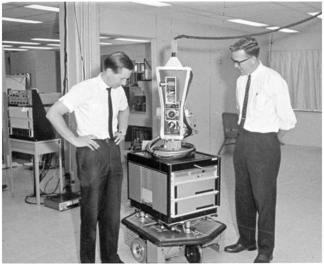
# Aproximácia funkcie ohodnotení v algoritmoch Q-learning

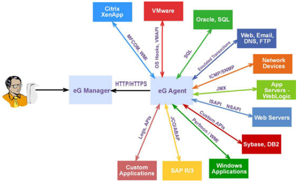
Ing. Michal CHOVANEC
Fakulta riadenia a informatiky

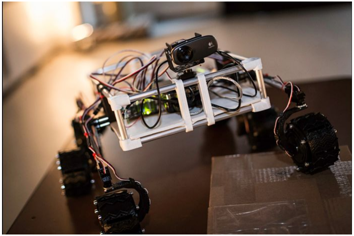
*Marec 2016*

# Obsah

# Agentové systémy

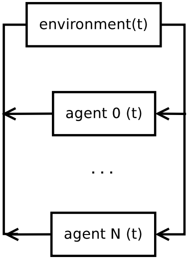


# Agentové systémy

Racionálny agent :

- Schopný vnímať prostredie
- Robiť rozhodnutia
- Pre každú možnú postupnosť vstupov vyberá akciu maximalizujúcu očakavaný výkon




Obr. : Multiagentný systém

# Adaptívne a učiace sa systémy

**Adaptívny systém**

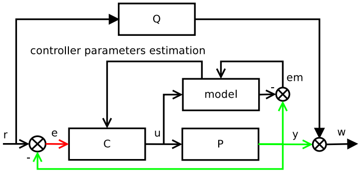
- reaktívne správanie
- dôraz na čas
- nemá pamäť - bez očakávaní
- rychlá dynamika

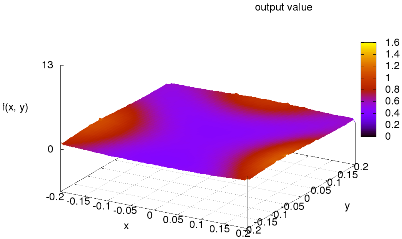
**Učiaci sa systém**

- konštruktívne správanie
- dôraz na priestor
- má pamäť - očakávania
- pomalá dynamika

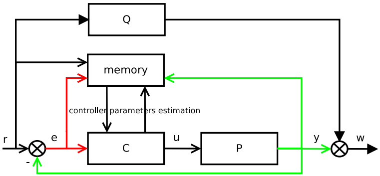
# Adaptívne a učiace sa systémy

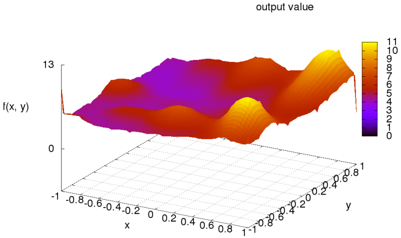
## Adaptívny systém




## Učiaci sa systém

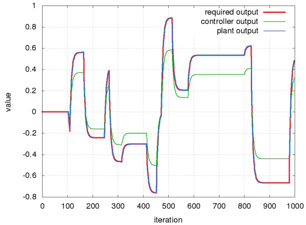


**Adaptívny systém**

PID regulátor

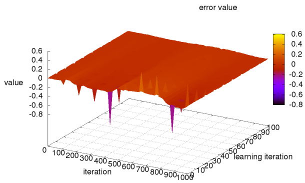
$$C(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - z^{-1}}$$




**Učiaci sa systém**

Iterative learning controll

$$C(w) = \frac{\gamma}{1 - w^{-1}}$$



# Q-learning algoritmus

Daná je množina stavov a akcií

$$s \in \mathbb{S}$$
$$a \in \mathbb{A}$$

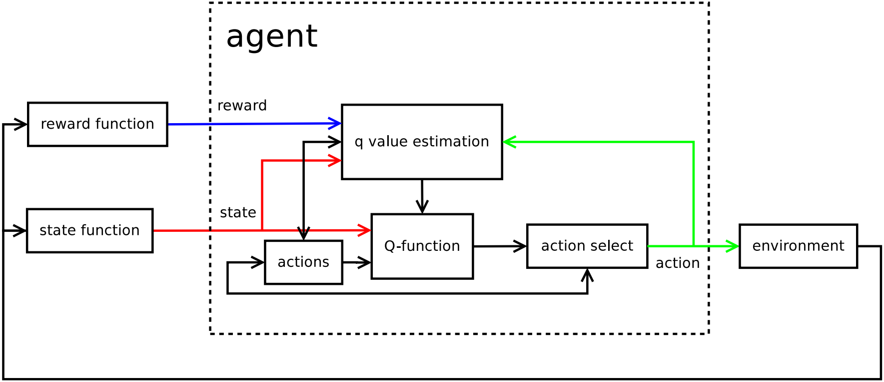
kde $\mathbb{S} \in \mathbb{R}^{N_s}$ a $\mathbb{A} \in \mathbb{R}^{N_a}$ .
Predpoklad : v prostedí existuje funkcia

$$s(n+1) = \lambda(s(n), a(n)) \tag{1}$$

prechodová funkcia zo stavu $s(n)$ použitím akcie $a(n)$ - táto funkcia je ale agentovi neznáma.
Cieľom je nájsť takú postupnosť akcií $a \in \mathbb{A}$ pre ktorú bude maximálne

$$y = \prod_{i=1} Q_i(s_i, a_i) \tag{2}$$

# Q-learning algoritmus - agent začlenený do prostredia



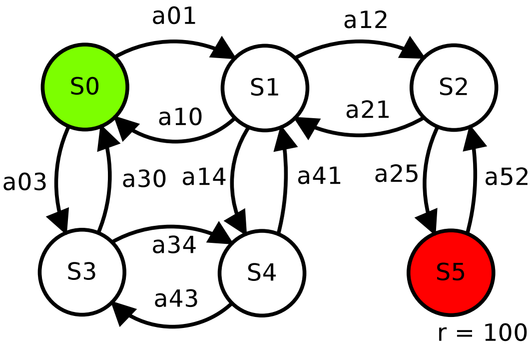
# Q-learning algoritmus

Daná je funkcia ohodnotení

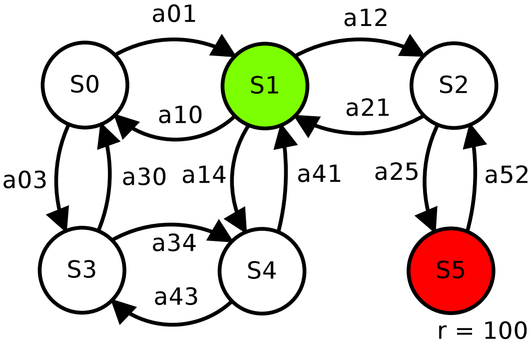
$$Q(s,a) = R(s,a) + \gamma \max_{a' \in \mathbb{A}} Q'(s',a') \tag{3}$$

kde
$R(s,a)$ je odmeňovacia funkcia,
$Q'(s',a')$ je odmeňovacia funkcia z ktorej sa agent dostal zo stavu "$s'$" vykonaním "$a$" do stavu "$s$",
$\gamma$ je odmeňovacia konštanta a platí $\gamma \in (0,1)$.

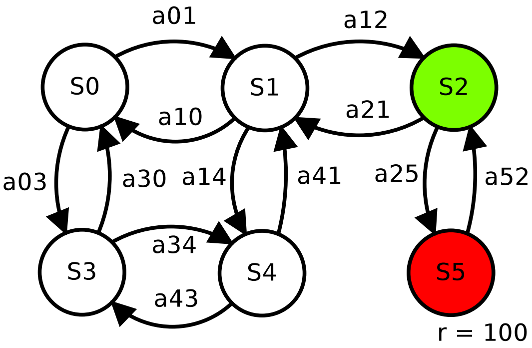
# Q-learning algoritmus



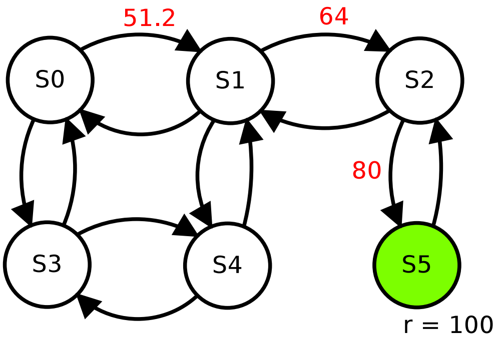
# Q-learning algoritmus



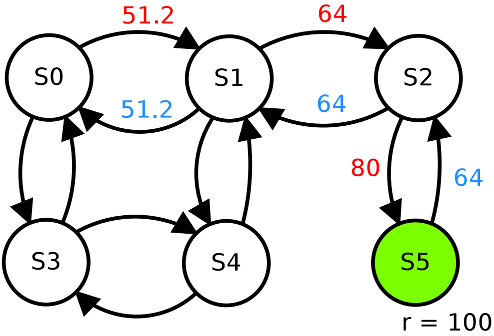
# Q-learning algoritmus



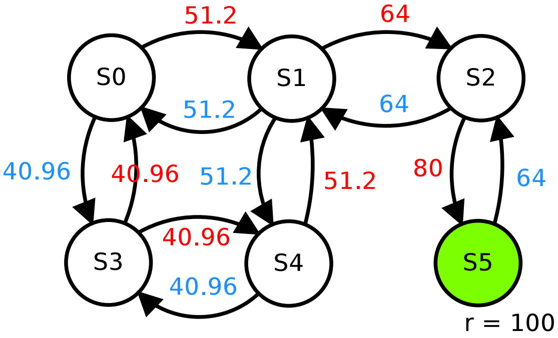
# Q-learning algoritmus



# Q-learning algoritmus



# Q-learning algoritmus



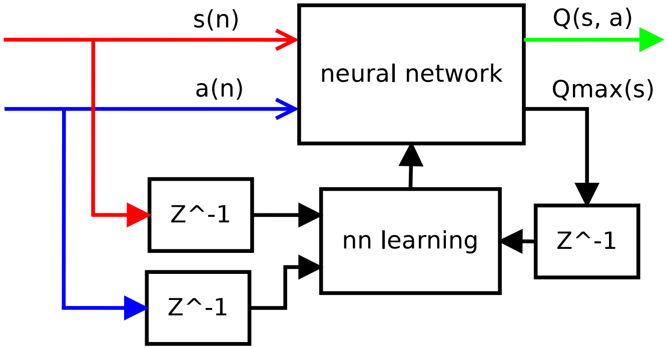
# Q-learning algoritmus

Problémy tabuľkovej interpretácie $Q(s, a)$

- pre veľké $N_s$ $alebo N_a$ narastajú pamäťové nároky
- o nevyplnených $Q(s, a)$ vieme povedať nič
- pre rozsiahle stavové priestory nevypočítateľné
- ako aproximovať $Q(s, a)$?

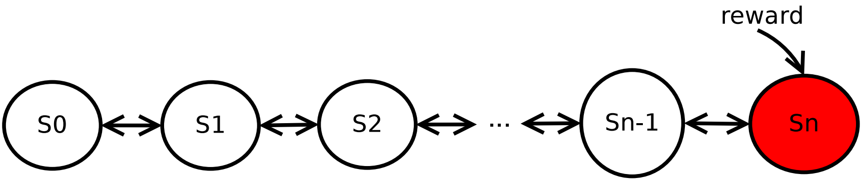
# Q-learning algoritmus - aproximácia

Neurónová sieť? Utopická predstava :




prečo nedáva správne výsledky?

# Q-learning algoritmus - aproximácia




Pre korektné vyplnenie hodnôt v $s_{n-1}$ sa vyžaduje korektá hodnota v $s_n$

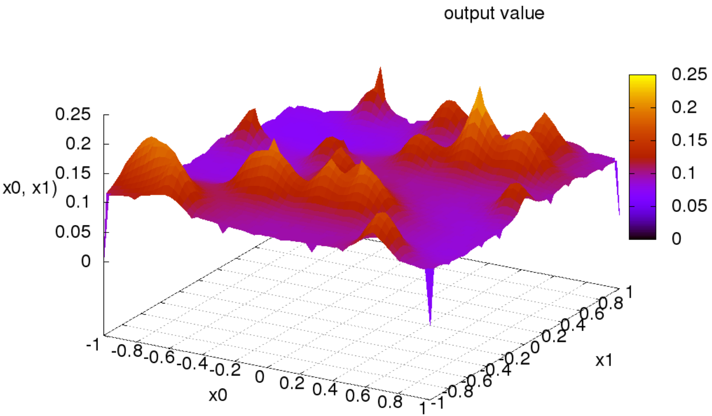
$$
\begin{aligned}
Q(s_0, a_0) &= R(s_0, a_0) + \gamma \max_{a'_1 \in \mathbb{A}} Q'(s_1, a'_1) \\
Q(s_1, a_1) &= R(s_1, a_1) + \gamma \max_{a'_2 \in \mathbb{A}} Q'(s_2, a'_2) \\
Q(s_2, a_2) &= R(s_2, a_2) + \gamma \max_{a'_3 \in \mathbb{A}} Q'(s_3, a'_3) \\
&\dots
\end{aligned} \tag{4}
$$

# Q-learning algoritmus - aproximácia

Učenie doprednej siete nie je homogénne!
- v priebehu učenia $Q(s, a)$ chaoticky osciluje okolo požadovanej hodnoty
- ani po 10-tkach milónoch iterácií sa hodnota neustáli na požadovanej hodnote

# Q-learning algoritmus - aproximácia

Je možné zostaviť neurónovú sieť ktorá sa dá učiť lokálne?



Rozklad na bázické funkcie

$$
\begin{aligned}
f_j^a(X) &= e^{\sum_{i=1}^{N_s} -b_{ji}(x_i - a_{ji})^2} \\
f_j^b(X) &= b_j \frac{1}{1 + \sum_{i=1}^{N_s} (x_i - a_{ji})^2} \\
f_j^c(X) &= e^{\sum_{i=1}^{N_s} -b_{ji}|x_i - a_{ji}|} \\
f_j^d(X) &= \sum_{i=1}^{N_s} b_{1ji}(x_i - a_{1ji})^1 + b_{2ji}(x_i - a_{2ji})^2 + \ldots
\end{aligned}
\tag{5}
$$

# Q-learning algoritmus - aproximácia

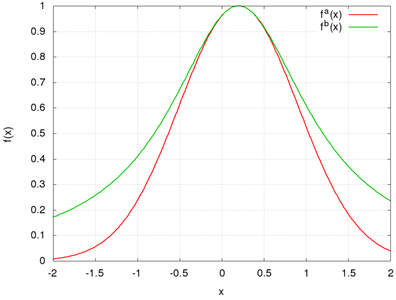
Ohliadnúc na charakter učiaceho algoritmu

$$Q(s,a) = R(s,a) + \gamma \max_{a' \in \mathbb{A}} Q'(s',a')$$

boli zvolené bázické funkcie

$$f_j^a(X) = e^{\sum_{i=1}^{N_s} -b_{ji}(x_i - a_{ji})^2}$$

$$f_j^b(X) = b_j \frac{1}{1 + \sum_{i=1}^{N_s} (x_i - a_{ji})^2}$$



# Q-learning algoritmus - aproximácia

Pre symetrické prechody medzi stavmi možno zjednodušiť na

$$f_j^a(X) = e^{-b_j \sum_{i=1}^{N_s} (x_i - a_{ji})^2}$$

$$f_j^b(X) = b_j \frac{1}{1 + \sum_{i=1}^{N_s} (x_i - a_{ji})^2}$$

A ich lineárna kombinácia

$$y(X) = \sum_{j=1}^{N} w_j f_j(X)$$

# Q-learning algoritmus - aproximácia

Stavenie parametrov :

- bázicke funkcie musia rovnomerne pokryť stavový priestor
- parameter $a_{ji}$ reprezentuje posunutie Gaussovej krivky - bod s najväčou funkčnou hodnotou.
- parameter $b_j$ reprezentuje strmosť krivky

# Q-learning algoritmus - aproximácia

Paramatre $a_{ji}$ - pokrytie stavového priestoru do oblastí podľa veľkosti $R(s,a)$ Využije sa princíp Kohonenovej siete - najbližšie vzory $a_j$ sa posunú podľa vstupných vektorov tak aby vrchol Gaussovej krivky ležal v ťazisku.

- na začiatku sa zvolia $a_{ji}$ náhodne
- spočítajú sa vzdialenosti od predloženého vstupu $d_j = \mid X - a_j \mid$
- nájde sa také $k$ kde $\forall j : d_k \leq d_j$
- spočíta sa krok učenia $\eta' = \eta_1 \mid y_r \mid$
- upravia sa parametre $a_{ki} = (1-\eta')a_{ki} + \eta' x_i$

kde
$X$ je vstupný vektor
$y_r$ je požadovaný výstup
$\eta_1$ je krok učenia

# Q-learning algoritmus - aproximácia

Paramatre $b_j$ - určuje strmosť krivky

- stanoví sa chyba $e(n) = y_r(n) - y(n)$
- pre každú bázickú funkciu $b_j(n+1) = b_j(n) - \eta_2 e(n) w_j(n)$
- skontroluje sa $b_j \in (0, -\infty)$

kde
$y_r$ je požadovaný výstup
$y$ je výstup
$\eta_2$ je krok učenia

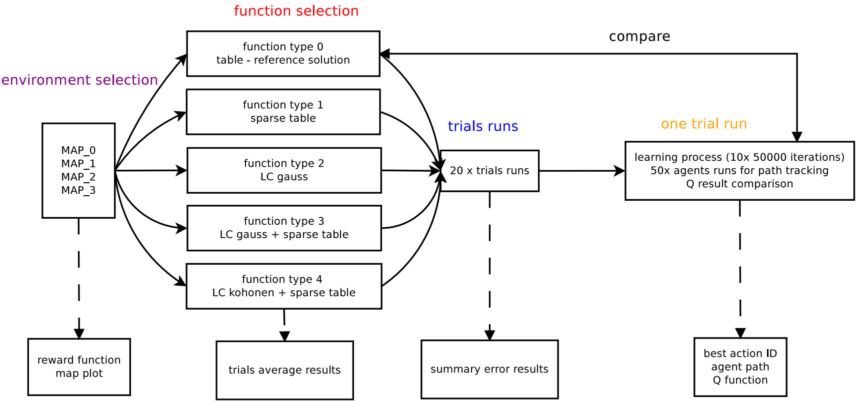
# Q-learning algoritmus - aproximácia

Paramatre $w_j$ - váhové parametre

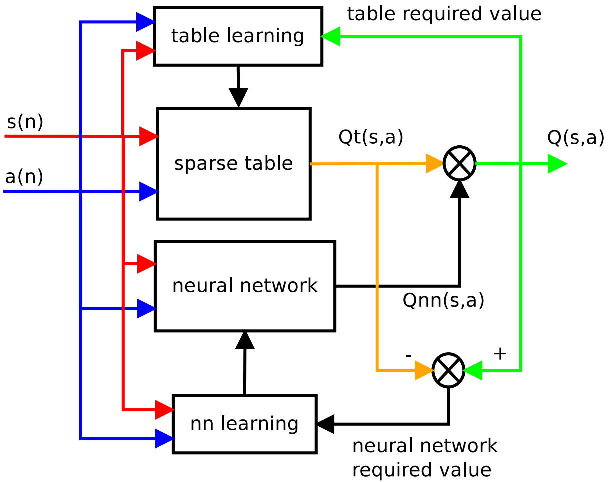
- stanoví sa chyba $e(n) = y_r(n) - y(n)$
- pre každé $w_j$ : $w_j(n+1) = w_j(n) - \eta_3 e(n) y_j(n)$
- skontroluje sa $w_j \in (-a, a)$

kde
$\eta_3$ je krok učenia
$a$ je maximálny rozsah váh

# Výsledky experimentov - schéma priebehu experimentov



# Výsledky experimentov - bloková schéma



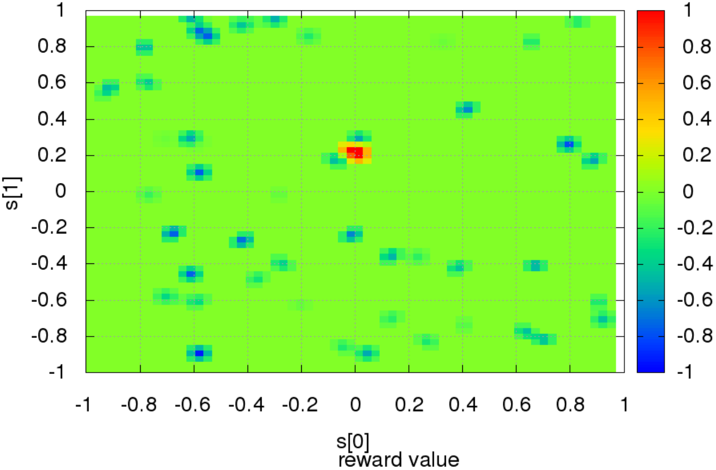
# Výsledky experimentov - podmienky

- 50000 iterácií učenia
- Q-learning

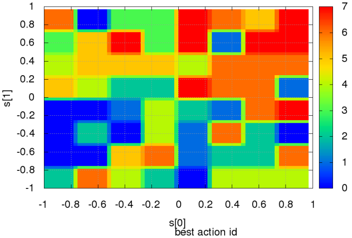
$$Q(s,a) = \alpha(Q'(s,a)) + (1-\alpha)(R(s,a) + \gamma \max_{a' \in \mathbb{A}} Q'(s',a'))$$

- $R(s,a) =$ náhodná mapa s 1 cieľovým stavom
- hodnoty $R(s,a) \in \langle -1, 1 \rangle$
- $\gamma = 0.98$ a $\alpha = 0.7$
- hustota referenčného riešenia $= 1/32$ (4096 stavov)
- počet akcií v každom stave $= 8$
- hustota riedkej tabuľky $= 1/8$ (1:16 pomer)
- počet bázických funkcií $= 64$
- rozsah parametrov
  - a_range = 1.0
  - b_range = 200.0
  - w_range = 10.0

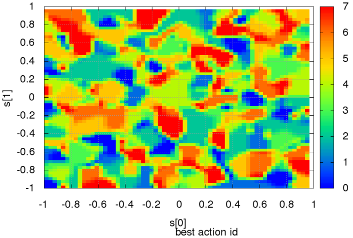
# Výsledky experimentov - funkcia R(s), mapa 1



# Výsledky experimentov - mapa najlepších akcií

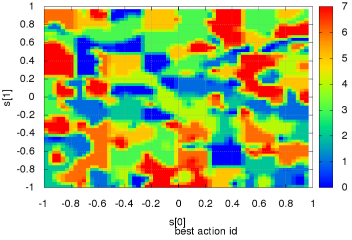



Obr. : sparse table

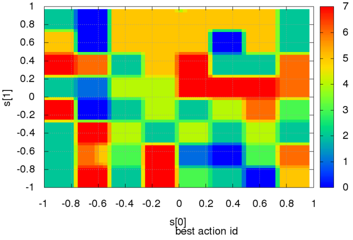



Obr. : linear combination Gauss

# Výsledky experimentov - mapa najlepších akcií

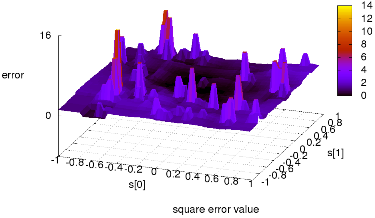



Obr. : sparse table + linear combination Gauss

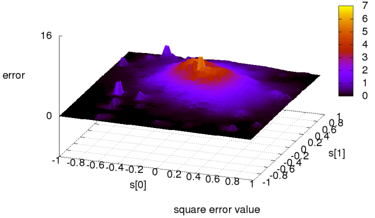



Obr. : sparse table + linear combination Kohonen function

# Výsledky experimentov - chybové funkcie

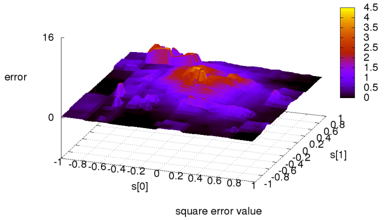



Obr. : sparse table

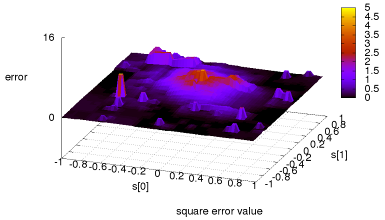



Obr. : linear combination Gauss

# Výsledky experimentov - chybové funkcie

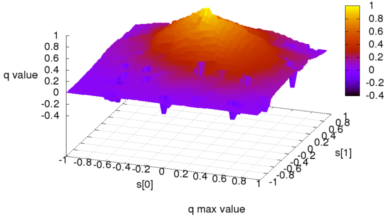



Obr. : sparse table + linear combination Gauss

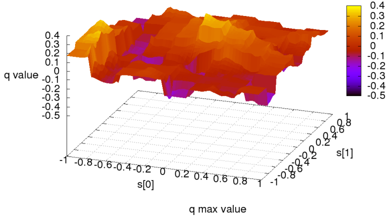



Obr. : sparse table + linear combination Kohonen function

# Výsledky experimentov - max Q(s, a)

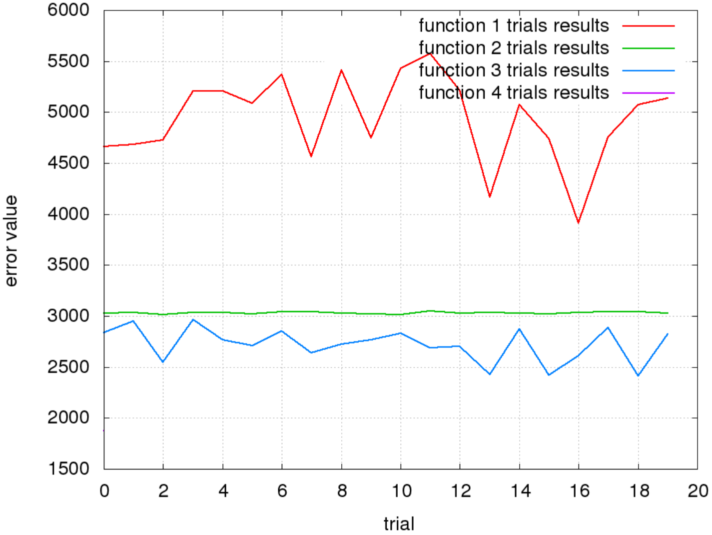



Obr. : reference table

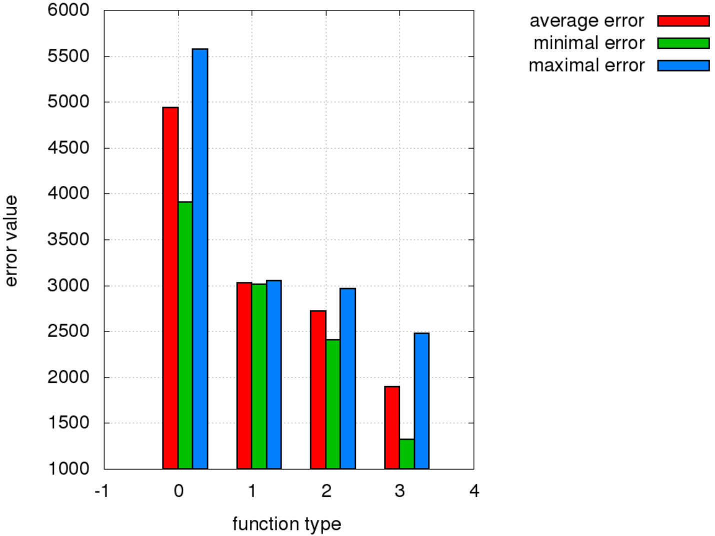



Obr. : sparse table + linear combination Gauss

# Výsledky experimentov - trials progress



# Výsledky experimentov - trials average



# Ďakujem za pozornosť

michal.chovanec@yandex.ru
https://github.com/michalnand/q_learning