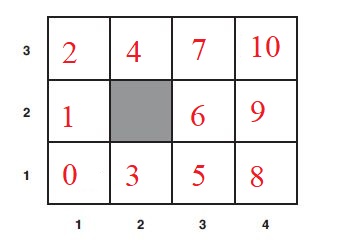
**UČENJE PODSTICANJEM**

1. **Rešavanje poznatog MPO planiranjem:**

Stanja su numerisana prema sledećoj slici, gde je sa 0 obeleženo startno stanje a sa 9 i 10 terminalna stanja:



Akcije su obeležene na sledeći način:

Levo → 0

Desno → 1

Gore → 2

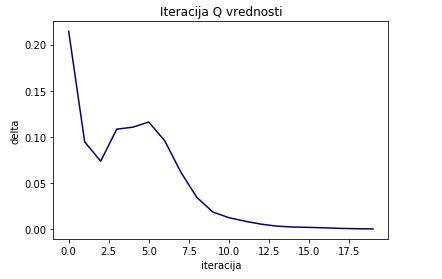
Dole → 3

Matrica *Q* dimenzija *Ns*x*Na* sadrži Q-vrednosti svih parova (stanje,akcija).

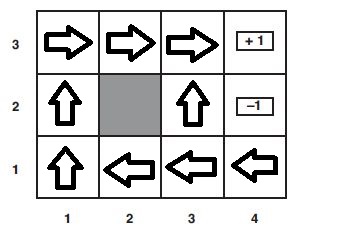
Vektor *R* dužine *Ns* sadrži nagrade svakog od stanja.

Matrica *P* dimenzija *Ns*x*Na*x*Ns* za svaki par (stanje s, akcija a) sadrži verovatnoću prelaza iz stanja s u bilo koje drugo stanje primenom akcije a.

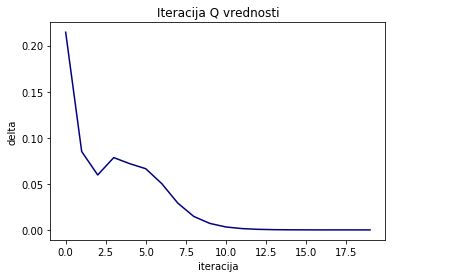
Primenom algoritma iteracije Q-vrednosti sa parametrom gama = 1 dobijen je sledeći grafik:



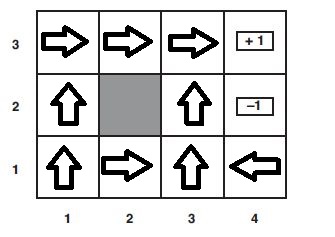
Dobijena politika:



Primenom algoritma iteracije Q-vrednosti sa parametrom gama = 0,9 dobijen je sledeći grafik:



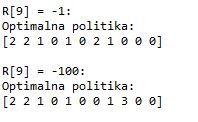
Dobijena politika:



Na prethodnim graficima je sa delta označeno koliko se Q vrednost promenila u odnosu na prethodnu iteraciju (usrednjeno po svim parovima (stanje, akcija)). Može se primetiti da smanjenje parametra gama ubrzava konvergenciju.

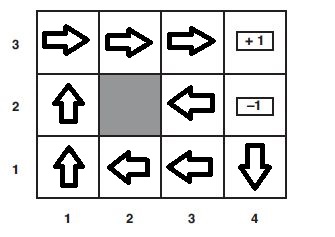
Kod kojim su dobijena prethodna dva grafika se nalazi u fajlu q\_iteracija.py.

Promenom nagrade „lošeg“ neterminalnog stanja (stanje 9) sa -1 na -100 politika se promenila za dva stanja:



Politika se promenila za stanja 6 i 8, odnosno za stanja iz kojih se moglo preći u stanje 9. Politika za stanje 6 se promenila iz gore u levo, a za stanje 8 iz levo u dole. Prethodna politika je sa malom verovatnoćom iz ovih stanja vodila u stanje 9. Posle drastičnog pogoršanja nagrade za stanje 9, politika je korigovana tako da se šansa prelaska u stanje 9 minimizuje (u ovom slučaju potpuno izbegne). Kod u kome je ovo urađeno se nalazi u fajlu q\_iteracija1.py.

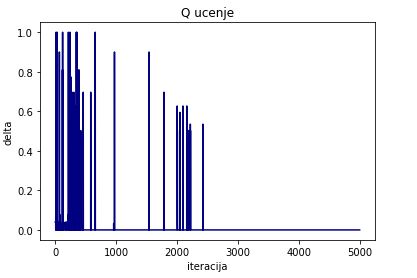
Slika za politiku nakon promene R[9] = -100:

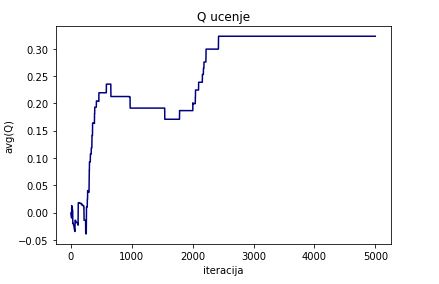


1. **Rešavanje nepoznatog MPO kroz učenje podsticanjem:**
2. Istraživanje uticaja parametra epsilon:

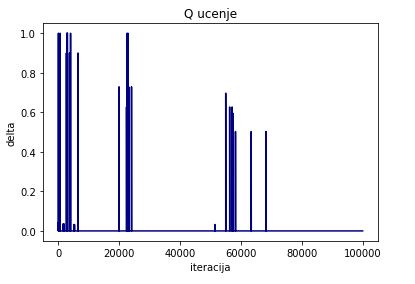
Za stopu učenja je ovom delu uzeto alfa = 1.

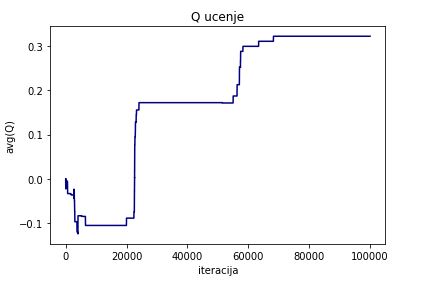
Za epsilon = 0,5 dobijeni su sledeći grafici, gde delta predstavlja promenu Q vrednosti u određenoj iteraciji a avg(Q) srednju vrednost matrice Q:



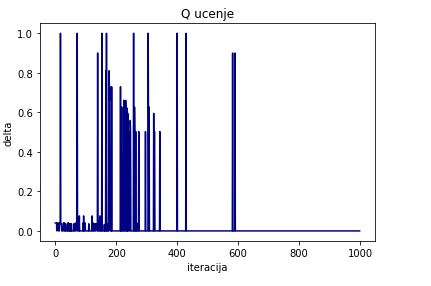


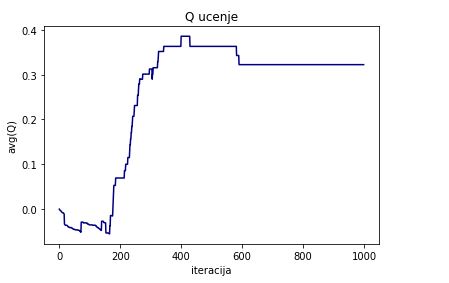
Za epsilon = 0,1 dobijeni su sledeći grafici:



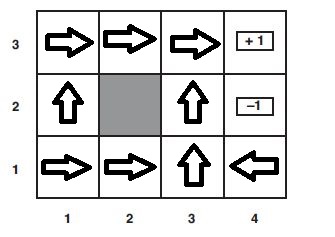


Za epsilon = 0,9 dobijenisu sledeći grafici:





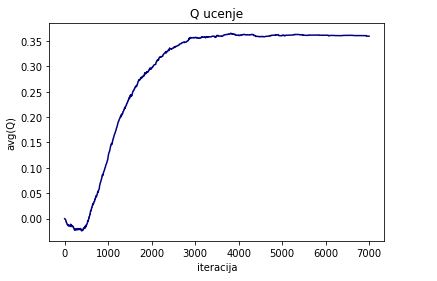
Uočeno je da se povećanjem parametra epsilon ubrzava konvergencija, što ima smisla s obzirom na to da povećanje epsilona znači da češće biramo akcije nasumično što znači da više istražujemo. Kada se algoritmu zada dovoljan broj iteracija da se dođe do konvergencije, u sva tri slučaja dobijamo istu politiku:



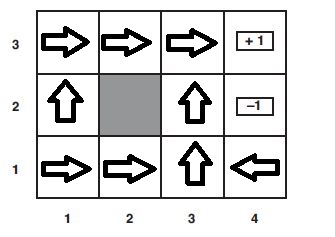
Kod kojim su prethodni grafici generisani se nalazi u fajlu q\_ucenje\_eps.py.

1. U ovom delu je korišćeno eps = 0,5.

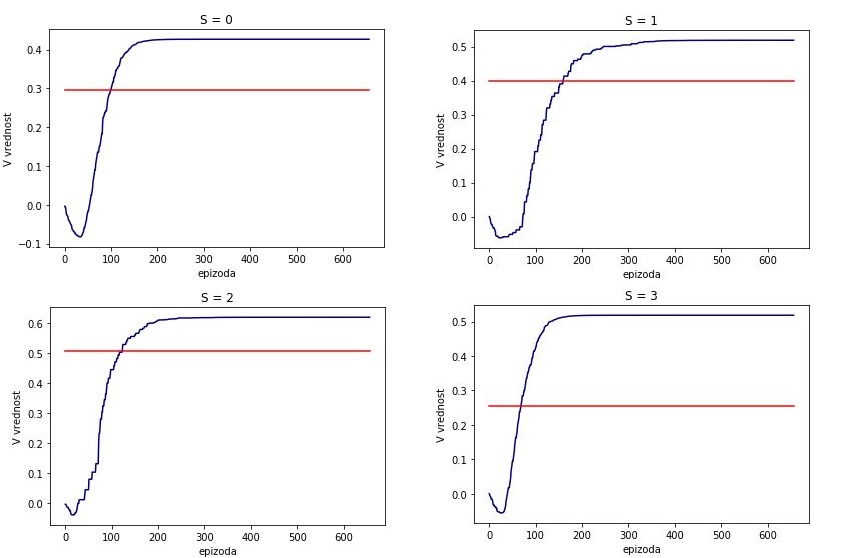
(a) Isprobavanjem različitih vrednosti za parametar alfa (stopu učenja), uočeno je da smanjenje ovog parametra usporava učenje ali smanjuje varijacije Q varijacije Q vrednosti u susednim iteracijama, odnosno smanjuje uticaj pojedinacnog odbirka na Q vrednost. Kao konstantna stopa učenja je usvojeno alfa = 0,1 i dobijen je sledeći grafik (kod u fajlu q\_ucenje\_alfa\_const.py):

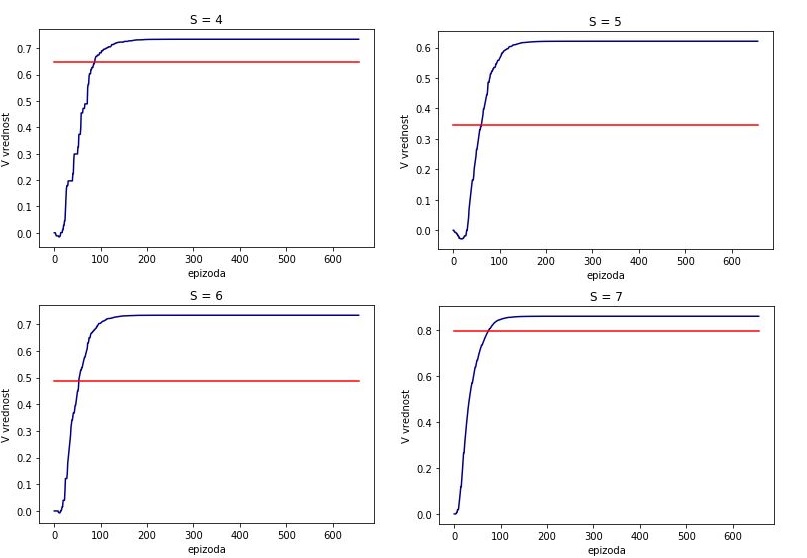


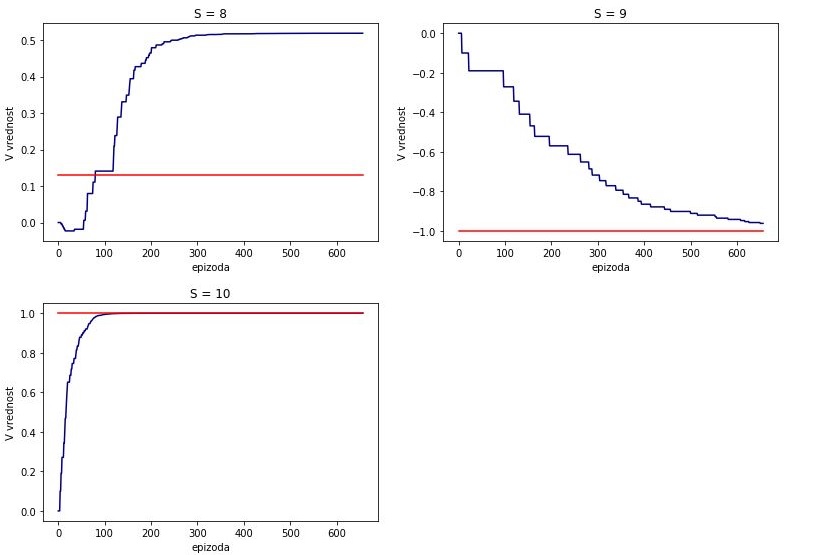
Dobijena politika:



Grafici za V vrednosti svih stanja su dati u nastavku (crvenom linijom je obeležena Vvrednost dobijena u prethodnom zadatku):

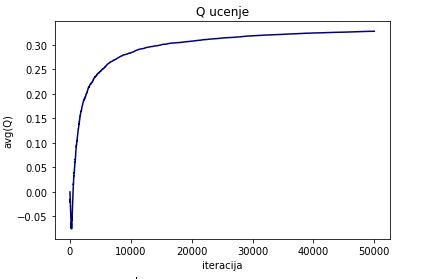




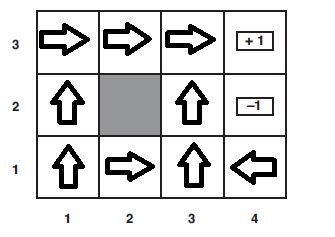


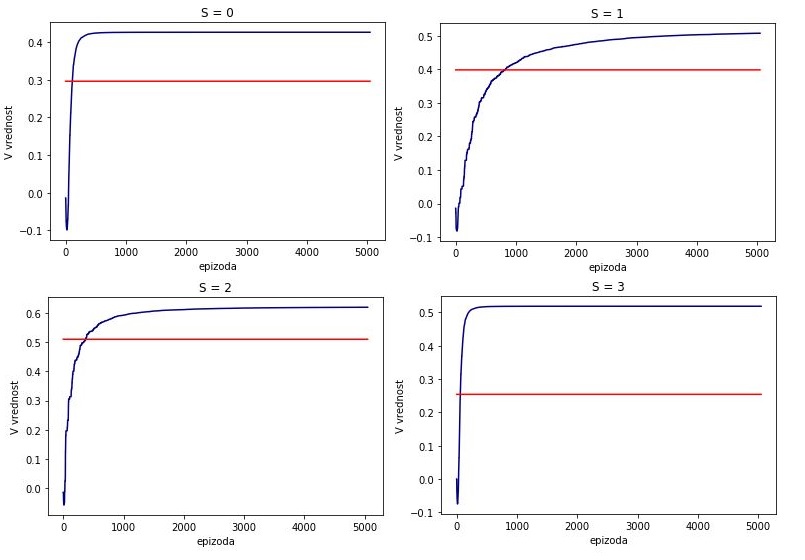
(b) Korišćenje adaptivne stope učenja po formuli

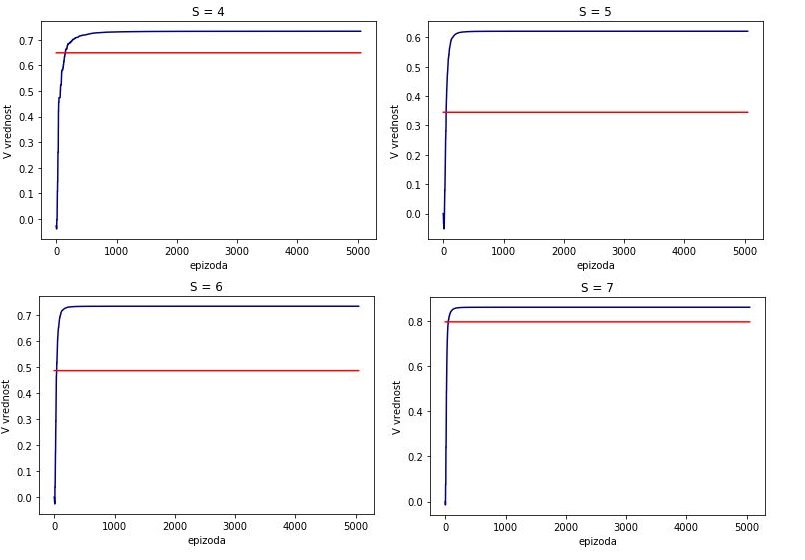
dalo je sledeće rezultate (kod u fajlu q\_ucenje\_alfat.py):

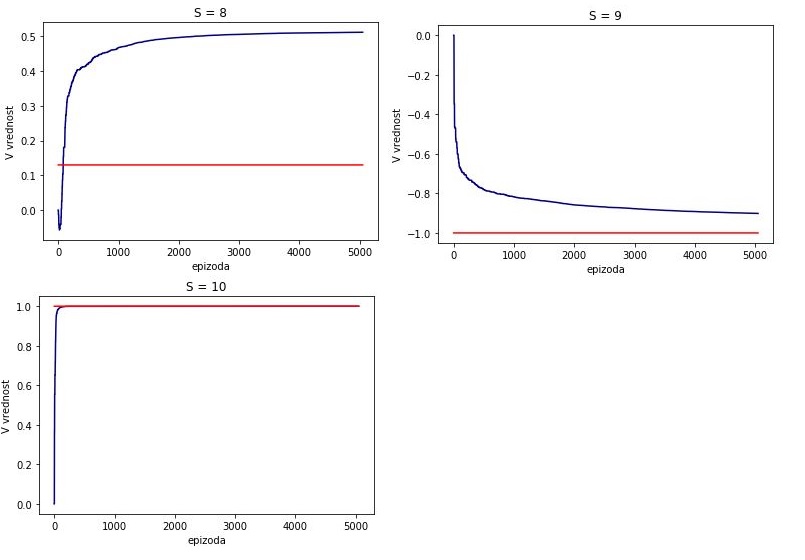


Dobijena politika:









1. Za konstantnu stopu učenja alfa = 0.1 potrebno je oko 600 epizoda da bi se došlo do konvergencije, dok je u slučaju adaptivne stope po gornjoj formuli taj broj mnogo veći, oko 5000 epizoda. (Ovi rezultati su dobijeni korišćenjem gama = 0,9 i eps = 0,5)