

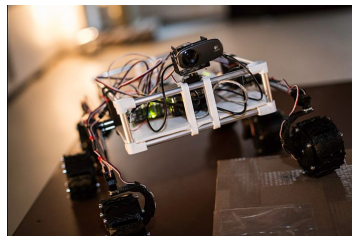
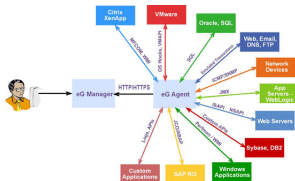
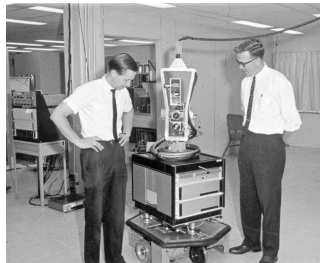
# Aproximácia funkcie ohodnotení v algoritmoch Q-learning

Ing. Michal CHOVANEC  
Fakulta riadenia a informatiky

*Marec 2016*

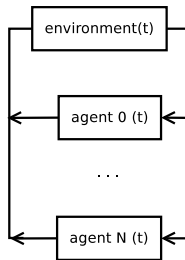
- Úvod
  - Agentové systémy
  - Adaptívne a učiace sa systémy
- Q-learning algoritmus
- Možnosti aproximácie
- Výsledky experimentov

# 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

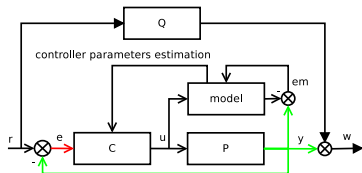
- reaktívne správanie
- dôraz na čas
- nemá pamäť - bez očakávaní
- rýchla dynamika

## Učiaci sa systém

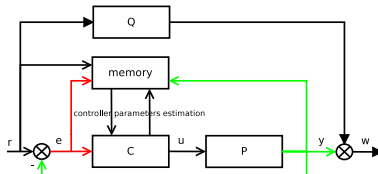
- konštruktívne správanie
- dôraz na priestor
- má pamäť - očakávania
- pomalá dynamika

# Adaptívne a učiace sa systémy

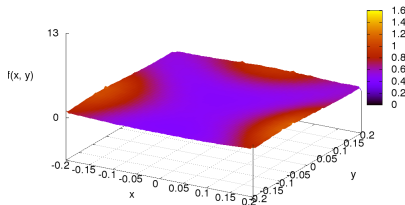
## Adaptívny systém



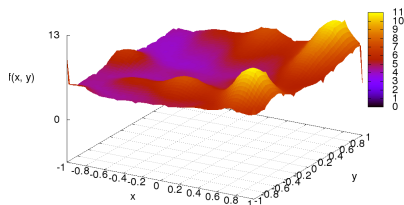
## Učiaci sa systém



output value



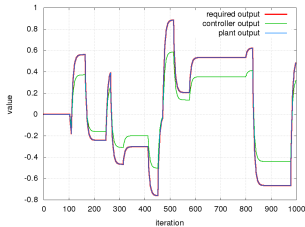
output value



# Adaptívne a učiace sa systémy

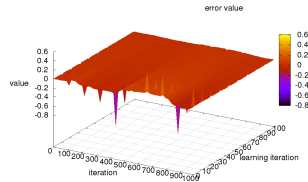
## Adaptívny systém PID regulátor

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



## 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 prostredí existuje funkcia

$$s(n+1) = \lambda(s(n), a(n)) \quad (1)$$

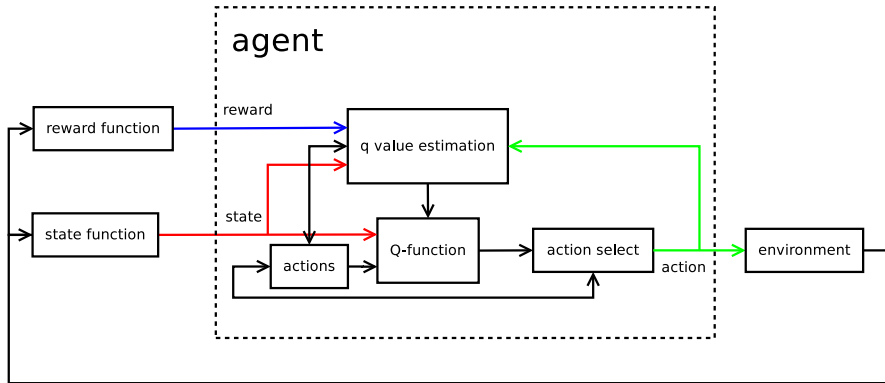
prechodová funkcia zo stavu  $s(n)$  použitím akcie  $a(n)$  - táto funkcia je ale agentovi neznáma.

Cieľom je nájsť takú postupnosť akcií  $a \in \mathbb{A}$  pre ktorú bude maximálne

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



# Q-learning algoritmus - agent začlenený do prostredia



Daná je funkcia ohodnotení

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

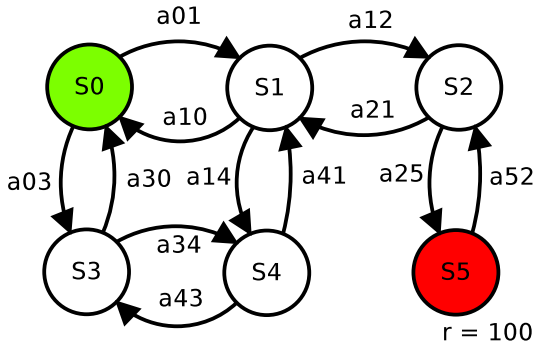
kde

$R(s, a)$  je odmeňovacia funkcia,

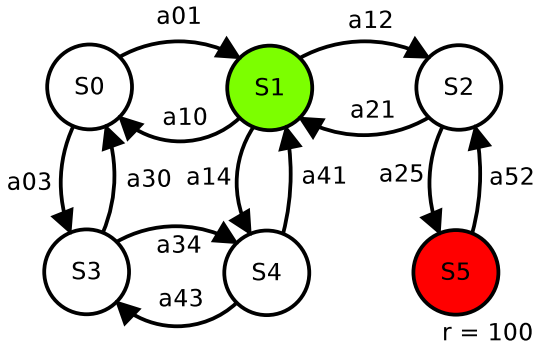
$Q'(s', a')$  je odmeňovacia funkcia z ktorej sa agent dostal zo stavu " $s'$ " vykonaním " $a$ " do stavu " $s$ ",

$\gamma$  je odmeňovacia konštanta a platí  $\gamma \in (0, 1)$ .

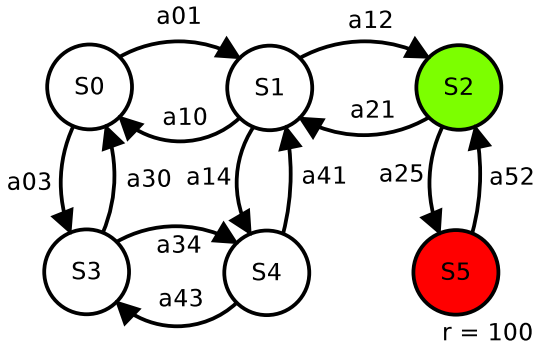
# Q-learning algoritmus



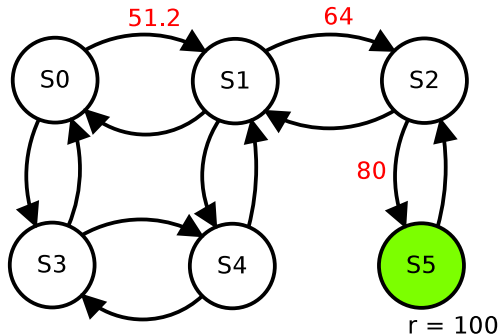
# Q-learning algoritmus



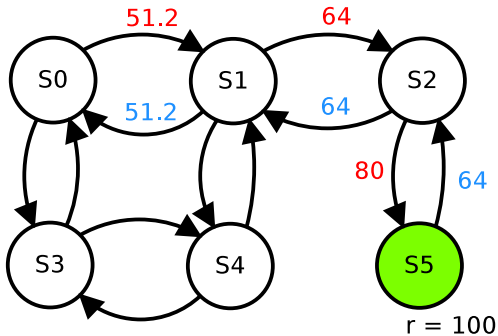
# Q-learning algoritmus



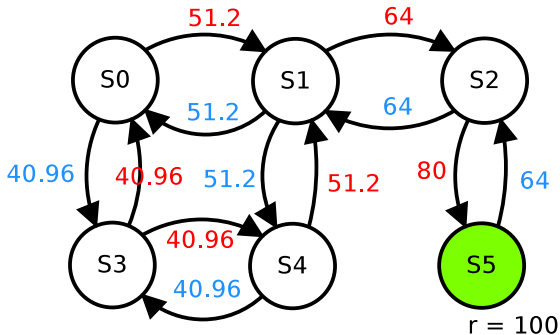
# Q-learning algoritmus



# Q-learning algoritmus



# Q-learning algoritmus



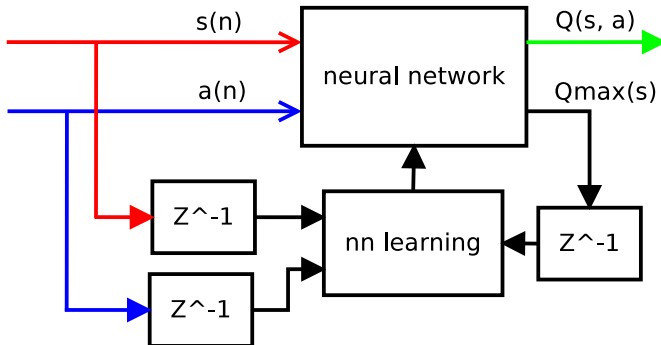


## Problémy tabuľkovej interpretácie $Q(s, a)$

- 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  $Q(s, a)$ ?

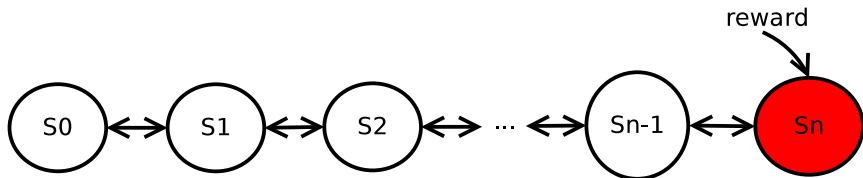
# Q-learning algoritmus - aproximácia

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  $s_n$

$$Q(s_0, a_0) = R(s_0, a_0) + \gamma \max_{a'_1 \in \mathbb{A}} Q'(s_1, a'_1)$$

$$Q(s_1, a_1) = R(s_1, a_1) + \gamma \max_{a'_2 \in \mathbb{A}} Q'(s_2, a'_2)$$

$$Q(s_2, a_2) = R(s_2, a_2) + \gamma \max_{a'_3 \in \mathbb{A}} Q'(s_3, a'_3)$$

...

(4)

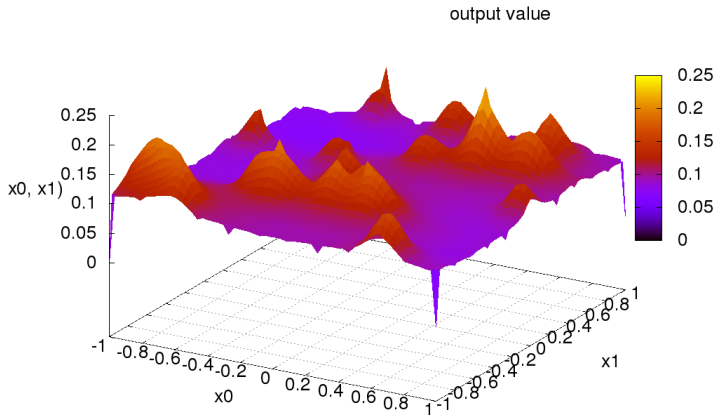
# Q-learning algoritmus - aproximácia

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?



# Q-learning algoritmus - aproximácia

Rozklad na bázické funkcie

$$f_j^a(X) = \sum_{i=1}^{N_s} b_{1ji}(x_i - a_{1ji})^1 + b_{2ji}(x_i - a_{2ji})^2 + \dots \quad (5)$$

$$f_j^a(X) = \sum_{i=1}^{N_s} \max(0, b_{ji}(x_i - a_{ji})) \quad (6)$$

$$f_j^b(X) = \cos\left(\sum_{i=1}^{N_s} b_{ji}(x_i - a_{ji})\right) \quad (7)$$

$$f_j^c(X) = e^{\sum_{i=1}^{N_s} -b_{ji}|x_i - a_{ji}|} \quad (8)$$

$$f_j^d(X) = e^{\sum_{i=1}^{N_s} -b_{ji}(x_i - a_{ji})^2} \quad (9)$$

# Q-learning algoritmus - aproximácia

Ohliadnúc na charakter učiaceho algoritmu

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

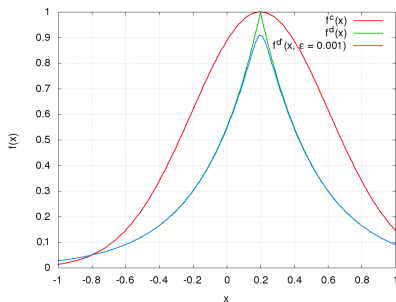
boli zvolené bážické funkcie

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

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

kde

$$|x| = \lim_{\epsilon \rightarrow 0} \sqrt[2]{x^2 + \epsilon}$$



# Q-learning algoritmus - aproximácia

Rozklad na bázické funkcie

$$f_j(X) = e^{\sum_{i=1}^{N_s} -b_{ji}(x_i - a_{ji})^2} \quad (10)$$

pre symetrické prechody medzi stavmi možno zjednodušiť na

$$f_j(X) = e^{-b_j \sum_{i=1}^{N_s} (x_i - a_{ji})^2} \quad (11)$$

A ich kombinácia

$$\begin{aligned} y(X) &= \sum_{j=1}^N w_j f_j(X) \\ &+ \sum_{j=1}^N \sum_{i=1}^j v_{ji} f_j(X) f_i(X) \end{aligned} \quad (12)$$



Stavenie parametrov :

- bázicke funkcie musia rovnomerne pokryť stavový priestor
- parameter  $a_j$  reprezentuje posunutie Gaussovej krivky - bod s najväčšou funkčnou hodnotou.
- parameter  $b_j$  reprezentuje strmosť krivky

# Q-learning algoritmus - aproximácia

Parametre  $a_{ji}$  - 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  $d_j = |X - a_j|$
- nájde sa také  $k$  kde  $\forall j : d_k \leq d_j$
- spočíta sa krok učenia  $\eta' = \eta_1 |y_r|$
- 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

Parametre  $b_j$  - určuje strmosť krivky

- stanoví sa chyba  $e(n) = y_r(n) - y(n)$
- pre každú bázičku 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

Parametre  $w_j$  - váhové parametre

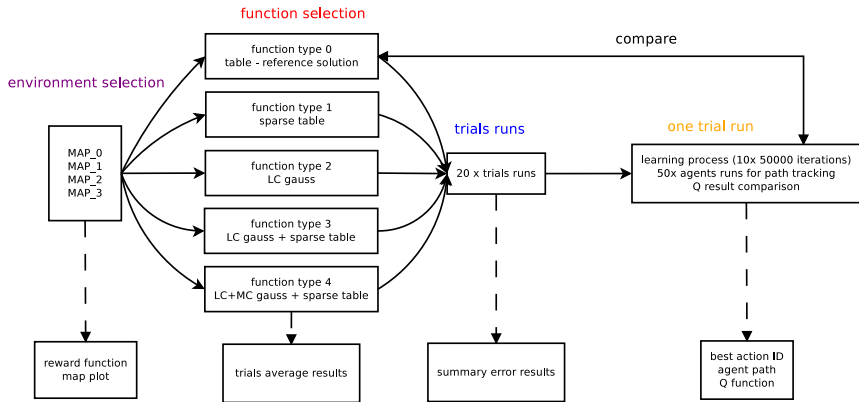
- stanoví sa chyba  $e(n) = y_r(n) - y(n)$
- pre každé  $w_j$  :  $w_j(n+1) = w_j(n) - \eta_3 e(n) y_j(n)$
- skontroluje sa  $w_j \in (-a, a)$

kde

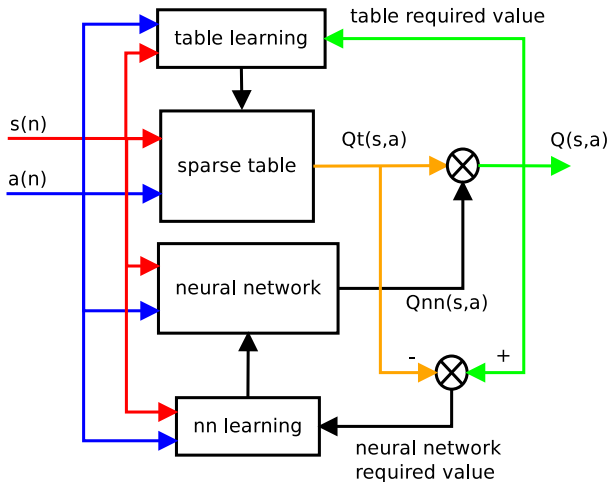
$\eta_3$  je krok učenia

$a$  je maximálny rozsah váh

# Výsledky experimentov - schéma priebehu experimentov



# Výsledky experimentov - bloková schéma



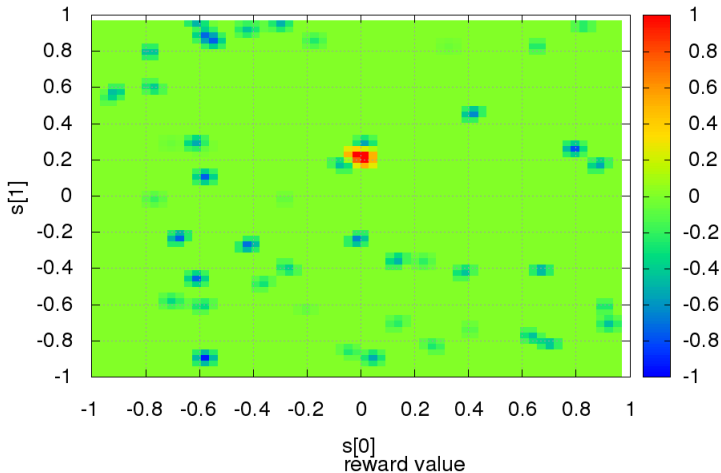
# Výsledky experimentov - podmienky

- 50000 iterácií učenia
- Q-learning

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

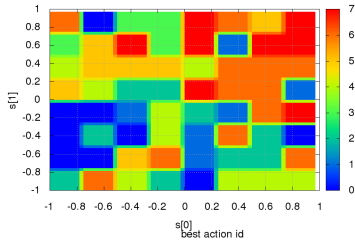
- $R(s, a)$  = náhodná mapa s 1 cieľovým stavom
- hodnoty  $R(s, a) \in \langle -1, 1 \rangle$
- $\gamma = 0.98$  a  $\alpha = 0.7$
- hustota referenčného riešenia =  $1/32$  (4096 stavov)
- počet akcií v každom stave = 8
- hustota riedkej tabuľky =  $1/8$  (1:16 pomer)
- počet bazických funkcií = 64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

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

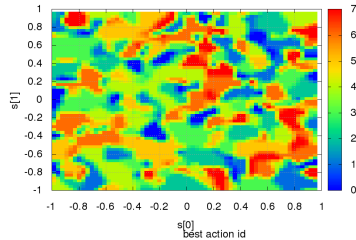




# Výsledky experimentov - mapa najlepších akcií

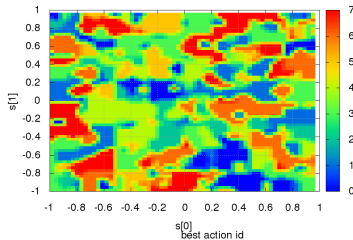


Obr. : sparse table

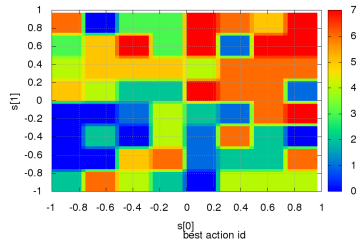


Obr. : linear combination Gauss

# Výsledky experimentov - mapa najlepších akcií

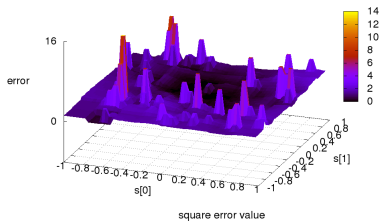


Obr. : sparse table + linear combination Gauss

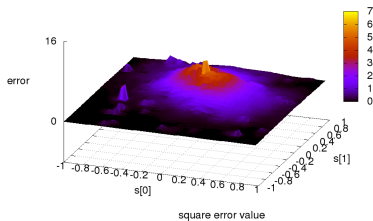


Obr. : sparse table + linear combination + multiplication Gauss

# Výsledky experimentov - chybové funkcie

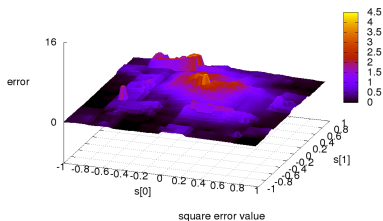


Obr. : sparse table

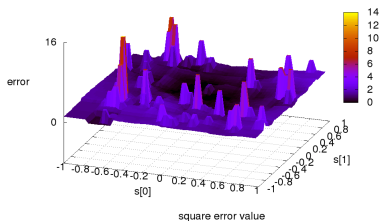


Obr. : linear combination Gauss

# Výsledky experimentov - chybové funkcie

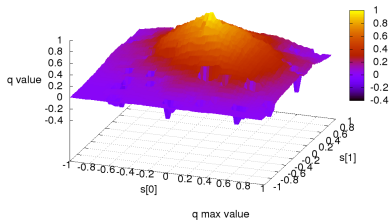


Obr. : sparse table + linear combination Gauss

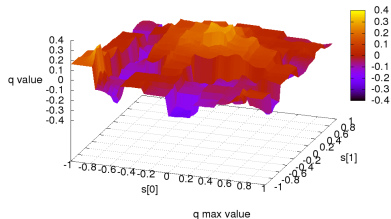


Obr. : sparse table + linear combination + multiplication Gauss

# Výsledky experimentov - max $Q(s, a)$

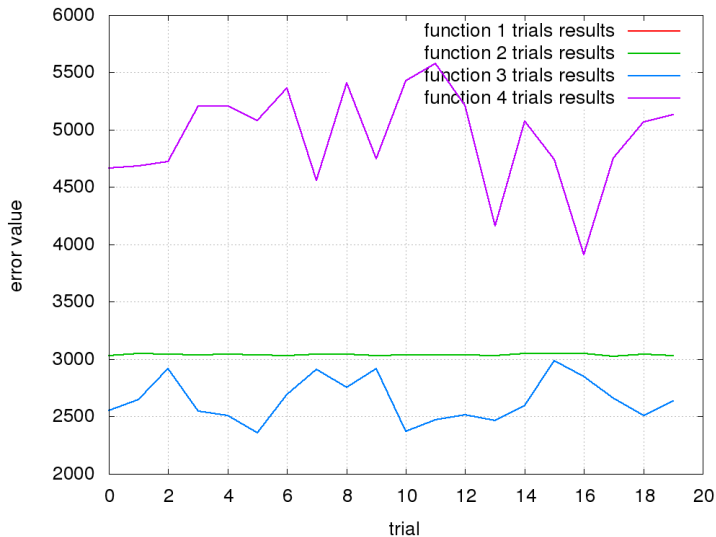


Obr. : reference table

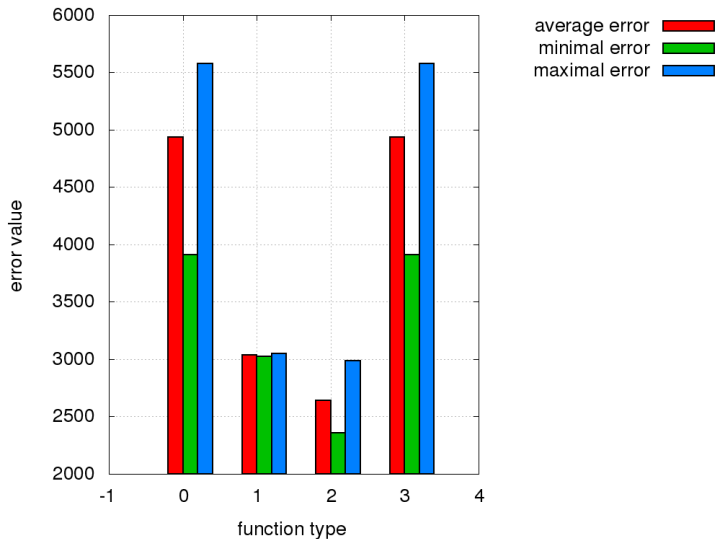


Obr. : sparse table + linear combination Gauss

# Výsledky experimentov - trials progress



# Výsledky experimentov - trials average



# Ďakujem za pozornosť



[michal.chovanec@yandex.ru](mailto:michal.chovanec@yandex.ru)