Power System Cascading Failure Mitigation by Reinforcement Learning

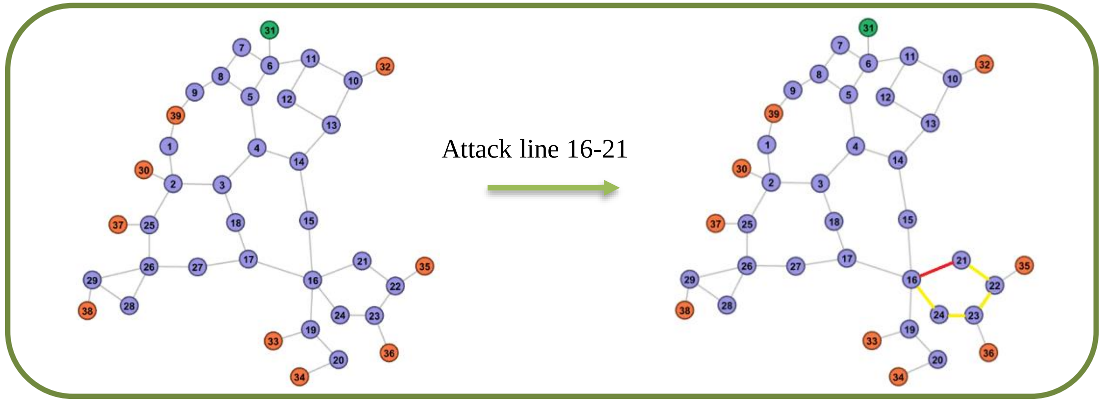
Yongli Zhu
Texas A&M University, USA
June.27.2021

Outline

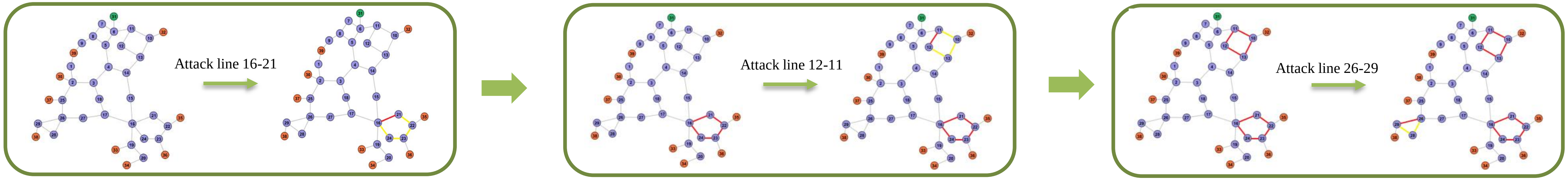
- 1. Motivation of Multi-Stage Cascading Failure
- 2. Formulation of Multi-Stage Cascading Failure
- 3. Mitigation Strategy by RL
- 4. Case Study
- 5. Conclusions and Future Work

1. Motivation of Multi-stage Cascading Failure

- Single-Stage Cascading Failure problem has been widely studied by power systems community



- However, succeeding outage stages can happen one by one closely, e.g. a wind storm happens first, **then** followed by the mis-operation of human operators ➔ Thus, **Multi-Stage Cascading Failure (MSCF)** problem is proposed.

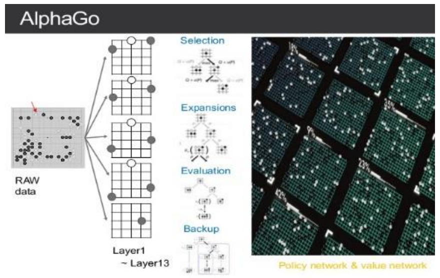


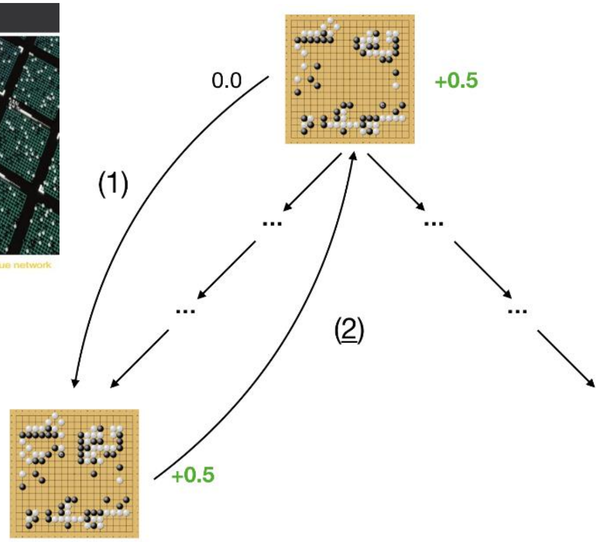
Stage-1 *Stage-2* *Stage-3*

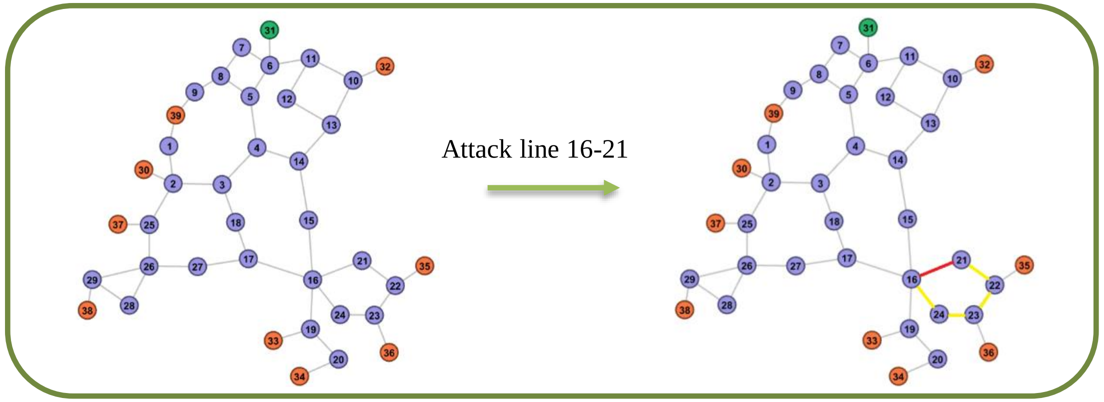
1. Motivation of Multi-stage Cascading Failure

- Can we use any control strategy to mitigate (limit or reduce) such kind of cascading failures? => Yes
  - Strategy options: load shedding, generation adjustment, line switching, transformer tap-ratio change, etc.
- How to determine which control strategy to use and when to use?
  - 1) Conventional approach like SCOPF may be useful for Single-Stage Cascading Failure problem
  - 2) However, for Multi-Stage Cascading Failure, both the timing (order) and type of the consecutive attacks (e.g. faults) can be unknown or stochastic. Only using SCOPF may not handle the MSCF problem well.

- We can resort to data-driven / machine learning methods
- Inspiration from *Alpha-Go* by Google







2. Formulation of Multi-Stage Cascading Failure

- *Generation*: one "event" of the cascading failures within one stage, e.g. a line tripping.

- *Stage*: after an attack (e.g. one line is broken by a natural disaster), the grid evolves with a series of potential *generations*. Finally, the power system will either reach a new equilibrium point if it exists; or the system collapses.

- Example simulation results of the IEEE 118-bus (= node) system for a two-stage MSCF problem in two independent episodes:

* ACPF (alternative current power flow): a set of nonlinear equations that a power grid needs to satisfy when it reaches steady state.

Table 1. Result of Episode-1

		ACPF converge	Over limit Lines
***Stage*-1**			
	Generation-1	Yes	0
***Stage*-2**		**ACPF converge**	**Over limit Lines**
	Generation-2	Yes	0
Result	Win		

Table 2. Result of Episode-2

		ACPF converge	Over limit Lines
***Stage*-1**			
	Generation-1	Yes	2
	Generation-2	Yes	0
***Stage*-2**		**ACPF converge**	**Over limit Lines**
	Generation-1	Yes	4
	Generation-2	Yes	2
	Generation-3	Yes	2
	Generation-4	Yes	3
	Generation-5	Yes	10
	Generation-6	Yes	20
	Generation-7	No	--
Result	Lose		

$$0 = -P_i + \sum^{N} |V_i||V_k|(G_{ik}\cos\theta_{ik} + B_{ik}\sin\theta_{ik})$$

$$0 = -Q_i + \sum_{k=1}^{N} |V_i||V_k|(G_{ik}\sin\theta_{ik} - B_{ik}\cos\theta_{ik})$$

* https://en.wikipedia.org/wiki/Power-flow_study

2. Formulation of Multi-stage Cascading Failure

- **Mimicking the corrective controls by DCOPF**

"Load shedding amount" of each load bus (MW)

$$\min_{p_i, p_j} \quad \sum_{i \in G} c_i p_i + \sum_{j \in D} d_j (p_j - P_{dj})$$

Objective function

$$\text{s.t.} \quad \mathbf{F} = \mathbf{A}\mathbf{p}$$

Branch flow representation

$$\sum_{k=1}^{n} p_k = 0$$

Power balance constraint

$$P_{dj} \le p_j \le 0, \qquad j \in D$$

Load power constraint

$$P_{gi}^{min} \le p_j \le P_{gi}^{max}, \qquad i \in G$$

Generator power constraint

$$-F_L^{max} \le F_l \le F_L^{max}, \qquad l \in L$$

Branch power constraint

- c_i , d_j : generation cost / load shedding cost per unit power (e.g., \$/MW); p_i : generator power (MW)
- P_{dj} : original load power (MW); p_j : load power (MW) (here the sign of electric power is *negative* for load)
- **A**: a constant matrix to associate the net nodal power injections with the branch power flows.
- **F**: a vector of all the branch flows; $\mathbf{p} = [p_k]$, $k = 1 \ldots n$: represents the net nodal power injections.
- n: the total bus number; G, D, L: respectively the generator set, load set and branch set

3. Mitigation Strategy by RL

Applying RL/DRL in Cascading Failure Mitigation

- 1) ***Reward design*** (of each Stage)
  - −Total generation cost (i.e. the negative objective function value of DCOPF) (if converge);
  - −1000, if DCOPF or ACPF diverge;
  - +1000, if system finally reaches a new steady state at the last stage.
- 2) ***Action design***
  - In the previous DCOPF formulation, the "branch flow limit" $\boldsymbol{F_L^{max}}$ is adopted as the action.
- 3) ***State design***
  - [branch_loading_status, V_1, θ_1, P_1, Q_1,…,V_n, θ_n, P_n, Q_n] (voltage magnitude, voltage angle, active power, reactive power)

Environment:
MATLAB + power grid simulation engine

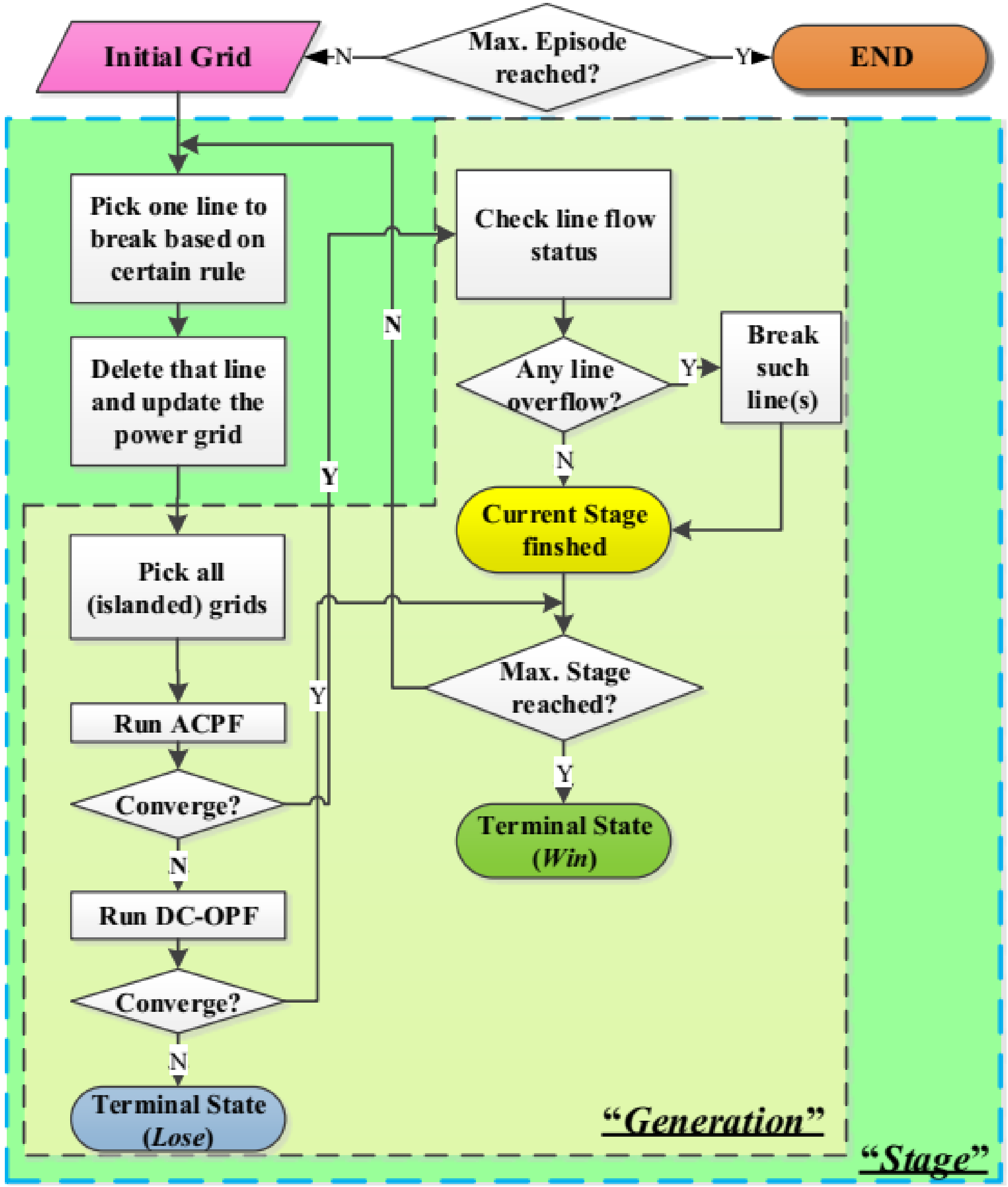
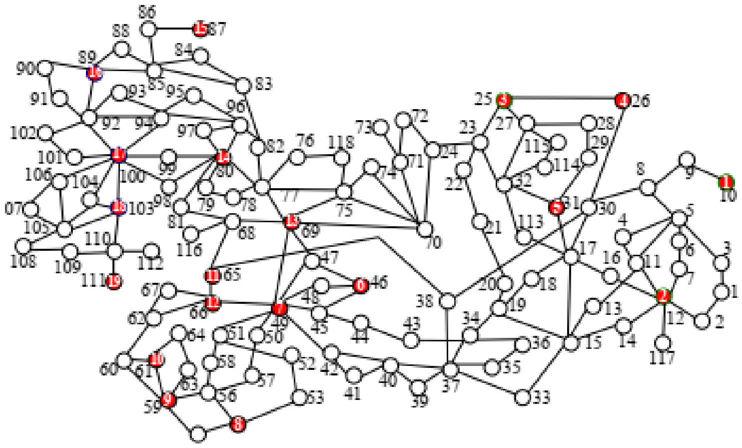


Figure 1. The overall workflow of grid simulation for MSCF study.

4. Case Study

- **Test power grid**:
- IEEE 118-bus system



It contains:

137 buses (nodes)

- 19 generators buses **(red dots)**
- 91 loads buses

186 lines (parallel lines included)

- **Network-1:** *SARSA* (On-policy TD)
- **Shallow Neural Network (RL)**

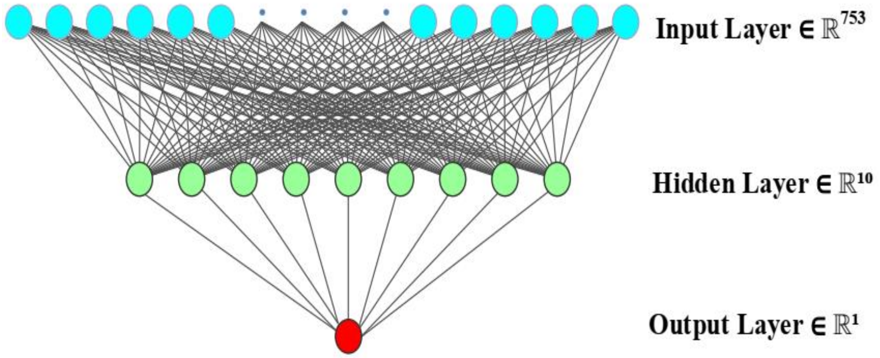


Figure 4. The shallow neural network structure used in RL.

Network architecture is:

- one input layer, one output player
- one hidden layer with 10 neuron units

Input:

- a 1-D vector with 753 (=137 ×4+177+28) elements

Output:

- the action in the RL framework (i.e., the line flow limit F_L^{max})

Action is bounded by [0.80, 1.25]

- **Network-2:** *Q-learning* (Off-policy TD)
- **Deep Neural Network (DRL)**

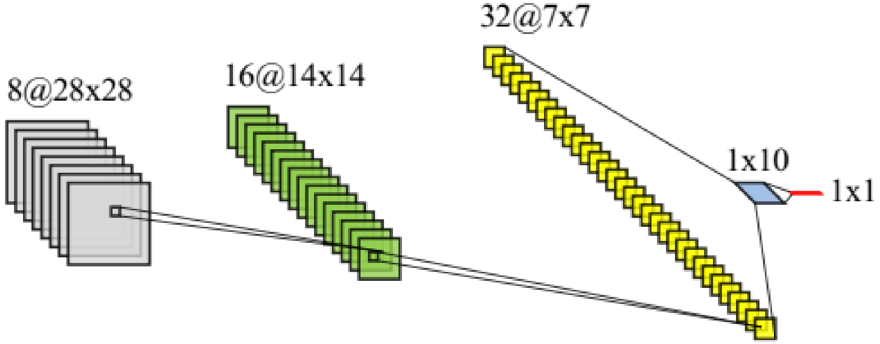


Figure 3. The network structure used in Deep RL.

Image-like input: 784 = 28×28 (extend the original input (length = 753) by padding extra zeros

The output of the 2nd-last layer (dim 1×10) is used in both ε - *greedy* and *greedy* policies

The candidate set of *Action*:
[0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.20, 1.25]

4. Case Study

Table 3. Learning Performance

PERFORMANCE	SHALLOW NETWORK	DEEP NETWORK
Win rate	78.00%	78.07%
Avg. reward	640.08	630.46

Maximum episode number = 10000 (for both networks)

Learning rate = 0.0001, and the discount rate γ = 0.7

Maximum stage number = 3

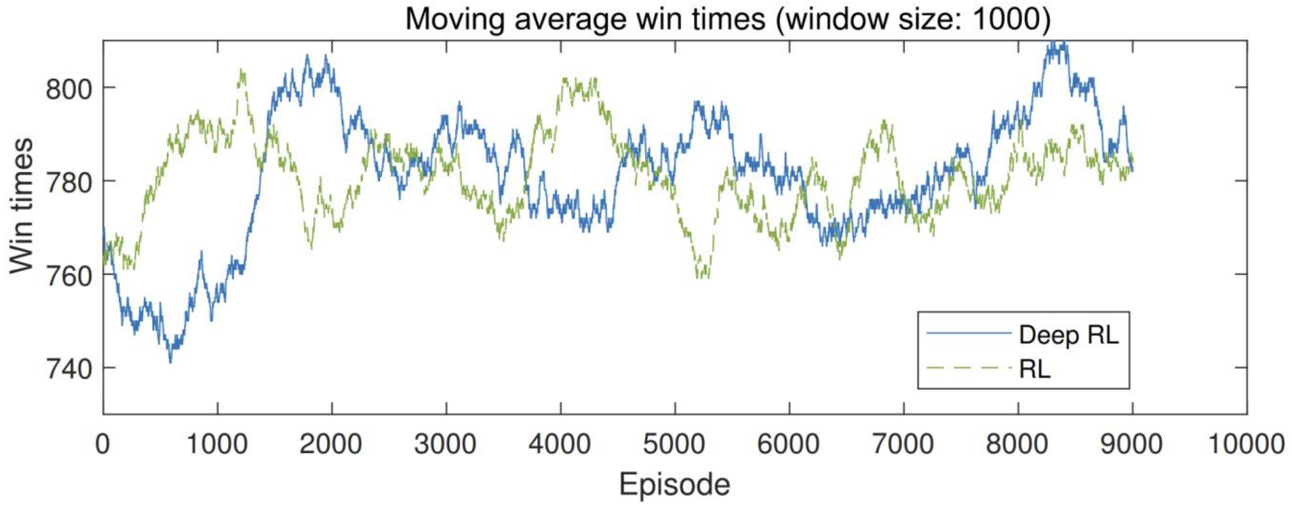


Figure 4. Moving average win times by RL and DRL

It can be observed that:

1) Both RL and Deep RL have achieved satisfactory results in terms of winning rates (i.e., *fewer system collapses*).

2) The higher the average winning rate, the lower the average reward may become; and vice versa.
   - One explanation is: if the system operator (RL agent) is willing to shed (cut) more load then the system typically recovers faster (i.e. toward *winning*); but that way will also increase the obj. function (thus reduce the average reward).

5. Conclusions and Future Work

- A Multi-Stage Cascading Failure (MSCF) problem is proposed and formulated
- A systematic (deep) RL framework is designed for the mitigation of MSCF problem.
- The proposed RL-based mitigation strategy works effectively on the IEEE 118-bus system under both shallow and deep architectures.
- Future work
  - Investigate effects of hyper-parameters (layer numbers, learning rate, discount factor, etc.) of the neural networks on the mitigation performance
  - Consider more control options e.g. transformer tap ratio, energy storge, etc.