Machine Learning Guided Optimal Transmission Switching to Mitigate Wildfire Ignition Risk

Weimin Huang, Bistra Dilkina
Department of Computer Science
University of Southern California
Los Angeles, CA, USA
{weiminhu, dilkina}@usc.edu

Ryan Piansky, Daniel K. Molzahn
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA, USA
{rpiansky3, molzahn}@gatech.edu

Abstract—To mitigate acute wildfire ignition risks, utilities de-energize power lines in high-risk areas. The Optimal Power Shut-off (OPS) problem optimizes line energization statuses to manage wildfire ignition risks through de-energizations while reducing load shedding. OPS problems are computationally challenging Mixed-Integer Linear Programs (MILPs) that must be solved rapidly and frequently in operational settings. For a particular power system, OPS instances share a common structure with varying parameters related to wildfire risks, loads, and renewable generation. This motivates the use of Machine Learning (ML) for solving OPS problems by exploiting shared patterns across instances. In this paper, we develop an ML-guided framework that quickly produces high-quality de-energization decisions by extending existing ML-guided MILP solution methods while integrating domain knowledge on the number of energized and de-energized lines. Results on a large-scale realistic California-based synthetic test system show that the proposed ML-guided method produces high-quality solutions faster than traditional optimization methods.

Index Terms—Machine Learning, Optimization, Power Shut-off, Transmission, Wildfire Ignition Risk

I. INTRODUCTION

Wildfires are a growing threat under a worsening global climate. For instance, the fire intensity potential in California is expected to significantly increase by 2050 [1], leading to more destructive and widespread wildfires. Fires ignited by power grid equipment are among the most destructive in California [2]. Wildfires ignited from power grid equipment tend to burn more area than fires ignited from other sources [3], likely because conditions that lead to ignitions from power equipment (dry vegetation, high temperatures, and strong winds) also contribute to faster spreading fires.

To mitigate the acute risk of wildfire ignition, utilities proactively de-energize lines in high wildfire ignition risk areas through Public Safety Power Shutoff (PSPS) events [4], [5]. While de-energizing lines eliminates their ignition risk, this can lead to load shedding when the partially de-energized system cannot supply all loads. To optimally manage wild-fire ignition risk and load shedding, researchers study Optimal Power Shutoff (OPS) problems to determine line de-energization decisions [6], [7]. Typically formulated as Mixed-Integer Linear Programs (MILP), OPS problems pose significant computational challenges that we address in this paper via machine learning (ML) guided MILP solving techniques.

Utilities consider a number of factors, such as vegetation, equipment age, and Fire Potential Indices (FPI), to assign an ignition risk to each line [8]. In common risk mitigation practices, lines that surpass a risk threshold are de-energized. Relative to risk thresholds applied to each line individually, OPS formulations typically permit flexibility in managing system-wide risk simultaneously across all lines, thus resulting in de-energization decisions that yield lower load shedding compared to less flexible threshold-based decisions [7].

FPIs and other conditions change daily, necessitating quick responses from utilities. However, solving OPS problems with realistic large-scale power system models can require hours to days of computation time [7], rendering the resulting solutions operationally irrelevant. Thus, new computational methods are needed to operationalize OPS formulations for practically relevant system models.

Recent advances in ML-guided MILP solving [9], [10] hold significant promise for addressing the computational challenges of OPS problems. Recognizing that MILPs with similar structures (i.e., MILPs of the same problem formulation constructed from data parameters sampled from a given distribution) are solved repeatedly in many applications, this body of research focuses on using ML to guide existing algorithmic policies or build new ones customized for specific instance distributions.

Two key features of OPS problems make the application of these recent ML advances particularly attractive. First, practical implementations of OPS problems would require repeated solution of very similar problems to determine daily line de-energization decisions. With limited changes to the network and conventional generators' characteristics, differences between OPS problems on a daily basis are primarily driven by varying wildfire ignition risks, load demands, and renewable generation, all of which have temporal and spatial patterns. Second, OPS formulations inherently involve significant mod-

eling assumptions (e.g., the use of power flow approximations [11]) and parameter uncertainties associated with forecasts for wildfire ignition risk, load demands, and renewable generator outputs [12]–[14]. By permitting the use of shorter forecast horizons, nearly optimal decisions obtained more quickly can potentially achieve better outcomes in practice than high-precision optima that require long computing times. The limited variation across OPS problems and modest value of high-precision global optima make it worthwhile to spend computational time in offline training of ML-guided MILP solvers to speed up day-of solution times when uncertainties are realized close to real-time operations. Thus, based on these two features, recent advances in ML-guided MILP solving such as those in [15]–[18] have the potential to determine OPS de-energization decisions much more quickly than traditional MILP solvers while significantly outperforming traditional threshold-based de-energization methods.

Extensions to the OPS problem consider long-term infrastructure investments to mitigate wildfire ignition risks [19]–[22], microgrids for maintaining service during PSPS events [23], and methods for improving the equity of wildfire mitigation outcomes [23], [24], each of which impose further computational challenges. While not all of these extensions require rapid solutions for near-real-time operations, they could still benefit from faster computation times achieved using ML-guided solvers. For instance, many methods for long-term infrastructure hardening problems use decomposition techniques that repeatedly solve similar subproblems such that time spent training ML guided solvers may be worthwhile [20], [21].

For large-scale MILPs, a large body of ML-guided MILP solving research focuses on ML-guided primal heuristics, where the goal is to obtain high-quality solutions within a short amount of time, instead of proving optimality. These approaches typically learn to predict the near-optimal assignment for variables in the problem [15]–[17]. Since an ML model's direct prediction may be infeasible, these methods refine the prediction by solving another MILP at inference time. Two major frameworks for this refinement are Neural Diving (ND) [15] and Predict-and-Search (PAS) [16], [17]. ND fixes a subset of variables to their predicted values, creating a smaller, easier-to-solve sub-MILP. While computationally efficient, this aggressive reduction of the search space may yield highly suboptimal solutions. In contrast, PAS searches for near-optimal solutions within a neighborhood based on the prediction. PAS operates on the original MILP and has a neighborhood parameter that specifies the number of variables that are allowed to change from their prescribed assignment suggested by the ML model [16], [17]. This approach offers more flexibility to correct prediction errors but is more computationally demanding, as the problem size remains unchanged.

While ML has previously been used for other wildfire applications (see, e.g., [25] for ML predicted ignitions and [26] for data-driven operation predictions under PSPS events), this paper is, to the best of our knowledge, the first to propose using ML-guided MILP methods to quickly identify high-quality

line de-energization decisions. As most lines are energized in the context of OPS, many binary variables take 1 as the solution value (as opposed to 0). This creates a class imbalance in the ML-guided framework, which is uncommon in other domains where ML-guided primal heuristics such as PAS and ND have been applied. Thus, in this paper, we extend the PAS framework used in [16], [17] by introducing two separate neighborhood parameters for the number of variables that are allowed to change from their ML-prescribed assignment, one for the variables predicted to be 0 and another one for the variables predicted to be 1. This allows us to integrate domain knowledge on the number of binary variables that take 0 and 1, respectively, as their solution value in the OPS problem. Finally, we propose a variant that combines PaS and ND, where the domain-informed PaS constraint acts as a safety net to restore feasibility. Results on a realistic synthetic test system of the California transmission grid [27] show that our proposed ML-guided method significantly outperforms Gurobi.

The remainder of this paper is organized as follows. Section II details the OPS model used to generate the training set used for the ML model in Section III. Section IV details the specific power system studied in this paper to inform the training data set. Section V shows results from the ML model. Finally, Section VI concludes the paper.

II. PROBLEM FORMULATION

This paper focuses on computing transmission line de-energization decisions to mitigate wildfire ignition risk. Traditional optimization approaches to this problem scale very poorly with both network size and the number of considered time periods. Machine learning methods provide a faster way to arrive at switching decisions. To establish notation and define the power shutoff formulation, we next describe the optimization model for the optimal power shutoff problem. As a benchmark, we then provide a methodology that uses a simple threshold-based rule to determine de-energization decisions.

A. Optimal Power Shutoff Model

The Optimal Power Shutoff (OPS) problem minimizes the total amount of load shed across the system as transmission lines are de-energized to mitigate the risk of wildfire ignition. In Model 1, we present a single time period version of the OPS problem derived from [7].

The objective (1a) minimizes the load shed (p_{ls}^n) over all buses $n \in \mathcal{N}$ as well as a penalty term with parameter ϵ (selected as 0.01 per unit) to minimize the number of lines de-energized. For notational brevity, the objective function in the remainder of the paper is represented by $h(\boldsymbol{z} \mid M)$ to indicate the objective optimized for a given model M dependent on the vector of de-energization decisions $\boldsymbol{z}$. Constraints (1b) and (1c) bound the generation (p_g^i) at each generator $i \in \mathcal{G}$ and load shed at each bus, respectively. Constraints (1d) and (1e) constrain the flow (f^ℓ) on each line $\ell \in \mathcal{L}$ to be within the rated limits. We group lines at high risk of igniting a fire into the set $\mathcal{L}^{\text{switch}}$. Lines in this set have their rated limits modified by the binary de-energization decision variable, z^ℓ in (1d). A

Model 1 Optimal Power Shutoff (OPS)

$$\min \ \sum_{n \in \mathcal{N}} p_{ls}^n + \epsilon \sum_{\ell \in \mathcal{L}^{\text{switch}}} z^\ell \tag{1a}$$

s.t.

$$0 \leqslant p_g^i \leqslant \overline{p}_g^i \qquad\qquad \forall i \in \mathcal{G} \tag{1b}$$

$$0 \leqslant p_{ls}^n \leqslant p_l^n \qquad\qquad \forall n \in \mathcal{N} \tag{1c}$$

$$-\overline{f}^\ell z^\ell \leqslant f^\ell \leqslant \overline{f}^\ell z^\ell \qquad\qquad \forall \ell \in \mathcal{L}^{\text{switch}} \tag{1d}$$

$$-\overline{f}^\ell \leqslant f^\ell \leqslant \overline{f}^\ell \qquad\qquad \forall \ell \in \mathcal{L} \setminus \mathcal{L}^{\text{switch}} \tag{1e}$$

$$f^\ell \geqslant -b^\ell(\theta^{n^{\ell,\text{fr}}} - \theta^{n^{\ell,\text{to}}}) + |b^\ell|\underline{\theta}(1 - z^\ell) \quad \forall \ell \in \mathcal{L}^{\text{switch}} \tag{1f}$$

$$f^\ell \leqslant -b^\ell(\theta^{n^{\ell,\text{fr}}} - \theta^{n^{\ell,\text{to}}}) + |b^\ell|\overline{\theta}(1 - z^\ell) \quad \forall \ell \in \mathcal{L}^{\text{switch}} \tag{1g}$$

$$f^\ell = -b^\ell(\theta^{n^{\ell,\text{fr}}} - \theta^{n^{\ell,\text{to}}}) \qquad \forall \ell \in \mathcal{L} \setminus \mathcal{L}^{\text{switch}} \tag{1h}$$

$$\sum_{\ell \in \mathcal{L}^{n,\text{fr}}} f^\ell - \sum_{\ell \in \mathcal{L}^{n,\text{to}}} f^\ell = \sum_{i \in \mathcal{G}^n} p_g^i - p_l^n + p_{ls}^n \quad \forall n \in \mathcal{N} \tag{1i}$$

$$\sum_{\ell \in \mathcal{L}^{\text{switch}}} z^\ell R_\ell \leq R^{\text{PSPS}} \qquad\qquad \forall \ell \in \mathcal{L}^{\text{switch}}. \tag{1j}$$

value of $z^\ell = 0$ indicates that line ℓ is de-energized while $z^\ell = 1$ indicates a line is energized. Constraints (1f) and (1g) implement the Bθ DC power flow approximation with a big-M linearization. Constraint (1h) models the DC power flow approximation for non-switchable lines. Constraint (1i) models power balance to ensure the flow of power into and out of each bus matches the power demanded (p_l^n) and produced at that bus. Constraint (1j) ensures that the sum of risk per line (R_ℓ) on energized lines is below an acceptable system-wide threshold (R^{PSPS}).

Note that we can infer two informative bounds on the number of de-energized lines from Model 1. First, we can find the minimum number of lines that must be de-energized to be feasible, $N_0^{\min}$. By de-energizing the k riskiest lines until we meet the constraint in (1j), we can determine the minimum number of lines that must be de-energized to have a feasible solution. Likewise, we can determine the maximum number of lines that must be de-energized to guarantee a feasible solution, $N_0^{\max}$. Here, the m least risky lines can be de-energized until the constraint is met in (1j). The number of de-energized lines in the optimal solution must be between $N_0^{\min}$ and $N_0^{\max}$. This domain-specific knowledge informs the machine learning methodology discussed in Section III.

B. Thresholded De-energization Decisions

As an alternative to an MILP formulation that optimizes de-energization decisions, one could de-energize lines based solely on a pre-specified threshold on their associated wildfire ignition risk, without considering the power flows in the network. The variable z^ℓ is then treated as a *parameter* in Model 1 to mimic current PSPS approaches which primarily consider wildfire ignition risk on a line-by-line basis. Similarly, the objective (1a) is reduced to only consider load shedding. This reduces the problem to a linear program (LP) which is far less

computationally challenging. For a fair comparison between OPS and threshold-based de-energization decisions, we set the value of R^{PSPS} to match the remaining risk in the network after de-energizing lines based on the threshold method.

III. MACHINE LEARNING ENHANCED OPTIMIZATION

Each instance of the OPS problem is parameterized by the wildfire risk per line, R_ℓ, renewable generation availability impacting the upper limit of some generation, p_g^i, and load demand at each bus, p_l^n. Parameters of Model 1 drawn from the same distribution result in a set of MILP instances with similar structures, which motivates the use of ML to accelerate MILP solvers. Building on existing techniques [15]–[17], this section introduces our ML-guided framework for faster solution of OPS problems.

A. Solution Prediction

1) Learning task: We use ML to predict the solution for the set of de-energization variables, $\boldsymbol{z} = [z^\ell]_{\forall \ell \in \mathcal{L}^{\text{switch}}}$ in (1d), as the complexity of MILPs significantly depends on these binary decisions. Following [15] and [16], we learn the probability distribution of the solution space given a problem instance. This approach learns from a set of multiple solutions given a problem instance and considers the solution quality of each solution in the training data. Specifically, given an MILP instance M of the OPS problem, let $h(\boldsymbol{z} \mid M)$ denote the objective value obtained by a solution $\boldsymbol{z}$ (computed according to (1a)). A solution quality function $E(\boldsymbol{z} \mid M)$ is defined as $h(\boldsymbol{z} \mid M)$ if $\boldsymbol{z}$ is feasible or ∞ otherwise. Given M, the conditional distribution of a solution $\boldsymbol{z}$ is modeled as

$$\mathbb{P}(\boldsymbol{z} \mid M) \equiv \frac{\exp(-E(\boldsymbol{z} \mid M))}{\sum_{\boldsymbol{z}' \in Z} \exp(-E(\boldsymbol{z}' \mid M))}, \tag{2}$$

where Z is the set of solutions in the solution space of M, so that solutions with better objective values are assigned higher probabilities. The learning task is to train an ML model parameterized by ω that approximates $\mathbb{P}(\boldsymbol{z} \mid M)$. Specifically, we learn a policy $\pi(\boldsymbol{z} \mid \omega, M)$ that outputs a prediction for the solution value for each variable in $\boldsymbol{z}$, given M as the input. The policy $\pi(\boldsymbol{z} \mid \omega, M)$ produces the probabilities for which the binary variables $\boldsymbol{z}$ in Model 1 take values of 1 in the solution. The output for a variable z^ℓ should be close to 1 if its value in the optimal solution is 1, and close to 0 otherwise.

2) Loss function: Following [15] and [16], we use Weighted Cross-Entropy loss [16] to learn the policy $\pi(\boldsymbol{z} \mid \omega, M)$. We collect a training dataset that contains N MILPs instances $\{(M_i, S_{M_i})\}_{i=1}^N$, where S_{M_i} is a set of high-quality solutions for the instance M_i. Let $\pi(\boldsymbol{z} \mid \omega, M_i)$ denote the probability of solution $\boldsymbol{z}$ given instance M_i as the input. Based on the Kullback-Leibler divergence which measures the distance between the conditional distribution in (2) and the learned policy, the loss function to be minimized is:

$$\mathcal{L}(\omega) \equiv -\sum_{i=1}^N \sum_{\boldsymbol{z} \in \mathcal{S}_{M_i}} w(\boldsymbol{z} \mid M_i) \log(\pi(\boldsymbol{z} \mid \omega; M_i)), \tag{3}$$

where $w(\boldsymbol{z} \mid M_i) \equiv \frac{\exp(-h_i(\boldsymbol{z}|M_i))}{\sum_{\boldsymbol{z}' \in \mathcal{S}_{M_i}} \exp(-h_i(\boldsymbol{z}'|M_i))}$ is the weight applied to the solution $\boldsymbol{z}$ for instance M_i. Following prior work, we assume conditional independence between variables to make the training task tractable. Given an instance M_i, let $\boldsymbol{z}_l$ denote the lth element of a solution vector $\boldsymbol{z}$. Conditional independence leads to $\pi(\boldsymbol{z} \mid \omega, M) = \prod_{l=1}^{|\boldsymbol{z}|} \pi(\boldsymbol{z}_l \mid \omega, M)$.

B. Neural Architecture

We learn the policy $\pi(\boldsymbol{z} \mid \omega, M)$ using a Graph Attention Network (GAT) [28]. We encode each input MILP instance as a graph before passing it to the GAT. We convert an MILP instance M into a featured graph $\Gamma = (\mathcal{G}, V, C, E)$ following [29], where $\mathcal{G} = (\mathcal{V}, \mathcal{C}, \mathcal{E})$ is a bipartite graph that contains two sets of nodes: the variable nodes $\mathcal{V} = \{1, \ldots, |\mathcal{L}^{\text{switch}}|\}$ representing the decision variables $\boldsymbol{z}$, and the constraint nodes $\mathcal{C}$ representing the set of all constraints in Model 1. The edges $\mathcal{E}$ represent the non-zero entries in the constraint-coefficient matrix associated with Model 1, connecting the nodes in $\mathcal{V}$ and $\mathcal{C}$. The matrices $V \in \mathbb{R}^{|\mathcal{V}| \times v'}$, $C \in \mathbb{R}^{|\mathcal{C}| \times c'}$, and $E \in \mathbb{R}^{|\mathcal{E}| \times e'}$ are the feature vectors for $\mathcal{V}$, $\mathcal{C}$, and $\mathcal{E}$, respectively. We adopt the feature set from [29], resulting in $v' = 15$ variable features (e.g., variable type, coefficients, bounds, root-LP statistics), $c' = 4$ constraint features (e.g., constant term, sense), and $e' = 1$ edge feature (coefficient value).

The GAT module processes the graph Γ before outputting the predictions $\pi(\boldsymbol{z} \mid \omega, M)$. It first uses learned linear projections to embed the original features (V, C, E) into a common K-dimensional space, yielding

$$V^1 \in \mathbb{R}^{n \times K}, \quad C^1 \in \mathbb{R}^{m \times K}, \quad E^1 \in \mathbb{R}^{|\mathcal{E}| \times K}.$$

The GAT module then performs two rounds of message passing: in round one, each constraint node in C^1 attends over its incident edges in Γ to produce updated constraint embeddings C^2; in round two, each variable node in V^1 attends over its incident edges to produce updated variable embeddings V^2. The message passing module is followed by a multi-layer perceptron with a sigmoid activation function, which outputs a single value as the solution prediction for each variable $\boldsymbol{z}_l$.

C. Inference Time Refinement

As the predictions $\pi(\boldsymbol{z} \mid \omega, M)$ can be infeasible after rounding, the predictions need to be refined at inference time. Neural Diving (ND) [15] creates the final solution by fixing a subset of variables to their predicted values and creating a smaller sub-MILP. While computationally efficient, this aggressive search space reduction may yield suboptimal solutions. Predict-and-Search (PaS) [16], [17] is another framework that searches for near-optimal solutions within a neighborhood based on the prediction. The PaS [16], [17] framework includes three parameters: k_0, k_1, and Δ. Denote $\mathcal{I}_0$ as a set with k_0 variables that contains the indices for the binary variables with the lowest $\pi(\boldsymbol{z}_l \mid \omega, M)$ values. Similarly, $\mathcal{I}_1$ is a set with k_1 variables that contains the indices for the binary variables with the highest $\pi(\boldsymbol{z}_l \mid \omega, M)$ values. The scalar Δ specifies the number of variables for which the

solution value of the variable is allowed to change in $\mathcal{I}_0 \cup \mathcal{I}_1$. PaS assigns all variables in $\mathcal{I}_0$ to 0 and all variables in $\mathcal{I}_1$ to 1 in the MILP but also allows Δ of the assigned variables to be flipped by the solver. Accordingly, PaS adds the following constraint to the original problem:

$$B(\mathcal{I}_0, \mathcal{I}_1, \Delta) = \{\boldsymbol{z} : \sum_{z_l \in \mathcal{I}_0} z_l + \sum_{z_l \in \mathcal{I}_1} 1 - z_l \leq \Delta\}. \quad (4)$$

1) Domain-informed PaS: As discussed in Section II-A, the OPS problem is infeasible when the number of zero entries in $\boldsymbol{z}$ is less than $N_0^{\min}$ and the optimal solution for $\boldsymbol{z}$ has a number of zero entries between $N_0^{\min}$ and $N_0^{\max}$. This domain knowledge informs our choice of k_0 and k_1, as well as motivates us to break the parameter Δ into Δ_0 and Δ_1 as separate parameters for the number of binary variables allowed to change from their prescribed assignment from those assigned to 0 ($\mathcal{I}_0$) and those assigned to 1 ($\mathcal{I}_1$). We set $k_0 = N_0^{\max}$ (in order not to exclude the optimal solution) and $\Delta_0 = N_0^{\max} - N_0^{\min}$ (in order not to cause infeasibility due to too many variables assigned to 0).

Additionally, based on typical wildfire risk profiles, a large percentage of variables take values of 1 in the solution (i.e., most lines are energized). Based on this observation, we set k_1 as a fraction of the total number of binary variables ($k_1 = \phi |\mathcal{L}^{\text{switch}}|$). We also identify a lower bound for Δ_1. We know that the problem is infeasible when the number of one entries is greater than $|\mathcal{L}^{\text{switch}}| - N_0^{\min}$. In this case, at least $k_1 - (|\mathcal{L}^{\text{switch}}| - N_0^{\min})$ variables assigned to 1 need to flip in order to maintain feasibility. Therefore, the lower bound for Δ_1 is $\max(k_1 - (|\mathcal{L}^{\text{switch}}| - N_0^{\min}), 0)$. We introduce an offset parameter ϕ' so that $\Delta_1 = \max(k_1 - (|\mathcal{L}^{\text{switch}}| - N_0^{\min}), 0) + \phi' k_1$. Our domain-informed PaS framework then adds the following two constraints to the original MILP:

$$B_0(\mathcal{I}_0, \Delta_0) = \{\boldsymbol{z} : \sum_{z_l \in \mathcal{I}_0} z_l \leq \Delta_0\}, \quad (5\text{a})$$

$$B_1(\mathcal{I}_1, \Delta_1) = \{\boldsymbol{z} : \sum_{z_l \in \mathcal{I}_1} 1 - z_l \leq \Delta_1\}. \quad (5\text{b})$$

2) Domain-informed PaS+ND with safety net: We also consider a special case where we set Δ_1 to its lower bound $\max(k_1 - (|\mathcal{L}^{\text{switch}}| - N_0^{\min}), 0)$. In the case when $k_1 \leq |\mathcal{L}^{\text{switch}}| - N_0^{\min}$, no variables in $\mathcal{I}_1$ are allowed to flip. This is similar to ND, where a subset of variables ($\mathcal{I}_1$) is fixed, creating a smaller sub-MILP. In this case, this variant can be viewed as a combination of PaS and ND, where we use the extended PaS with $B_0(\mathcal{I}_0, \Delta_0)$ to infuse information on the number of zero entries and use ND to encode our belief on the number of one entries in the solution. Additionally, when $k_1 > |\mathcal{L}^{\text{switch}}| - N_0^{\min}$, the PaS constraint $B_1(\mathcal{I}_1, \Delta_1)$ acts as safety net for ND that can help restore feasibility, in case setting too many variables to 1 leads to infeasibility.

IV. CASE STUDY AND DATASETS

To demonstrate the performance of ML-guided solvers for the OPS problem in Model 1, we use the California Test System (CATS), a synthetic transmission network with

line routings accurate to the actual transmission corridors in California [27]. This network has hourly load and renewable generation data at each bus derived from real world sources providing a realistic-but-not-real synthetic transmission network. This network was augmented in [7] with daily real world wildfire ignition risk data derived from the US Geological Survey's (USGS) Wildland Fire Potential Index (WFPI) [30]. The USGS WFPI provides a 1 km by 1 km map of the US on each day with wildfire potential values ranging from 0 (low risk) to 247 (high risk). Each transmission line can intersect multiple pixels. Figure 1 shows the CATS network with USGS WFPI risk information.

Based on the metrics defined in [7], we look at the maximum intersected risk on each line. In this paper, we solve a single time period instance of the OPS problem. Specifically, we consider the hour with the most load shed from the thresholded version of the problem run on each hour of the day as defined in [7] and discussed in Section II-B. This threshold is based on the 95th-percentile of pixels intersected by all lines in the network over 2019 as defined in [7]. After determining the hour of the day with the most load shed, we solve the OPS problem to reach the same amount of acceptable risk while minimizing the amount of load shed. Some days have no load shed or no wildfire risk so we do not solve an OPS problem on each day. We solve these single-hour OPS problems for 2020 and 2021, resulting in 651 optimization problems for use in our ML methodology.

These OPS problems are solved using Gurobi 12.0.2 [31]. We limit the computing time to 24-hours for each problem and save the ten incumbent solutions with the best objective values encountered through the solution process for use in the ML training as discussed in Section III-A.
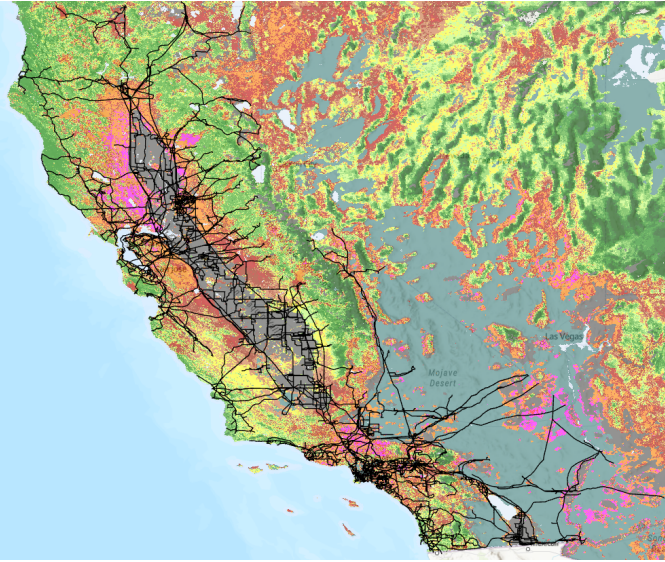

Fig. 1: California's transmission line paths on a Wildland Fire Potential Index map for Oct. 26, 2020. Pixels are individually colored with lower values in green and higher values in red and pink.

For the ML method described in Section III, we classify the difficulty of each daily problem instance as either "easy" or "hard". Easy problems can be solved using conventional MILP solvers such as Gurobi in a reasonable amount of time (on the order of an hour). Hard problems under this classification tend to take hours to days of computing time to reach an optimal solution, making them a good target for ML-guided optimization techniques to provide high-quality switching decisions on a day-ahead operational time scale.

We determine an easy/hard classification based on the amount of load shed seen in the thresholded problem since, as an LP, this problem can be solved within several minutes. If the thresholded de-energization problem yields relatively low load shed (less than 100 MWh based on our prior experience in [7]), the OPS problem is typically able to find an optimal solution quickly. Classifying days based on the initial load shed leads to 111 easy problems and 540 hard problems. We apply our ML methods on problems classified as hard, as the easy problems can be solved to optimality with Gurobi within a short amount of time. We use 80%, 10%, and 10% of the instances for training, validation, and test. This results in 432, 54, and 54 instances for train, validation, and test, respectively.

V. Results

This section presents our evaluation metrics, implementation setup, the results for the ML-guided optimization method alongside several benchmark methods.

A. Evaluation Metrics

We use the following metrics to evaluate the effectiveness of different methods: (1) The *Objective Value* (OV) is the best known objective value of the MILP found by the time cutoff. (2) The *Primal Gap* (PG) [32] is the normalized difference between the objective value v found by a method and a best known objective value v^*, defined as $\text{PG} = \frac{|v-v^*|}{\max(|v|,|v^*|)}$, when $vv^* > 0$. When no feasible solution is found or when $vv^* < 0$, PG is defined to be 1. PG is 0 when $|v| = |v^*| = 0$. (3) The *Primal Integral* (PI) [32] is the integral of the primal gap over time, which captures the speed at which better solutions are found. (4) The *number of wins in terms of PI* (# wins) is the number of MILP instances in the test set for which the method results in the lowest PI across all other methods. Note that lower values for the objective, primal gap, and primal integral indicate superior performance.

B. Implementation Setup

We use the PyTorch framework to train the ML model, and training is done on an NVIDIA A100 GPU with 128 GB of memory. We use the Adam optimizer [33] with a learning rate of 0.0001. At inference time, we compare our ML-guided approaches with Gurobi and set the time limit to 1,800 seconds for each MILP instance for all methods. We conduct testing on a cluster with epyc-7542 CPUs with 64 GB RAM. To find the hyperparameters ϕ and ϕ' for the ML guided approaches in Section III-C, we experiment with $\phi \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ and $\phi' \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35\}$. We select the

TABLE I: Results of ML-guided methods vs Gurobi. Average Objective Value (OV), Primal Gap (PG), Primal Integral (PI), and number of wins in terms of PI (# wins) across 54 hard test instances.

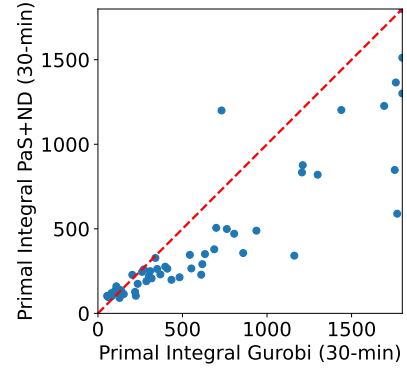| Method	OV	PG	PI	# wins
Gurobi (30-min)	29.60	0.29	627.63	11
PaS (30-min)	29.66	0.27	686.91	1
PaS+ND (30-min)	**15.14**	**0.02**	**433.46**	**42**

Fig. 2: Scatter plot of primal integral with Gurobi vs primal integral with PAS+ND.

Fig. 3: Average primal gap as a function of time (seconds).

values that result in the lowest PI on the validation set and then apply the selected value on the test set at inference time. For the the domain-informed PaS method (PAS), the final hyperparameters used are $\phi = 0.9$ and $\phi' = 0.05$. For the domain-informed PaS+ND with safety net (PAS+ND), $\phi = 0.7$ is used.

C. Results and Discussions

We compare the proposed ML guided approaches (PAS and PAS+ND) against Gurobi. As shown in Table I, PAS+ND performs the best among the three methods in all the evaluation metrics considered, resulting in the lowest average objective value, primal gap, and primal integral and the highest number of wins in terms of primal integral on the test instances. PAS+ND significantly outperforms Gurobi, reducing the primal integral by 30.94% relative to Gurobi. Figure 2 shows a scatter plot of the primal integral with PAS+ND and with Gurobi, where each point represents an MILP instance in the test set. Many points in the scatter plot lie below the 45-degree line, indicating that PAS+ND often outperforms Gurobi. Notably, the biggest gains of PAS+ND are on the hardest instances. Additionally, Figure 3 shows the primal gap, averaged over the test set, as a function of time. PAS+ND produces high-quality solutions at a faster speed across most time steps compared to Gurobi.

PAS performs worse than PAS+ND and Gurobi in most of the metrics considered. A possible explanation is that PAS solves an MILP that contains the same number of binary variables as the original MILP at inference time, as discussed in Section III-C. Therefore, it takes a longer time for PAS to produce solutions with lower objective values, although it provides more flexibility to correct errors from the ML predictions by solving a larger MILP, compared to PAS+ND. The success of PAS+ND on this problem can likely be attributed to the high *recall* value of the ML prediction, which is the ratio of true positives to all actual positives. The *recall* of the ML model with respect to the best ground truth solution is 98.99%, meaning that a high number of variables that take 1 as the solution value in the ground truth data are correctly predicted to 1 by the ML model. Therefore, aggressively fixing a large percentage of variables that has high prediction scores leads to good performance.

D. PSPS Outcomes

To better characterize the physical impacts of the solutions from each solution approach, we detail results from the thresholded version of this problem (denoted as *threshold*, see Section II-B), the ML guided outcomes at a 30-minute time limit (denoted as *ML-guided 30-min*, see Section III, the Gurobi output at a 30-minute time limit (denoted as *Gurobi 30-min*), and extended results at a 24-hour time limit or optimality at a 1% MIP gap (denoted as *Gurobi extended*, see Section II-A). For the ML guided results in this section, we consider the PAS+ND outcomes as this ML methodology achieves better performance, as shown in Section V.
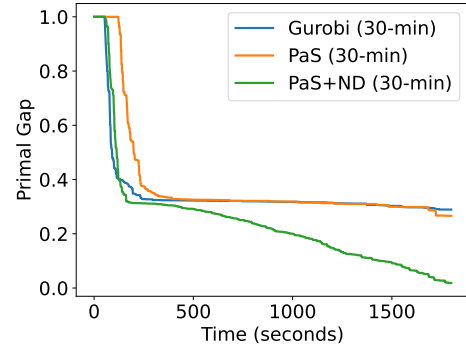
In Figure 4, we show the switching outcomes from these four different solution methodologies on a representative day (March 2, 2021) with high wildfire ignition risk in southern California. We can see that the switching decisions from the Gurobi solution at a 30-minute time limit (Figure 4b) are similar to those from the thresholded version (Figure 4a) with large amounts of load shed. The ML-guided outcomes (Figure 4c) have different switching decisions with much lower load shed when compared to the thresholded and 30-minute Gurobi solutions. The extended Gurobi solution, run to a 24-hour time limit (3.23% MIP gap), achieves much lower load shed with a different network topology. Across the network, there are a large number of lines that remain energized under all solution methodologies, highlighting that the choice of line energization status can greatly impact the amount of load shed.
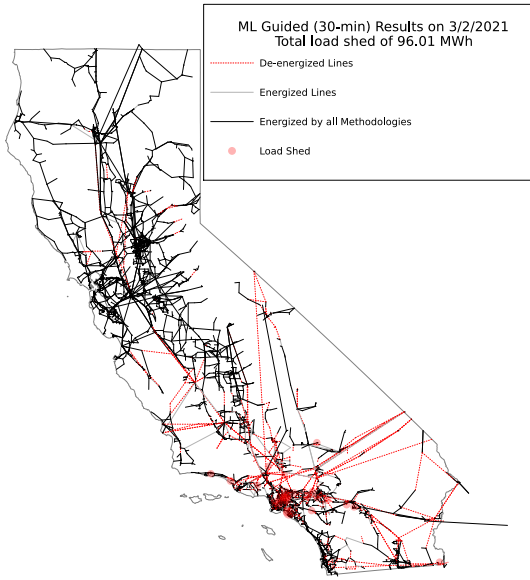
The box plots in Figure 5 show the distribution of objective values across the four different methodologies. The
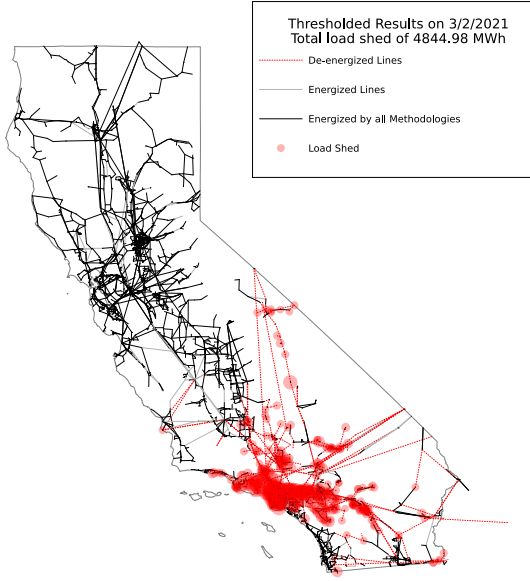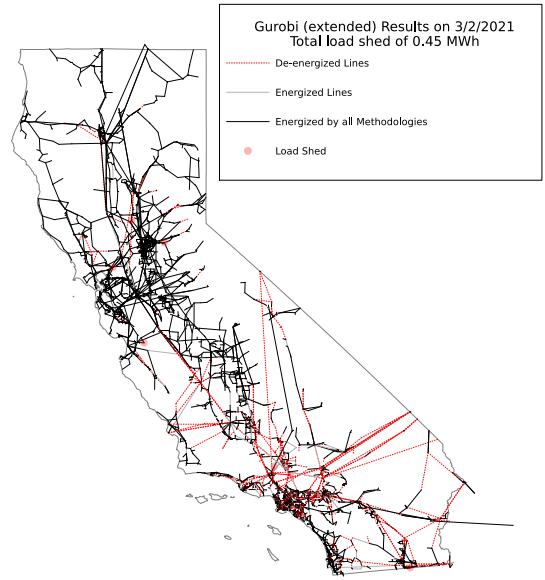
(a) Thresholded line statuses.
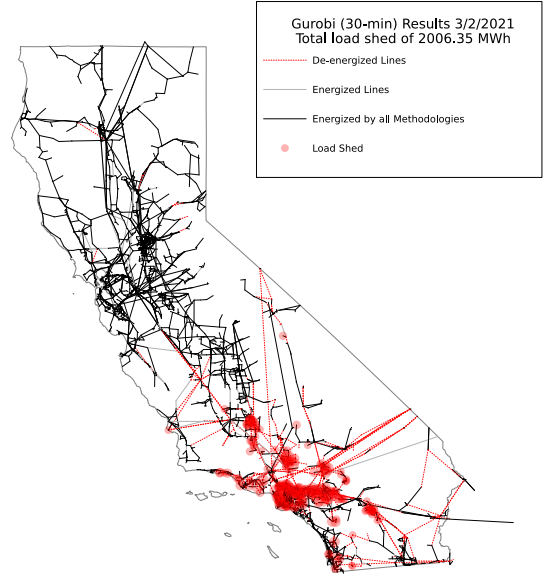
(b) Line statuses from the Gurobi solution at 30 minutes.

(c) Line statuses from the ML-guided solution at 30 minutes.

(d) Line statuses from Gurobi solution at 24 hours

Fig. 4: Geographic plots showing the line statuses and load shed from different solution methodologies for March 2$^{\text{nd}}$, 2021, one of the hard instances. Red dashed lines show lines de-energized under that solution methodology. Black solid lines show lines that are energized across all solution methodologies for this day. Grey lines show lines that are energized in that specific methodology. Red circles indicate load shed at a bus n with the size of the circle corresponding to the amount of load shed.
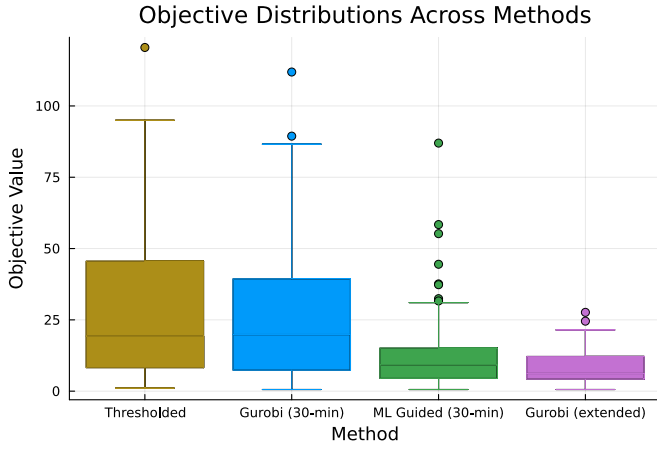
Fig. 5: Box plots showing the distribution of objective outcomes across the four considered solution methodologies. Here, we only include objective values in the distribution from dates that fall in to the hard test data set discussed in Section IV.

thresholded and 30-minute Gurobi results have similar values with the 30-minute Gurobi solutions having a slightly lower objective on average. The ML-guided and extended Gurobi solutions have similar distributions of objective values. While the extended Gurobi does reach more optimal solutions on average, this requires a 24-hour time limit as opposed to the ML guided solutions which are only run for 30 minutes.

VI. CONCLUSIONS

This paper has proposed a domain-informed ML-guided solution method for OPS problems. Compared to Gurobi, the proposed domain-informed PAS+ND method finds high-quality de-energization decisions at a faster speed and reduces the primal integral by 30.94% and primal gap by 93.10% at the 30-minute computational time cutoff. PAS+ND also achieves large reductions in load shedding compared to traditional optimization methods with the same amount of time.

The results in this paper demonstrate that ML-guided MILP methods have significant potential to improve solution speed for important power grid resilience problems. In the context of mitigating wildfire ignition risks, future work aims to extend this work by considering more sophisticated problem formulations that incorporate alternative power flow models [11], multiple time periods [34], restoration models [35], and security constraints [36]. ML-guided MILP methods, especially algorithms based on large-neighborhood search [18], [37], also have potential applications in OPS-based long-term planning problems that optimize infrastructure investment decisions under wildfire ignition risk [19] and other hazards [38], particularly when decomposition methods yield many closely related subproblems [20], [21].

REFERENCES

[1] P. T. Brown, S. J. Strenfel, R. B. Bagley, and C. B. Clements, "The potential for fuel reduction to reduce wildfire intensity in a warming california," *Environmental Research Letters*, vol. 20, no. 2, 2025.

[2] California Department of Forestry and Fire Protection (CalFire), "Top 20 destructive california wildfires," July 2025. [Online]. Available: https://34c031f8-c9fd-4018-8c5a-4159cdff6b0d-cdn-endpoint. azureedge.net/-/media/calfire-website/our-impact/fire-statistics/top20_ destruction_072525.pdf

[3] A. D. Syphard and J. E. Keeley, "Location, timing and extent of wildfire vary by cause of ignition," *International Journal of Wildland Fire*, vol. 24, no. 1, pp. 37–47, 2015.

[4] G. Wong-Parodi, "Support for public safety power shutoffs in California: Wildfire-related perceived exposure and negative outcomes, prior and current health, risk appraisal and worry," *Energy Research & Social Science*, vol. 88, p. 102495, 2022.

[5] D. A. Z. Vazquez, F. Qiu, N. Fan, and K. Sharp, "Wildfire mitigation plans in power systems: A literature review," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 3540–3551, 2022.

[6] N. Rhodes, L. Ntaimo, and L. Roald, "Balancing wildfire risk and power outages through optimized power shut-offs," *IEEE Transactions on Power Systems*, vol. 36, no. 4, 2021.

[7] R. Piansky, S. Taylor, N. Rhodes, D. K. Molzahn, L. A. Roald, and J.-P. Watson, "Quantifying metrics for wildfire ignition risk from geographic data in power shutoff decision-making," *58th Hawaii International Conference on System Sciences (HICSS)*, January 2025.

[8] Pacific Gas and Electric Company, "PG&E wildfire mitigation plan R1 (2026–2028), volume 1: Base WMP," Pacific Gas and Electric Company, Tech. Rep. Vol. 1, February 2026. [Online]. Available: https://www.pge.com/assets/pge/docs/outages-and-safety/ outage-preparedness-and-support/pge-base-wmp-vol-1.pdf

[9] W. Huang, T. Huang, A. M. Ferber, and B. Dilkina, "Distributional MIPLIB: A multi-domain library for advancing ML-guided MILP methods," *arXiv:2406.06954*, 2024.

[10] L. Scavuzzo, K. Aardal, A. Lodi, and N. Yorke-Smith, "Machine learning augmented branch and bound for mixed integer linear programming," *Mathematical Programming*, pp. 1–44, 2024.

[11] E. Haag, N. Rhodes, and L. Roald, "Long solution times or low solution quality: On trade-offs in choosing a power flow formulation for the optimal power shutoff problem," *Electric Power Systems Research*, vol. 234, p. 110713, 2024, *23rd Power Systems Computation Conference (PSCC)*.

[12] J. C. Eidenshink and S. M. Howard, "United States Geological Survey fire science: Fire danger monitoring and forecasting," *US Geological Survey*, 2012, https://www.usgs.gov/publications/united-states-geological-survey-fire-science-fire-danger-monitoring-and-forecasting.

[13] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.

[14] C. Xu, Y. Xie, D. A. Z. Vazquez, R. Yao, and F. Qiu, "Spatio-temporal wildfire prediction using multi-modal data," *IEEE Journal on Selected Areas in Information Theory*, vol. 4, pp. 302–313, 2023.

[15] V. Nair, S. Bartunov, F. Gimeno, I. Von Glehn, P. Lichocki, I. Lobov, B. O'Donoghue, N. Sonnerat, C. Tjandraatmadja, P. Wang *et al.*, "Solving mixed integer programs using neural networks," *arXiv:2012.13349*, 2020.

[16] Q. Han, L. Yang, Q. Chen, X. Zhou, D. Zhang, A. Wang, R. Sun, and X. Luo, "A GNN-guided predict-and-search framework for mixed-integer linear programming," in *11th International Conference on Learning Representations (ICLR)*, 2023.

[17] T. Huang, A. M. Ferber, A. Zharmagambetov, Y. Tian, and B. Dilkina, "Contrastive predict-and-search for mixed integer linear programs," in *41st International Conference on Machine Learning (ICML)*, July 2024.

[18] T. Huang, A. M. Ferber, Y. Tian, B. Dilkina, and B. Steiner, "Searching large neighborhoods for integer linear programs with contrastive learning," in *40th International Conference on Machine Learning (ICML)*, 2023, pp. 13 869–13 890.

[19] A. Kody, R. Piansky, and D. K. Molzahn, "Optimizing transmission infrastructure investments to support line de-energization for mitigating wildfire ignition risk," in *11th IREP Symposium on Bulk Power System Dynamics and Control*, July 2022.

[20] R. Piansky, G. Stinchfield, A. Kody, D. K. Molzahn, and J. P. Watson, "Long duration battery sizing, siting, and operation under wildfire risk using progressive hedging," *Electric Power Systems Research*, vol. 235, Oct 2024, *23rd Power Systems Computation Conference (PSCC)*.

[21] R. Piansky, R. K. Gupta, and D. K. Molzahn, "Optimizing battery and line undergrounding investments for transmission systems under wildfire risk scenarios: A Benders decomposition approach," *Sustainable Energy,*

Grids and Networks, vol. 43, no. 101838, September 2025, *12th IREP Bulk Power system Dynamics and Control Symposium*.

[22] S. Taylor and L. A. Roald, "A framework for risk assessment and optimal line upgrade selection to mitigate wildfire risk," *Electric Power Systems Research*, vol. 213, 2022, *22nd Power Systems Computation Conference (PSCC)*.

[23] S. Taylor, G. Setyawan, B. Cui, A. Zamzam, and L. A. Roald, "Managing wildfire risk and promoting equity through optimal configuration of networked microgrids," in *ACM e-Energy*, June 2023, pp. 189–199.

[24] M. Pollack, R. Piansky, S. Gupta, and D. K. Molzahn, "Equitably allocating wildfire resilience investments for power grids—The curse of aggregation and vulnerability indices," *Applied Energy*, vol. 388, p. 125511, 2025.

[25] M. Yao, M. Bharadwaj, Z. Zhang, B. Jin, and D. S. Callaway, "Predicting electricity infrastructure induced wildfire risk in California," *Environmental Research Letters*, vol. 17, no. 9, p. 094035, 9 2022.

[26] W. Hong, B. Wang, M. Yao, D. Callaway, L. Dale, and C. Huang, "Data-driven power system optimal decision making strategy under wildfire events," in *55th Hawaii International Conference on System Sciences (HICSS)*, January 2022.

[27] S. Taylor, A. Rangarajan, N. Rhodes, J. Snodgrass, B. Lesieutre, and L. A. Roald, "California test system (CATS): A geographically accurate test system based on the California grid," *IEEE Transactions on Energy Markets, Policy and Regulation*, vol. 2, no. 1, pp. 107–118, 2024.

[28] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv:2105.14491*, 2021.

[29] M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi, "Exact combinatorial optimization with graph convolutional neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[30] U.S. Geological Survey, "Wildland Fire Potential Index," 2024. [Online]. Available: https://www.usgs.gov/fire-danger-forecast/wildland-fire-potential-index-wfpi

[31] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2025. [Online]. Available: www.gurobi.com

[32] T. Berthold, "Measuring the impact of primal heuristics," *Operations Research Letters*, vol. 41, no. 6, pp. 611–614, 2013.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[34] H. Yang, N. Rhodes, H. Yang, L. Roald, and L. Ntaimo, "Multi-period power system risk minimization under wildfire disruptions," *IEEE Transactions on Power Systems*, vol. 39, no. 5, pp. 6305–6318, 2024.

[35] N. Rhodes and L. A. Roald, "Co-optimization of power line shutoff and restoration under high wildfire ignition risk," in *IEEE Belgrade PowerTech*, 2023.

[36] N. Rhodes, C. Coffrin, and L. Roald, "Security constrained optimal power shutoff for wildfire risk mitigation," *IET Generation, Transmission & Distribution*, vol. 18, no. 18, pp. 2972–2986, 2024.

[37] J. Song, R. Lanka, Y. Yue, and B. Dilkina, "A general large neighborhood search framework for solving integer linear programs," *33rd Advances in Neural Information Processing Systems (NeurIPS)*, pp. 20012–20023, 2020.

[38] R. Piansky, D. K. Molzahn, N. D. Jackson, and J. K. Skolfield, "Evaluating undergrounding decisions for wildfire ignition risk mitigation across multiple hazards," *IEEE Kiel PowerTech*, June 2025.