Optimization Techniques for Machine Learning
A Study of Linear Programming and Reinforcement Learning for One-Shot Game in Smart Grid Security

CS22B2051 ANSHU SAINI
CS22B2052 KATHIRAVAN
CS22B2053 DHIVYA DHARSHAN V
CS22B2054 ABISHEK CHAKRAVARTHY

09 December 2023

Group 13

Contents

- Introduction
- Problem Formulation and Implementation
- Gaming: Attacker-Defender Interaction
- Simulation Results
- Conclusion

Definition:-

- A smart grid is an advanced electrical infrastructure that utilizes digital technology for improved efficiency, reliability, and sustainability in power generation, distribution, and consumption.

Problem Definition:-

- Integrating smart grid systems with cyber operations introduces security threats, including potential hacker attacks. These pose risks such as cascading failures, leading to significant losses in power, financial, and social domains.

Objective:-

- Creating a simulation model to analyze the dynamic interaction between attackers and defenders, aiming to identify potential attack vectors and devise optimal defense strategies for enhancing the resilience of smart power grid systems.

Utilized Algorithms:-

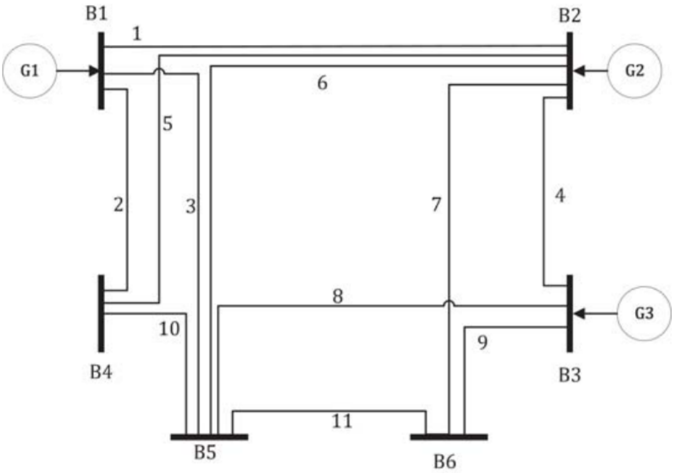
- Linear programming in one-shot game provides multiple solutions for attack-defense sets with probabilities in multi-line-switching attacks.
- Reinforcement learning offers optimal attacker's perspective with single line-switching attack and adaptive defender actions based on historical attacks.

Problem Formulation

Systems used:-

- 6 Bus System for Linear Programming and Reinforcement Learning.
- The topological information is employed to calculate generation loss, with transmission line indices representing the identified attacks in the system.

One-line diagram of 6 Bus System



Problem Formulation

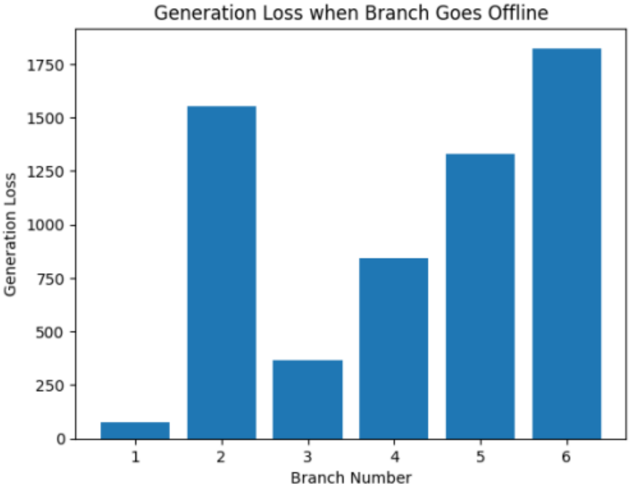
Calculation of Concurrent Overload:

$$\Delta o_j(t, \Delta t) = \begin{cases} \int_t^{t+\Delta t} (f_j(t) - \bar{f}_j)\, dt & \text{if } f_j(t) > \bar{f}_j \\ 0 & \text{otherwise} \end{cases}$$

- For branch j, outage occurs when the concurrent overload o_j surpasses the limit $\bar{o}_j$ based on power flow f_j and flow limit $\bar{f}_j$.
- The simulator determines Δt, advances time, switches branches offline on relay trips, and calculates generation losses for linear programming and reinforcement learning solutions.

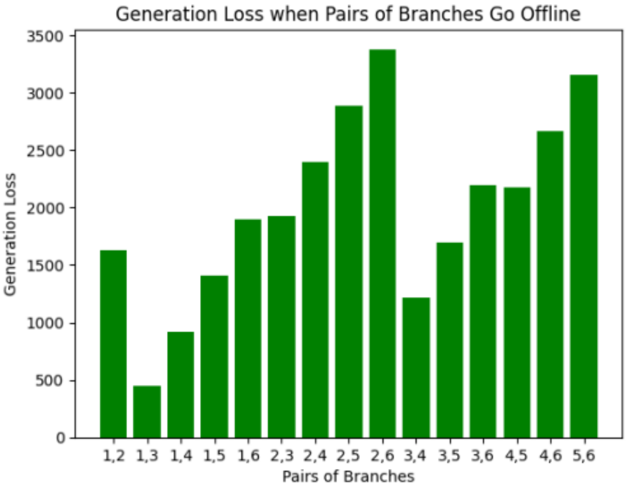
Problem Implementation

Generation Losses:-



Problem Implementation

Target Branches Combinations:-

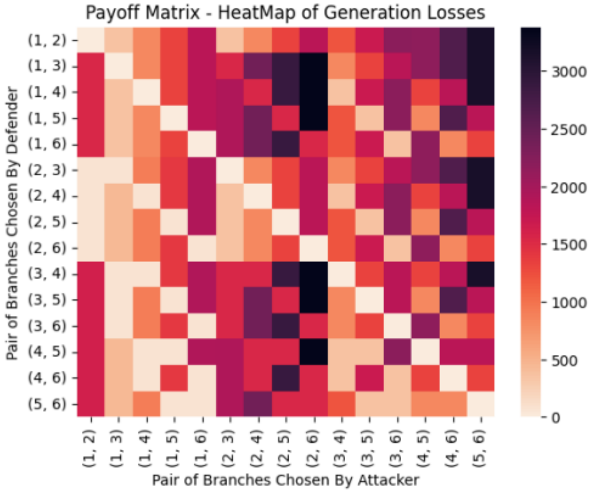


Problem Implementation

Game Matrix:-

```
[[  0  70 115  72 422  70 115  72 422 185 142 492 187 537 494]
 [435   0 115  72 422 435 550 507 857 115  72 422 187 537 494]
 [435  70   0  72 422 505 435 507 857  70 142 492  72 422 494]
 [435  70 115   0 422 505 550 435 857 185  70 492 115 537 422]
 [435  70 115  72   0 505 550 507 435 185 142  70 187 115  72]
 [404 404 519 476 826   0 115  72 422 115  72 422 187 537 494]
 [404 474 404 476 826  70   0  72 422  70 142 492  72 422 494]
 [404 474 519 404 826  70 115   0 422 185  70 492 115 537 422]
 [404 474 519 476 404  70 115  72   0 185 142  70 187 115  72]
 [839 404 404 476 826 435 435 507 857   0  72 422  72 422 494]
 [839 404 519 404 826 435 550 435 857 115   0 422 115 537 422]
 [839 404 519 476 404 435 550 507 435 115  72   0 187 115  72]
 [839 474 404 404 826 505 435 435 857  70  70 492   0 422 422]
 [839 474 404 476 404 505 435 507 435  70 142  70  72   0  72]
 [839 474 519 404 404 505 550 435 435 185  70  70 115 115   0]]
```

Payoff Matrix (Heatmap):-



Linear Programming Approach

Calculation of Average Value:

$$J(y, w) = \sum_{i=1}^{m} \sum_{j=1}^{n} y_i a_{ij} w_j$$
$$= \mathbf{y}^T \mathbf{A} \mathbf{w}$$

Probability distribution vectors:

$$y = (y_1, \ldots, y_m)^T, \quad w = (w_1, \ldots, w_n)^T$$

Here, A is a $m \times n$ matrix, $A = \{a_{i,j} : i = 1, \ldots, m; j = 1, \ldots, n\}$.
y and w are defender's and attacker's probability distribution vectors, respectively.

Gaming: Attacker-Defender Interaction

- The defender aims to minimize $J(y, w)$ by selecting an optimal probability distribution vector $y \in Y$, while the attacker seeks to maximize the same quantity by choosing $w \in W$.

$$Y = \{y \in \mathbb{R}^m \mid y \geq 0, \sum_{i=1}^{m} y_i = 1\}$$

$$W = \{w \in \mathbb{R}^n \mid w \geq 0, \sum_{j=1}^{n} w_j = 1\}$$

Gaming: Attacker-Defender Interaction

- The average game value in the mixed strategies for the attacker-defender zero-sum game can be written as:

$$V_m(A) = \min_{Y} \max_{W} y^T A w = \max_{W} \min_{Y} y^T A w$$

- This equation can be written as:

$$\min_{y \in Y} v_1(y)$$

where

$$v_1(y) = \max_{w \in W} y^T A w \geq y^T A w, \quad \forall w \in W$$

- Re-writting we get:

$$A^T y \leq \mathbf{1}_n v_1(y), \quad \mathbf{1}_n \in \mathbb{R}^n \text{ where } \mathbf{1}_n = (1, \ldots, 1)^T$$

Gaming: Attacker-Defender Interaction

- With the conditions mentioned, we get a maximization problem:

$$\max_{\tilde{y}} \tilde{y}^T \mathbf{1}_m$$

subject to

$$\begin{cases} A^T \tilde{y} \leq \mathbf{1}_n, \\ \tilde{y} \geq 0 \end{cases}$$

here $\tilde{y}$ is defined as $\frac{y}{v_1(y)}$

- The function mentioned above is defender's objective function.

Gaming: Attacker-Defender Interaction

- Similarly, we get attacker's objective function:

$$\min_{\tilde{w}} \tilde{w}^T \mathbf{1}_n$$

subject to

$$\begin{cases} A^T \tilde{w} \geq \mathbf{1}_m, \\ \tilde{w} \geq 0 \end{cases}$$

- For solving these equations, pay-offs from game matrix is considered as input, subject to the constraints.

Reinforcement Learning Approach

- The attacker and defender interact with the power system as the environment.
- The reward is feedback for their actions.
- In a two-person zero-sum game, optimal mixed strategies maximize long-term rewards.
- The attacker's and defender's probabilities of taking actions remain constant over time (stationary policy).

Gaming: Attacker-Defender Interaction

- Quality of state:

$$Q_A(a, d, s) = R_A(a, d, s) + \gamma \sum_{s' \in S} Q_A(s') T(a, d, s, s')$$

- $R_A(a, d, s)$: Attacker's reward for actions 'a' and 'd' by attacker and defender, respectively.
- γ: Impact of current decisions on long-term rewards, ranges from 0 to 1.
- s': Next state.
- $T(a, d, s, s')$: State transition probability, considered equal for all state transitions.

Gaming: Attacker-Defender Interaction

- Value of game:

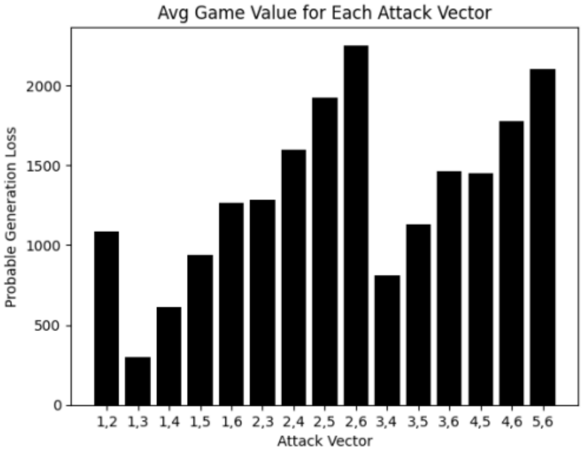
$$V_A(s) = \max_{\pi_A(s)} \min_{\pi_D(s)} \sum_{a \in M_A(s)} \sum_{d \in M_D(s)} \pi_A(s) Q_A(a, d, s) \pi_D(s)$$

where, $\pi_A(s) = \pi_a(s) \mid a \in M_A(s)$, $\pi_d(s) \mid d \in M_D(s)$.

- We assume a fixed defender's action throughout the game, initially determined randomly.

Simulation Results

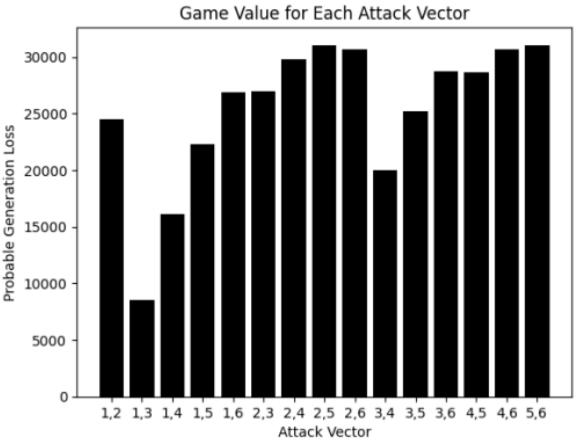
Linear Programming:-



Defender is Static

Simulation Results

Linear Programming:-



Defender is Dynamic

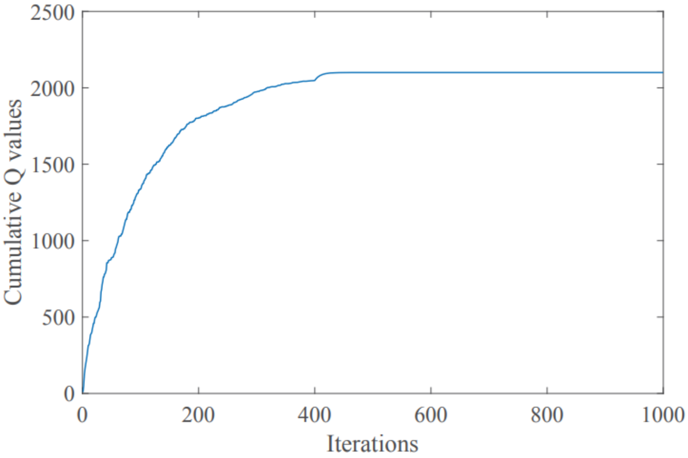
Reinforcement Learning:-

Parameter	Values
Test Case	6 bus system
Number of total transmission lines	11
Number of target transmission lines	4 (30% of total transmission lines)
Maximum generation loss	210 MW
Attacker's optimal action	Transmission line - 5
Defender's fixed action	Transmission line - 2
Gamma, γ	0.9
Epsilon, ϵ	0.4
Total iterations	1000

Reinforcement Learning:-

- Epsilon (ϵ) ranges from 0 to 1, ensuring sufficient exploration in the game environment. With $\epsilon = 0.4$, the attacker explores for 40% of 1000 total iterations.
- Generation loss serves as the reward in solving the two-person zero-sum game through reinforcement learning.

Reinforcement Learning:-



Conclusion

- In conclusion, we explore a dual approach to smart grid security: a pre-calculated linear programming algorithm for multi-line-switching attacks and an online reinforcement learning method for single-line-switching attacks.
- These strategies uncover optimal actions and mixed strategies for both attacker and defender, offering a robust solution for grid security.

Thank You