, we define our problem as one section of a wind farm that is represented by a grid of size

3×5; our task is to decide which $m = 2$ of these 15 possible turbine placements are active during a given timestep. Note that the space complexity of this problem increases drastically as we increase the grid size or the possible number of active turbines. In the example of a 3×5 grid with $m = 2$ active turbines, we would have ${}_{15}C_2 = 105$ states in our state space.

To minimize the computational complexity of our implementation, we leverage numpy arrays to create a "flattened" representation of each state in the state space. For instance, consider the flattened states [0, 1] and [3, 4]. Each integer represents a flattened grid index, such that turbine index 0 = (0,0), turbine index 1 = (0,1), turbine index 3 = (0,3), and so on. On a 3×5 grid, the expanded grid representation of the state [0, 1] would look like the following:

$$\begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

In this example, the complete state space consists of all possible combinations of $m = 2$ active turbines in a grid of 3×5 turbines. As described previously, in this case the state space would be of size ${}_{15}C_2 = 105$, and each state would be represented as an integer array of length 2 where each integer is the flattened representation of grid coordinates (i, j) where $0 \leq i \leq 2$ and $0 \leq j \leq 4$ for a grid of size 3×5.

2. Action Space

At each timestep, a turbine can either change its state or remain in the same state that it was in at the previous timestep. Effectively, this results in the following four cases:

1) A turbine that was inactive at one timestep can become active at the next timestep
2) A turbine that was active at one timestep can become inactive at the next timestep
3) A turbine that was active at one timestep can remain active at the next timestep
4) A turbine that was inactive at one timestep can remain inactive at the next timestep

Because our main constraint is that m turbines in the grid may be active at once, we simplify this problem by making our action space identical to our state space. For instance, if the grid representation of our current state is

$$\begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

and the grid representation of our action is

$$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

then our next state is precisely the action we took. This action encapsulates all changes that occurred to get from our original state to the next state (our current action). In this case, this means making no changes to the turbine at position $(0, 0)$ by leaving it active, changing the turbine at position $(0, 1)$ to be inactive, and changing the turbine at position $(0, 2)$ to be active, and leaving all other turbines as is.

3. Transition Probabilities

The transition probabilities are straightforward given our previously defined state and action spaces. After all, given the current state and action, the next state is deterministic; the next state is precisely the action we took.

Note that because our transition to the next state is always deterministic given our action,

$$T(s'|s, a) = \begin{cases} 1 & s' = a \\ 0 & otherwise \end{cases}$$

Examples of $T(s'|s, a)$ are as follows, where we refer to s, a, and s' using their flattened rather than grid representations:

- $s = [0, 1], a = [2, 3], s' = [2, 3] \rightarrow T(s'|s, a) = 1$.
- $s = [0, 1], a = [2, 3], s' = [1, 2] \rightarrow T(s'|s, a) = 0$.

Therefore, in our implementation we represent our transition probabilities as a matrix of dimension $|A| \times |S| \times |S|$, where A corresponds to our action space and S corresponds to our state space. Each action $a \in A$ maps to a $|S| \times |S|$ matrix, where element (i, j) in this matrix corresponds to $T(s' = j|s = i, a)$. All values in this matrix are 0 except for the column j where $j = a$ (this column is filled with 1s).

For instance, for a given action a corresponding to index 0 in the transition probability matrix, if we have 5 states in our state space S, the $|S| \times |S|$ transition probability matrix for action a is as follows:

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{matrix}$$

4. Reward Function

In our model, the reward at some state represents the amount of money in U.S. dollars earned from power generation by the wind turbines in that state during one timestep. We define one timestep to be 1 week (168 hours), as this is a reasonable time frame over which to run active turbines continuously before turning some turbines on or off. To calculate the reward for an entire state, we compute the reward generated by each turbine in the state, and save these values in a matrix (our reward matrix). The reward for an individual turbine is the sum of three components:

1) The amount of revenue generated from selling the power produced by the turbine to consumers. We calculated this using the following pieces of information:
   - The *power curve* for the Vestas V80-1.8 wind turbine, which is one off the turbines that is installed at multiple sites in Northern California according to the U.S. Department of Energy's Wind Turbine Data Set [7]. This power curve provides power output in kilowatts per hour for the Vestas turbine at different wind velocity ranges.
   - The *average wind velocity* at the site of the turbine. To calculate this, we leveraged wind velocity data from the U.S. Western Wind Dataset [6] from the National Renewable Energy Laboratory. This dataset provides minimum and maximum expected wind velocities across the Western United States, and we were able to use it to generate average wind velocities for a region in Northern California that contains a sizeable wind energy farm.
   - The *price* per kilowatt-hour of electricity in California
   - The *number of hours* over which to calculate revenue generation (i.e., the number of hours in one timestep)
2) The operational cost of switching a wind turbine on and off (only in cases when a turbine changes state from active $\rightarrow$ inactive or inactive $\rightarrow$ active). We experimented with different values for this cost, as it was difficult to find reliable information that quantified this cost.
3) The cost of performing corrective maintenance or preventative maintenance on an inactive turbine (since maintenance is one of the largest causes of wind turbines being shut off). We experimented with different values for the cost of corrective and preventative maintenance, and used the figures presented in [3] as a starting point. We added the cost of performing corrective maintenance and preventative maintenance during separate runs, so that we could compare the optimal policies generated when all maintenance is done correctively (i.e., only when a turbine breaks) vs. when all maintenance is done preventatively (i.e., when a turbine is expected to break, but it hasn't broken yet). We also performed some runs without adding any additional cost for PM or CM, so that we could compare the policy generated by this run to runs that accounted for PM and CM (described in more detail in the Experimental Methodology section).

III. Experimental Methodology

A. Datasets

We used two datasets to model our problem. The *U.S. Wind Turbine Dataset* [7] contains data on U.S. wind turbine placements and characteristics of wind turbines (such as physical specifications and power output) and the *Wind Integration National Dataset* [6] contains data on wind velocity, wind power production, wind resources, infrastructure, and site analyses across the continental United States. As described in the previous section, we utilized these datasets when calculating the reward for each state (since reward is based on power output, which in turn depends on a wind turbine's specifications and the wind velocity range in a particular geographic region).

B. MDP Toolbox

We used the MDP Toolbox: *mdp* module [1] for Python to formulate and run algorithms against our MDP. Once installed, this module provides functions for performing value iteration, policy iteration, and Q-learning on a pre-defined MDP. In order to use these functions, we had to construct matrices to represent our states, actions, transitions, and rewards. This requirement partially motivated the MDP structure that we described earlier (in which the states, actions, transitions, and rewards are all represented in matrix form).

C. Experiments

After defining the MDP using the Python *mdp* module, we ran the MDP under various state and action space constraints, with the following algorithms: Value Iteration, Value Iteration GS (Gauss Seidel), Finite Horizon, Relative Value Iteration, Policy Iteration, Policy Iteration Modified, and Q-Learning. We were able to run three different "scenarios", with different reward functions depending on the specification of turbine inactive state. They are named and described in high-level below:

1) "Normal": when a turbine is active, it generates revenue based on power output. When a turbine is inactive, the only cost is the amount of lost revenue, and any operational cost of turning a turbine off.
2) "Corrective Maintenance (CM)": when a turbine is active, it generates revenue based on power output. When a turbine is inactive, it accrues a penalty that is relatively large (in addition to any operational cost), to reflect the high cost of having to undergo corrective maintenance [2], or maintenance that may require a significant amount of time and resources.
3) "Preventative Maintenance (PM)": when a turbine is active, it generates revenue based on power output. When a turbine is inactive, it accrues a small penalty (in addition to any operational cost), to reflect the minimal cost of doing preventative maintenance [2].

Having these three distinctions will be interesting to explore how the MDP solvers may choose optimal policies given the scenarios, especially since turbine inactivity due to maintenance in the real world is rarely a predictable occurrence and may require additional foresight and data analytics to help wind farm managers draw conclusions on how they should place their wind turbines.

IV. Data and Results

A. Optimal Action Frequencies and Connectivity

We started with a qualitative approach to visualize and reason about optimal policies given the various algorithms we used to solve our MDP. With a small enough state space (in this case we used ${}_5C_2$, where 5 is the total number of available turbine placements, and we are limited to choosing 2 turbine locations as "active"), all of the algorithms converged to the same optimal policy. We compare the optimal policy across our three different scenarios.

Below are graphs generated using Value Iteration:

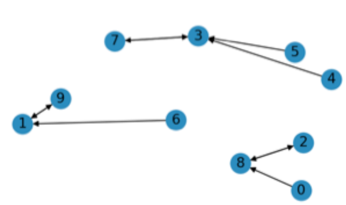

Fig. 1. Value Iteration for Normal Scenario.

It is interesting to see how clusters begin to form in the graphical representation of the optimal policy, where multiple states converge to the same action and next state. Qualitatively, this may denote that there are certain actions that are more favorable than others, and we are able to sort the optimal policy into a frequency count instead, and analyze the most frequent state transition as the final optimal state for turbine placements.

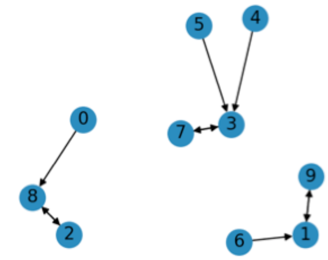

Fig. 2. Value Iteration for CM Scenario.

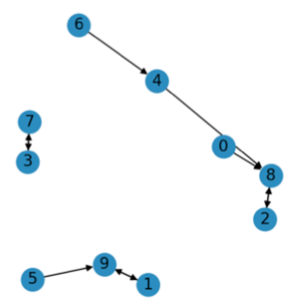

Fig. 3. Value Iteration for PM Scenario.

B. Optimal Policy for ${}_5C_2$

We needed to limit our state space due to memory and runtime constraints, so started off with analyzing the optimal policy for a ${}_5C_2$ state and action space.

Below are the most frequent actions taken by all the algorithms, sorted by descending frequency from left to right. The tables show the top frequent actions by scenario, and highlighted in green is the most common frequent action across all algorithms (and those highlighted in yellow are second common, and highlighted in red are uncommon).

Beneath each table is a graphical representation of the most frequent action, to show visually what the optimal placement looks like in grid form, juxtaposed with the actual wind speed data at each location.

Value Iteration	3	8	1	9	7	2
Value Iteration GS	3	8	2	9	7	1
Finite Horizon	3	8	1	9	7	2
Relative Value Iteration	3	8	1	9	7	2
Policy Iteration	3	8	7	9	2	1
Policy Iteration Modified	8	7	9	3	2	1
Q-Learning	3	5	8	9	7	2
Most Frequent (Left to Right)	**3**	**8**	**1**	**9**	**7**	**2**

Fig. 4. Table of most frequent actions by solver, for "Normal" scenario.

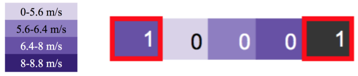

Fig. 5. Visualization of most frequent action (a=3) for Fig. 4.

Value Iteration	6	4	5
Value Iteration GS	6	4	5
Finite Horizon	6	4	5
Relative Value Iteration	6	4	5
Policy Iteration	6	4	5
Policy Iteration Modified	6	4	5
Q-Learning	5	6	4
Most Frequent (Left to Right)	**6**	**4**	**5**

Fig. 6. Table of most frequent actions by solver, for "CM" scenario.

Fig. 7. Visualization of most frequent action for Fig. 6.

Value Iteration	8	9	7	4	3	2	1
Value Iteration GS	8	9	7	4	3	2	1
Finite Horizon	8	9	7	4	3	2	1
Relative Value Iteration	8	9	1	7	3	2	N/A
Policy Iteration	8	9	7	4	3	2	1
Policy Iteration Modified	8	9	7	4	3	2	1
Q-Learning	9	0	8	1	5	N/A	N/A
Most Frequent (Left to Right)	**8**	**9**	**7**	**4**	**3**	**2**	**1**

Fig. 8. Table of most frequent actions by solver, for "PM" scenario.

Fig. 9. Visualization of most frequent action for Fig. 8.

C. Optimal Policies for ${}_{15}C_x$

After running the MDP solvers for a ${}_5C_2$ state and action space, we expanded to both ${}_{15}C_2$ and ${}_{15}C_3$ state/action spaces, to see if the MDP solvers were still able to find optimal solutions given the added complexity of additional space. Below are the visual representations of the most frequent action across all solvers for the respective state and action spaces, and for each scenario. The shading of the grid indicates the velocity of wind speed at each location.

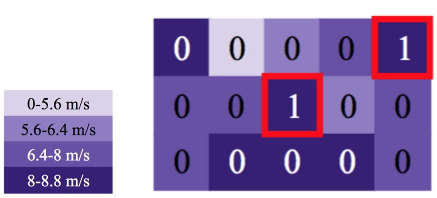

Fig. 10. Most frequent action for ${}_{15}C_2$, same for "Normal" and "PM" scenarios.

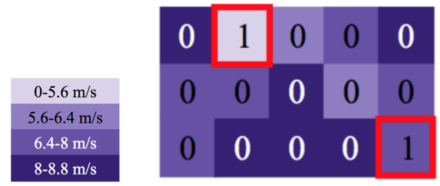

Fig. 11. Most frequent action for ${}_{15}C_2$, "CM" scenario.

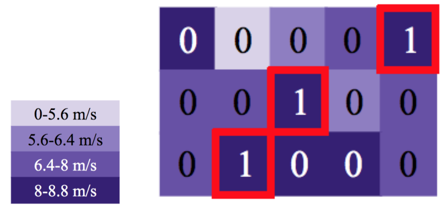

Fig. 12. Most frequent action for ${}_{15}C_3$, same for "Normal" and "PM" scenarios.

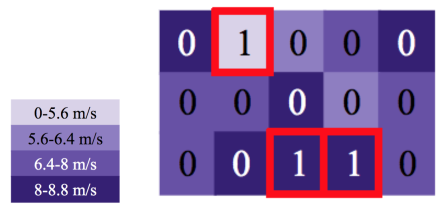

Fig. 13. Most frequent action for ${}_{15}C_3$, "CM" scenario.

V. Discussion and Conclusion

A. Notable Trends

From our various experiments, we were able to draw some insights regarding the strategic placement of wind turbines.

1) Optimal Placements at High Wind Speeds: In general, the MDP solvers all choose optimally in the "Normal" scenario, meaning they all place active wind turbines in the cells with the highest wind speeds. This trend makes sense, as high wind speeds result in higher power outputs and thus greater reward.

2) "Normal" & "PM" Scenarios vs. "CM" Scenario: Notice that from Figures 10 and 12, we observe that the MDP solvers perform similarly for the "Normal" and "PM" scenarios for the ${}_{15}C_2$ and ${}_{15}C_3$ experiments, respectively. In both of these scenarios, the solvers choose active turbines optimally by choosing the turbines in high wind speed locations. However, note that from Figures 11 and 13 for the ${}_{15}C_2$ and ${}_{15}C_3$ experiments, respectively, the "CM" scenario selects active turbines sub-optimally in terms of wind speed.

This makes sense intuitively, as the "Normal" scenario should determine the optimal wind turbine placement based on wind speed, and the "PM" scenario should assign a small penalty when turbines must undergo preventative maintenance. If this penalty is negligable, the "PM" scenario's optimal turbine placements should not differ much from those of the "Normal" scenario. However, the "CM" scenario assigns a significantly higher penalty when turbines must undergo corrective maintenance, meaning that wind speed may not be the determining factor in this scenario when optimizing turbine placement.

Although our experiments do not definitively suggest the reasoning behind the optimal placements in the "CM" scenario, we understand why this difference exists between the "Normal" & "PM" scenarios and the "CM" scenario.

B. Future Work

There are several ways in which we could extend this project to more effectively model wind turbine placement and maintenance policies.

1) POMDP Representation: To begin, we could turn our model into a POMDP which could account for observations based on wind turbine sensor data. This is a more realistic representation of how anomalies and failures are actually detected in wind turbines: in practice, sensors capture information about wind turbine failures and alert wind turbine operators to the need for maintenance. However, sensors are not always accurate, and can sometimes incorrectly report a turbine anomaly or failure (either by reporting a failure when there isn't one, or by not reporting a failure when one occurs). A POMDP representation would allow us to use a more sophisticated transition function that captures the probabilities of false positives and false negatives in sensor data.

It would also enable us to encode for different degrees of failure (i.e., a minor failure requiring minimal maintenance work vs. a major failure requiring significant maintenance work). Coding for different degrees of failure would allow us us to construct a more sophisticated reward function that takes the cost and time of required maintenance work into account when computing the net reward generated at each timestep. With a more sophisticated reward function, we could then compare interesting tradeoffs between various maintenance policies, such as performing a lot of preventative maintenance up-front versus handling maintenance issues only when a wind turbine breaks.

2) Additional Variables: Apart from leveraging a POMDP representation to better model wind turbine failures, it could be beneficial to take variables like seasonality and wind angle into account when constructing our model.

Seasonality is important because the rates of wind turbine failures (and likely of sensor accuracy) change with the seasons, particularly in geographic regions with dramatically different weather in different seasons. Areas with harsh winters are especially interesting, as severe winter weather conditions can adversely affect wind turbine performance.

Wind angle is an interesting factor to consider when comparing different placements of turbines: the angle at which a wind turbine's blades face the wind can have a pronounced affect on the turbine's power output, and the draft from nearby turbines can affect the wind velocity and power output at a particular turbine. These factors are quite complex to model, but could prove useful in constructing a model that more accurately represents how wind turbines operate in practice.

Acknowledgment

We would like to thank the CS238/AA228 teaching staff for their helpful guidance and support throughout our project.

Distribution of Work

We split up the coding and writing evenly among the three of us. We did a significant portion of the work simultaneously (with more than one person working on a piece of code or part of the paper at the same time).

References

[1] Markov Decision Process (MDP) Toolbox. URL: https://pymdptoolbox.readthedocs.io/en/latest/api/mdptoolbox.html.

[2] Sebastian Brusca, Rosario Lanzafame, and Michele Messina. Wind turbine placement optimization by means of the monte carlo simulation method. *Modelling and Simulation in Engineering*, 2014:35, 2014.

[3] Eunshin Byon and Yu Ding. Season-dependent condition-based maintenance for a wind turbine using a partially observed markov decision process. *IEEE Transactions on Power Systems*, 25(4):1823–1834, 2010.

[4] Adam Chehouri, Rafic Younes, Adrian Ilinca, and Jean Perron. Wind turbine design: multi-objective optimization. *Wind Turbines: Design, Control and Applications*, page 121, 2016.

[5] Milad Memarzadeh, Matteo Pozzi, and J Zico Kolter. Optimal planning and learning in uncertain environments for the management of wind farms. *Journal of Computing in Civil Engineering*, 29(5):04014076, 2014.

[6] U.S. Department of Energy. Western Wind Data Set. URL: https://www.nrel.gov/grid/western-wind-data.html.

[7] Office of Energy Efficiency and Renewable Energy. Massive U.S. Wind Turbine Data Set Released for Public Use. URL: https://www.energy.gov/eere/wind/articles/massive-us-wind-turbine-data-set-released-public-use.

[8] John Tzanos, Kostas Margellos, and John Lygeros. Optimal wind turbine placement via randomized optimization techniques. In *Proceedings of the 17th Power Systems Computation Conference, Stockholm, Sweden*, pages 22–26, 2011.