

소비 패턴을 고려한 그리드 환경에서의 배터리 최적화 비교 분석

노하진¹, 임유진²

¹숙명여자대학교 IT공학과 석사과정

²숙명여자대학교 인공지능공학부 교수

hajins@sookmyung.ac.kr, yujin91@sookmyung.ac.kr

Comparative Analysis of Battery Optimization in Grid Considering Consumption Patterns

Hajin Noh¹, Yujin Lim²

¹Dept. of IT Engineering, Sookmyung Women's University

²Div. of Artificial Intelligence Engineering, Sookmyung Women's University

요약

현재 전력망에서는 불규칙하거나 낭비되는 전력 문제를 해결하기 위한 한 방법으로 ESS(Energy Storage System)를 활용하는 방법이 많은 관심을 받고 있다. 본 연구에서는 업종별로 시간대에 따라 요금을 다르게 부과하는 배전망 시스템에서, 배터리를 보다 경제적으로 사용하는 동시에 여유 용량을 유지하도록 하는 DQN 기반 강화학습 기법을 제안하였다. 또한, 업종별로 다른 전력 소비 패턴을 에이전트의 동작성과 함께 그 성능을 분석하고 비교하였다.

1. 서론

탄소 감축의 목소리가 커지면서 탄소 배출이 비교적 적은 재생 에너지 사용이 확대되고 있다. 하지만, 재생 에너지는 기상 조건에 따라 발전량이 불규칙적으로 달라진다는 단점이 있다. 반면 화력, 원자력 등의 발전은 전기를 지속적으로 공급할 수 있지만, 전력 수요가 적은 시간대에는 전력이 낭비된다는 문제점이 있다. 이러한 에너지의 불안정한 수급 및 낭비 문제를 극복하기 위해 전기를 저장하였다가 필요시 사용할 수 있도록 하는 ESS가 많은 주목을 받고 있다. 이를 통해 전력 공급을 원활하게 함과 동시에 버려지는 에너지를 활용할 수 있게 되었다.

현재 배전망은 소비자 업종과 시간대에 따라 요금에 차등을 두고 있다. 하지만, 동일 요금대로 나뉘더라도 세부 업종별로 전력 소비 패턴이 다르다. 소비 패턴과 비용 문제를 고려하여 ESS를 활용한다면 더 효율적이고 경제적인 전력 소비가 될 것이다.

2. 관련 연구

[1]은 스마트 그리드 환경에서 전기 구매 및 판매 비용을 최소화하고, ESS가 시간대에 따라 일정 범위 내에서 충전 및 방전하는 양을 정하도록 하였다. 또한, 강화학습 기법인 DQN, DDPG, TD3의 성능

비교를 진행하였다. 하지만, 가정에서 사용하는 에너지 환경만을 고려하였다는 한계점이 있다. [2]에서는 스마트 그리드 환경에서 강화학습 기법인 SA-RSA 및 Q-learning 방식의 멀티 에이전트 강화학습을 이용하여 일반 전기 요금과 PAR(Peak to Average Ratio)를 줄이고자 하였다. 하지만 이 역시 주택 환경만을 가정한다는 한계점이 있다. [3]은 ESS와 태양광 발전이 포함된 그리드 환경에서 수요량과 태양광 발전량을 예측하고, 예측한 데이터를 멀티 에이전트 강화학습 PPO에 사용하였다. 해당 연구는 운영 비용 절감과 탄소 배출 감소를 목표로 하였다. 다양한 건물의 데이터를 사용하였으나, 모두 주택 데이터라는 한계점도 존재한다.

본 연구에서는 다양한 업종에서 태양광 에너지 발전이 존재한다고 가정한 후, ESS 충전 및 방전을 수행하기 위해 불연속적인 환경에서 높은 성능을 보이는 강화학습 기반 DQN 알고리즘을 사용한다. 이를 통해 전기 요금을 절감하고, 배터리 충전 수준을 유지하여 에너지를 효율적으로 관리한다. 더불어, 시간대별 요금 및 다양한 업종별 전력 소비 특성과 연관 지어 에이전트의 동작성을 분석한다.

3. 시스템 모델

3-1. 목표 및 환경 설정

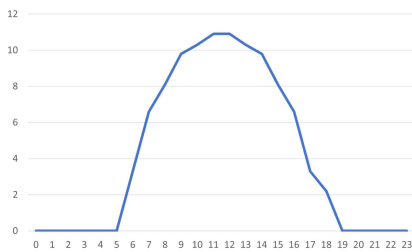
본 논문에서 DQN 에이전트는 소비자의 누적 전기 요금을 감소시키는 동시에 배터리 충전량을 일정 수준 이상 주중인 5일간 유지하는 것을 목표로 한다. 실험 환경은 병원, 공장, 학교, 가게 형태의 서로 다른 소비자가 각각 태양광 발전기와 배터리를 가지고 있다고 가정한다. 또한, 태양광 에너지를 1순위로 사용하여 남은 태양광 에너지는 없다고 가정한다.

일일 수요량은 전력통계정보시스템이 제공하는 2020~2022년 7, 8월 계약종별 전력 사용량[4]의 평균과 최대, 최소를 사용하였으며 <표 1>과 같다.

<표 1> 업종별 일일 수요량 범위(단위: kWh)

	병원	공장	학교	가게
최소	249	1900	962	82
평균	287.9	1998.1	1047.6	95.8
최대	314	2115	1218	109

각 업종의 배터리 총용량은 일일 수요량 평균의 10%로 설정하였다. 비상시를 대비하여 유지해야 할 최소 용량은 배터리의 40%로, 이는 정전 시 1-2시간을 운영할 수 있는 전력량이다. 또한, 자연 방전을 고려하여 배터리의 80% 이하를 유지하도록 설정하였다. 일일 태양광 발전량은 각 일일 수요량의 5-10% 범위에서 무작위로 설정하였다. (그림 1)은 시간대별 태양광 발전 비율로, 여름철 일출과 일몰 시간을 고려하여 임의로 설정하였다.



(그림 1) 시간대별 태양광 발전량 비율

<표 2>와 <표 3>은 한국전력공사[5]가 제공하는 데이터로, 이를 기준으로 부하 시간과 업종별 전력 요금을 설정하였다. 기본요금은 고려하지 않았다.

<표 2> 여름철(6월~8월) 시간대별 구분

경부하	중간부하	최대부하
22:00~08:00	08:00~11:00 12:00~13:00 18:00~22:00	11:00~12:00 13:00~18:00

<표 3> 시간대에 따른 업종별 전력 요금(단위: 원/kWh)

	병원	공장	학교	가게
경부하	87.2	82.3	76.5	87.2
중간부하	113.0	108.1	121.2	113.0
최대부하	146.5	141.6	187.1	146.5

시간대별 소비 비율은 한국전력공사에서 제공하는 2021년 전력소비행태분석[6] 데이터를 이용하였

다. 산업 분류 중 보건업, 음료 제조업, 교육 서비스업, 음식 및 주점업을 각각 병원, 공장, 학교, 가게로 간주하였다.

3-2. 제안하는 알고리즘

DQN 알고리즘은 경험 버퍼와 타겟 네트워크를 이루며, 경험 버퍼에는 학습 과정 중 얻어지는 정보들이 저장되어 더욱 효과적으로 가중치를 업데이트하도록 한다. 또한, 타겟 네트워크를 통해 학습이 안정적으로 진행되도록 한다.

배터리가 취할 수 있는 액션은 총 9가지로, 충전, 방전, 유지이며, 그 단위는 B_{\max} 에 0.025, 0.05, 0.075, 0.1을 곱한 것과 같다. 유지의 경우는 0이다.

변수 및 정의는 <표 4>와 같다. 이때, 배터리 관련 변수의 범위는 $0 < BG_{\min} < BG_{\max} < B_{\max}$ 이며, $0 \leq B_t \leq B_{\max}$ 이다. G_t 와 pg_t , pgb_t 의 계산식은 식 (1), (2), (3)과 같다.

<표 4> 변수 및 정의

변수	정의
L_d	d 일에서의 일일 수요량
r_t	시간 t 에서의 수요량 비율
RE_d	d 일에서의 일일 태양광 에너지 발전량
rer_t	시간 t 에서의 태양광 에너지 발전량 비율
G_t	시간 t 에서의 일반 전력 소비량
c_t	시간 t 에서의 단위 요금
pg_t	배터리를 고려하지 않았을 때의 요금
pgb_t	배터리를 고려했을 때의 요금
a_t	시간 t 에서의 배터리 충/방전 액션
B_t	시간 t 에서의 배터리 충전 상태
B_{\max}	배터리 최대 충전량
BG_{\min}	목표 최소 충전량
BG_{\max}	목표 최대 충전량

$$G_t = L_t \times r_t - RE_d \times rer_t + a_t \quad (1)$$

$$pg_t = (L_t \times r_t - RE_d \times rer_t) \times c_t \quad (2)$$

$$pgb_t = G_t \times c_t \quad (3)$$

보상 함수는 세 가지 경우로 나뉘며, 식 (4)와 같다. 첫 번째 경우는 B_t 가 목표 범위 내에서 유지될 때로, 가격 보상만을 고려한다. 이때, 가격은 초기부터 현재 시간 t 까지의 누적 요금이다. 두 번째와 세 번째 경우는 목표 범위를 벗어났을 때로, 목표 충전량과 현재 충전량의 차이를 패널티로 부과한다. 두 패널티의 분모는 최대-최소 스케일링으로 정규화하는 과정에서 설정된 값이다. 두 번째 경우, 패널티의 가중치인 2.5는 가격 보상과 패널티의 영향력이 비

숫하도록 설정한 값이다.

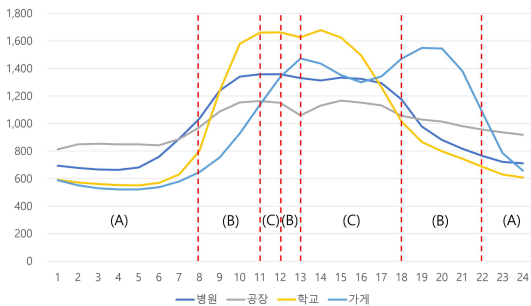
$$penalty = \begin{cases} \frac{\sum_{t=0}^t pg_t}{\sum_{t=0}^t pgb_t} & (\text{if } BG_{\min} \leq B_t \leq BG_{\max}) \\ \frac{\sum_{t=0}^t pg_t}{\sum_{t=0}^t pgb_t} - 2.5 \times \frac{BG_{\min} - B_t}{BG_{\min}} & (\text{if } 0 \leq B_t < BG_{\min}) \\ \frac{\sum_{t=0}^t pg_t}{\sum_{t=0}^t pgb_t} - \frac{B_t - BG_{\max}}{B_{\max} \times 0.2} & (\text{if } BG_{\max} < B_t \leq B_{\max}) \end{cases} \quad (4)$$

4. 실험 결과 및 분석

총 500개의 에피소드를 학습하였으며, ESS의 설정값은 <표 5>와 같다. (그림 2)는 8월 업종별 소비 전력 형태이며, 시간대별 수요량은 각 패턴을 따른다. (A)는 경부하, (B)는 중간부하, (C)는 최대부하 시간대이다.

<표 5> ESS 목표 용량 및 초기 용량(단위: kWh)

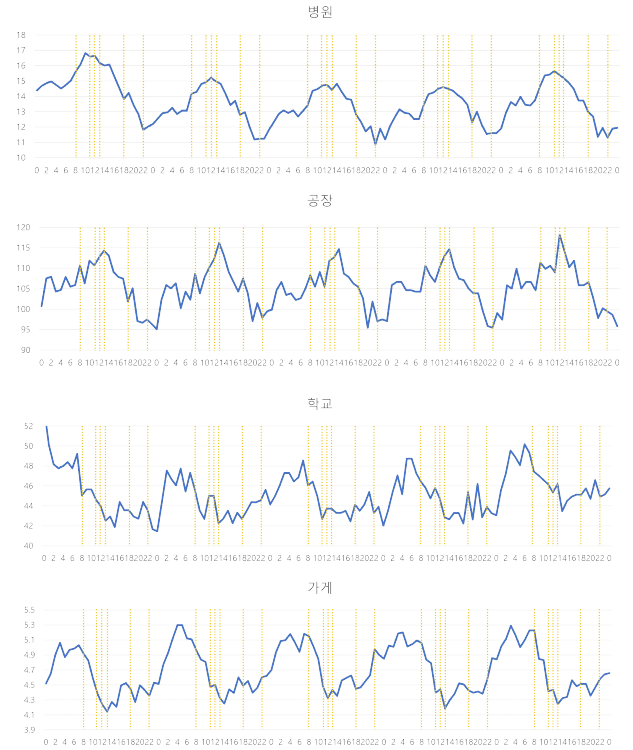
	병원	공장	학교	가게
BG_{\max}	23.0	159.8	83.8	7.7
B_0	15.0	103.9	54.5	5.0
BG_{\min}	11.5	79.9	41.9	3.8



(그림 2) 8월 업종별 소비 전력 행태

병원, 공장, 학교는 소비가 낮에 주로 집중되어 있으며, 가게는 점심부터 저녁 시간에 집중되어 있다. 병원의 경우, 5시부터 전력 소비가 증가하다 10-17시에 가장 많은 전력을 소비하며, 17시 이후에는 감소한다. 공장을 제외한 다른 업종과 비교했을 때, 영업이 끝난 새벽 시간대에도 비교적 높은 전력 소비를 보인다. 공장의 경우, 최대 소비 전력과 최소 소비 전력의 차이가 가장 낮다. 6시부터 소비가 증가하며, 점심시간인 12-13시에는 약간 감소한다. 학교도 마찬가지로 6-11시까지 급격하게 증가하다가 점심시간에 약간 감소한다. 가게의 경우, 6시부터 점심인 13시 전까지 증가하다가 16시까지 감소한다. 이후 저녁 시간대에 다시 전력 소비가 증가한다. 나머지 업종에 비해 비교적 늦은 시간대에 소비가 집

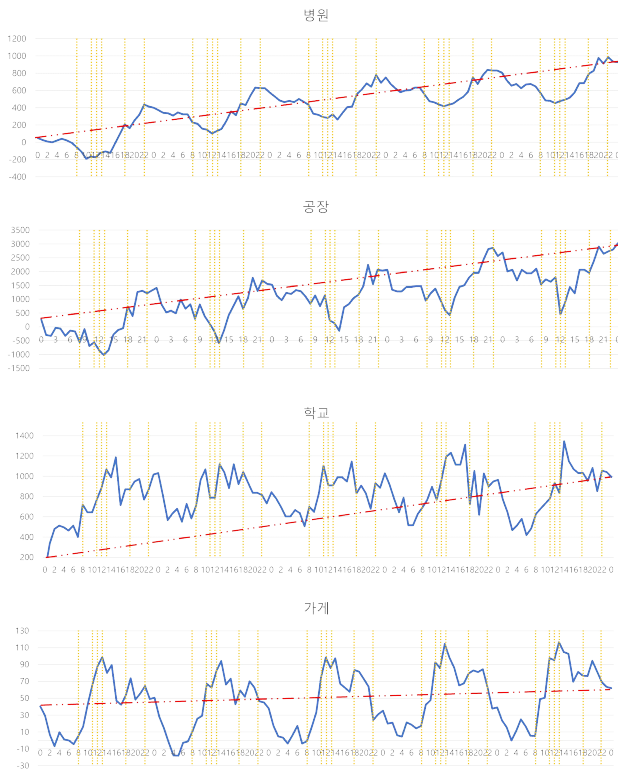
중되어 있다.



(그림 3) 업종별 시간에 따른 ESS 충전량 상태

(그림 3)은 업종별 에피소드 10개의 5일간의 ESS 충전량 상태 평균 수치이다. 병원의 ESS 상태 추이를 살펴보면, 경부하 시간은 물론 오전 중간부하와 최대부하 시간을 포함한 12시까지도 약 15kWh까지 배터리를 충전하는 것을 확인할 수 있다. 하지만, 이후 22시 이전까지는 계속 방전하며 약 4kWh의 충전량 차이를 보인다. 병원은 다른 업종에 비해 최대 수준의 전력을 오랫동안 소비하므로, 오전 최대부하 시간에 충전하더라도 오후 최대부하 시간에 더 오래 방전함으로써 요금 절약이 가능하다. 공장 배터리의 동작성을 관찰했을 때, 0시부터 13시까지 충전하여 하루 중에 가장 큰 약 115kWh 충전량을 보인다. 이후 22시까지 약 20kWh를 사용하며, 경부하 시간대에는 다시 충전된다. 학교는 약 6시부터 점심시간인 12-13시까지 방전이 지속된다. 이후 18시까지 소규모의 충전과 방전으로 일정 충전 수준을 유지하며 0시부터 다시 충전을 시작한다. 가게는 전체적으로 약 6시부터 방전하다가 약 13-6시에 대부분 충전하는 모습을 보인다. 하지만, 오후 최대부하와 중간부하 시간대에는 경부하 시간대에 비해 충전하는 양이 적고, 전력 소비가 감소하기 시작하는 시간대이다. 또한, 소비량이 다시 증가하기 시작하는 16-18시에는 약간의 방전을 관찰할 수 있다. 전체적으로 B_t 가 BG_{\min} 와 BG_{\max} 사이에서 유지되며, 22

시-8시에 충전하는 양상을 보인다. 또한, 병원과 가게는 시간대별 요금에 같지만, 병원은 오전 중간부하-최대부하 시간대에 충전하는 데에 비해 가게는 방전하는 것을 볼 수 있다. 또 다른 차이점으로는 다른 업종은 전체적으로 12시까지 충전하지만 공장은 13시까지 충전하는데, 이는 공장 소비 전력이 점심시간에 비교적 감소하기 때문에 미리 충전하는 것으로 추측할 수 있다.



(그림 4) 업종별 시간에 따른 누적 요금 차이

(그림 4)는 배터리를 사용하지 않았을 때와 사용했을 때의 누적 전기 요금의 차이이다. 해당 그래프를 통해 각 환경에서 전기 요금을 얼마나 절약했는지를 알 수 있다. 배터리를 충전할 때는 그만큼 일반 전력을 더 사용한 것이기 때문에 전기 요금 차이가 줄어들게 된다. 하지만, 배터리를 방전할 때는 방전한 만큼 일반 전력을 사용하지 않기 때문에 요금 차이가 커진다. 요금 차이는 배터리의 충전, 방전 패턴에 따라 자연스럽게 달라지나, 시간이 지날수록 점점 증가하는 것을 관찰할 수 있으며, 이를 통해 절약한 금액이 점점 커지고 있음을 알 수 있다.

5. 결론

본 연구에서는 누적 전기 요금 감소와 배터리 충전 수준 유지를 목표로 한 알고리즘을 제안하고, 업종별로 학습을 진행한 결과를 비교 분석하였다. 실

험 결과, 업종별 소비 패턴에 따라 같은 보상 설계 이더라도 충전 및 방전 패턴이 달라지는 것을 알 수 있었다. 또한, 배터리를 사용했을 때, 요금 절감을 확인하였다. 이를 통해, 각 환경에 보다 적합한 알고리즘을 설계하여 효율적이고 경제적인 전력망 구축을 기대할 수 있을 것이다.

한편, 본 연구에서는 2020~2022년 데이터를 사용하였는데, COVID-19 시기로 인한 특수한 상황이었으므로 향후 전력 소비 패턴 보완이 필요하다. 이와 더불어, 추후 상호작용할 수 있는 멀티 에이전트를 이용하여 성능을 높이는 연구를 진행하고자 한다.

사사문구

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2021R1F1A1047113).

참고문헌

- [1] A. Kahraman and G. Yang, "Home Energy Management System based on Deep Reinforcement Learning Algorithms," IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Novi Sad, Serbia, pp. 1-5, 2022.
- [2] S. Aladdin, S. El-Tantawy, M. M. Fouda and A. S. Tag Eldien, "MARLA-SG: Multi-Agent Reinforcement Learning Algorithm for Efficient Demand Response in Smart Grid," IEEE Access, vol. 8, pp. 210626-210639, 2020.
- [3] C. Yang, J. Zhang, F. Lin, L. Wang, W. Jiang and H. Zhang, "Combining Forecasting and Multi-Agent Reinforcement Learning Techniques on Power Grid Scheduling Task," IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, pp. 1576-1580, 2023.
- [4] 전력통계정보시스템, <https://epsis.kpx.or.kr/epsisnew/selectEksaAscAscChart.do?menuId=060401>
- [5] 한국전력공사, <https://cyber.kepco.co.kr/ckepco/foront/jsp/CY/E/E/CYEEHP00108.jsp>
- [6] KEMRI한전경영연구원, https://home.kepco.co.kr/kepco/KR/ntcob/ntcobView.do?pageIndex=1&boardSeq=21057234&boardCd=BRD_000456&menuCd=FN311802&parnScrpSeq=0&searchCondition=total&searchKeyword=