Robustifying machine-learned algorithms for efficient grid operation

Nicolas Christianson
California Institute of Technology
Pasadena, CA, USA
nchristi@caltech.edu

Christopher Yeh
California Institute of Technology
Pasadena, CA USA
cyeh@caltech.edu

Tongxin Li
The Chinese University of Hong Kong (Shenzhen)
Shenzhen, Guangdong, China
litongxin@cuhk.edu.cn

Mahdi Torabi Rad
Beyond Limits
Glendale, CA, USA
mtorabi@beyond.ai

Azarang Golmohammadi
Beyond Limits
Glendale, CA, USA
agolmohammadi@beyond.ai

Adam Wierman
California Institute of Technology
Pasadena, CA USA
adamw@caltech.edu

Abstract

We propose a *learning-augmented* algorithm, ROBUSTML, for operation of dispatchable generation that exploits the good performance of a machine-learned algorithm while providing worst-case guarantees on cost. We evaluate the algorithm on a realistic two-generator system, where it exhibits robustness to distribution shift while enabling improved efficiency as renewable penetration increases.

1 Introduction

The need to reduce greenhouse gas emissions to mitigate the impacts of anthropogenic climate change is driving an energy transition characterized by large amounts of renewable generation resources being added to the grid. During this transition, the variability of solar and wind energy will require the operation of dispatchable generation to balance out the fluctuations in renewable energy production and maintain reliable grid operation. However, conventional fossil fuel generators incur significant added costs from the frequent cycling and ramping they must perform under high penetration of renewables, due both to decreased fuel efficiency and increased operations/maintenance required from operating in this regime [29]. Moreover, most dispatchable resources are limited in their ramp rate, and thus under high penetration of renewables they must be operated in a manner that anticipates system ramp needs, taking into account the high costs of frequent ramping while still meeting demand.

Operating generation optimally requires minimizing fuel costs while taking account of intertemporal coupling of decisions, including both ramp costs and ramp limits. A natural approach for this problem is model predictive control (MPC), an algorithm that utilizes near-term forecasts of demand and other conditions to choose decisions that minimize aggregate cost over a fixed lookahead horizon [20]. In addition to theoretical work confirming its good performance [16], MPC works well in practice and has been studied in a number of energy and sustainability-related domains, including control of wind turbines and solar photovoltaics [17, 28], smart buildings [11, 3], and energy storage [19, 21]. Moreover, several regional power system operators in the US use MPC to settle the real-time electricity market [7], and it is widely understood that such lookahead algorithms will play an

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2022.

increasingly important role in enabling power systems to reliably absorb renewable energy volatility [9, 31].

However, MPC suffers computational complexity exponential in the lookahead horizon if the system model/costs are nonconvex, necessitating the use in practice of heuristic solvers that operate on a faster timescale but generally produce suboptimal decisions [10, 6, 2]. One promising avenue for overcoming the computational complexity of nonconvex MPC to enable improved performance in practice is the development of machine learning (ML) models that imitate its behavior, bypassing the need to solve an independent nonconvex optimization problem to generate each decision. This approach of "learning to control/optimize" has seen wide recent interest in the ML, control, and power systems communities [13, 24, 27, 8, 23, 22]. However, these learning-based approaches come with no *a priori* guarantees on their incurred cost under distribution shift or on out-of-sample problem instances, jeopardizing their performance at deployment time. To counter this potential for poor performance and enable confident deployment of ML proxies for MPC in real-world settings, this work proposes an algorithm to *robustify* the behavior of such an ML proxy. We follow the paradigm of the emerging *learning-augmented* algorithms literature (e.g., [18, 25, 14]), specifically building upon the line of work [1, 26, 5, 15] designing algorithms for *online optimization with switching costs* (a generalization of the dispatch problem) that can exploit the performance of an ML algorithm while providing worst-case guarantees on cost.

Our contributions are twofold: *first*, we propose a learning-augmented algorithm RobustML for online optimization with switching costs that achieves the best deterministic performance bound in the setting of general nonconvex cost functions. Specifically, when provided with an ML algorithm for the problem as well as a heuristic baseline algorithm, for any desired $\epsilon, \delta > 0$, our algorithm achieves cost at most $(1 + \epsilon + \delta)$ times the cost incurred by the ML algorithm, while maintaining a worst-case cost bond of $\mathcal{O}(\frac{\mathrm{C}}{\delta} + \frac{D}{\epsilon})$, where C is the cost of the heuristic baseline and D is the diameter of the decision space. This is the best known tradeoff for deterministic algorithms, as all prior deterministic learning-augmented algorithms paid at least 3 times the cost of the ML algorithm [5, 1]. *Second*, we empirically evaluate the performance of RobustML on a realistic two-generator system under increasing penetration of renewable energy. We find that using a learning-based approach can improve computation time over MPC by 5 orders of magnitude, and our algorithm RobustML ensures robustness to distribution shift while improving cost by ~3% over the heuristic baseline under no distribution shift, with this difference widening under increasing renewable penetration.

Our work has potential for both direct and downstream impact on the problem of climate change. Our results indicate that using RobustML for real-world grid operation could yield modest but tangible efficiency improvements, leading to reduced emissions. Moreover, RobustML's robustness guarantees and lookahead use could enable greater penetration of renewables while maintaining grid reliability. More generally, we see great promise in using learning-augmented algorithms like RobustML to achieve efficiency improvements without sacrificing robustness in other energy and sustainability-related domains where MPC is widely used [17, 28, 11, 3, 19, 21].

2 Model and Preliminaries

We consider the problem of dispatching generation to meet both electricity and steam demand in the presence of variable renewable generation (see Figure 1a for a diagram illustrating the problem). Specifically, we consider an array of several heterogeneous thermal generators, and at each time $t \in \{1, \dots, T\}$, the system operator must choose how much steam and electricity each generator will produce, subject to the constraint that aggregate generation must meet demand in every time interval. Each generator incurs a cost due to fuel consumption, which depends on its production level and environmental factors (temperature, pressure, humidity, etc.), as well as costs due to ramping.

We formulate this problem as an instance of online optimization with switching costs [4]. In an abstract setting, online optimization with switching costs can be considered as a game in which at each time $t \in \{1, \dots, T\}$, a decision-maker receives a vector $\boldsymbol{\theta}_t \in \mathbb{R}^n$ parametrizing a cost function $f(\cdot; \boldsymbol{\theta}_t)$ and then must choose some decision $\mathbf{x}_t \in \mathbb{R}^d$, paying the *hitting* cost $f(\mathbf{x}_t; \boldsymbol{\theta}_t)$ as well as the *switching* cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$ incurred by that decision, where $\|\cdot\|$ is some norm. We assume that the decision $\mathbf{x}_t$ does not impact future parameters $\boldsymbol{\theta}_\tau$ for $\tau > t$. In the context of our application to generation dispatch, $\boldsymbol{\theta}_t$ is a vector containing all ambient factors such as temperature, pressure, humidity, and power/steam demand at time t, $\mathbf{x}_t$ collects the system operator's dispatch decisions

at time t, and f maps ambient conditions and generator dispatches to a fuel cost while penalizing violation of any constraints on the decision. The switching term $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$ acts as the ramp cost. The problem is *online*, so when making a decision $\mathbf{x}_t$ at time t, the decision-maker only has access to the parameters $\theta_1, \dots, \theta_t$ that have been revealed so far. However, the decision-maker may have access to (possibly inaccurate) forecasts $\hat{\theta}_{t+1|t}, \dots, \hat{\theta}_{t+w|t}$ of parameters within a lookahead window of length $w \in \mathbb{N}$. Such forecasts could be obtained using standard ML methods for predicting near-term weather or energy demand.

We consider two standard algorithms for the problem of online optimization with switching costs. The first, GREEDY, is a myopic algorithm that simply chooses the decision $\mathbf{x}_t$ that minimizes $f(\cdot; \theta_t)$ at each time t. This algorithm has worst-case cost guarantees under mild assumptions on the structure of the cost function f [30], and resembles the single-stage dispatch algorithm used widely by power system operators. Its behavior is characterized formally as follows:

$$\text{GREEDY} : \theta_t \mapsto \operatorname*{arg\,min}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}; \theta_t) =: \mathbf{x}_t.$$

That is, GREEDY can be viewed as a function that, when provided with parameter vector $\theta_t \in \mathbb{R}^n$, returns the minimizer of $f(\cdot; \theta_t)$ as a dispatch decision. The second algorithm we consider is model predictive control (MPC). It solves a lookahead optimization problem using near-term predictions of parameters to choose a decision. Formally, at time t, given a (fixed) prior dispatch $\mathbf{x}_{t-1}$, perfect knowledge of the parameter vector θ_t, and forecasts $\hat{\theta}_{t+1|t}, \dots, \hat{\theta}_{t+w|t}$ of parameters over the next w timesteps, MPC chooses its decision as follows:

$$\text{MPC} : \mathbf{\Theta} \mapsto \operatorname*{arg\,min}_{\substack{\mathbf{x} \in \mathbb{R}^d \\ \mathbf{y}_1, \dots, \mathbf{y}_w \in \mathbb{R}^d}} f(\mathbf{x}; \theta_t) + \|\mathbf{x} - \mathbf{x}_{t-1}\| + \sum_{\tau=1}^{w} f(\mathbf{y}_\tau; \hat{\theta}_{t+\tau|t}) + \|\mathbf{y}_\tau - \mathbf{y}_{\tau-1}\| =: \mathbf{x}_t$$

where $\mathbf{y}_0 \coloneqq \mathbf{x}$ and we have stacked the vectors $(\theta_t, \hat{\theta}_{t+1|t}, \dots, \hat{\theta}_{t+w|t})$ into a single entity $\mathbf{\Theta} \in (\mathbb{R}^n)^{w+1}$ for brevity. Note that only the minimizer $\mathbf{x}$ corresponding to the decision made for time t is binding, i.e., the chosen decision $\mathbf{x}_t$ is just the optimal $\mathbf{x}$ in the above optimization; all of the other variables $\mathbf{y}_1, \dots, \mathbf{y}_w$ are ignored after the solution is obtained.

As discussed in the introduction, MPC can be computationally prohibitive if $f(\cdot; \theta)$ is nonconvex. Thus, in our work, we train a machine learning model to approximate the input-output behavior of MPC. That is, given some dataset $\mathcal{D} = \{(\mathbf{\Theta}_i, \mathbf{x}_i)\}_{i=1}^N$ of parameter-decision pairs generated by MPC (*i.e.*, $\mathbf{x}_i = \text{MPC}(\mathbf{\Theta}_i)$), we train a neural network $\text{ML} : (\mathbb{R}^n)^{w+1} \to \mathbb{R}^d$ to minimize the error $\sum_{i=1}^N \|\text{ML}(\mathbf{\Theta}_i) - \mathbf{x}_i\|_2^2$. We seek for ML to approximate MPC well, so that $\|\text{ML}(\mathbf{\Theta}) - \text{MPC}(\mathbf{\Theta})\|_2$ is small in general. However, while we may obtain low empirical error on the training set, this does not guarantee that ML will be a good proxy for MPC on out-of-sample instances or under distribution shift. This motivates the development in the next section of an approach to robustify ML.

Finally, we introduce some notation. For an algorithm ALG producing decisions $\mathbf{x}_1, \dots, \mathbf{x}_T$, define $\text{ALG}_t \coloneqq \mathbf{x}_t$ as ALG's decision at time t, and define $\text{C}_{\text{ALG}}(s, t) \coloneqq \sum_{\tau=s}^t f(\mathbf{x}_\tau; \theta_\tau) + \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|$ as ALG's cost from time s through t. For brevity, we write the total cost as $\text{C}_{\text{ALG}} \coloneqq \text{C}_{\text{ALG}}(1, T)$.

3 Algorithm

We propose in Algorithm 1 (Appendix A) a novel algorithm, ROBUSTML, that robustifies the algorithm ML. It behaves as follows: it starts by following ML's decisions, but if GREEDY is performing well relative to ML and ML surpasses a cost threshold, then ROBUSTML will switch to following GREEDY's decisions (line 8). However, if GREEDY then starts performing worse relative to ML, ROBUSTML will switch back to following ML (line 15). The specific thresholds for switching are determined by the parameters $\epsilon, \delta > 0$, and it is assumed that the decision space has diameter D, so that $\|\text{ML}_t - \text{GREEDY}_t\| \le D$ for all t. Our main analytic result is the following performance bound.

Theorem 1. *The algorithm* ROBUSTML *(Algorithm 1) achieves cost*

$$\text{C}_{\text{ROBUSTML}} \le \min\left\{(1 + \epsilon + \delta)\text{C}_{\text{ML}}, \left(1 + \frac{1+\epsilon}{\delta}\right)\text{C}_{\text{GREEDY}} + \left(1 + \frac{2}{\epsilon}\right)D\right\}.$$

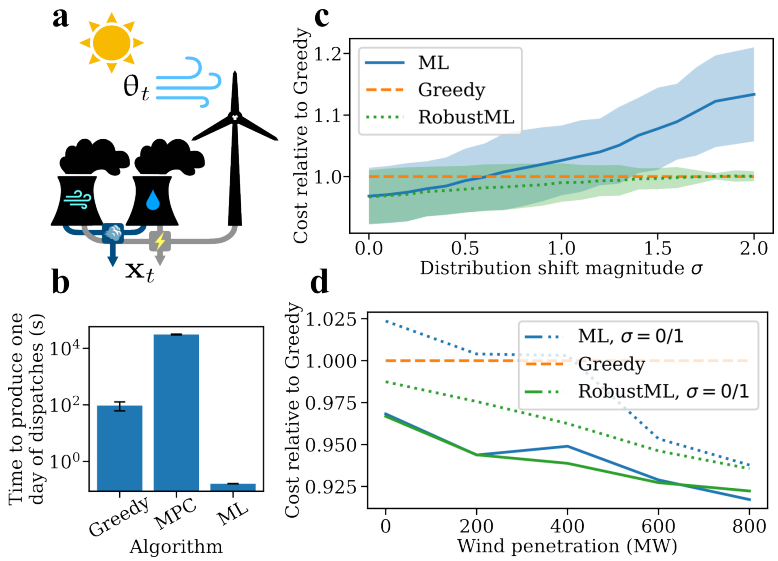

Figure 1: **(a)** A simplified depiction of the power plant setup. In our example, there is an air-cooled unit and a water-cooled unit. At each time t, ambient conditions (temperature, demand, etc.) are aggregated in the vector θ_t, and dispatches are aggregated in the vector $\mathbf{x}_t$. **(b)** Number of seconds (mean and std. dev.) for each algorithm to produce a day's worth of dispatch decisions. **(c)** Cost (mean $\pm$ std. dev.) of each algorithm on the test set under distribution shift, normalized by GREEDY's cost. **(d)** Mean cost of each algorithm (normalized by GREEDY's cost) on the test set under increasing wind penetration and two distribution shift scenarios (solid line is $\sigma = 0$, dotted is $\sigma = 1$).

We prove the theorem in Appendix B. In particular, Theorem 1 tells us that by selecting ϵ, δ arbitrarily small, ROBUSTML can achieve performance arbitrarily close to ML, at the cost of possibly worse performance relative to GREEDY. However, by selecting moderate ϵ, δ, it is possible to trade off exploitation of ML with robustness in cost performance relative to GREEDY.

4 Experimental Results and Discussion

We deploy ROBUSTML with parameters $\epsilon = \delta = 1$ on a small but realistic system with two thermal generators and varying levels of wind generation. The generator models are proprietary and their costs are modeled in a black-box fashion via neural networks. Wind generation data was obtained from the WIND Toolkit [12]. We use a proprietary dataset of 269 days of ambient conditions (temperature, pressure, humidity) and municipal demands for energy and steam on a 15 minute basis. After splitting into training days (200 days) and test days (69 days), we generate a dataset of MPC decisions on the training days with lookahead $w = 12$ using differential evolution, and train a 3-layer neural network as the algorithm ML to imitate the behavior of MPC on this training set.

We begin by examining the performance of ML in comparison to MPC. We find that ML approximates the decisions of MPC well, achieving cost only 0.55% worse on the test set. Moreover, ML is five orders of magnitude faster, producing a day's worth of dispatch decisions in less than a second, while MPC takes upwards of 8 hours, both on 4 virtual CPU cores (Figure 1b).

We next examine the performance of ML, GREEDY, and ROBUSTML on the baseline system (no renewables) when there is distribution shift on the lookahead predictions. That is, we compare the setting of perfect predictions ($\hat{\theta}_{t+1|t} = \theta_{t+1}, \ldots, \hat{\theta}_{t+w|t} = \theta_{t+w}$) to settings with increasing magnitudes of noise σ on the predictions. We show the results in Figure 1c. In particular, we observe that while ML performs better than GREEDY when predictions are good ($\sigma \approx 0$), its performance degrades as the noise grows ($\sigma \to 2$). Nonetheless, ROBUSTML gracefully transitions between the good performance of ML for small σ to matching the performance of GREEDY in the large σ regime. Thus, even though the quality of predictions is unknown *a priori*, ROBUSTML preserves robustness.

We further examine the performance of the algorithms under increasing penetration of wind energy and two distribution shift scenarios ($\sigma = 0$ and 1), displaying the results in Figure 1d. We find that the efficiency improvement of ML over GREEDY widens as wind penetration increases, highlighting the value of using lookahead to increase efficiency under high renewable generation. Moreover, ROBUSTML parallels this improvement while achieving better performance than ML when $\sigma = 1$.

Acknowledgments and Disclosure of Funding

The authors acknowledge support from NSF grants CNS-2146814, CPS-2136197, CNS-2106403, and NGSDI-2105648, Beyond Limits, and Amazon AWS. Nicolas Christianson was supported by an NSF Graduate Research Fellowship (DGE-1745301). Tongxin Li was supported by the start-up funding UDF01002773 of CUHK-Shenzhen.

References

[1] ANTONIADIS, A., COESTER, C., ELIAS, M., POLAK, A., AND SIMON, B. Online metric algorithms with untrusted predictions. In *Proceedings of the 37th International Conference on Machine Learning* (Nov. 2020), PMLR, pp. 345–355. ISSN: 2640-3498.

[2] CAO, G., LAI, E. M.-K., AND ALAM, F. Gaussian process model predictive control of unknown non-linear systems. *IET Control Theory & Applications 11*, 5 (2017), 703–713. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-cta.2016.1061.

[3] CARLI, R., CAVONE, G., BEN OTHMAN, S., AND DOTOLI, M. IoT Based Architecture for Model Predictive Control of HVAC Systems in Smart Buildings. *Sensors 20*, 3 (Jan. 2020), 781. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[4] CHEN, N., COMDEN, J., LIU, Z., GANDHI, A., AND WIERMAN, A. Using Predictions in Online Optimization: Looking Forward with an Eye on the Past. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science* (Antibes Juan-les-Pins France, June 2016), ACM, pp. 193–206.

[5] CHRISTIANSON, N., HANDINA, T., AND WIERMAN, A. Chasing Convex Bodies and Functions with Black-Box Advice. In *Proceedings of Thirty Fifth Conference on Learning Theory* (June 2022), PMLR, pp. 867–908. ISSN: 2640-3498.

[6] DENG, K., SUN, Y., LI, S., LU, Y., BROUWER, J., MEHTA, P. G., ZHOU, M., AND CHAKRABORTY, A. Model Predictive Control of Central Chiller Plant With Thermal Energy Storage Via Dynamic Programming and Mixed-Integer Linear Programming. *IEEE Transactions on Automation Science and Engineering 12*, 2 (Apr. 2015), 565–579. Conference Name: IEEE Transactions on Automation Science and Engineering.

[7] ELA, E., AND O'MALLEY, M. Scheduling and Pricing for Expected Ramp Capability in Real-Time Power Markets. *IEEE Transactions on Power Systems 31*, 3 (May 2016), 1681–1691. Conference Name: IEEE Transactions on Power Systems.

[8] FIORETTO, F., MAK, T. W., AND VAN HENTENRYCK, P. Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods. *Proceedings of the AAAI Conference on Artificial Intelligence 34*, 01 (Apr. 2020), 630–637.

[9] HUA, B., SCHIRO, D. A., ZHENG, T., BALDICK, R., AND LITVINOV, E. Pricing in Multi-Interval Real-Time Markets. *IEEE Transactions on Power Systems 34*, 4 (July 2019), 2696–2705.

[10] KELMAN, A., MA, Y., AND BORRELLI, F. Analysis of local optima in predictive control for energy efficient buildings. *Journal of Building Performance Simulation 6*, 3 (May 2013), 236–255.

[11] KILLIAN, M., AND KOZEK, M. Ten questions concerning model predictive control for energy efficient buildings. *Building and Environment 105* (Aug. 2016), 403–412.

[12] KING, J., CLIFTON, A., AND HODGE, B. Validation of Power Output for the WIND Toolkit. Tech. Rep. NREL/TP-5D00-61714, 1159354, Sept. 2014.

[13] KOTARY, J., FIORETTO, F., AND VAN HENTENRYCK, P. Learning Hard Optimization Problems: A Data Generation Perspective. In *Advances in Neural Information Processing Systems* (2021), vol. 34, Curran Associates, Inc., pp. 24981–24992.

[14] LEE, R., MAGHAKIAN, J., HAJIESMAILI, M., LI, J., SITARAMAN, R., AND LIU, Z. Online Peak-Aware Energy Scheduling with Untrusted Advice. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems* (Virtual Event Italy, June 2021), ACM, pp. 107–123.

[15] LI, P., YANG, J., AND REN, S. Expert-Calibrated Learning for Online Optimization with Switching Costs. In *Abstract Proceedings of the 2022 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems* (Mumbai India, June 2022), ACM, pp. 85–86.

[16] LIN, Y., HU, Y., SHI, G., SUN, H., QU, G., AND WIERMAN, A. Perturbation-based Regret Analysis of Predictive Control in Linear Time Varying Systems. In *Advances in Neural Information Processing Systems* (2021), vol. 34, Curran Associates, Inc., pp. 5174–5185.

[17] LIO, W. H., ROSSITER, J., AND JONES, B. L. A review on applications of model predictive control to wind turbines. In *2014 UKACC International Conference on Control (CONTROL)* (July 2014), pp. 673–678.

[18] LYKOURIS, T., AND VASSILVTISKII, S. Competitive Caching with Machine Learned Advice. In *Proceedings of the 35th International Conference on Machine Learning* (July 2018), PMLR, pp. 3296–3305. ISSN: 2640-3498.

[19] MENG, K., DONG, Z. Y., XU, Z., AND WELLER, S. R. Cooperation-Driven Distributed Model Predictive Control for Energy Storage Systems. *IEEE Transactions on Smart Grid 6*, 6 (Nov. 2015), 2583–2585. Conference Name: IEEE Transactions on Smart Grid.

[20] MORARI, M., AND LEE, J. H. Model predictive control: past, present and future. *Computers and Chemical Engineering* (1999), 16.

[21] MORSTYN, T., HREDZAK, B., AGUILERA, R. P., AND AGELIDIS, V. G. Model Predictive Control for Distributed Microgrid Battery Energy Storage Systems. *IEEE Transactions on Control Systems Technology 26*, 3 (May 2018), 1107–1114. Conference Name: IEEE Transactions on Control Systems Technology.

[22] NELLIKKATH, R., AND CHATZIVASILEIADIS, S. Physics-Informed Neural Networks for AC Optimal Power Flow. *Electric Power Systems Research 212* (Nov. 2022), 108412.

[23] PAN, X., CHEN, M., ZHAO, T., AND LOW, S. H. DeepOPF: A Feasibility-Optimized Deep Neural Network Approach for AC Optimal Power Flow Problems, July 2022. arXiv:2007.01002 [cs, eess].

[24] PON KUMAR, S. S., TULSYAN, A., GOPALUNI, B., AND LOEWEN, P. A Deep Learning Architecture for Predictive Control. *IFAC-PapersOnLine 51*, 18 (Jan. 2018), 512–517.

[25] PUROHIT, M., SVITKINA, Z., AND KUMAR, R. Improving Online Algorithms via ML Predictions. In *Advances in Neural Information Processing Systems* (2018), vol. 31, Curran Associates, Inc.

[26] RUTTEN, D., CHRISTIANSON, N., MUKHERJEE, D., AND WIERMAN, A. Online Optimization with Untrusted Predictions. *arXiv:2202.03519 [cs]* (Feb. 2022). arXiv: 2202.03519.

[27] SPIELBERG, S., GOPALUNI, R., AND LOEWEN, P. Deep reinforcement learning approaches for process control. In *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)* (May 2017), pp. 201–206.

[28] SULTANA, W. R., SAHOO, S. K., SUKCHAI, S., YAMUNA, S., AND VENKATESH, D. A review on state of art development of model predictive control for renewable energy applications. *Renewable and Sustainable Energy Reviews 76* (Sept. 2017), 391–406.

[29] TROY, N., DENNY, E., AND O'MALLEY, M. Base-Load Cycling on a System With Significant Wind Penetration. *IEEE Transactions on Power Systems 25*, 2 (May 2010), 1088–1097. Conference Name: IEEE Transactions on Power Systems.

[30] ZHANG, L., JIANG, W., LU, S., AND YANG, T. Revisiting Smoothed Online Learning. In *Advances in Neural Information Processing Systems* (2021), vol. 34, Curran Associates, Inc., pp. 13599–13612.

[31] ZHAO, J., ZHENG, T., AND LITVINOV, E. A Multi-Period Market Design for Markets With Intertemporal Constraints. *IEEE Transactions on Power Systems 35*, 4 (July 2020), 3015–3025.

A The RobustML algorithm

We specify the algorithm RobustML below in Algorithm 1. Note we assume that all algorithms begin in the same initial state $\mathbf{x}_0$, so $\mathbf{a}_0 = \mathbf{r}_0 = \mathbf{x}_0$ (where $\mathbf{a}_0 = \mathrm{ML}_0$ and $\mathbf{r}_0 = \textsc{Greedy}_0$).

Algorithm 1: RobustML(ϵ, δ)

Input: Algorithms ML, Greedy; hyperparameters $\epsilon, \delta > 0$, space diameter D
Output: Decisions $\mathbf{x}_1, \ldots, \mathbf{x}_T$ chosen online

$s \leftarrow 1$
$\mathbf{x}_1 \leftarrow \mathbf{a}_1 \coloneqq \mathrm{ML}_1$
for $t = 2, 3, \ldots, T$ **do**
  Observe f_t, $\mathbf{a}_t \coloneqq \mathrm{ML}_t$, and $\mathbf{r}_t \coloneqq \textsc{Greedy}_t$
  if $\mathbf{x}_{t-1} = \mathbf{a}_{t-1}$ **then** `// Case where the algorithm coincides with` ML_{t-1}
    if $\mathrm{C}_{\mathrm{ML}}(s,t) \geq \frac{2D}{\epsilon}$ *and* $\mathrm{C}_{\textsc{Greedy}}(1,t) < \delta \cdot \mathrm{C}_{\mathrm{ML}}(1,t)$ **then**
      $s \leftarrow t+1$
      $\mathbf{x}_t \leftarrow \mathbf{r}_t$
    else
      $\mathbf{x}_t \leftarrow \mathbf{a}_t$
  else `// Case where the algorithm coincides with` $\textsc{Greedy}_{t-1}$
    if $\mathrm{C}_{\textsc{Greedy}}(1,t) < \delta \cdot \mathrm{C}_{\mathrm{ML}}(1,t)$ **then**
      $\mathbf{x}_t \leftarrow \mathbf{r}_t$
    else
      $\mathbf{x}_t \leftarrow \mathbf{a}_t$
end

B Proof of Theorem 1

We begin by showing $\mathrm{C}_{\textsc{RobustML}} \leq (1 + \epsilon + \delta)\mathrm{C}_{\mathrm{ML}}$. Note that the algorithm consists of *phases* in which RobustML first coincides with ML, and then switches to following Greedy, before switching back to ML, and so on. We will assume that RobustML ends the instance coinciding with ML, so $\mathbf{x}_T = \mathbf{a}_T$; the case in which RobustML ends at $\mathbf{r}_T$ is similar. Let t_i denote the timestep in which RobustML switches from Greedy back to ML for the ith time, with $t_0 \coloneqq 1$ since RobustML always begins by following ML. Similarly, let m_i denote the timestep in which RobustML switches from ML to Greedy for the ith time. Clearly we have $1 = t_0 < m_1 < t_1 < \cdots < m_k < t_k \leq T$, for some $k \in \mathbb{N}$. Even though RobustML ends at ML, define $m_{k+1} \coloneqq T + 1$ for notational simplicity. Then the cost of RobustML may be written as

$$
\begin{aligned}
\mathrm{C}_{\textsc{RobustML}} &= \sum_{\tau=1}^{m_1-1} f_\tau(\mathbf{a}_\tau) + \|\mathbf{a}_\tau - \mathbf{a}_{\tau-1}\| \\
&\quad + \sum_{i=1}^{k} \Bigg(f_{m_i}(\mathbf{r}_{m_i}) + \|\mathbf{r}_{m_i} - \mathbf{a}_{m_i-1}\| + \sum_{\tau=m_i+1}^{t_i-1} f_\tau(\mathbf{r}_\tau) + \|\mathbf{r}_\tau - \mathbf{r}_{\tau-1}\| \\
&\qquad + f_{t_i}(\mathbf{a}_{t_i}) + \|\mathbf{a}_{t_i} - \mathbf{r}_{t_i-1}\| + \sum_{\tau=t_i+1}^{m_{i+1}-1} f_\tau(\mathbf{a}_\tau) + \|\mathbf{a}_\tau - \mathbf{a}_{\tau-1}\| \Bigg) \\
&\leq \mathrm{C}_{\mathrm{ML}}(1, m_1 - 1) + \sum_{i=1}^{k} \Big(\mathrm{C}_{\textsc{Greedy}}(m_i, t_i - 1) + \|\mathbf{r}_{m_i-1} - \mathbf{a}_{m_i-1}\| \\
&\qquad + \mathrm{C}_{\mathrm{ML}}(t_i, m_{i+1} - 1) + \|\mathbf{a}_{t_i-1} - \mathbf{r}_{t_i-1}\| \Big) \qquad (1) \\
&\leq \mathrm{C}_{\mathrm{ML}}(1, m_1 - 1) + 2kD + \sum_{i=1}^{k} \mathrm{C}_{\textsc{Greedy}}(m_i, t_i - 1) + \mathrm{C}_{\mathrm{ML}}(t_i, m_{i+1} - 1) \qquad (2) \\
&\leq (1+\epsilon)\mathrm{C}_{\mathrm{ML}} + \sum_{i=1}^{k} \mathrm{C}_{\textsc{Greedy}}(m_i, t_i - 1) \qquad (3) \\
&\leq (1+\epsilon+\delta)\mathrm{C}_{\mathrm{ML}} \qquad (4)
\end{aligned}
$$

where (1) follows from the triangle equality on $\|\mathbf{r}_{m_i} - \mathbf{a}_{m_i-1}\|$ and $\|\mathbf{a}_{t_i} - \mathbf{r}_{t_i-1}\|$, and (2) follows by the diameter bound. The inequality (3) follows by line 6 of the algorithm, which states that the algorithm will switch from following ML to following GREEDY at time t only if $\mathrm{C}_{\mathrm{ML}}(s,t) \geq \frac{2D}{\epsilon}$. Noting that at the start of any timestep t, s is exactly

$$
s = \max_{i : m_i + 1 \leq t} m_i + 1
$$

(with $m_0 \coloneqq 0$ for notational convenience), it follows that for each $i \in [k]$, $\mathrm{C}_{\mathrm{ML}}(m_{i-1}+1, m_i) \geq \frac{2D}{\epsilon}$. Thus

$$
2kD \leq \epsilon \sum_{i=1}^{k} \mathrm{C}_{\mathrm{ML}}(m_{i-1}+1, m_i) = \epsilon \cdot \mathrm{C}_{\mathrm{ML}}(1, m_k) \leq \epsilon \cdot \mathrm{C}_{\mathrm{ML}}.
$$

Finally, (4) follows from

$$
\sum_{i=1}^{k} \mathrm{C}_{\textsc{Greedy}}(m_i, t_i - 1) \leq \mathrm{C}_{\textsc{Greedy}}(1, t_k - 1) < \delta \cdot \mathrm{C}_{\mathrm{ML}}(1, t_k - 1) \leq \delta \cdot \mathrm{C}_{\mathrm{ML}},
$$

since by definition, $\mathbf{x}_{t_k-1} = \mathbf{r}_{t_k-1}$, which by line 12 of the algorithm means that $\mathrm{C}_{\textsc{Greedy}}(1, t_k - 1) < \delta \cdot \mathrm{C}_{\mathrm{ML}}(1, t_k - 1)$. Thus we have proved the desired bound $\mathrm{C}_{\textsc{RobustML}} \leq (1+\epsilon+\delta)\mathrm{C}_{\mathrm{ML}}$.

We now turn to showing $\mathrm{C}_{\textsc{RobustML}} \leq \left(1 + \frac{1+\epsilon}{\delta}\right) \mathrm{C}_{\textsc{Greedy}} + \left(1 + \frac{2}{\epsilon}\right) D$. First suppose ROBUSTML finishes the instance coinciding with ML, so $\mathbf{x}_T = \mathbf{a}_T$. Let $\tau \in \{0, \ldots, T-1\}$ denote the last time at which ROBUSTML coincided with GREEDY, or that $\mathbf{x}_\tau = \mathbf{r}_\tau$. Thus the cost can be bounded as

$$
\begin{aligned}
\mathrm{C}_{\textsc{RobustML}} &= \mathrm{C}_{\textsc{RobustML}}(1, \tau+1) + \mathrm{C}_{\textsc{RobustML}}(\tau+2, T) \\
&\leq (1+\epsilon+\delta)\mathrm{C}_{\mathrm{ML}}(1, \tau+1) + \mathrm{C}_{\mathrm{ML}}(\tau+2, T) \qquad (5) \\
&\leq \max\left\{ \left(1 + \frac{1+\epsilon}{\delta}\right) \mathrm{C}_{\textsc{Greedy}}(1, \tau+1) + \frac{2D}{\epsilon}, \left(1 + \frac{1+\epsilon}{\delta}\right) \mathrm{C}_{\textsc{Greedy}} \right\} \qquad (6) \\
&\leq \left(1 + \frac{1+\epsilon}{\delta}\right) \mathrm{C}_{\textsc{Greedy}} + \frac{2D}{\epsilon} \qquad (7)
\end{aligned}
$$

where (5) follows via the previously proved inequality $\mathrm{C}_{\textsc{RobustML}} \leq (1+\epsilon+\delta)\mathrm{C}_{\mathrm{ML}}$, and (6) follows by the fact (according to line 14 of the algorithm) that ROBUSTML switching from GREEDY to ML at time $\tau+1$ means that $\mathrm{C}_{\textsc{Greedy}} \geq \delta \cdot \mathrm{C}_{\mathrm{ML}}(1, \tau+1)$, as well as from the following observation:

since RobustML coincides with ML between times $\tau + 1$ and T, line 6 of the algorithm tells us that either $\mathrm{C}_{\mathrm{ML}}(\tau + 2, T) < \frac{2D}{\epsilon}$ or $\mathrm{C}_{\textsc{Greedy}} \geq \delta \cdot \mathrm{C}_{\mathrm{ML}}$.

Finally, suppose RobustML finishes the instance coinciding with Greedy, so $\mathbf{x}_T = \mathbf{r}_T$. Let $\sigma \in \{0, \ldots, T-1\}$ denote the last time at which RobustML coincided with ML, or that $\mathbf{x}_\sigma = \mathbf{a}_\sigma$. By the previous case's inequality (7), we have

$$\begin{aligned}
\mathrm{C}_{\textsc{RobustML}} &= \mathrm{C}_{\textsc{RobustML}}(1, \sigma) + \mathrm{C}_{\textsc{RobustML}}(\sigma + 1, T) \\
&\leq \left(1 + \frac{1+\epsilon}{\delta}\right) \mathrm{C}_{\textsc{Greedy}}(1, \sigma) + \frac{2D}{\epsilon} + f_{\sigma+1}(\mathbf{r}_{\sigma+1}) + \|\mathbf{r}_{\sigma+1} - \mathbf{a}_\sigma\| + \mathrm{C}_{\textsc{Greedy}}(\sigma + 2, T) \\
&\leq \left(1 + \frac{1+\epsilon}{\delta}\right) \mathrm{C}_{\textsc{Greedy}}(1, \sigma) + \frac{2D}{\epsilon} + D + \mathrm{C}_{\textsc{Greedy}}(\sigma + 1, T) \\
&\leq \left(1 + \frac{1+\epsilon}{\delta}\right) \mathrm{C}_{\textsc{Greedy}} + \left(1 + \frac{2}{\epsilon}\right) D.
\end{aligned}$$

□