ELEKTROTEHNIŠKI VESTNIK 93(1-2): 63–68, 2026
IZVIRNI STROKOVNI ČLANEK

SARSA ali DQN za krmiljenje baterijskega hranilnika

Andraž Sivec, Marko Meža

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: andraz.sivec@fe.uni-lj.si

Povzetek. Zaradi porasta proizvodnje električne energije iz sončnih in vetrnih virov se pojavlja problem viškov generacije in negativnih cen elektrike, kar vpliva na stabilnost elektroenergetskega sistema. Kot možna rešitev se vse bolj uveljavlja uporaba baterijskih hranilnikov, ki pa zahtevajo učinkovito upravljanje za najboljši učinek. V članku primerjamo delovanje dveh metod spodbujevalnega učenja pri optimizaciji delovanja baterije: tabelarično metodo SARSA in metodo z nevronsko mrežo DQN. Agenta se učita na podlagi zgodovinskih podatkov o proizvodnji sončne elektrarne, porabi gospodinjstva in tržnih cenah elektrike, njun cilj pa je minimizacija stroškov nakupa energije. Rezultati kažejo, da uporaba obeh pristopov bistveno zmanjša stroške v primerjavi s primerom brez baterije. DQN dosega najboljše rezultate, saj se bolje prilagodi kompleksnim vzorcem v podatkih in boljše deluje v večjih prostorih stanj. Raziskava potrjuje, da lahko spodbujevalno učenje učinkovito podpira odločanje pri upravljanju hranilnikov in prispeva k večji ekonomski učinkovitosti ter stabilnosti energetskega sistema ter prikaže nekatere izzive pri implementaciji.

Ključne besede: Spodbujevalno učenje, zeleni prehod, primerjava algoritmov, optimizacija krmiljenja baterije po ceni

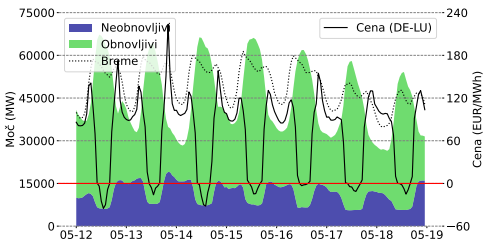
A Comparison of Tabular and Deep Reinforcement Learning for Battery Management

The increasing share of solar and wind energy production introduces challenges related to excess electricity generation and its negative prices, which affect the stability of power systems. Battery storage is emerging as a promising solution, but its optimal operation requires advanced control strategies. This paper compares two reinforcement learning methods for battery management: the tabular SARSA and the deep Q-Network (DQN) method. Both methods are trained on historical data from a photovoltaic plant and electricity market prices, with the objective of minimizing energy purchase costs. The results demonstrate that both methods significantly reduce the costs compared to the baseline without storage, with DQN outperforming SARSA due to its ability to capture complex patterns and handle continuous data input. The findings confirm that reinforcement learning can effectively support decision-making in energy storage management, contributing to improved economic efficiency and system stability while also highlighting some implementation challenges, such as training instability and sensitivity to hyperparameters.

Keywords: reinforcement learning, green transition, algorithm comparison, optimization of battery control based on price

1 UVOD

Delež električne energije, proizvedene z obnovljivimi viri, v zadnjem času močno narašča, to pa za elektroenergetski sistem prinaša nove izzive. Klasične elektrarne lahko zagotavljajo enostavno centralno krmiljenje in imajo zelo predvidljivo proizvodnjo, skoraj neodvisno od zunanjih dejavnikov, kar nam omogoča dobro prilagajanje proizvodnje porabi. Proizvodnja obnovljivih virov pa je v precejšnji meri pogojena z vremenskimi pogoji, poleg tega predvsem v primeru sončnih elektrarn tudi močno razpršena in običajno ni centralno krmiljena s strani večjih družb ali lokalnih operaterjev. Pri manjšem deležu takšnih elektrarn to ne povzroča težav, če je pa delež obnovljivih virov prevelik, pa lahko pride do prevelike proizvodnje električne energije in posledično tudi negativnih cen na trgu [1], saj je ob optimalnih pogojih proizvedene električne energije preveč. To se lepo vidi na sliki 1, za prikaz smo izbrali nemški trg zaradi tamkajšnjega velikega deleža obnovljivih virov.



Slika 1 Proizvodnja, poraba in cena električne energije v Nemčiji v tednu 20 v letu 2025.

Dodatna povečava deleža sončnih in vetrnih elektrarn zahteva večjo fleksibilnost omrežja. To lahko storimo s povečanjem števila plinskih elektrarn, ki so izredno fleksibilen vir s hitrim reakcijskim časom, in z do-

Prejet 1. oktober, 2025
Odobren 12. februar, 2026

dajanjem baterijskih kapacitet v omrežje [2], ki nam omogočijo, da odvečno energijo shranimo in porabimo pozneje, ko je proizvodnja manjša. Take hranilnike je najbolj smiselno postaviti blizu proizvodnje, da ne obremenjujemo omrežja s prenosom. Ker je v Sloveniji 64 % sončnih kapacitet iz stanovanjskih inštalacij [3], v Nemčiji pa okoli 67 % [4], ni presenetljivo, da delež stanovanjskih sončnih elektrarn z baterijo raste, takšni sistemi pa zahtevajo krmiljenje.

Obstaja mnogo različnih algoritmov, kako krmiliti tak sistem. Od najbolj enostavnega principa, ko se odvečna proizvodnja vedno shrani in porabi za čas, ko je te premalo, tako da se kupljena električna energija minimizira, pa vse do naprednih optimizacijskih algoritmov, ki poskušajo končno ceno optimizirati tudi s trgovanjem na trgu. Seveda je potreba po krmilnem algoritmu ali bateriji nasploh odvisna od plačilne sheme, ki jo uporabnik ima. Če je namreč dogovorjeno za fiksno ceno odkupljene električne energije ali NET-metering, ni pri tem nikakršne iniciative za uporabo baterije, saj tudi pri prodaji, ko je električne energije preveč (negativni ceni), uporabnik enako zasluži.

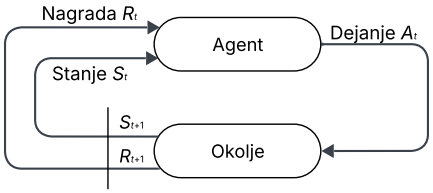
Cilj tega dela je prikazati uporabo spodbujevalnega učenja kot krmilnega algoritma za upravljanje baterij v okviru stanovanjskih sončnih elektrarn. Najprej v drugem poglavju predstavimo osnovno tabelarično obliko spodbujevalnega učenja, nato pa še njegovo nadgradnjo – globoko spodbujevalno učenje – ter pojasnimo, zakaj je takšna nadgradnja potrebna. Poleg tega izpostavimo prednosti in slabosti globokega spodbujevalnega učenja ter opišemo postopek njegove implementacije. V tretjem poglavju predstavimo rezultate, ki smo jih v analizi dobili, nato v četrtem poglavju svoje delo povzamemo in predstavimo možnosti za nadaljnje delo. Pri tem smo se osredotočili na razmeroma enostavne algoritme, ki jih lahko končni uporabniki vsaj okvirno razumejo in preizkusijo na svojih zgodovinskih podatkih. Glavni cilj je bil razviti preproste metode, ki lahko učijo iz zgodovine delovanja lastne sončne elektrarne.

2 SPODBUJEVALNO UČENJE

Spodbujevalno učenje (angl. Reinforcement learning – RL) je področje strojnega učenja, ki je pogosto uporabljeno za inteligentni nadzor in optimizacijo. V nasprotju s klasičnim nadzorovanim ali nenadzorovanim učenjem se algoritem RL uči iz izkušenj [5].

Spodbujevalno učenje je sestavljeno iz strategije, nagrade in vrednostne funkcije ter občasno tudi modela okolja. Ti elementi sestavljajo agenta, ki vidi nekatere lastnosti okolja (S) in na njihovi podlagi sprejema odločitve (A), nato pa dobi povratno informacijo v obliki nagrade (r), ki mu pove, kako dobra je bila ta odločitev [5], [6], kar je prikazano na sliki 2. Hkrati okolje uporabi dejanje, ki ga je izbral agent, za spremembo stanja, novo stanje pa nato skupaj z nagrado vrne agentu.

S treningom se lahko agent ne nauči samo tega, v katerem stanju je posamezno dejanje najboljše za trenutno nagrado, ampak celo to, kakšno dejanje bo pripeljalo do največje nagrade na dolgi rok. To opisuje Q-funkcija (enačba 1), ki dodeli vrednost vsaki kombinaciji stanja in dejanja, kjer se predvideva, da bi po izbranem dejanju agent sledil optimalni strategiji. Ko ima agent tako funkcijo, mora za optimalni rezultat samo v vsakem stanju izbrati dejanje z najvišjo vrednostjo.



Slika 2 Povezava agenta in okolja.

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \cdot \max_{a'} Q(s',a') - Q(s,a) \right] \tag{1}$$

- s — trenutno stanje (angl. *state*),
- a — izbrano dejanje (angl. *action*),
- r — prejeta nagrada (angl. *reward*),
- s' — naslednje stanje po dejanju (angl. *next state*),
- a' — dejanje, ki ga agent izbere v stanju s' (angl. *next action*),
- α — hitrost učenja (angl. *learning rate*),
- γ — faktor popusta (faktor vrednotenja prihodnjih nagrad).

Da se to funkcijo agent nauči, pa mora najprej raziskovati po prostoru in preizkušati različna dejanja za dana stanja - temu delu delovanja rečemo raziskovanje. Vsak agent ima torej dve fazi. Prva je raziskovalna, saj se v njej bolj ali manj naključno sprehaja po prostoru in preizkuša različna dejanja. Druga preide na izkoriščanje tega naučenega znanja, ko izbira dejanja s ciljem, da bodo na dolgi rok prinesla največjo nagrado. Seveda želimo, da večina raziskovanja mine pred končno implementacijo algoritma, a pri RL se agent lahko uči tudi na končni izvedbi, le da običajno le z zelo okrnjenim raziskovanjem.

2.1 Tabelarično spodbujevalno učenje in problem dimenzionalnosti

Najosnovnejša izvedba in na področju krmiljenja mikro-omrežij tudi najbolj raziskana je tabelarična izvedba spodbujevalnega učenja [7] z najpogostejšima podvrstama algoritmov q-učenje (angl. *Q-learning*) in SARSA. V tej izvedbi našo Q-funkcijo predstavlja tabela stanj in dejanj, kjer je vsaka vrstica unikatna kombinacija stanja in dejanja s pripadajočo vrednostjo. Ta izvedba odlično deluje v okoljih, kjer imamo majhno število diskretnih stanj in dejanj (npr. šahovnica z nekaj figurami).

Toda ta pristop ima pomembne omejitve. Ne le da morajo biti stanja in dejanja diskretizirana, tudi njihovo število je močno omejeno. Če na primer upoštevamo štiri spremenljivke in vsako diskretiziramo na deset vrednosti, dobimo 10^4 možnih stanj. Če temu dodamo štiri možna dejanja, Q-tabela naraste na 40 000 polj. To je sicer še obvladljivo, a s povečevanjem števila spremenljivk in dejanj ali z bolj fino diskretizacijo število stanj hitro eksplodira in Q-tabela postane neobvladljiva. Kljub temu lahko s premišljeno izbiro stanj in dejanj ter z uporabo več med seboj povezanih agentov dosežemo dobre rezultate tudi pri zahtevnejših problemih [8].

2.2 Globoko spodbujevalno učenje

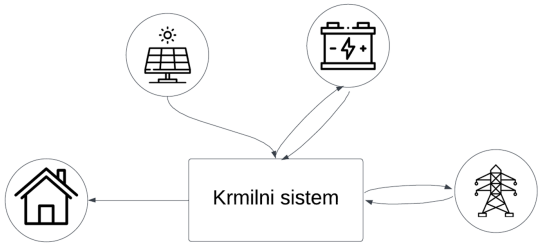
Ko se ločljivost in število vhodnih podatkov povečuje, velikost Q-tabele postane neobvladljiva, zato takrat uporabimo drugo metodo za opis Q-funkcije. V RL je to običajno umetna nevronska mreža (angl. *Artificial Neural Network – ANN*), ki preko med seboj povezanih plasti nevronov povezuje vhodne spremenljivke z izhodnimi [6]. Takšni nevronski mreži, ki ima vhodno, izhodno in več vmesnih (skritih) plasti nevronov, pogosto rečemo tudi globoka nevronska mreža (angl. *Deep Neural Network – DNN*), tako je bila predstavljena tudi globoka Q-mreža (angl. *Deep Q-Network – DQN*), ki uporabi ANN, da nadomesti Q-tabelo in aproksimira Q-funkcijo. Ta pristop nam omogoča uporabo zveznih prostorskih spremenljivk, a ostane omejen na diskretna dejanja, kar je za naš problem povsem dovolj.

Čeprav nam nevronska mreža omogoča sprejemanje zveznih vhodnih podatkov, pa ima tak algoritem v primerjavi s tabelarnim spodbujevalnim učenjem tudi nekaj slabosti. V glavnem so to sami parametri nevronske mreže, sej je zahtevno določiti hiperparametre, ki bi bili optimalni za uporabljeno nevronsko mrežo in problem. Pri tem nam delo zaradi naključnosti v raziskovanju [9] dodatno otežujejo težave s stabilnostjo [10] in razlikami med samimi ponovitvami, kar lahko povzroči velike razlike med ponovitvami učenja pri istih parametrih.

2.3 Okolje

Za to delo smo uporabili enako okolje kot v [11], to je digitalni dvojček pametne hiše s sončnimi paneli, baterijo in priklopom na omrežje (slika 3). Za podatke o porabi hiše in proizvodnji sončne elektrarne smo uporabili surove podatke iz grške pametne hiše [12] ter jih primerno prečistili in obdelali. Te podatke smo izbrali, ker so podatki, če so primerno citirani dovoljeni za uporabo v člankih. Podatke o ceni električne energije pa smo dobili iz zgodovinskih podatkov o ceni električne energije pri trgovanju za dan vnaprej [13].

Pri simulaciji smo upoštevali, da pri polnjenju in praznjenju baterij prihaja do izgubi, zato smo v vseh simulacijah upoštevali učinkovitost polnjenja in praznjenja 95 %s čimer dobimo skupno učinkovitost $90,3\,\%$. Omejili smo tudi maksimalno energijo, ki jo baterija sme prejeti in oddati v eni uri na 6 kWh.



Slika 3 Krmiljen sistem.

2.4 Dejanja

Naše okolje smiselno pokrijejo štiri dejanja, ki določajo količino kupljene in prodane električne energije ter spremembo stanja shranjene električne energije v bateriji. Ta štiri dejanja tvorijo minimalen nabor, ki agentu še vedno omogoča izbiro med vsemi smiselno različnimi načini upravljanja. Agent nima vpliva na porabo gospodinjstva. Dejanja, med katerimi lahko agent izbira, so:

1) **Baterija se polni s polno močjo:** Baterija se polni s polno močjo. Če sončna elektrarna (SE) proizvede več, se presežek proda. Če je proizvodnja nezadostna, se električna energija dokupi. Baterija se ne prazni.
2) **Baterija se polni iz odvečne električne energije SE:** Kupimo le toliko električne energije, kot jo potrebujemo za oskrbo hiše. Če obstaja presežek iz SE, se ta uporabi za polnjenje baterije, sicer baterija počiva.
3) **Baterija napaja hišo:** Hišo najprej napaja SE, nato baterija. Če to ne zadošča, se preostanek električne energije kupi iz omrežja. Baterija se ne polni, le prazni.
4) **Baterija se prazni s polno močjo:** Baterija se prazni s polno močjo za pokritje porabe in prodajo presežka v omrežje. Električna energija se iz omrežja kupi le, če SE in baterija ne pokrijeta vse porabe.

2.5 Nagrade

Naša nagrada je sestavljena iz treh ločenih komponent, pri čemer ima vsaka svoj namen in vpliv na vedenje agenta.

Prva komponenta $R_1(t)$ kaznuje agenta, kadar je napolnjenost baterije zunaj želenih meja – nad zgornjo mejo α ali pod spodnjo mejo β. Cilj je zaščititi življenjsko dobo baterije ter ohraniti del kapacitete za morebitne tržne priložnosti. Funkcija $R_1(t)$ ima eksponentno obliko in je zasnovana tako, da kazen eksponentno narašča s preseganjem meje. Tako dopušča manjša odstopanja ob izjemnih priložnostih, hkrati pa vseeno ohranja baterijo znotraj želenih obratovalnih meja. Obenem agent v želenem delovnem območju baterije dobi majhno pozitivno nagrado, ki sili proti sredini, kar pomaga pri učenju.

Druga komponenta $R_2(t)$ spodbuja praznjenje baterije v času visokih cen in polnjenje ob nizkih oziroma negativnih cenah električne energije. Obratno pa kaznuje polnjenje pri visokih cenah in praznjenje pri nizkih. Pri tem se trenutna cena $C(t)$ primerja s 30-dnevno srednjo vrednostjo cen $C_{\text{med}}(t)$. Namen te komponente je pospešiti iskanje optimalne strategije ter se izogniti lokalnemu optimumu, kjer bi agent baterijo enostavno prenehal uporabljati, da ne bi prišlo do izgub pri polnjenju in praznjenju.

Tretja komponenta $R_3(t)$ neposredno upošteva strošek električne energije v trenutnem koraku. Izražena je kot $-\delta \cdot \text{Pla}(t)$, kjer je δ faktor, ki nam pomaga pri normiranju nagrade na želeno velikost, $\text{Pla}(t)$ pa predstavlja plačilo v trenutku t.

Skupna nagrada, ki jo agent prejme v trenutku t, je torej vsota vseh treh komponent: $R(t) = R_1(t)+R_2(t)+R_3(t)$.

3 Rezultati

Algoritem smo izvajali v okolju Google Colab in za algoritem DQN uporabili knjižnico pytorch. Najprej smo vhodne podatke prečistili ter odstranili dneve, ko so se zaradi prekinitev pojavljale NaN-vrednosti ali druge napake. Nato smo podatke razdelili na dve polovici vsako dolgo približno 1 leto. Prva je bila uporabljena za učenje, druga pa za preverjanje.

Metrika, ki smo jo uporabljali za določanje uspešnosti posameznega algoritma, je bila kumulativna cena električne energije v validacijskem območju.

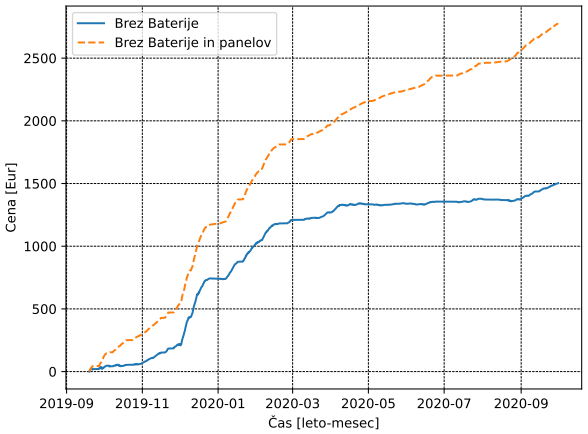
3.1 Namestitev panelov

Preden se osredotočimo na vpliv baterije, je smiselno najprej analizirati vpliv samih sončnih panelov brez baterijskega sklopa, saj tako lažje ocenimo dodano vrednost baterije. Kot je razvidno s slike 4, se pri namestitvi sistema z močjo 9,57 kWp in ob polni ceni odkupa električne energije letni stroški zmanjšajo za približno 1200 EUR. Skupni letni stroški električne energije brez sončnih panelov znašajo 2776 EUR, z nameščenimi paneli in ob polni ceni odkupa pa 1503 EUR. Če za oddano električno energijo v omrežje ne bi prejeli plačila, bi letni strošek znašal 2034 EUR.

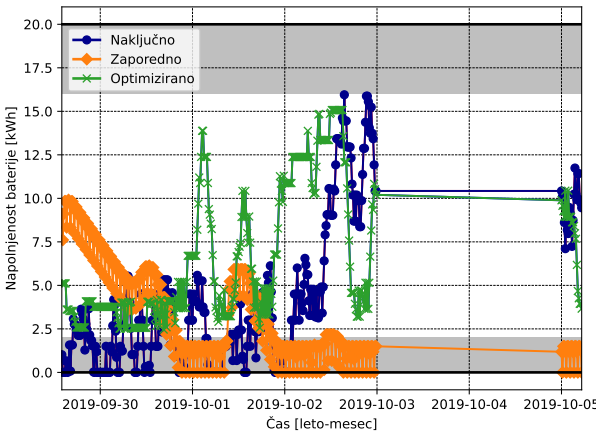
3.2 Tabelarično spodbujevalno učenje

Najprej smo problem krmiljenja rešili s tabelaričnim spodbujevalnim učenjem, specifično z algoritmom SARSA. Za vhodne podatke smo vzeli ceno električne energije v danem trenutku, razliko med proizvedeno in porabljeno električno energijo ter stanje baterije. Vse vhodne podatke smo diskretizirali na 20 stopenj v njihovem območju delovanja, kot izhod je imel algoritem zgornja štiri dejanja.

Relativno hitro smo dosegli, da se je algoritem naučil gibanja v želenem območju napolnjenosti baterije (slika 5). Tudi končne cene so bile že po eni ponovitvi učenja



Slika 4 Vpliv panelov na kumulativno ceno električne energije pri polni ceni.



Slika 5 Stanje baterije po učenju v primerjavi z naključno in zaporedno izbiro dejanj (neželeno stanje je obarvano sivo).

boljše kot pri naključni strategiji, ki smo jo poleg zaporedne menjave dejanj uporabljali za primerjavo. Mnogo boljše rezultate smo dobili, če smo odstranili prvi del nagrade in napolnjenost baterije ročno omejili znotraj želenega območja. Za naš primer to pomeni, da smo kapaciteto baterije zmanjšali za 30 % in tako ostali na stopnji od 10 do 80 % napolnjenosti baterije. S tem smo sicer nekoliko zmanjšali teoretične optimume, ki jih algoritem lahko doseže, a uspelo nam je izboljšati dobljene rezultate že po eni ponovitvi treninga.

Pregledali smo tudi vpliv velikosti baterije na končne dosežene stroške električne energije. Zaradi časovne zahtevnosti simulacij smo izvedli le en prelet učenja, pri čemer smo se omejili na učenje z navidezno omejitvijo kapacitete, saj v tem primeru hitreje dosežemo stabilne optimizacijske vrednosti.

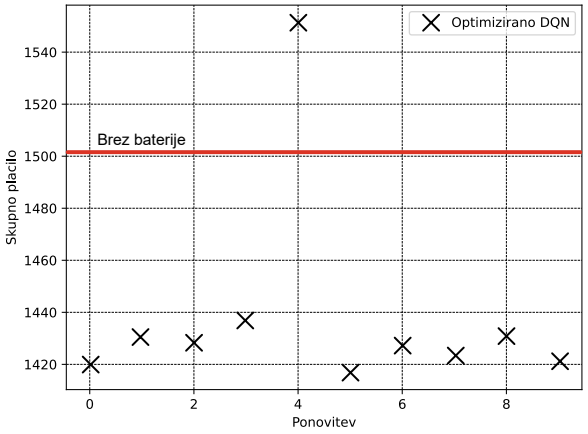
3.3 DQN

Nato smo v istem okolju preizkusili še agenta DQN s Q-mrežo, kar nam je omogočilo, da smo uporabili zvezne vhodne podatke. Majhen vpliv vhodnih spremenljivk

na hitrost izvajanja tega algoritma nam je omogočilo, da smo ločili porabo in generiranje energije ter lahko eksperimentirali z dodatnimi spremenljivkami. Zaradi boljše stabilnosti smo normirali vhodne vrednosti (generiranje, porabo in stanje baterije), ki jih naš algoritem vidi na območju med približno -1 in 1. Trenutno ceno električne energije pa smo normirali glede na srednjo ceno v zadnjem mesecu.

Algoritem DQN smo začeli uvajati podobno kot SARSA in tudi njemo je že pri dveh nevronih uspelo ohraniti kapaciteto baterije znotraj želenih meja. Število nevronov in njihovih plasti smo nato povečevali in dodajali drugi dve nagradi. Tudi pri algoritmu DQN smo dosegali najuspešnejše rezultate takrat, ko je bila kapaciteta baterije omejena ročno brez prve nagrade.

3.3.1 Nestabilnost delovanja: Opozoriti je treba, da tudi pri našem algoritmu DQN prihaja do nestabilnosti, predvsem pri neprimerni zasnovi nevronske mreže in drugih hiperparametrov, kot sta faktorja γ in α (1). To je lepo razvidno na sliki 6, kjer smo izvedli 10 ponovitev učenja pri povsem enakih nastavitvah in pri kapaciteti baterije 10 kWh.



Slika 6 Razlika med ponovitvami algoritma DQN pri kapaciteti 10 kWh.

Pri devetih izvedbah je bil končen rezultat dovolj blizu drugim, pri deseti pa je prišlo do tako velikega odstopanja, da je rezultat slabši, kot če baterije ne bi uporabili. To je problematično predvsem takrat, ko želimo s samo eno ponovitvijo hitro oceniti vpliv katere od sprememb v algoritmu, saj lahko dobimo povsem napačno informacijo. Žal ta nestabilnost ni nenavadna [9] in od nas zahteva, da po spremembi izvedemo več ponovitev za preverjanje, da je rezultat resnično dober predstavnik stanja.

3.4 Primerjava rezultatov algoritmov DQN in SARSA

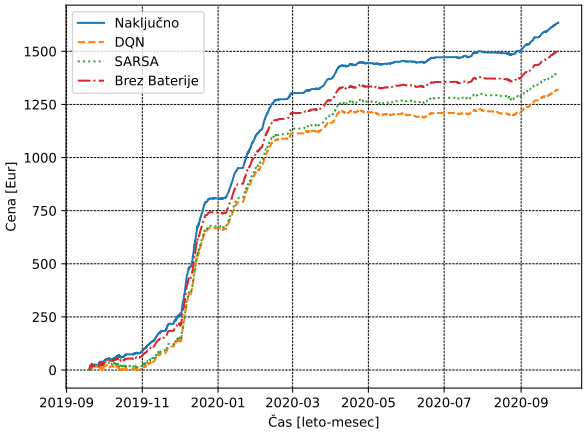
V tabeli 1 primerjamo končne rezultate obeh algoritmov pri različnih kapacitetah baterije. Vidimo, da nam je z obema algoritmoma pri vseh kapacitetah, kljub izgubam pri polnjenju in praznjenju baterije, uspelo preseči stanje brez baterije. Edini razliki med podatki algoritmov SARSA in DQN sta bila dodatno normiranje ter ločeno podajanje generirane in porabljene električne energije, kljub temu nam je z algoritmom DQN uspelo doseči občutno boljše rezultate kot pri algoritmu SARSA.

Tabela 1 Primerjava najboljših rezultatov algoritmov DQN in SARSA pri različnih kapacitetah baterije.

Kapaciteta (kWh)	**Kumulativna cena (EUR)**	
	SARSA	**DQN**
20	1388	1306
14	1439	1356
10	1477	1409
7	1481	1429
Brez baterije	1503	1503

Naredili smo še primerjavo kumulativne cene električne energije in nagrade, ki sta jo oba algoritma dosegla. Pri tem je bila kapaciteta baterije programsko omejena (ne z nagrado) in nastavljena na 20 kWh z namenom lepše razvidnosti razlik pri visokih kapacitetah.

Na sliki 7 je prikazana cena električne energije v drugi polovici podatkov, ki ni bila uporabljena za učenje. Vidimo, da pri naključni izbiri dejanj dosežemo slabši rezultat, kot če baterije sploh ne bi imeli, do tega pride pa zaradi izgub pri polnjenju in praznjenju baterije. A ker se del električne energije vendarle takoj proda, naključna izbira kljub 10 % izgubam ni toliko slabša, kot bi morda mislili.

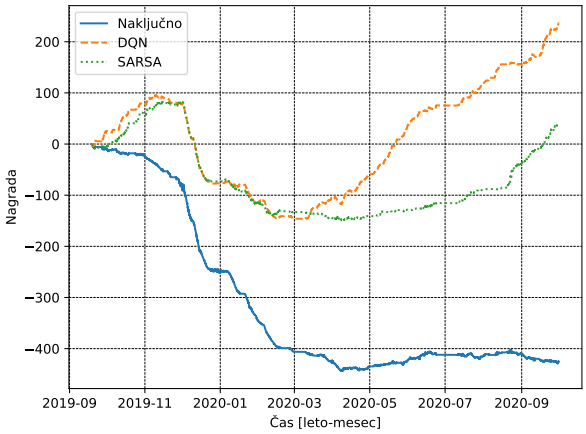


Slika 7 Kumulativna cena električne energije v drugi polovici podatkov DQN in SARSA pri kapaciteti 20 kWh.

Če si ogledamo sliko 8, vidimo, da je nagrada, ki jo dobi naš algoritem DQN po optimizaciji v vsakem trenutku boljša, kot bi jo dobili z naključno izbiro dejanj, in tudi občutno višja od tiste, ki jo uspe dobiti algoritem SARSA. Višjo nagrado dosežemo predvsem v poletnih mesecih, ko zna algoritem DQN bolje izkoristiti nihanja v ceni, kar se pozna tudi na kumulativni ceni (slika 7).

Tako kot nagrada je tudi končna cena, ki jo optimizirani algoritem plača, vedno boljša, kot če baterije ne

bi imeli, prav tako se na obeh slikah vidi, da sta se največje plačilo in posledično padec nagrade izvršila v decembru. V času zimskega solsticija je osvetlitev najslabša, ter posledično najmanjša proizvodnja sončnih panelov, obenem pa je v tistem času poraba električne energije velika in je ne moremo pokrivati s proizvedeno energijo.



Slika 8 Nagrada algoritma DQN druga polovica podatkov.

Pri slikah 7, 8 in posebno na 5 lahko vidimo linearne odseke opazovanih količin. Do njih pride zaradi napak v meritvah, ko shranjenih podatkov ni ali pa so ti nesmiselni (15-minutna poraba gospodinjstva v MWh). Dneve, ko so se napake pojavljale in je prišlo do daljših NaN-vrednosti, smo namreč izločili iz podatkov, a ker smo risali slike po časovni osi, so točke med dnevi, kjer je analiza potekala, vseeno linearno povezane.

4 Zaključek

Pri izvedbi obeh algoritmov smo bili uspešni, zlasti pri algoritmu DQN pa smo dosegli znatno izboljšavo. S tem nam je brez sprememb pri porabi ali proizvodnji električne energije uspelo doseči 13,1 odstotno zmanjšanje kumulativne cene v primerjavi s scenarijem brez baterije.

Idej za nadaljnje delo je še precej, a za začetek bi bilo algoritmu DQN zanimivo dodati določene podatke. Zanimiv bi bil vpliv časovnih podatkov, kot so ura, dan v tednu, mesec v letu, podatek ali je dan tisti praznik, ter podatek o napovedani ceni energije za dan vnaprej. Dobro bi bilo algoritme preizkusiti tudi na večji skupini gospodinjstev in to, v kolikšni meri so treningi prenosljivi, ter kako je splošni agent primerljiv s tistim, ki je treniran na specifičnih podatkih določenega gospodinjstva. Zanimivo bi bilo tudi razširiti primerjavo na druge algoritme krmiljenja in jih med seboj primerjati, zanima nas predvsem zahtevnost učenja ter implementacije v praksi, ter njihova uspešnost pri izboljšavi končne cene.

Zahvala

Zahvaljujemo se sodelavcem v laboratoriju LOEE za pomoč in podporo.

Literatura

[1] J. Viehmann, “Negative electricity prices are no longer an exception when photovoltaic feed-in increases in summer.” https://www.next-kraftwerke.com/energy-blog/risky-solar-peaks-negative-power-market-prices, 2025. Published: 16-05-2025. Accessed: 10-07-2025.

[2] A. Biber, M. Felder, C. Wieland, and H. Spliethoff, “Negative price spiral caused by renewables? electricity price prediction on the german market for 2030,” *The Electricity Journal*, vol. 35, no. 8, p. 107188, 2022.

[3] “Slovenia adds 298.8 MW of solar in 2024 — pv-magazine.com.” https://www.pv-magazine.com/2025/02/12/slovenia-adds-298-8-mw-of-solar-in-2024. [Accessed 12-09-2025].

[4] W. W. Dr. Simon Philipps, “Photovoltaicsreport.” https://www.ise.fraunhofer.de/content/dam/ise/de/documents/publications/studies/Photovoltaics-Report.pdf, 2025. [Accessed 11-09-2025].

[5] R. S. Sutton, A. G. Barto, *et al.*, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[6] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, “Deep reinforcement learning: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5064–5078, 2022.

[7] P. I. Barbalho, A. L. Moraes, V. A. Lacerda, P. H. Barra, R. A. Fernandes, and D. V. Coury, “Reinforcement learning solutions for microgrid control and management: a survey,” *IEEE access*, 2025.

[8] F.-D. Li, M. Wu, Y. He, and X. Chen, “Optimal control in microgrid using multi-agent reinforcement learning,” *ISA transactions*, vol. 51, no. 6, pp. 743–751, 2012.

[9] A. Irpan, “Deep reinforcement learning doesn’t work yet.” https://www.alexirpan.com/2018/02/14/rl-hard.html, 2018.

[10] E. O. Arwa and K. A. Folly, “Reinforcement learning techniques for optimal power control in grid-connected microgrids: A comprehensive review,” *IEEE Access*, vol. 8, pp. 208992–209007, 2020.

[11] A. Sivec and M. Meža, “Optimizacija cene električne energije gospodinjstva s spodbujevalnim učenjem,” in *Proceedings of the 34th International Electrotechnical and Computer Science Conference (ERK 2025)*, (Portorož, Slovenia), Faculty of Electrical Engineering, University of Ljubljana, 2025. Track: Pametni rezidenčni energetski sistemi / Smart Residential Energy Systems.

[12] L. Zyglakis, S. Zikos, K. Kitsikoudis, A. D. Bintoudi, A. C. Tsolakis, D. Ioannidis, and D. Tzovaras, “Greek smart house nanogrid dataset,” Nov. 2020.

[13] “European Wholesale Electricity Price Data — Ember — ember-energy.org.” https://ember-energy.org/data/european-wholesale-electricity-price-data/. [Accessed 18-08-2025].

Andraž Sivec je leta 2024 magistriral s področja elektrotehnike na Univerzi v Ljubljani fakulteti za elektrotehniko. Kjer sedaj deluje kot asistent. Njegovo področje raziskovanja obsega krmiljenje in integracija obnovljivih virov ter spodbujevalno učenje.

Marko Meža (Senior member, IEEE) je leta 2001 diplomiral in leta 2007 doktoriral na Fakulteti za elektrotehniko, Univerze v Ljubljani. Je izredni profesor na Fakulteti za elektrotehniko, kjer je trenutno član Laboratorija za osnove elektrotehnike in elektromagnetiko. Raziskovalno se ukvarja z uporabo sodobnih pristopov obdelave, strojnega učenja in podatkovnega rudarjenja na signalih iz tehnike, medicine in socialne interakcije.