

# Spodbujevalno učenje pri igranju namiznih iger (angl. *Reinforcement learning in board games*)

Tim Kalan

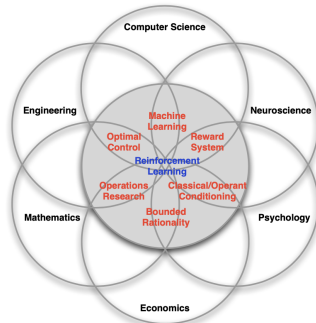
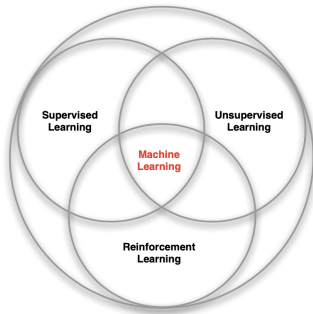
Mentor: izr. prof. dr. Marjetka Knez

Fakulteta za matematiko in fiziko

30. marec 2021

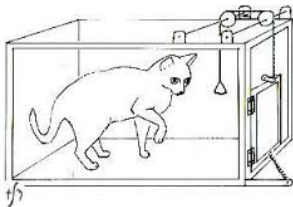
# Napovednik

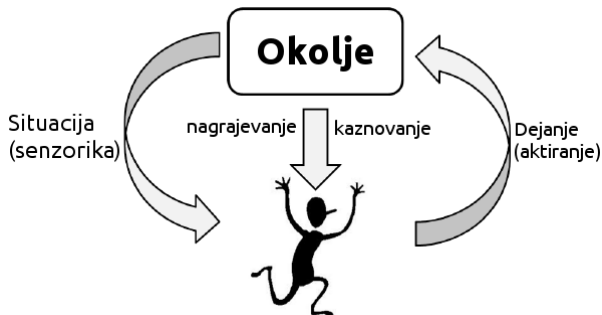
- ▶ Motivacija,
- ▶ problem spodbujevalnega učenja,
- ▶ algoritmi,
- ▶ namizne igre.



# Motivacija: Instrumentalno pogojevanje

- ▶ Psihološko motivirana podlaga.
- ▶ **Nagrade in kazni.**



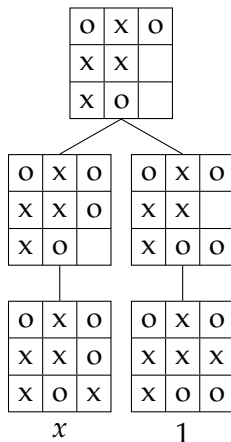


## Primer 1: robot se uči hoje

- ▶ **Situacija/Stanje:** položaj v sobi in stanje nog,
- ▶ **Nagrada:** 1 za doseg vrat, 2 za ključ,  $-0.5$  za časovni korak,
- ▶ **Okolje:** soba in senzorji, ki govorijo o položaju,
- ▶ **Akcija:** Premik noge.

## Primer 2: križci in krožci

- ▶ **Situacija/Stanje:** stanje na plošči,
- ▶ **Nagrada:** 1 za zmago,  $-1$  za poraz,  $x$  za izenačenje/potezo,
- ▶ **Okolje:** nasprotnik, plošča, sodnik, nagrajevalec,
- ▶ **Akcija:** postavitve X oz. O na ploščo.



# Ideja

- ▶ Agent »pade« v okolje.
- ▶ S poskušanjem se nauči pravih akcij.
- ▶ Svoje znanje izkoristi za maksimizacijo nagrade.



# Ideja

- ▶ Agent »pade« v okolje.
- ▶ S poskušanjem se nauči pravih akcij.
- ▶ Svoje znanje izkoristi za maksimizacijo nagrade.

## Hipoteza 1 (Hipoteza o nagradi).

*Vse cilje je mogoče opisati kot maksimizacijo neke kumulativne numerične nagrade.*

# Formalizacija: Markovski proces odločanja 1

## Definicija 2 (Markovska veriga).

*Slučajni proces  $(S_t)_{t=0}^T$  na končnem verjetnostnem prostoru  $(\Omega, \mathcal{F}, P)$  je **Markovska veriga**, če velja Markovska lastnost*

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, \dots, S_0 = s_0) = P(S_{t+1} = s_{t+1} \mid S_t = s_t)$$

# Formalizacija: Markovski proces odločanja 1

## Definicija 2 (Markovska veriga).

*Slučajni proces  $(S_t)_{t=0}^T$  na končnem verjetnostnem prostoru  $(\Omega, \mathcal{F}, P)$  je **Markovska veriga**, če velja Markovska lastnost*

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, \dots, S_0 = s_0) = P(S_{t+1} = s_{t+1} \mid S_t = s_t)$$

- ▶ Prihodnost je neodvisna od preteklosti, če poznamo sedanjost

# Formalizacija: Markovski proces odločanja 1

## Definicija 2 (Markovska veriga).

Slučajni proces  $(S_t)_{t=0}^T$  na končnem verjetnostnem prostoru  $(\Omega, \mathcal{F}, P)$  je **Markovska veriga**, če velja Markovska lastnost

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, \dots, S_0 = s_0) = P(S_{t+1} = s_{t+1} \mid S_t = s_t)$$

- ▶ Prihodnost je neodvisna od preteklosti, če poznamo sedanjost
- ▶  $p_{ss'} := P(S_{t+1} = s' \mid S_t = s) \rightarrow \mathcal{P} := [p_{ss'}]_{s,s' \in \mathcal{S}}$ ,  $\mathcal{S}$  je množica stanj
- ▶ Markovska veriga je torej dvojica  $(\mathcal{S}, \mathcal{P})$

# Formalizacija: Markovski proces odločanja 2

## Definicija 3 (Markovski proces nagrajevanja).

*Markovski proces nagrajevanja* je nabor  $(S, \mathcal{P}, \mathcal{R}, \gamma)$ , kjer je

- ▶  $S$  (končna) množica stanj,
- ▶  $\mathcal{P}$  prehodna matrika, kjer  $\mathcal{P}_{ss'} = P(S_{t+1} = s' \mid S_t = s)$ ,
- ▶  $\mathcal{R}$  nagradna funkcija  $\mathcal{R}_s = E[R_{t+1} \mid S_t = s]$ ,
- ▶  $\gamma \in [0, 1]$  je diskontni faktor.

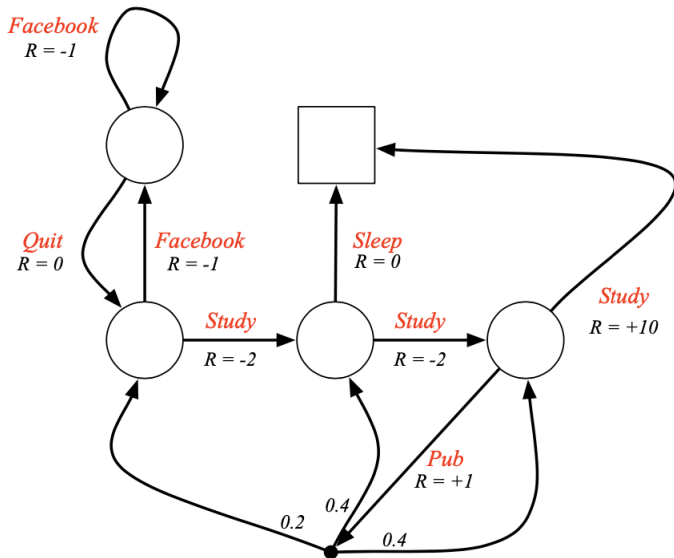
# Formalizacija: Markovski proces odločanja 3

## Definicija 4 (Markovski proces odločanja).

*Markovski proces odločanja (MDP) je nabor  $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ , kjer je*

- ▶  $\mathcal{S}$  (končna) množica stanj,
- ▶  $\mathcal{A}$  (končna) množica akcij oz. dejanj,
- ▶  $\mathcal{P}$  prehodna matrika, kjer  $\mathcal{P}_{ss'}^a = P(S_{t+1} = s' \mid S_t = s, \mathbf{A}_t = \mathbf{a})$ ,
- ▶  $\mathcal{R}$  nagradna funkcija  $\mathcal{R}_s^a = E[R_{t+1} \mid S_t = s, \mathbf{A}_t = \mathbf{a}]$ ,
- ▶  $\gamma \in [0, 1]$  diskontni faktor.

# Primer: MDP



# Agent 1

- ▶ Strategija (angl. *Policy*)
- ▶ Vrednostna funkcija (angl. *Value function*)
- ▶ (Model)



## Agent 2: strategija

### Definicija 5.

- ▶ *Deterministična strategija* stanju  $s$  priredi akcijo  $a$ ,

$$\pi(s) = a.$$

- ▶ *Stohastična strategija* za vsako stanje  $s$  pove verjetnosti vseh možnih akcij  $a$ ,

$$\pi(a|s) = P(A_t = a \mid S_t = s).$$

## Agent 3: vrednostna funkcija

### Definicija 6 (Povračilo).

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

### Definicija 7 (Vrednostna funkcija).

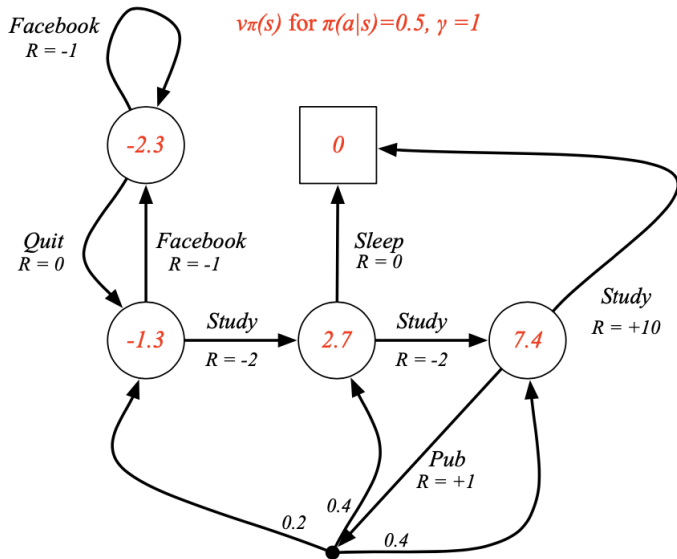
- ▶ *Vrednostna funkcija stanja je pričakovana vrednost povračila, če se vedemo skladno s strategijo  $\pi$*

$$v_{\pi}(s) = \mathbb{E}[G_t \mid S_t = s].$$

- ▶ *Vrednostna funkcija akcije je podobna prejšnji, le da sprosti prvo akcijo*

$$q_{\pi}(s, a) = \mathbb{E}[G_t \mid S_t = s, A_t = a].$$

# Primer: strategija in vrednostna funkcija



# Algoritmi

- ▶ Učenje prek strategije ali **vrednostne funkcije**.
- ▶ Celoten problem je **načrtovanje**:
  - ▶ Napovedovanje - ugotavljanje vrednosti.
  - ▶ Upravljanje - iskanje optimalne strategije.

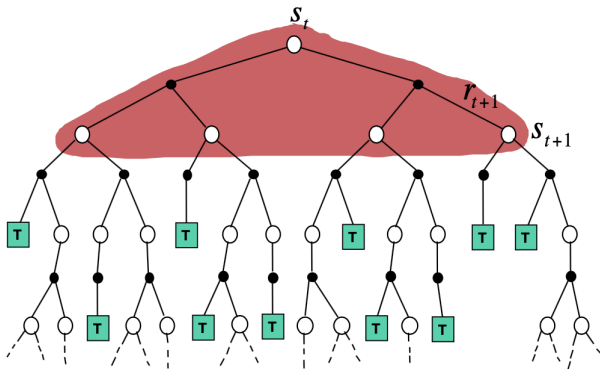
# Algoritmi: dinamično programiranje 1

- ▶ Poznamo  $\mathcal{P}_{ss'}^a$  in  $\mathcal{R}_s^a$ ,
- ▶ Bellmanove enačbe,
- ▶ vrednostna funkcija - ponovna uporaba rešitev,

$$\begin{aligned}v_{\pi}(s) &= \mathbb{E}[G_t \mid S_t = s] \\&= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right] \\&= \mathbb{E}\left[R_{t+1} + \sum_{k=1}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right] \\&= \mathbb{E}\left[R_{t+1} + \gamma \left(\sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k+1}\right) \mid S_t = s\right] \\&= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\&= \mathbb{E}\left[R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s\right].\end{aligned}$$

## Algoritmi: dinamično programiranje 2

$$V(S_t) \leftarrow \mathbb{E}_{\pi} [R_{t+1} + \gamma V(S_{t+1})]$$



# Algoritmi: Monte Carlo 1

- ▶ Nepoznan epizodični MDP,
- ▶ problem napovedovanja,
- ▶ empirično povračilo,
- ▶ štejemo obiske stanj.

## Algoritmi: Monte Carlo 2

- Ob prvem obisku stanja  $s$ :

$$N(s) \leftarrow N(s) + 1$$

$$S(s) \leftarrow S(s) + G_t$$

- Po koncu učenja:

$$V(s) \leftarrow S(s)/N(s)$$

- Pomni: Računanje povprečja zaporedja  $(X_i)_{i \in \mathbb{N}}$

$$\mu_k = \frac{1}{k} \sum_{j=1}^k X_j = \mu_{k-1} + \frac{1}{k} (X_k - \mu_{k-1})$$

- Inkrementalni Monte Carlo:

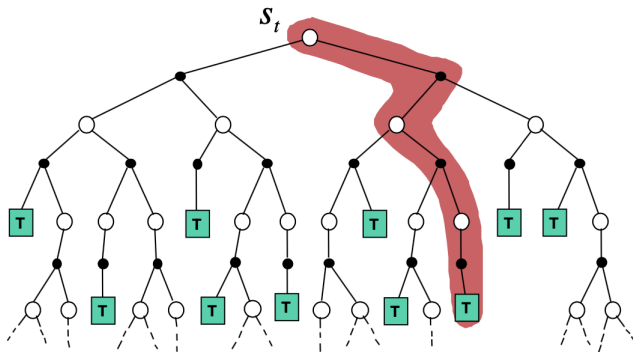
$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(t)} (G_t - V(S_t))$$



# Algoritmi: Monte Carlo 3

- Inkrementalni Monte Carlo:

$$V(S_t) \leftarrow V(S_t) + \alpha (G_t - V(S_t))$$



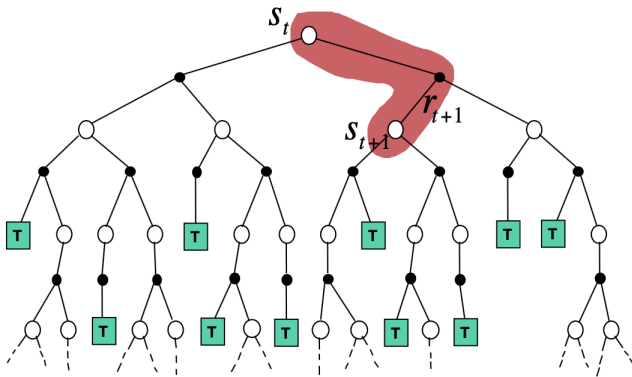
- Splošni obrazec:

$$nova\ ocena \leftarrow stara\ ocena + korak\ (tarča - stara\ ocena).$$

## Algoritmi: TD(0)

- ▶ Učenje s časovno razliko.
- ▶ *Bootstrapping*.
- ▶ Ne potrebujejo povračila.
- ▶  $G_t \approx R_{t+1} + \gamma V(S_{t+1})$ .

$$V(S_t) \leftarrow V(S_t) + \alpha (R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$

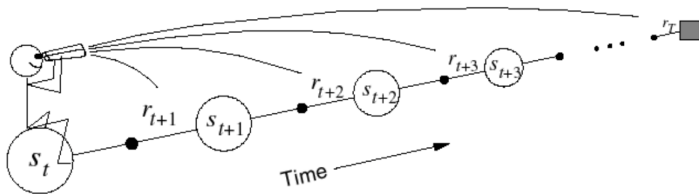


# Algoritmi: TD( $\lambda$ ) 1

- ▶ Povezava med MC in TD(0).
- ▶  $G_t^{(n)} = R_{t+1} + \dots + \gamma^{n-1}R_{t+n} + \gamma^n V(S_{t+n})$ .
- ▶ Povprečenje različnih  $G_t^{(n)}$ :  $G_t^\lambda = (1 - \lambda) \sum_{n=1}^{\infty} \lambda^{(n-1)} G_t^{(n)}$ .

TD( $\lambda$ ) s pogledom naprej:

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t^\lambda - V(S_t)).$$

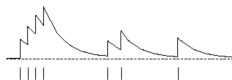


## Algoritmi: TD( $\lambda$ ) 2

- ▶ **Sledi upravičenosti** (angl. *eligibility traces*):

$$E_0(s) = 0,$$

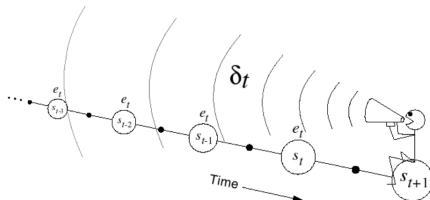
$$E_t(s) = \gamma \lambda E_{t-1}(s) + \mathbb{1}(S_t = s),$$



- $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ .

**TD( $\lambda$ ) s pogledom nazaj:**

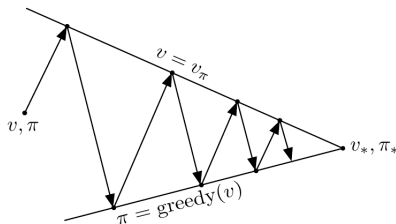
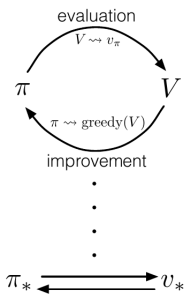
$$V(s) \leftarrow V(s) + \alpha \delta_t E_t(s).$$



# Spreminjanje strategije - upravljanje

- ▶ Potrebujemo vrednostno funkcijo akcij.
- ▶ raziskovanje in izkoriščanje.
- ▶  $\epsilon$ -požrešna izbira akcij:

$$\pi(a|s) = \begin{cases} \epsilon/m + 1 - \epsilon & \text{če } a^* = \arg \max_{a \in \mathcal{A}} Q(s, a) \\ \epsilon/m & \text{sicer} \end{cases}$$



# Konvergenca

► **GLIE:**

$$\lim_{k \rightarrow \infty} N_k(s, a) = \infty,$$

$$\lim_{k \rightarrow \infty} \pi_k(a|s) = \mathbb{1}(a = \arg \max_{a' \in \mathcal{A}} Q_k(s, a')).$$

► **Robbins-Monro** zaporedje *korakov*  $\alpha_t$ :

$$\sum_{t=1}^{\infty} \alpha_t = \infty,$$

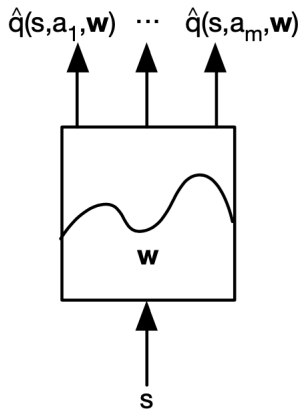
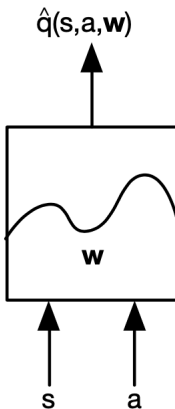
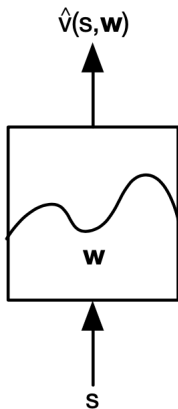
$$\sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

# Težave

- ▶ Veliki MDP-ji:
  - ▶ Križci in krožci:  $3^9$  / 4578 / 765 stanj,
  - ▶ Štiri v vrsto: 4.531.985.219.092 stanj,
  - ▶ Šah: približno  $10^{46}$  stanj,
  - ▶ Go:  $10^{170}$  stanj,
- ▶ Vsi zgornji algoritmi so tabelarični.
- ▶ Počasno učenje.

# Aproksimacija

- ▶ Linearna Aproksimacija.
- ▶ Nevronske mreže.





# Namizne igre: posebnosti

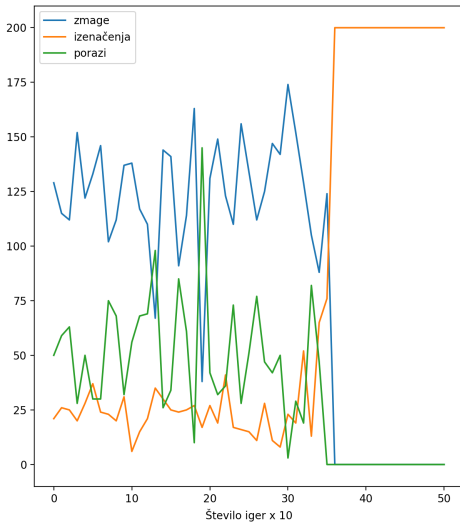
- ▶ »Postanja«.
- ▶ Trening:
  - ▶ Fiksiran nasprotnik,
  - ▶ naključni nasprotnik,
  - ▶ samoigra.
- ▶ Več agentov:  $\pi = \langle \pi^1, \pi^2 \rangle$ .
- ▶ Iskanje.

$$v_*(s) = \max_{\pi^1} \min_{\pi^2} v_{\pi}(s)$$

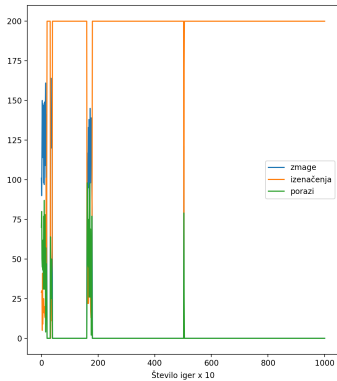
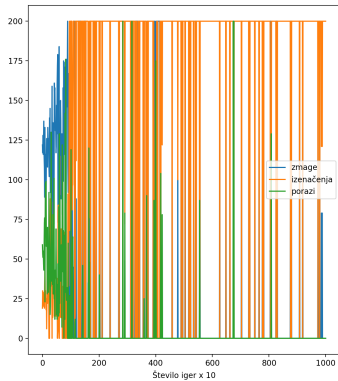
# m,n,k-igra

- ▶ Dva igralca, vsota nič, ekstenzivna,
- ▶  $m \times n$  plošča,
- ▶  $k$  v vrsto,
- ▶ pravila križcev in krožcev (3,3,3-igra),
- ▶ prilagoditve: gravitacija.

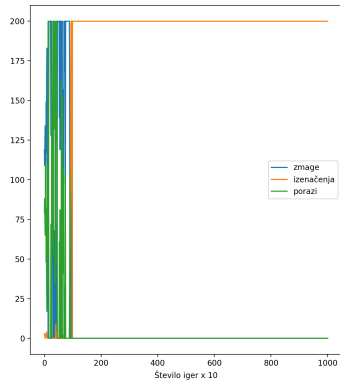
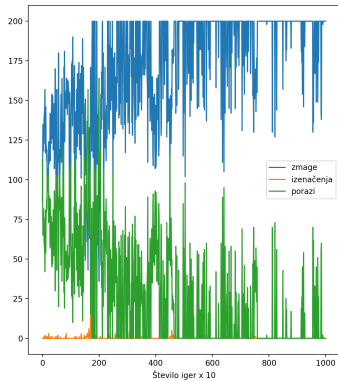
# 3,3,3-igra 1



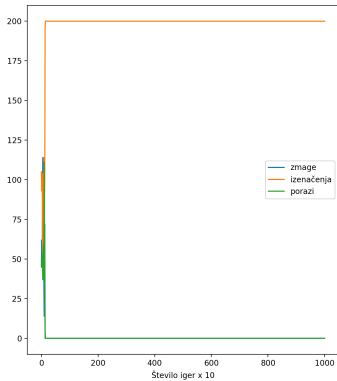
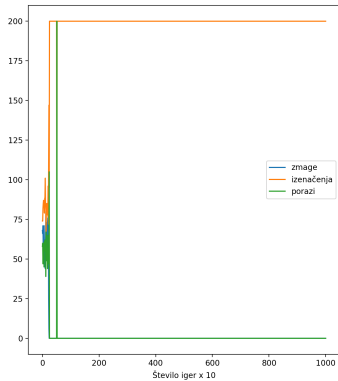
## 3,3,3-igra 2



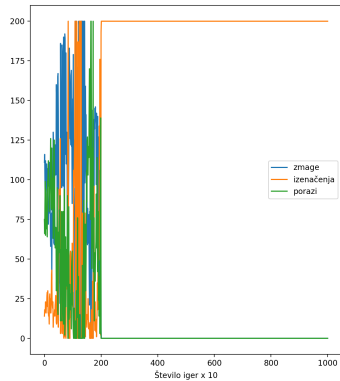
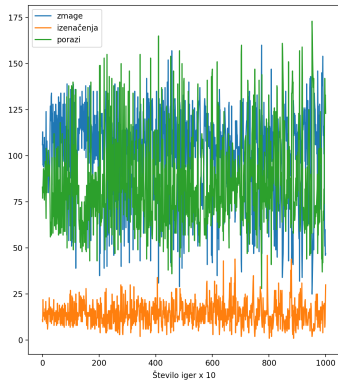
# 4,4,3-igra



# 4,4,4-igra



# 5,5,4-igra



# Literatura I



Richard E. Bellman.

Dynamic Programming.

Princeton University Press, Princeton, 1957.



Richard E. Bellman.

A markov decision process.

Journal of Mathematical Mechanics, (6), 1957.



Imran Ghory.

Reinforcement learning in board games.

2004.



David Silver.

Introduction to reinforcement learning.

[https://deepmind.com/learning-resources/  
-introduction-reinforcement-learning-david-silver](https://deepmind.com/learning-resources/-introduction-reinforcement-learning-david-silver)  
2015.



# Literatura II



Richard S. Sutton and Andrew G. Barto.

Reinforcement Learning: An introduction.

The MIT Press, Cambridge, Massachusetts, 2 edition, 2015.



Csaba Szepesvari.

Algorithms for Reinforcement Learning.

Morgan & Claypool Publishers, Alberta, Canada, 2009.