DQN BASED DYNAMIC DISTRIBUTION NETWORK RECONFIGURATION FOR ENERGY LOSS MINIMIZATION CONSIDERING DGS

***Se-Heon Lim*[1]*, Leon Fidele Nishimwe H.*[2]*, Sung-Guk Yoon*[3*]**

[1]*Department of Electrical Engineering, Soongsil University, Seoul, Korea*
[2]*Department of Electrical Engineering, Soongsil University, Seoul, Korea*
[3] *Department of Electrical Engineering, Soongsil University, Seoul, Korea*
**sgyoon@ssu.ac.kr*

Keywords: DISTRIBUTION NETWORK RECONFIGURATION (DNR), DEEP REINFORCEMENT LEARNING, ENERGY LOSS, CURTAILMENT OF RENEWABLE ENERGY.

Abstract

In a distribution network with a high penetration rate of solar photovoltaic (PV), curtailment of solar PV's output is inevitable to keep the network stable resulting in a high loss of renewable energy. Without network reinforcement, dynamic distribution network reconfiguration (DNR) that hourly controls the network topology by controlling sectionalizing and tie switches can reduce solar PV curtailment as changing the power flow in the network. This paper aims to minimize total energy loss by using DNR. Power loss is defined as the amount of curtailed output of PV and line losses from power flow. We formulate this problem as a Markov decision process (MDP) and use a deep *Q*-network (DQN) to solve it. DQN algorithm is a data-driven approach for MDP, so it does not require topology information in the dynamic DNR problem, i.e., a model-free characteristic. Furthermore, we adopt dropout to reduce overfitting the training data. A case study shows a performance improvement of the proposed DQN based dynamic DNR algorithm in terms of the total energy loss using a 33-bus distribution test feeder.

1 Introduction

To reduce greenhouse-gas-emission, in the power sector, renewable energy has been increased, mainly with solar and wind. Renewable energy-based distributed generators (DGs) are generally installed into distribution networks, i.e., near the customer side. It causes a bi-directional power flow, which is not considered before, resulting in instability of the network, such as voltage deviation, thermal limits, and protection issues. Therefore, distribution system operators (DSOs) restrict the portion of DGs in each feeder line, i.e., hosting capacity. For some DG installed more than the hosting capacity, renewable energy curtailment is inevitable under the violation of operational conditions. The amount of curtailed energy increases with the amount of DGs in the distribution network. Therefore, it is important to reduce energy loss, including the curtailment of renewable energy. Note that we only consider solar PV as renewable energy, but any renewable energy can be applied.

A fundamental solution to increase hosting capacity while minimizing curtailment is a reinforcement of the distribution network. However, this is a very costly solution in terms of time and budget. Therefore, related research works have proposed to use power devices such as on-load tap changing transformer, energy storage system, switches, inverter, etc [1-3]. Another important solution is dynamic distribution network reconfiguration (DNR), which changes the status of sectionalizing and tie switches in a day [3].

Recent research on dynamic DNR problem used linear programming [4], dynamic programming [5], and a heuristic algorithm [6]. Among them, model-based algorithms such as linear programming and dynamic programming require full information of the model and very accurate prediction. In addition, the computational complexity of the model-based algorithm increases exponentially with the number of switches. Therefore, model-based algorithms are not practical for the dynamic DNR problem. On the other hand, heuristic algorithms take much less computational time, but the performance of these algorithms might not good with a number of switches.

Another approach to solving the dynamic DNR problem is a data-driven algorithm, which does not require a model, i.e., model-free algorithm. In recent years, the capacity and speed of state-of-the-art computer systems enable data-driven approaches a great success in various domains [7], [8]. Therefore, the data-driven algorithm is a practical solution for complex problems like the dynamic DNR problem.

In this paper, we use reinforcement learning (RL), i.e., one sort of data-driven algorithms, to solve the dynamic DNR problem. We formulate this problem as a Markov decision process (MDP) and use a deep *Q*-network (DQN) to solve it. The problem is formulated as a minimization of energy loss defined as a sum of line loss, switching loss, and curtailment of renewable energy. To avoid overfitting the training data, dropout is applied. A case study using a 33-bus distribution test feeder verifies the proposed algorithm's performance and shows that the DQN algorithm is a good approach for the dynamic DNR problem.

2 Deep reinforcement learning

RL is one of the machine learning algorithms to solve the MDP, which describes a sequential decision-making problem using state, action, and reward. Let s_t, a_t, and r_t denote state, action, and reward at time t, respectively. Conventionally, MDPs are solved using dynamic programming, which is a model-based algorithm. On the other hand, RL is a model-free algorithm that solves MDPs using data.

The objective of RL is to find the optimal policy that maximizes expected future returns [9]. A policy π is a function which maps a state to action. In this paper, we use a DQN, which is a Q-learning based algorithm, to solve the dynamic DNR problem. Q-learning trains Q value, which is used to find the optimal policy without model information. The Q-learning algorithm updates the optimal action-value function Q^* as

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \\ -Q(s_t,\ a_t) \qquad (1)$$

where α is a learning rate and γ is the discount factor that determines the weight of future reward. After the Q-value converges via eq. (1), the optimal policy can be obtained by taking the action which gives the largest Q-value among possible actions in each state. That is

$$\pi^*\ : s_t\ \rightarrow \underset{a_t}{argmax}\, Q^*(s_t, a_t). \qquad (2)$$

DQN uses the ANN's output as its Q value. To minimize the loss function of DQN, ANN's weight θ^Q is updated using training data, which is expressed as

$$L(\theta^Q) = E[(r_{t+1} + \gamma \underset{a_{t+1}}{max}\, Q(s_{t+1}, a_{t+1}\ ; \theta^{Q-}) \\ -Q(s_t, a_t; \theta^Q))^2] \qquad (3)$$

where θ^{Q-} denotes the weight of the Q-target network. DQN uses experience tuples to calculate loss function. An experience tuple e_t consists of state, action, reward, and next state, that is

$$e_t = (s_t, a_t, r_{t+1}, s_{t+1}). \qquad (4)$$

3 Problem formulation

In this section, we formulate the objective function of the dynamic DNR as an energy loss minimization problem. The energy loss f is defined as a sum of line loss, switching loss, and curtailment of renewable energy. That is

$$f = \sum_{t=1}^{T}(l_t \times \Delta t + h_t + \sum_{n \in \bar{N}} c_t^n \times \Delta t), \qquad (5)$$

where l_t, h_t, and c_t^n denote line loss, switching loss, and curtailment of renewable energy at bus n at time t. Also, T, Δt, and $\bar{N}$ represent the operational period, time interval, and set of buses with DG, respectively.

3.1 Radiality constraint for dynamic DNR

Because most distribution networks use a radial network, we put an operational constraint to ensure system reliability. The radiality constraint is defined as

$$\sum_{s \in S} x_s = C_s, \qquad (6)$$

where x_s denotes a binary variable that models the on/off status of a line switch s, and S and C_s mean the switches set and a constant, respectively. This constraint implies the number of open [closed] switches should always be the same as the number of the initial radial configuration. However, it is insufficient to describe radiality because it does not consider the case where some buses were isolated from the network. Therefore, in this paper, we exclude the isolated cases by finding divergence of voltage as a result of load flow [3].

3.2 Curtailment of renewable energy

In this paper, we define the output power limit of bus n as maximum power $p_{t,max}^n$ that can be integrated into the distribution network under operational constraints eqs. (8) and (9). These are

$$p_{t,max}^n = \max\ p_t^n \qquad (7)$$

$$V_{min} < |v_t^n| < V_{max} \quad \forall\, n \in N \qquad (8)$$

$$|i_t^l| < I_{max} \quad \forall\, l \in L \qquad (9)$$

Let N and L denote sets of buses and power lines, respectively. Also, v_t^n and i_t^l are voltage at bus n and current on line l at time t, respectively. Constraint (8) describes the operational voltage range that the voltage magnitude for all buses and at any time should lie within the nominal voltage range $[V_{min}\,, V_{max}]$. Also, there is a constraint (9) for power lines, current should be kept less than I_{max}. Now, we can obtain an amount of curtailment c_t^n as

$$c_t^{\bar{n}} = \max(P^n \times \rho_t - p_{max}^n, 0) \qquad (10)$$

where P^n is the capacity of the DG at bus n, and ρ_t is the normalized power output of the DG at time t. normalized power output is obtained using formulas in [10] as

$$\rho_t = \max(\cos(\delta) \times \cos(\varphi) \times \cos\big(15°(t-12)\big) \\ + \sin(\delta)\sin(\varphi)\,, 0) \quad (11)$$

where δ and φ denote latitude and declination, respectively. It is assumed that the normalized power output peaks as 1 when the sun's altitude is 90°.

4 Dynamic DNR model using MDP

In this section, we formulate the dynamic DNR problem as MDP and propose DQN method to solve this problem.

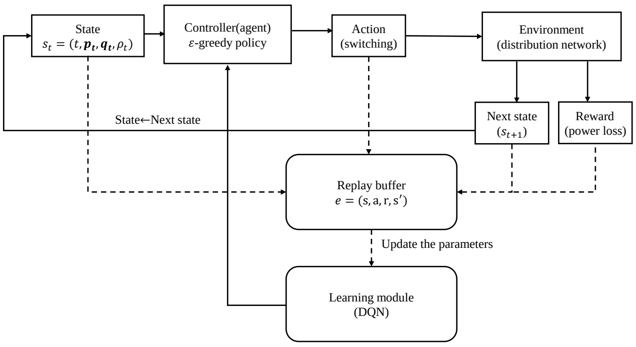

Fig. 1 Block diagram of the training process in DQN.

4.1 Action

In the dynamic DNR problem, an agent performs an action at each time t. It is switching movement among a feasible set of actions that keeps the distribution network radiality after switching events. Switching action x_t^s of switch s at time t can be described as

$$x_t^s = \begin{cases} 0 & \text{if switch off} \\ 1 & \text{if switch on} \end{cases} \tag{12}$$

$$a_t = \{x_s^t \mid s \in S\} \tag{13}$$

4.2 State

State in MDP is information on the environment that can describe the agent's status numerically. States changes according to the agent's action. In dynamic DNR problem, we define the state as

$$s_t = (t, \boldsymbol{p}_t, \boldsymbol{q}_t, a_{t-1}, \rho_t), \tag{14}$$

where vector $\boldsymbol{p}_t$ and $\boldsymbol{q}_t$ denote the active and reactive power of each bus except the slack bus, respectively. Also, ρ_t is included in the state because DG output significantly affects the status of the distribution network.

4.3 Reward

The reward is a function that maps state and action to a number, i.e., reward. The agent learns optimal policy that maximizes the expected future reward, a sum of the rewards received until the operational period ends. Therefore, the reward is very closely related to the objective function. Therefore, we define a reward at t as the negative value of energy loss, which is given as

$$r(s_t, a_t) = -(l_t \times \Delta t + h_t + \textstyle\sum_{n \in \bar{N}} c_t^n \times \Delta t) \tag{15}$$

4.4 DQN Training

Fig. 1 shows the training process of the proposed DQN based dynamic DNR algorithm. The algorithm is divided into two processes. The dotted line is the part where the agent learns parameters using the experience tuples from the replay buffer, and the solid line is the part that takes a virtual action based on the model learned so far. A detailed algorithm to train DQN is shown in Algorithm 1. Here, M U, and B are the number of episodes, the update period for Q-target parameters, a set of experience tuples stored in the replay buffer, respectively.

Algorithm 1 : DQN Training Algorithm

```
Initialize:
replay buffer $B$, DQN parameter $\theta^Q$
for $i = 1, \dots, M$
    $\varepsilon = \max(\varepsilon_{max} - k \cdot i, \varepsilon_{min})$
    for $t = 1, \dots, T$
        Select an action $a_t$ by $\varepsilon$-greedy policy eq. (15)
        Take action $a_t$ and observe $r(s_t, a_t)$ and $s_{t+1}$
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $B$
        Random sampling for minibatch from $B$
        Update $\theta^Q$, which minimize its loss $L(\theta^Q)$ eq. (3)
    end for
    if $mod(i, U) == 0$
        Update the target $Q$ network: $\theta^{Q-} \leftarrow \theta^Q$
    end if
end for
Output : policy $\pi : s_t \rightarrow \underset{a}{\operatorname{argmax}}\, Q(s_t, a, \theta^Q)$
```

4.5 Performance Improvement

To improve the performance of the proposed DQN algorithm, we use two techniques: ε-greedy policy [11] and the dropout [12].

In our work, we adopt an ε-greedy policy as a behavior policy to balance between exploration and exploitation. For a stable and efficient model training, we reduce ε linearly with k according to i, as shown in the third line, Algorithm 1. The efficiency of learning can be improved more by focusing on exploration in the early period, and more on exploitation as the algorithm proceed.

The real data of DG's output might have different characteristics to the training data. Therefore, if the DQN is overfitting to the training data, the actual performance is lower than that with training data. To enhance the generalization capability of the proposed DQN, we use the dropout technique. Dropout randomly removes some neurons between layers. The probability of removing neurons is called the dropout rate.

5 Case study

This section shows the performance of the proposed DQN based dynamic DNR in terms of energy loss compared to a myopic algorithm and dynamic programming. The myopic algorithm chooses an action that minimizes current energy loss, which is obtained through power flow calculation every hour. Dynamic programming can achieve the global optimal solution under the assumption of perfect knowledge of future information.

5.1 Test system description

We use a 33-bus distribution network test feeder, as shown in Fig. 2 [13]. Basic distribution network parameters are obtained from Korea Power Cooperation (KEPCO). The power base and nominal voltage are 15 MVA and 22.9 kV. In this distribution network, there is one substation (154/22.9 kV) which supplies power to 33 buses, five sectionalizing switches (solid lines), five tie switches (dotted line), and three solar PVs of 3 MW capacity. Note that solar PVs at the end of the feeder make the distribution network most vulnerable to voltage deviation.

It is assumed that the power factor for all buses is 0.9. We use real load data for 2017 in the Midwest in the US [14], and then the load is scaled up to match the load capacity of KEPCO standard as 10 MVA [15]. The voltage tolerance limit was set from 0.91 p.u to 1.04 p.u according to KEPCO standard [15]. The maximum allowable current is 395 A from ACSR-OC 160mm^2 line's specification. We set the switching loss as 5 kWh/action, which comes from a conversion using the cost of power [16].

5.2 Simulation setting

For simulations, we use 30 days (1/1/2017 - 1/30/2017). Among them, the first 25 days and the other five days are used for training and testing the proposed DQN, respectively. The learning rate α, discount factor γ, buffer size, batch size, and update period U are set as 0.005, 0.85, 5000, 128, and 60, respectively.

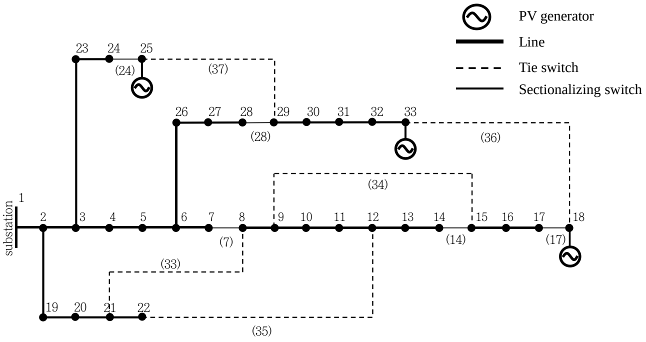

Fig. 2 A 33-bus distribution network test feeder.

Fig. 3 shows the simulation results. With the number of episodes, the total energy loss of the training period decreases, and then it converges.

5.3 Performance evaluation

Table 1 shows the performance of DNR algorithms in terms of energy loss. The bold font stands for the minimum loss. As shown in Table. 1, the average energy loss is 1229 kWh at fixed configuration which does not change switch status at all. When switches operate in a random manner, the loss is even greater than that of the initial configuration because of the switching loss. The energy loss of the proposed algorithm is 432 kWh, i.e., a reduction of 64.8% compared to the fixed configuration. It is almost the same performance as dynamic programming, which requires perfect future information.

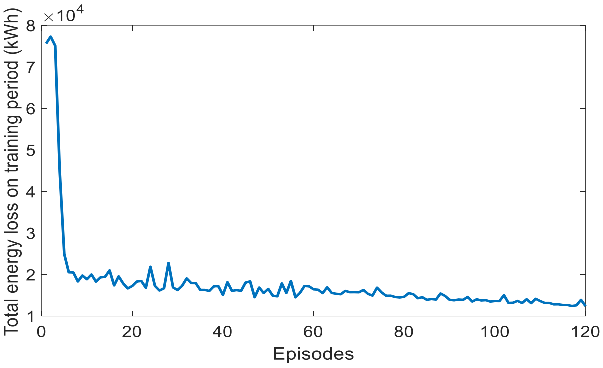

Fig. 3 Training result of DQN.

The proposed algorithm shows better performance than myopic for all the five days. A small number of switching actions is better because frequent actions reduce the lifespan of the switch. The average number of switching per day of the proposed algorithm is less than one per switch.

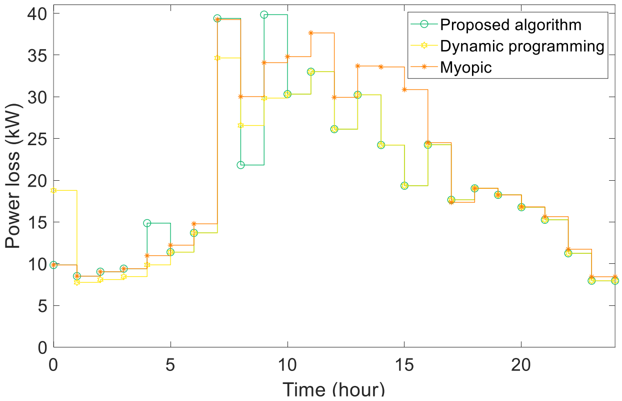

Fig. 4 Comparison real-time power loss of day 1

Energy loss for day 1 is shown in Fig. 4 to check the proposed algorithm's behavior. The proposed algorithm takes the same action as the Myopic algorithm from 0:00 to 4:00, and then it does as dynamic programming from 4:00 to 8:00 and 10:00 to 24:00. The proposed algorithm takes the same action as 75% of dynamic programming. With discount factor γ as 0.75, 0.85, and 0.95, the energy losses are 438 kWh, 432 kWh, and 451 kWh, respectively, so we set γ as 0.85. Discount factor γ as 0.85 is lower than the usual value of 1. That is, this algorithm focuses more on the current situation. This implies that the myopic algorithm also shows a good performance in this dynamic DNR problem. Because the proposed algorithm uses γ as 0.85, the agent partly takes the same actions as the myopic algorithm.

Table. 1 Energy loss comparison of different optimization methods

Method	Energy loss (kWh)						Switching number per day
	Day 1	Day 2	Day 3	Day 4	Day 5	average	(average)
Fixed configuration	872	615	1569	1565	1525	1229	0
Random action	1302	1930	1786	1179	1354	1510	108
Proposed algorithm	471	**435**	**423**	410	**419**	432	9.5
Myopic	510	463	449	412	430	453	8.8
Dynamic programming	**463**	**435**	**423**	**409**	**419**	**430**	**8.4**

6 Conclusion

With a high penetration of renewable energy, it might not be possible to use renewable energy fully. To minimize this energy loss as well as line loss, we propose a DQN based dynamic DNR algorithm in distribution networks. To improve the performance of DQN, ε-greedy policy and dropout are applied. Simulation results using 33 bus system confirms that the proposed algorithm reduces energy loss by 64.8% compared to fixed configuration. Its performance is very close to the ideal solution obtained by the dynamic programming method. In future work, more control variables such as shunt capacitors, smart inverters, smart transformer, and a multi-agent reinforcement learning algorithm are required.

7 Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1F1A1075137).

8 References

[1] Seuss, J., Reno, M. J., Broderick, R. J., & Grijalva, S.: 'Improving distribution network PV hosting capacity via smart inverter reactive power support', 2015 IEEE Power & Energy Society General Meeting, July 2015, pp. 1-5

[2] Kim, Y. H., Myung, H. S., Kang, N. H., Lee, C. W., Kim, M. J., & Kim, S. H.: 'Operation Plan of ESS for Increase of Acceptable Product of Renewable Energy to Power System', The Transactions of The Korean Institute of Electrical Engineers, 67(11), 2018, pp. 1401-1407

[3] Capitanescu, F., Ochoa, L. F., Margossian, H., & Hatziargyriou, N. D.: 'Assessing the potential of network reconfiguration to improve distributed generation hosting capacity in active distribution networks', IEEE Transactions on Power Systems, 30(1), 2014, pp. 346-356

[4] Novoselnik, B., & Baotić, M.: 'Dynamic reconfiguration of electrical power distribution networks with distributed generation and storage', IFAC-PapersOnLine, 48(23), 2015, pp. 136-141

[5] Feinberg, E. A., Hu, J., & Huang, K. (2011, October).: 'A rolling horizon approach to distribution feeder reconfiguration with switching costs', IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011, pp. 339-344

[6] Bernardon, D. P., Mello, A. P. C., Pfitscher, L. L., Canha, L. N., Abaide, A. R., & Ferreira, A. A. B.: 'Real-time reconfiguration of distribution network with distributed generation', Electric Power Systems Research, 107, 2014, pp. 59-67

[7] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D.: 'Mastering the game of go without human knowledge', nature, 550(7676), 2017, pp. 354-359

[8] Kober, J., Bagnell, J. A., & Peters, J.: 'Reinforcement learning in robotics: A survey', The International Journal of Robotics Research, 32(11), 2013, pp. 1238-1274

[9] Sutton, R. S., Barto, A. G.: 'Reinforcement learning: An introduction' (MIT press, 2nd edn. 2018)

[10] Lim. S, Kim. T, Yoon. S.: 'Distribution Network Reconfiguration to Minimize Power Loss Using Deep Reinforcement Learning', The Transaction of KIEE, 69(11), 2020, pp. 1659-1667

[11] Tokic, M., Palm, G.: 'Value-difference based exploration: adaptive control between epsilon-greedy and softmax'. In Annual Conference on Artificial Intelligence, Springer, Berlin, Heidelberg, October 2021, pp. 335-346.

[12] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R.: 'Improving neural networks by preventing co-adaptation of feature detectors', arXiv preprint arXiv:1207.0580, 2012

[13] Mishra, S., Das, D., & Paul, S.: 'A comprehensive review on power distribution network reconfiguration. Energy Systems', 8(2), 2017, pp. 227-284

[14] 'Z. Wang: Dr. Zhaoyu Wang's homepage', http://wzy.ece.iastate.edu/ Testsystem.html, accessed Jun. 2019

[15] Korea Electric Power Company (KEPCO).: 'Regulation on the Use of Electrical Equipment for Transmission Distribution System (Appendix4)', http://cyber.kepco.co.kr/ckepco/front/jsp/CY/H/C/CYHCHP00704.jsp (accessed on Jan. 8 2021).

[16] Gao, Y., Shi, J., Wang, W., & Yu, N.: 'Dynamic distribution network reconfiguration using reinforcement learning', IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), October 2019, pp. 1-7 IEEE.