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

Tim Kalan

Mentor: izr. prof. dr. Marjetka Knez

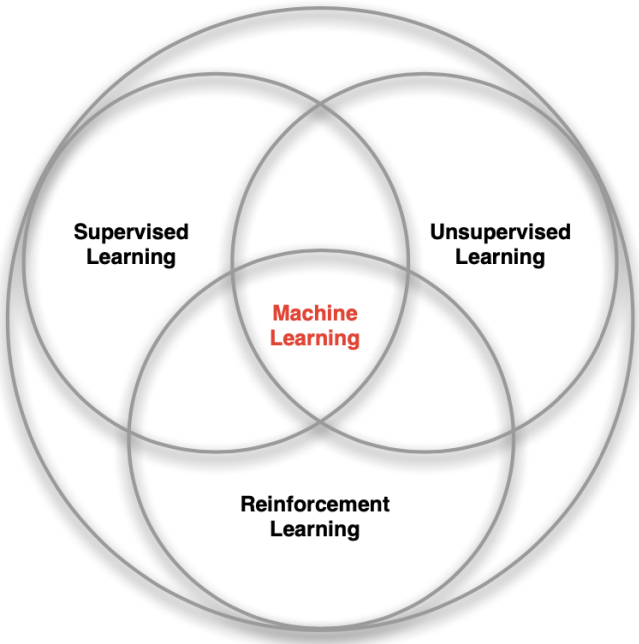
Fakulteta za matematiko in fiziko

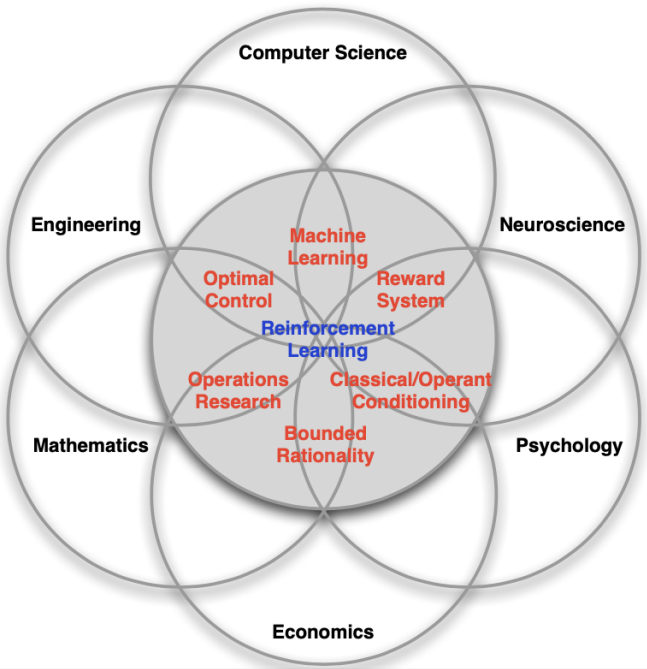
30. marec 2021

Napovednik

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

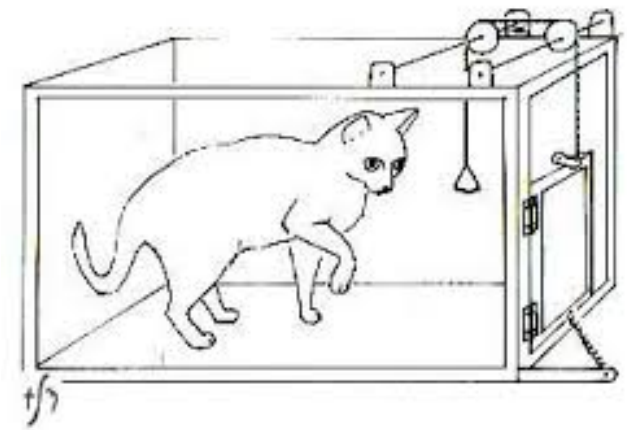
Umestitev


Supervised Learning
Unsupervised Learning
Machine Learning
Reinforcement Learning

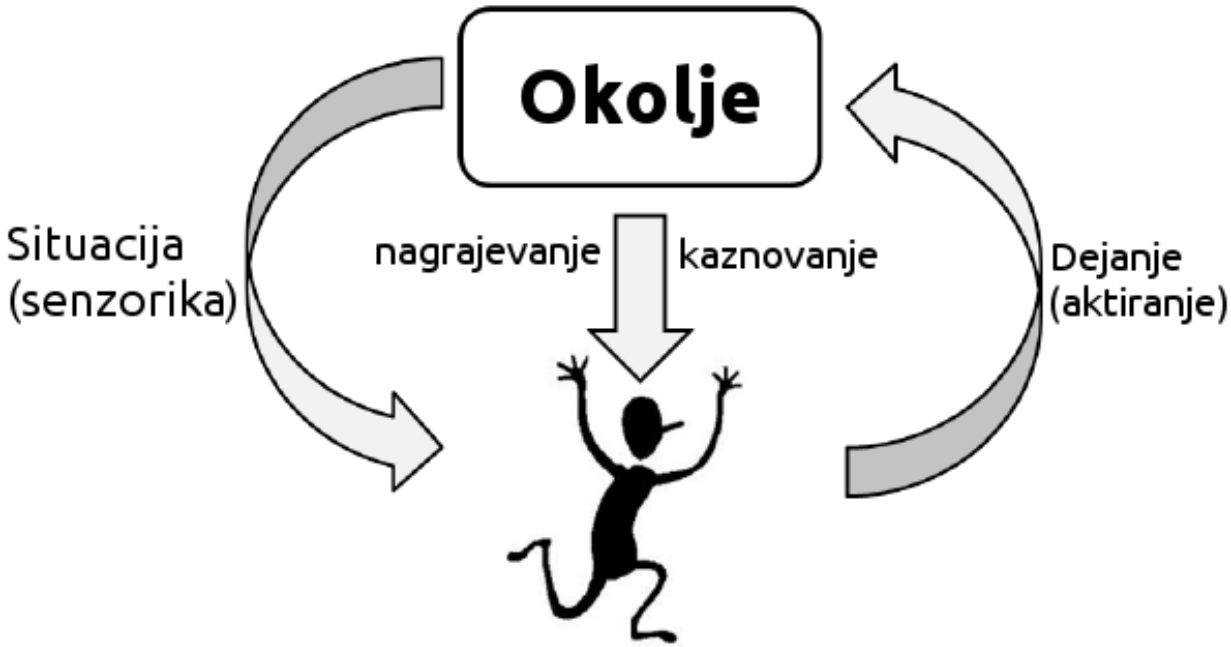

Computer Science
Engineering
Neuroscience
Machine Learning
Optimal Control
Reward System
Reinforcement Learning
Operations Research
Classical/Operant Conditioning
Bounded Rationality
Mathematics
Psychology
Economics

Motivacija: Instrumentalno pogojevanje

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



Okvir

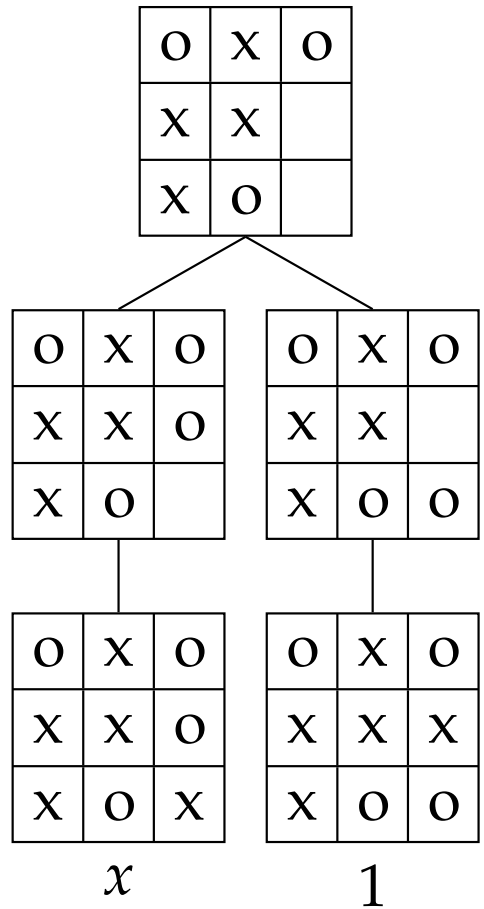

Okolje
Situacija
(senzorika)
nagrajevanje
kaznovanje
Dejanje
(aktiranje)

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**: postavitev X oz. O na ploščo.



Ideja

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

Ideja

- Agent »pade« v okolje.
- S poskušanjem se nauči pravilnih 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*

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

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*

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

- 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*

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

- 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* $(\mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma)$*, kjer je*

- $\mathcal{S}$ *(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.*

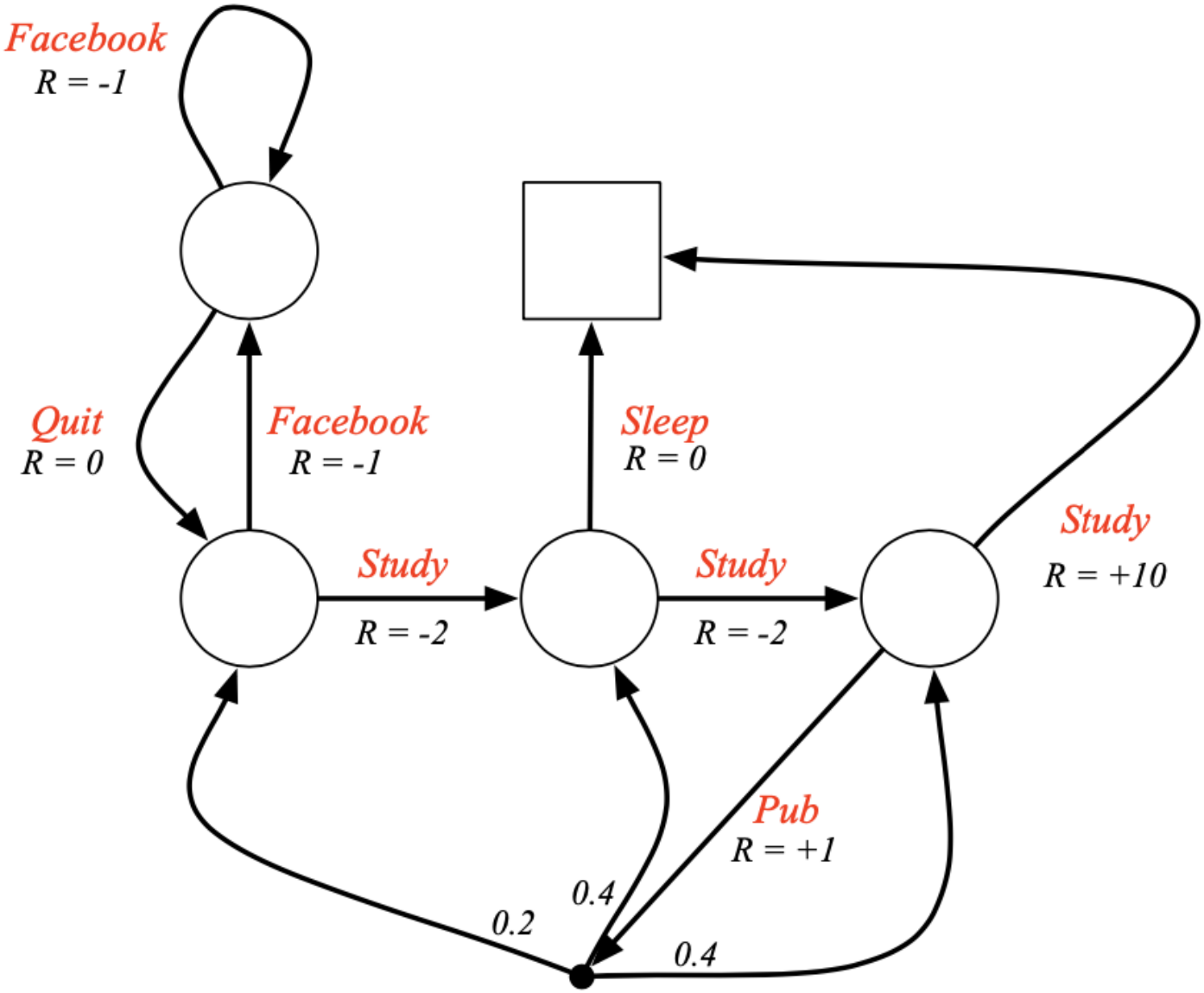
Formalizacija: Markovski proces odločanja 3

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* $\mathcal{P}^a_{ss'} = P(S_{t+1} = s' \mid S_t = s, \mathbf{A_t} = \mathbf{a})$*,*
- $\mathcal{R}$ *nagradna funkcija* $\mathcal{R}^a_s = E[R_{t+1} \mid S_t = s, \mathbf{A_t} = \mathbf{a}]$*,*
- $\gamma \in [0, 1]$ *diskontni faktor.*

Primer: MDP


Facebook
R = -1
Quit
R = 0
Facebook
R = -1
Sleep
R = 0
Study
R = -2
Study
R = -2
Study
R = +10
Pub
R = +1
0.2
0.4
0.4

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).

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

Definicija 7 (Vrednostna funkcija).

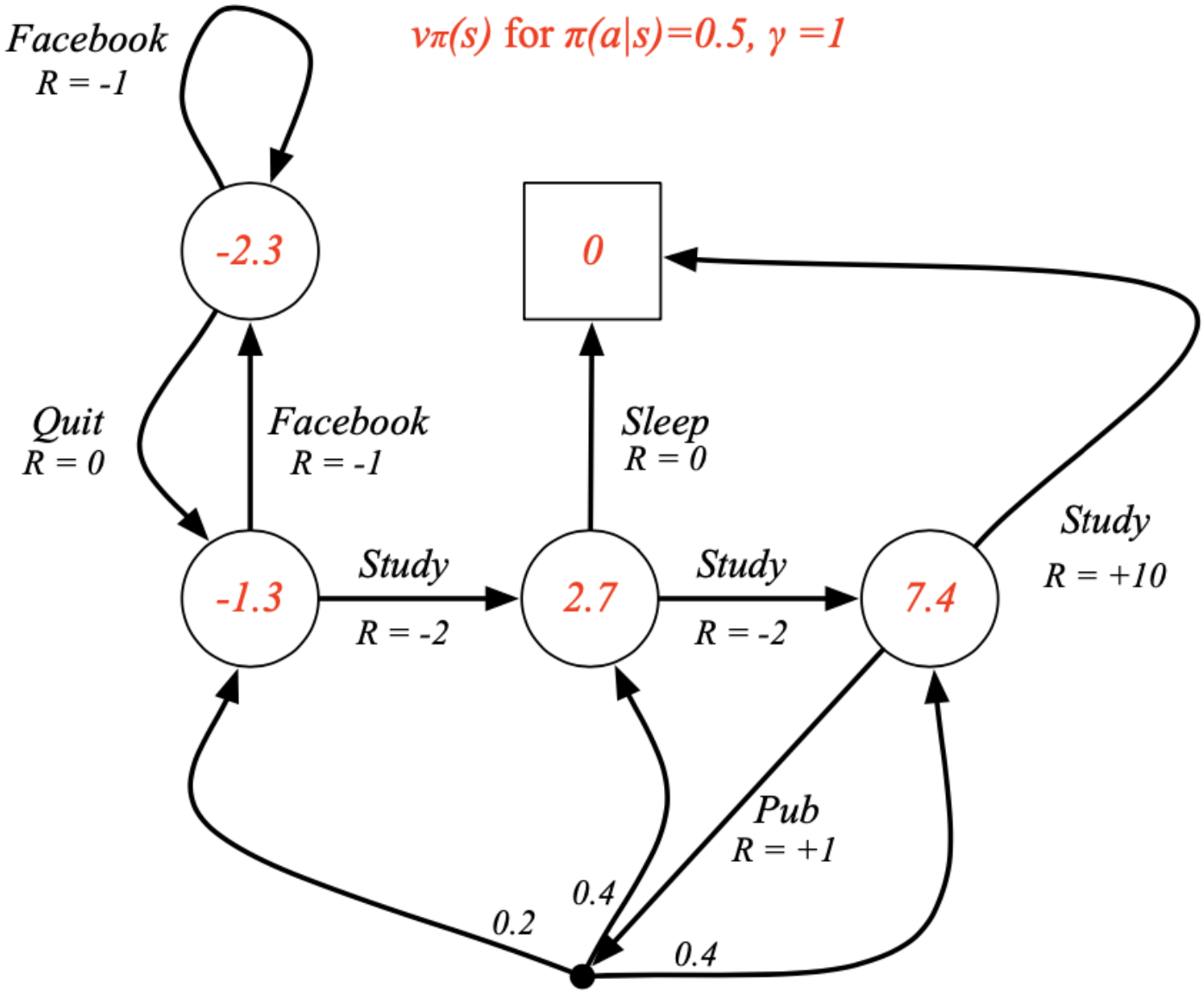
- ***Vrednostna funkcija stanja*** *je pričakovana vrednost povračila, če se vedemo skladno s strategijo* π

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

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

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

Primer: strategija in vrednostna funkcija



Algoritmi

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

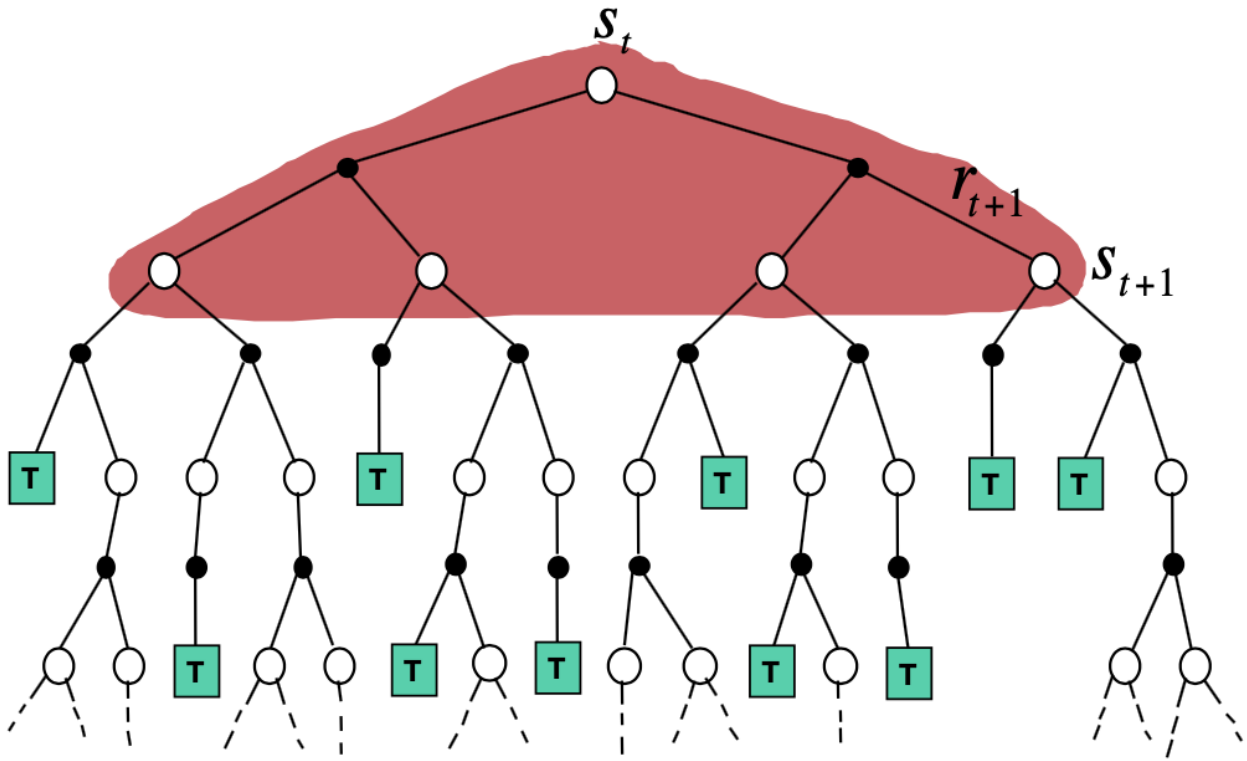
Algoritmi: dinamično programiranje 1

- Poznamo $\mathcal{P}^a_{ss'}$ in $\mathcal{R}^a_s$,
- Bellmanove enačbe,
- vrednostna funkcija - ponovna uporaba rešitev,

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

Algoritmi: dinamično programiranje 2

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



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 $(X_i)_{i \in \mathbb{N}}$

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

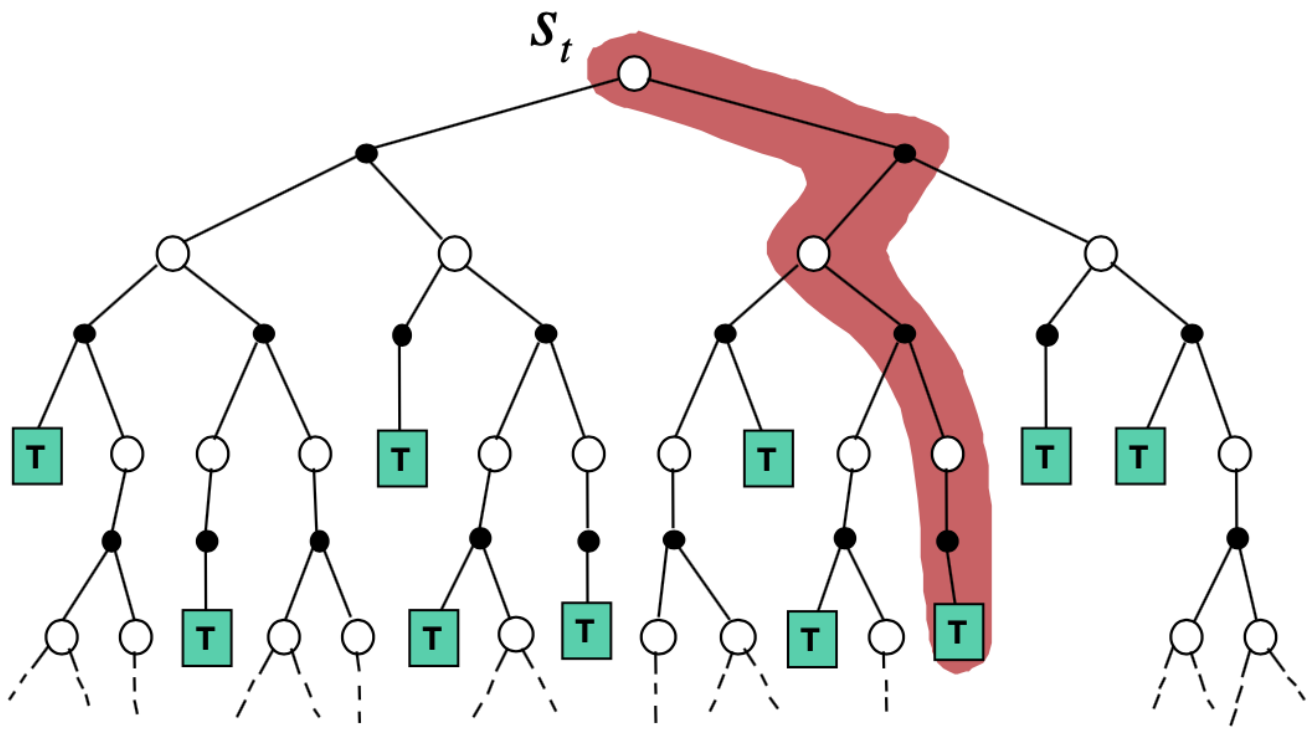
- Inkrementalni Monte Carlo:

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

Algoritmi: Monte Carlo 3

- Inkrementalni Monte Carlo:

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



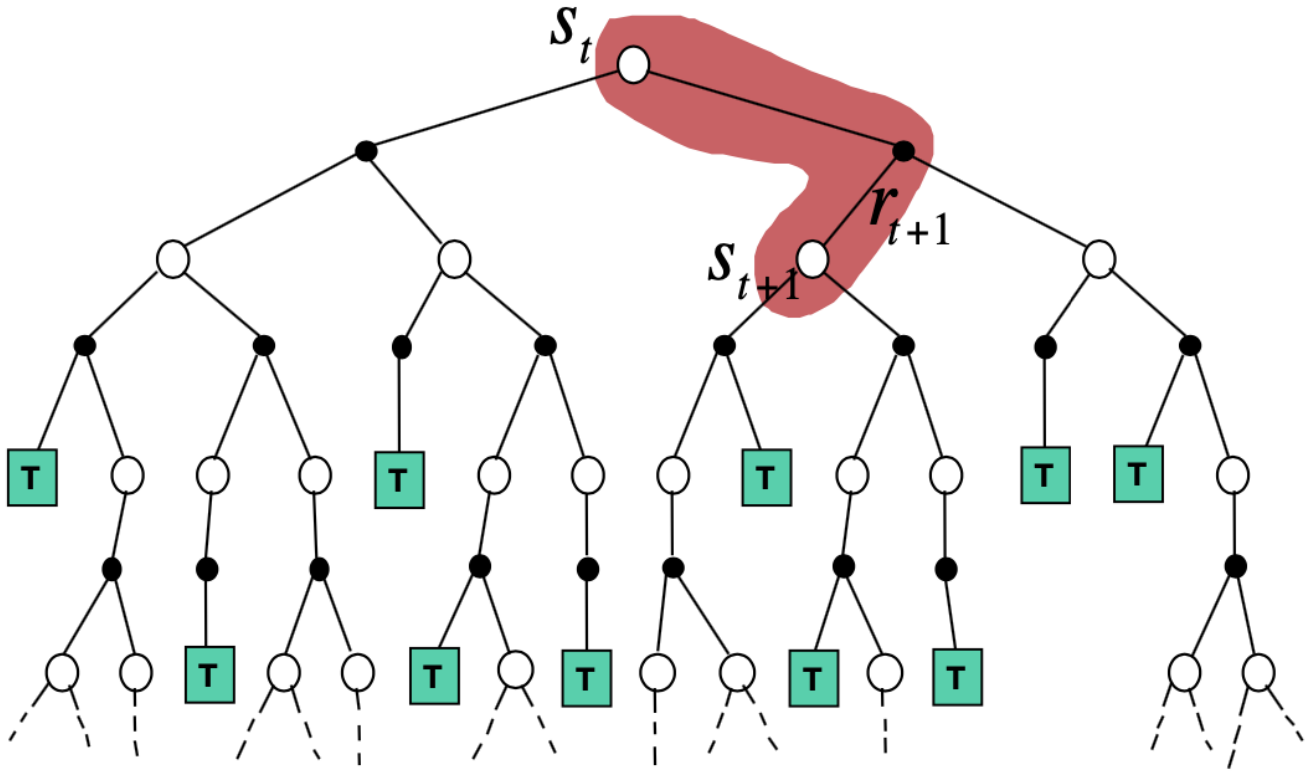
- Splošni obrazec:

$$\textit{nova ocena} \leftarrow \textit{stara ocena} + \textit{korak}\,(\textit{tarča} - \textit{stara ocena}).$$

Algoritmi: TD(0)

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

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

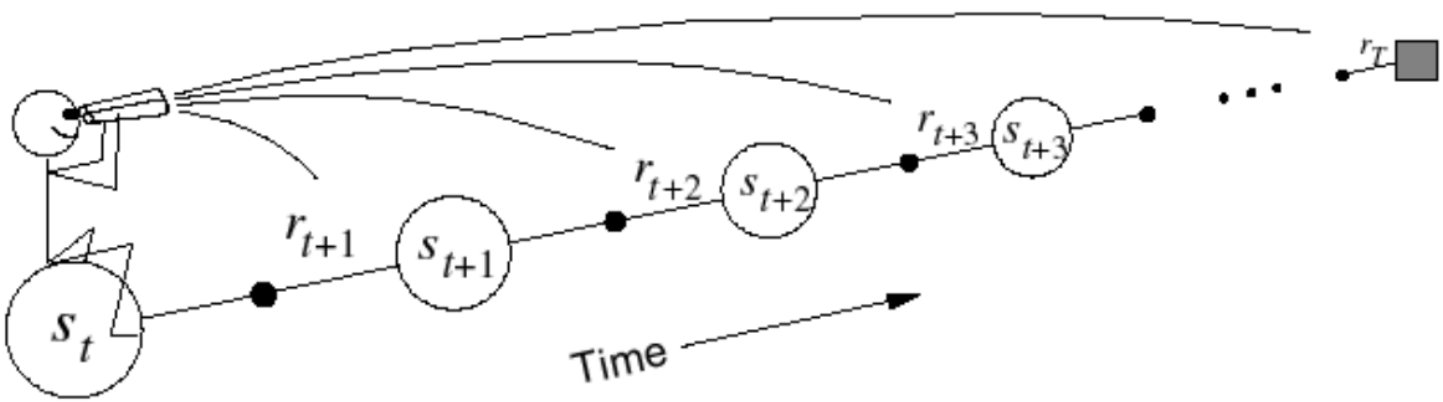


Algoritmi: TD(λ) 1

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

TD(λ) s **pogledom naprej**:

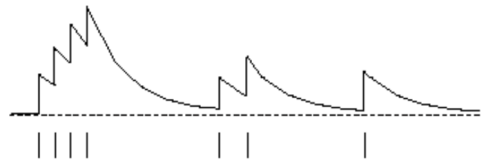
$$V(S_t) \leftarrow V(S_t) + \alpha(G_t^\lambda - V(S_t)).$$



Algoritmi: TD(λ) 2

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

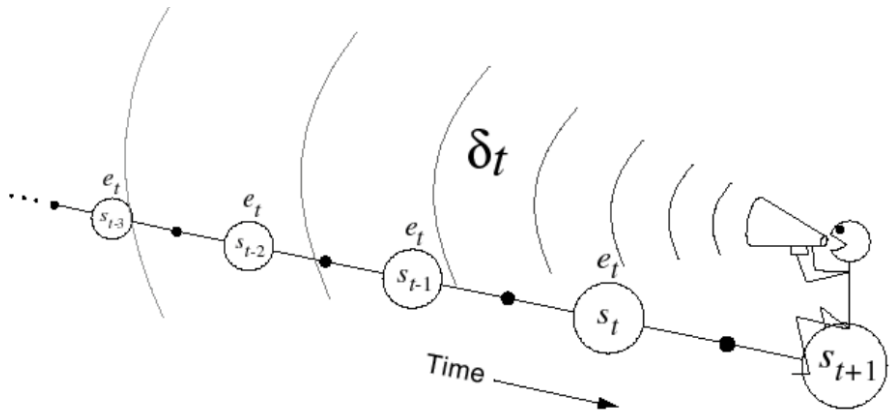
$$E_0(s) = 0,$$
$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbb{1}(S_t = s),$$



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

TD(λ) s **pogledom nazaj**:

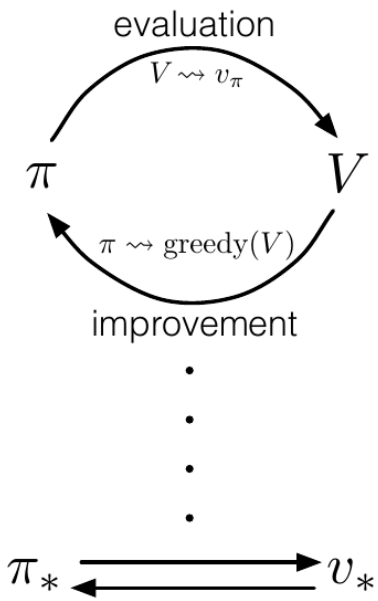
$$V(s) \leftarrow V(s) + \alpha\delta_t E_t(s).$$

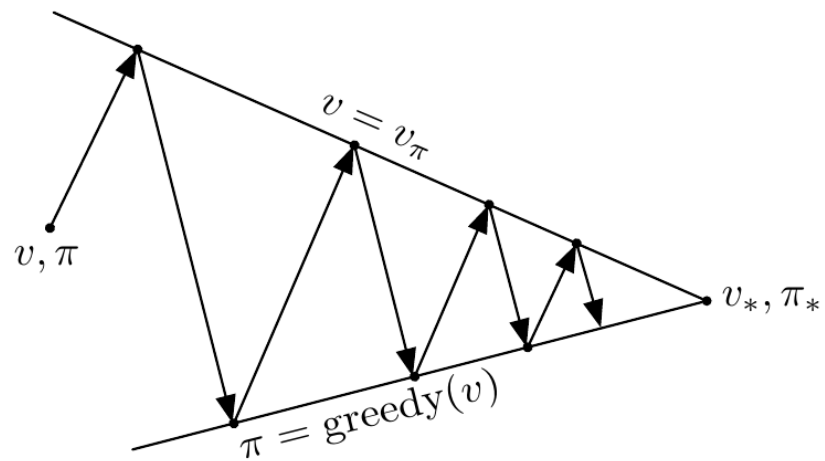


Spreminjanje strategije - upravljanje

- Potrebujemo vrednostno funkcijo akcij.
- raziskovanje in izkoriščanje.
- ϵ-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**:

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

- **Robbins-Monro** zaporedje *korakov* α_t:

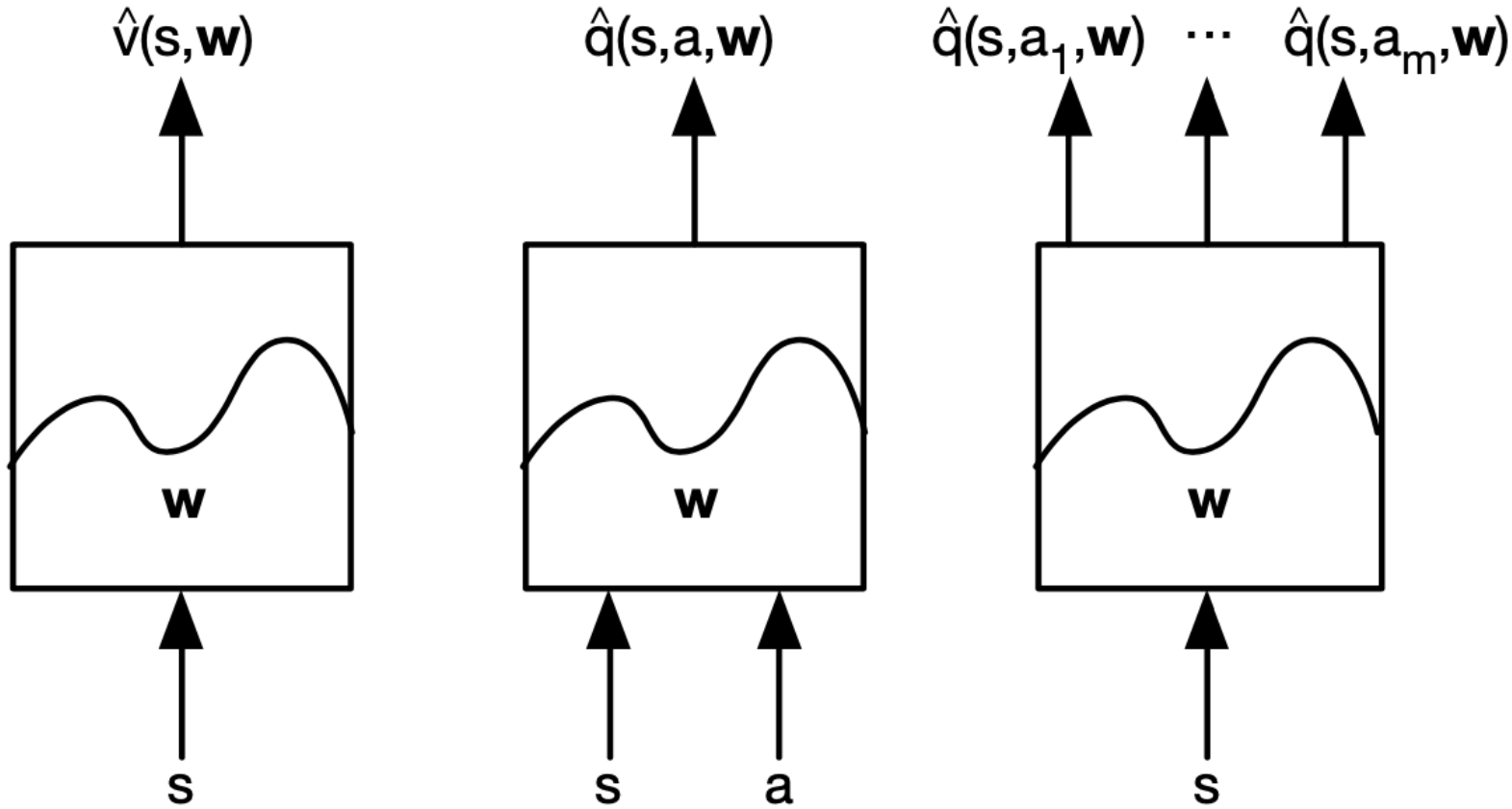
$$\sum_{t=1}^{\infty} \alpha_t = \infty,$$
$$\sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

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

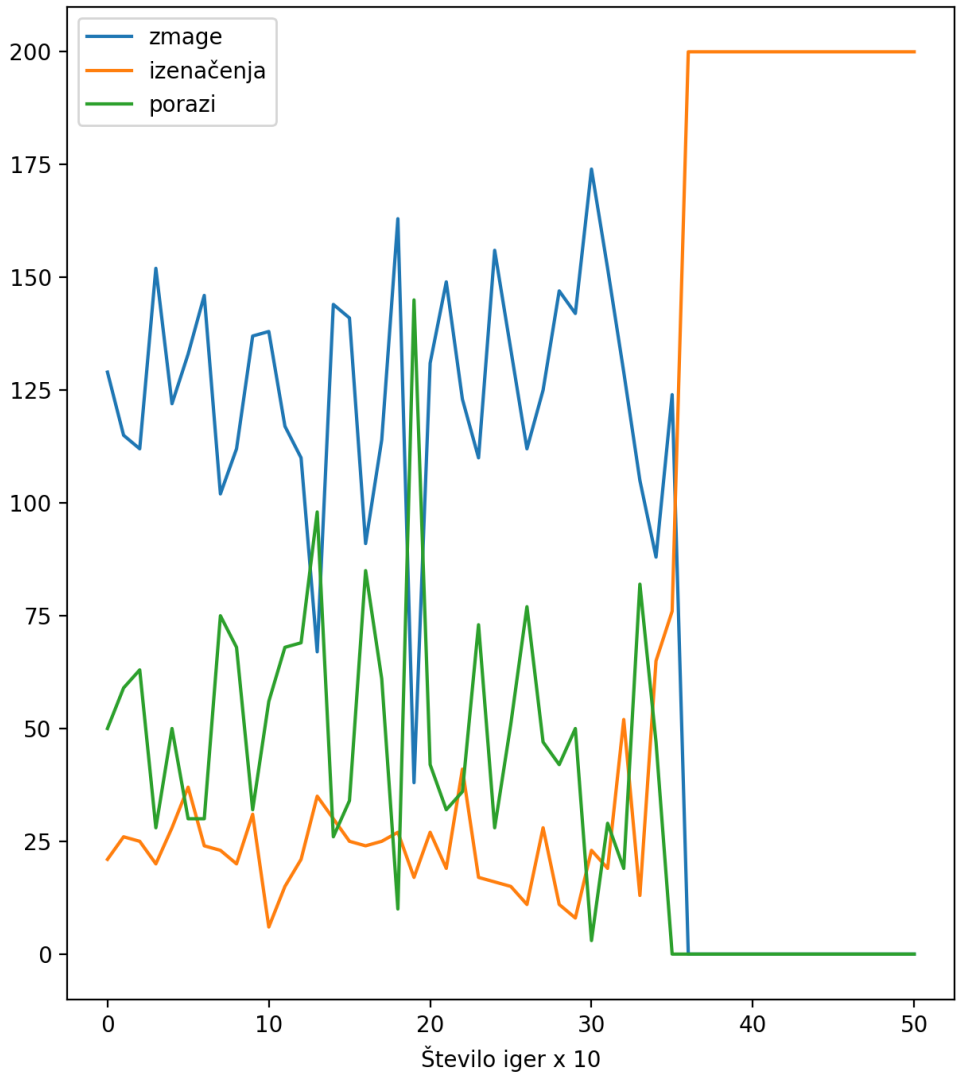
- »Postanja«.
- Trening:
  - Fiksiran nasprotnik,
  - naključni nasprotnik,
  - samoigra.
- Več agentov: $\pi = \langle \pi^1, \pi^2 \rangle$.
- Iskanje.

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

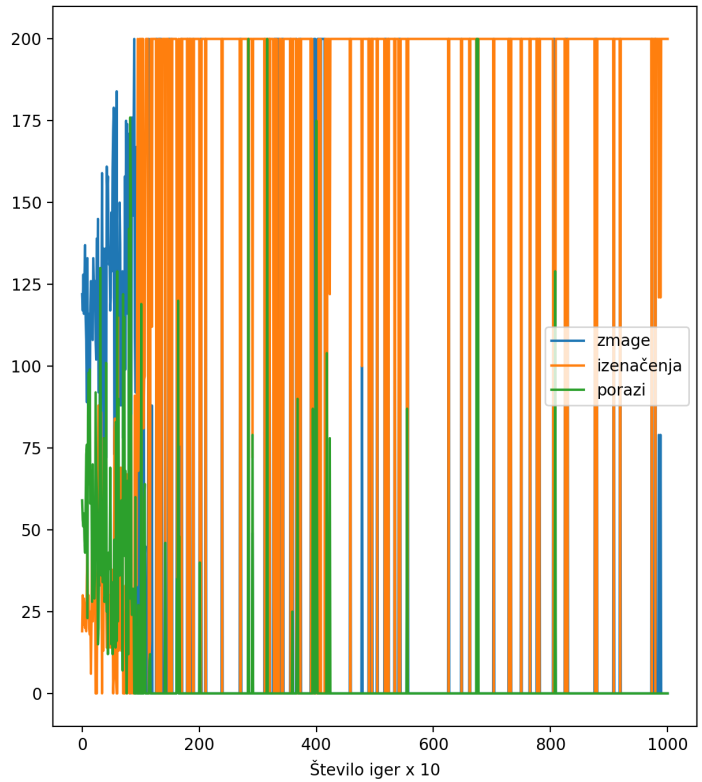
m,n,k-igra

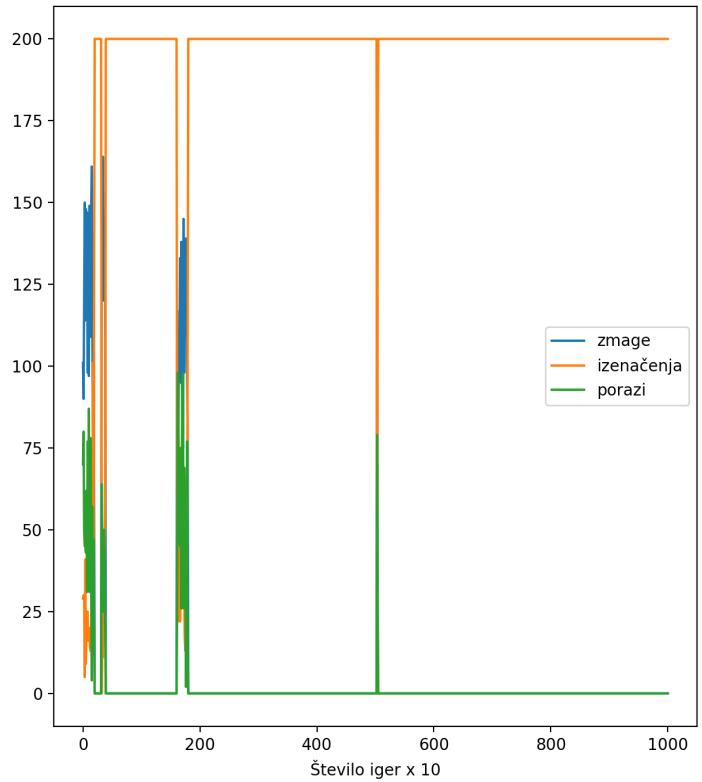
- Dva igralca, vsota nič, ekstenzivna,
- $m \times n$ plošča,
- k v vrsto,
- pravila križcev in krožcev (3,3,3-igra),
- prilagoditve: gravitacija.

3,3,3-igra 1

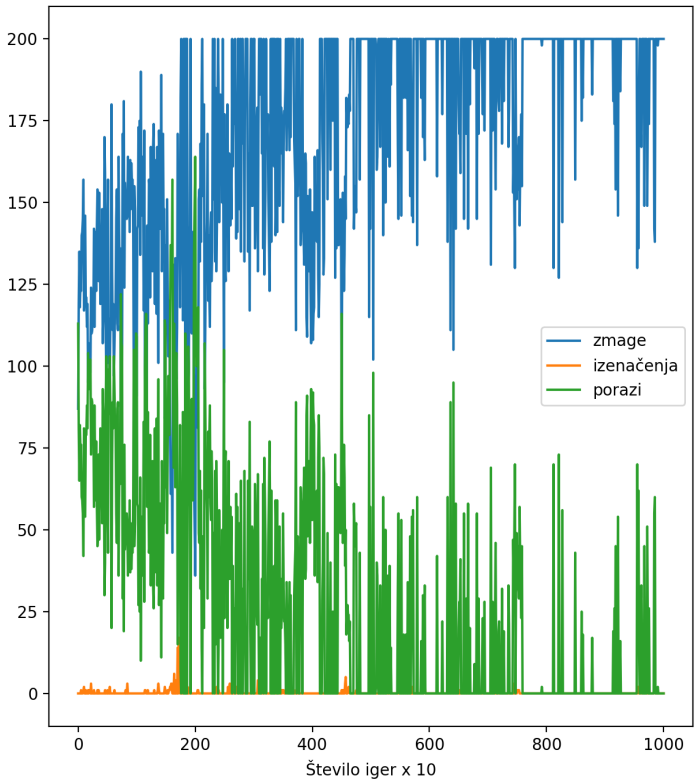

zmage
izenačenja
porazi
200
175
150
125
100
75
50
25
0
0
10
20
30
40
50
Število iger x 10

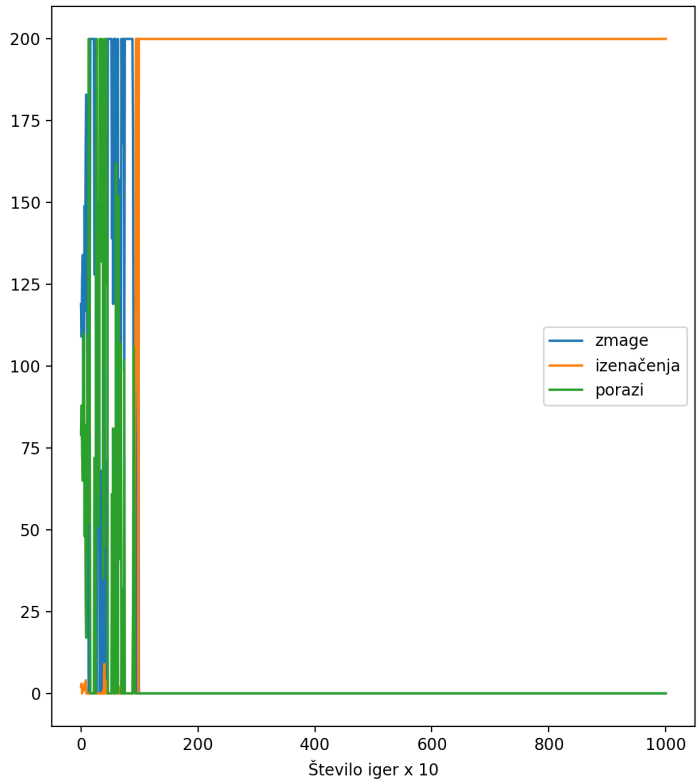
3,3,3-igra 2


200
175
150
125
100
75
50
25
0
0
200
400
600
800
1000
Število iger x 10
zmage
izenačenja
porazi

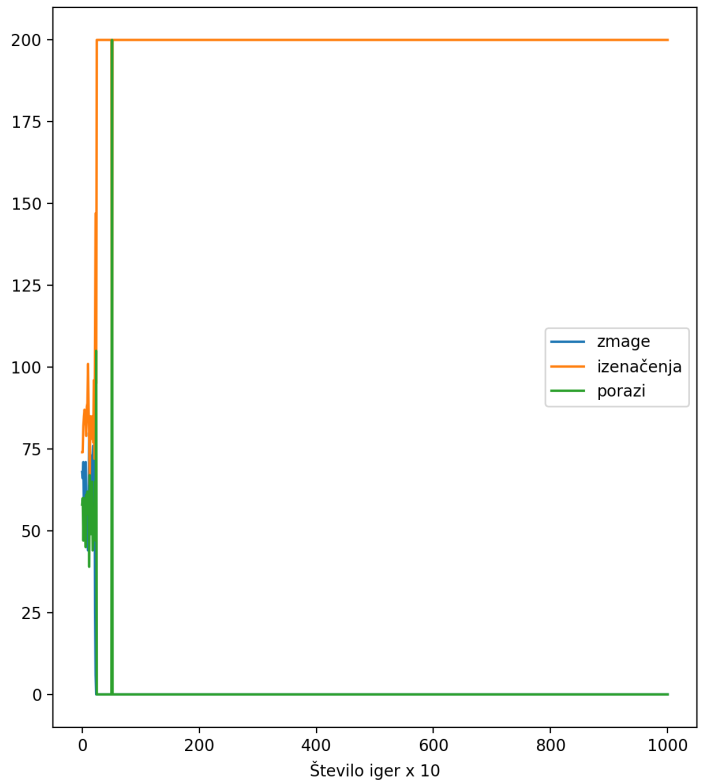

200
175
150
125
100
75
50
25
0
0
200
400
600
800
1000
Število iger x 10
zmage
izenačenja
porazi

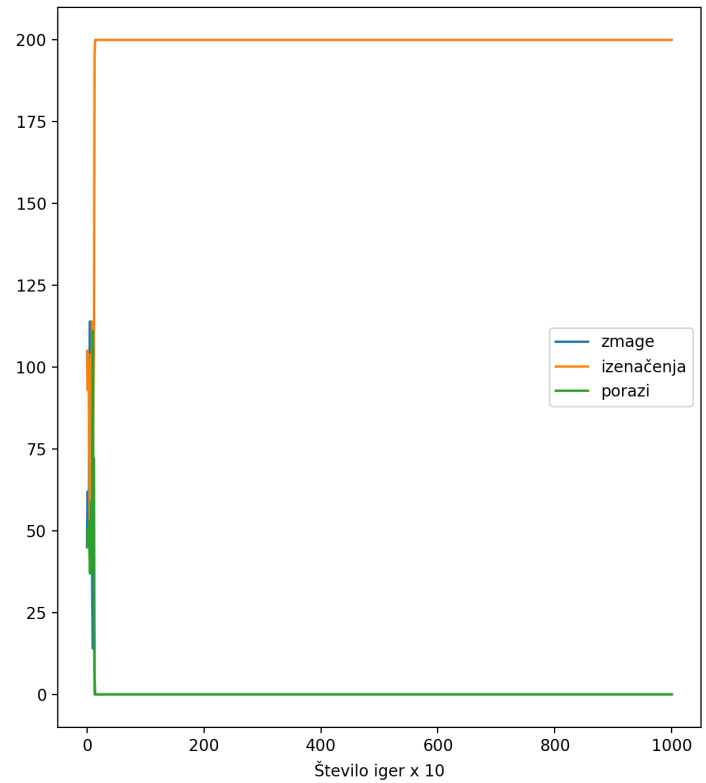
4,4,3-igra


200
175
150
125
100
75
50
25
0
0
200
400
600
800
1000
Število iger x 10
zmage
izenačenja
porazi

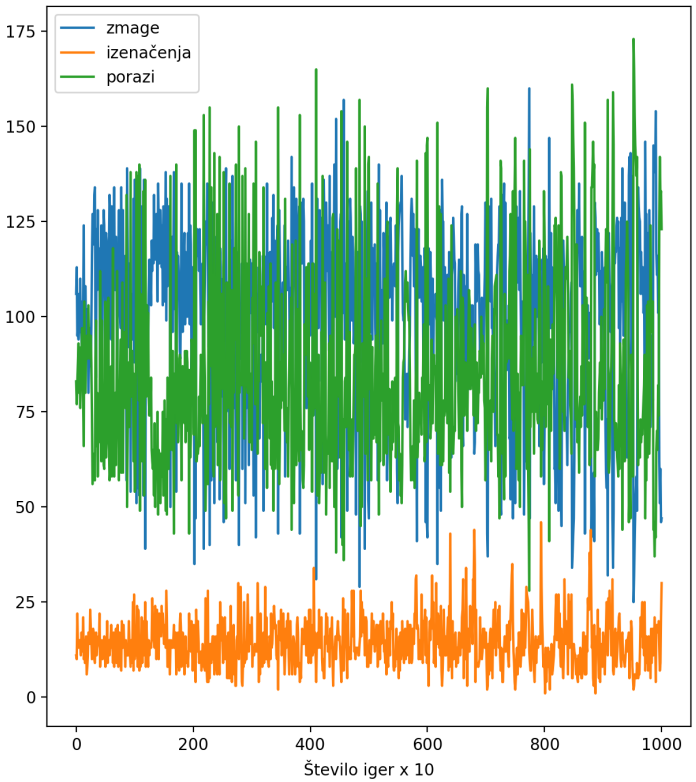

200
175
150
125
100
75
50
25
0
0
200
400
600
800
1000
Število iger x 10
zmage
izenačenja
porazi

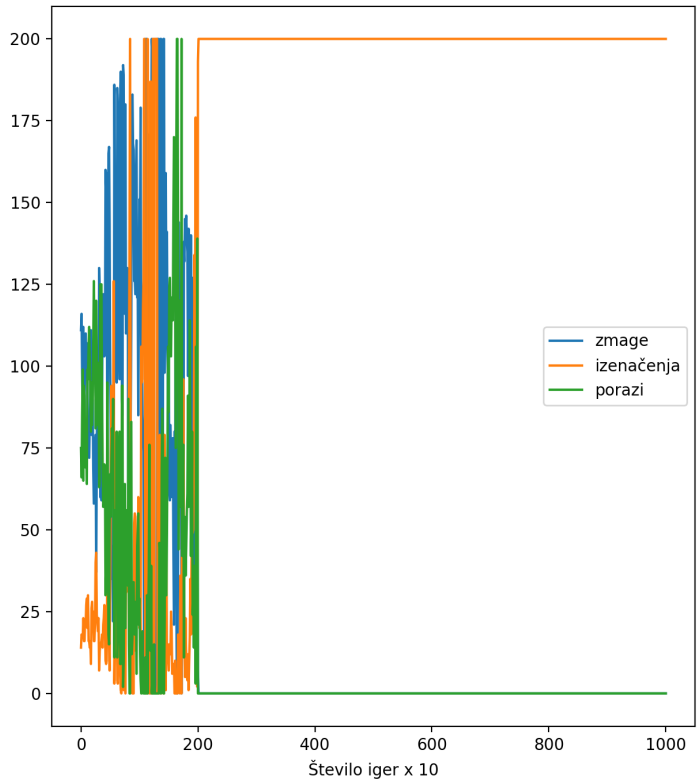
4,4,4-igra


200
175
150
125
100
75
50
25
0
0
200
400
600
800
1000
Število iger x 10
zmage
izenačenja
porazi


200
175
150
125
100
75
50
25
0
0
200
400
600
800
1000
Število iger x 10
zmage
izenačenja
porazi

5,5,4-igra


zmage
izenačenja
porazi
175
150
125
100
75
50
25
0
0
200
400
600
800
1000
Število iger x 10


200
175
150
125
100
75
50
25
0
zmage
izenačenja
porazi
0
200
400
600
800
1000
Število iger x 10

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-silve
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.