Empirical Analysis of Q(sigma)

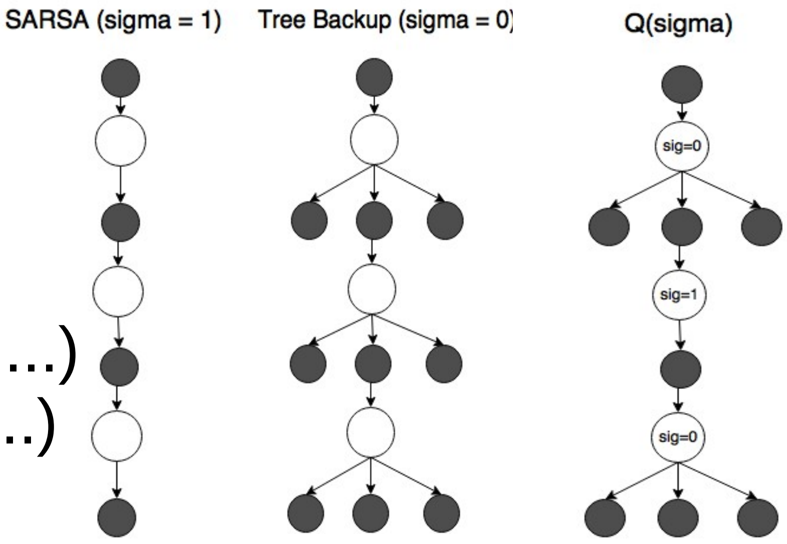
COMP 767 – Reinforcement Learning
February 24th

Code: https://github.com/NicolasAG/Q-sigma

~Nicolas Angelard-Gontier - 260532513

Algorithms

- All **OnPolicy**: bcs reduce variance & not expensive to sample from target policy
  - n-step **SARSA** (sigma = 1)
  - n-step **Tree Backup** (sigma = 0)
  - n-step **Q(sigma)**
    - alternating sigma (0,1,0,1,0,1,...)
    - decreasing sigma (1,1,...,1,0,1,0,...,0,0,...)
    - increasing sigma (0,0,...,0,1,0,1,...,1,1,...)



- number of episodes: 1,000 – repeat 10 times and take the average.
- no environment stochasticity
- gamma = 0.99

Q(sigma) Variations

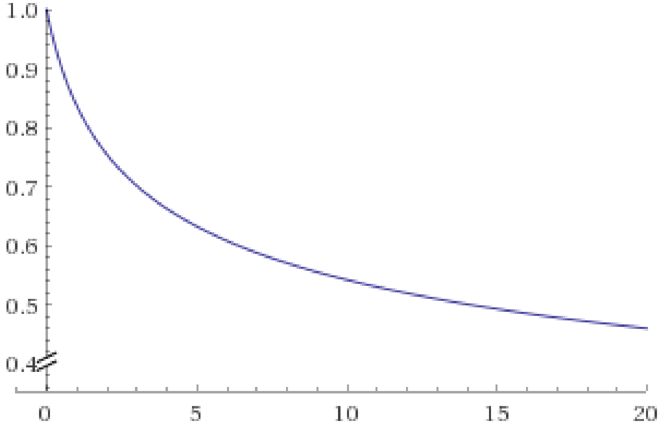
- alternating sigma (0,1,0,1,0,1,...)

```
    return 1 – sigmas[-1]
```

- decreasing sigma (1,1,1,...,1,0,1,0,...,0,0,0,...)

$$sample\ proba = e^{-\log_{base}(1+t)}$$

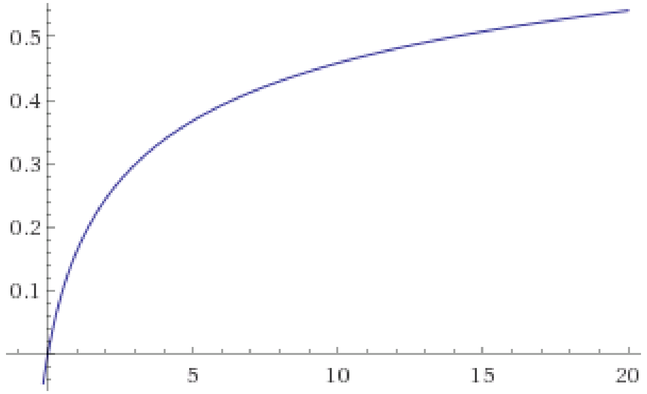
```
return 1 with proba sample_proba else 0
```



- increasing sigma (0,0,0,...,0,1,0,1,...,1,1,1,...)

$$sample\ proba = 1 - e^{-\log_{base}(1+t)}$$

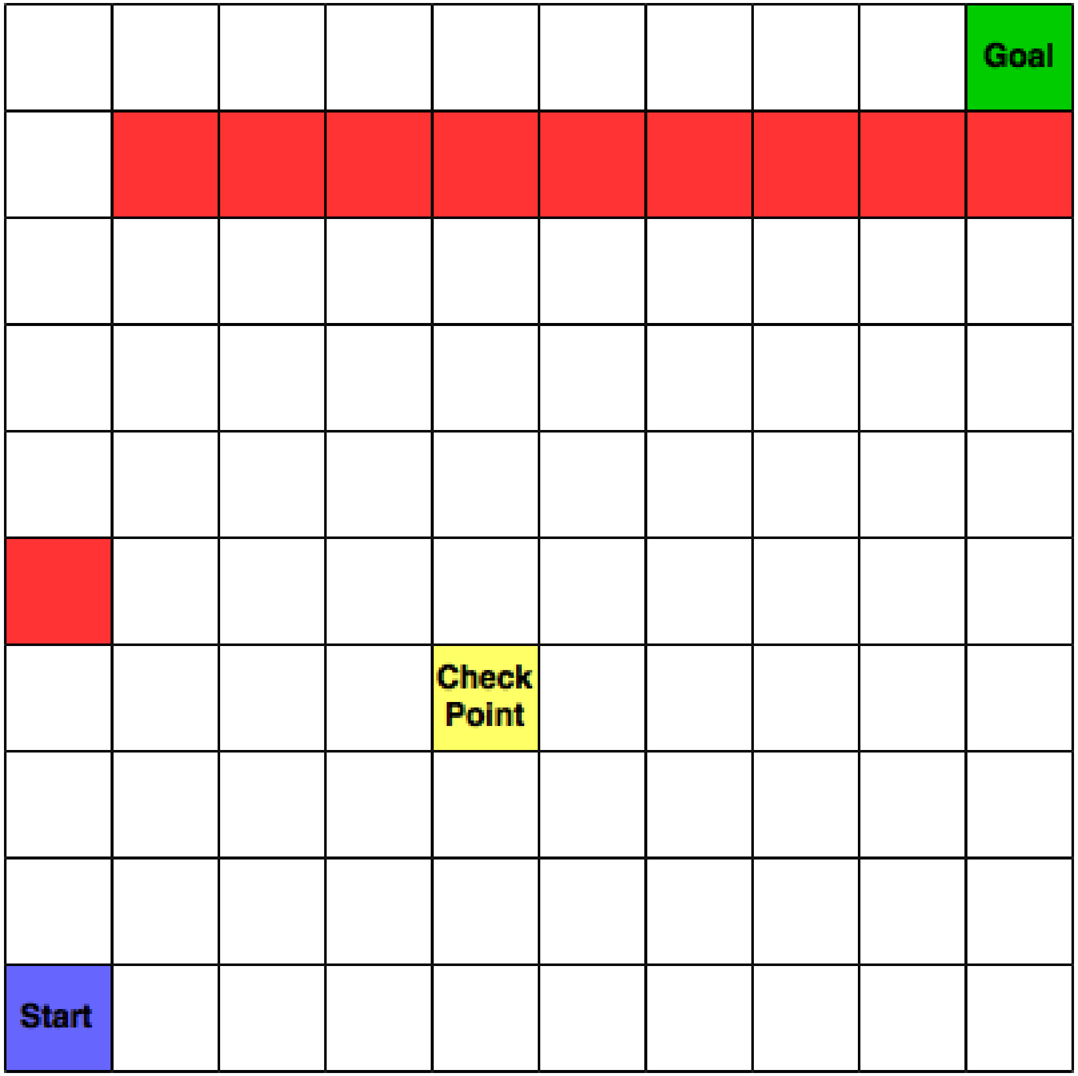
```
return 1 with proba sample_proba else 0
```



With base=50, p~0.5 at t=15

Extra parameter: log base!
(used 50 in these experimennts)

Grid World



Rewards:
STEP = -1
WALL = -10
CHECK POINT = +0
GOAL = +1,000

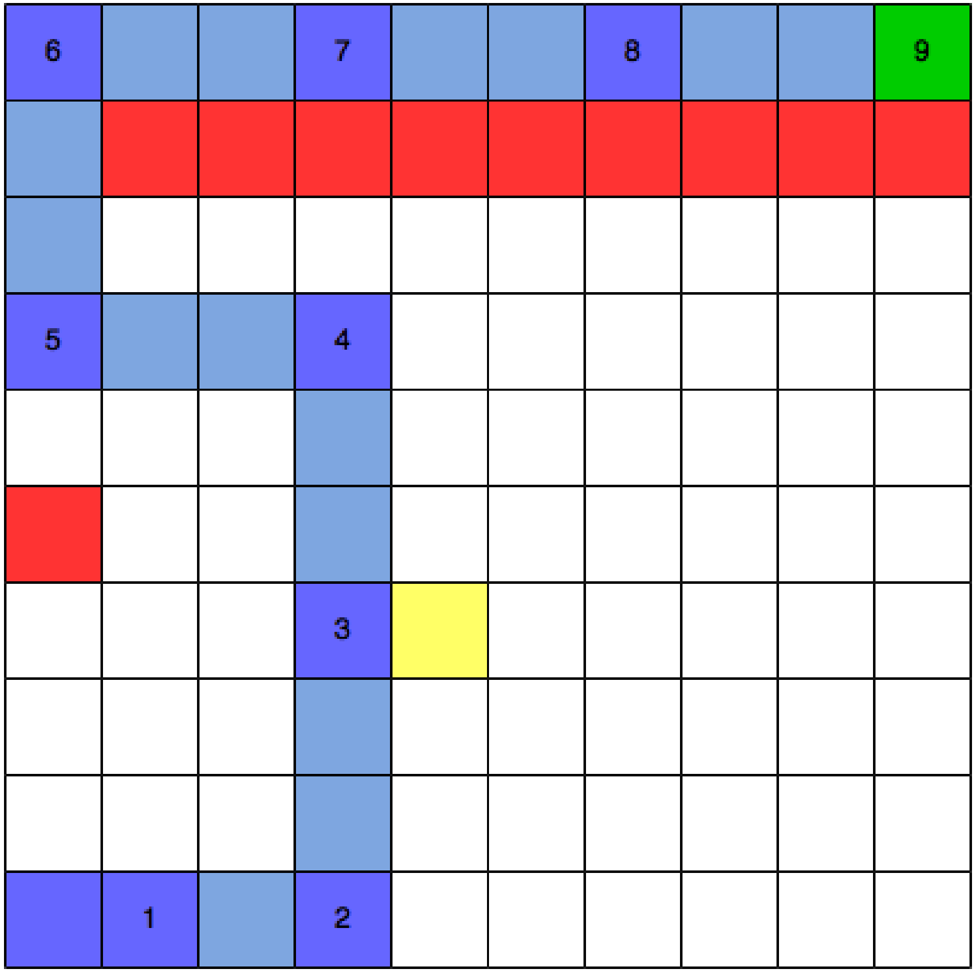
Actions: V in [0, V_MAX]

	V - 1	V + 0	V + 1
RIGHT	0	1	2
UP	3	4	5
LEFT	6	7	8

Crash:
Return to Start & V=0 & V_MAX=3

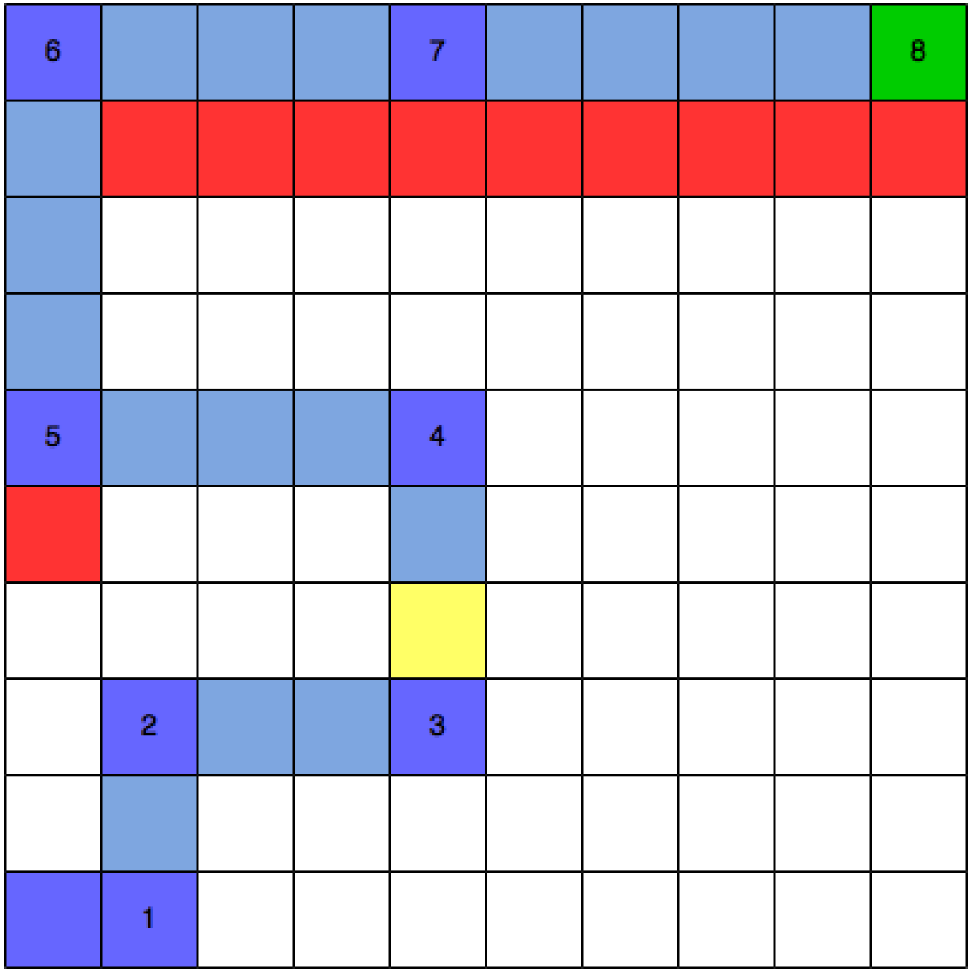
CheckPoint:
V_MAX = 5

Grid World



No Bonus: **V_MAX = 3**
Minumum nuber of steps: **9**

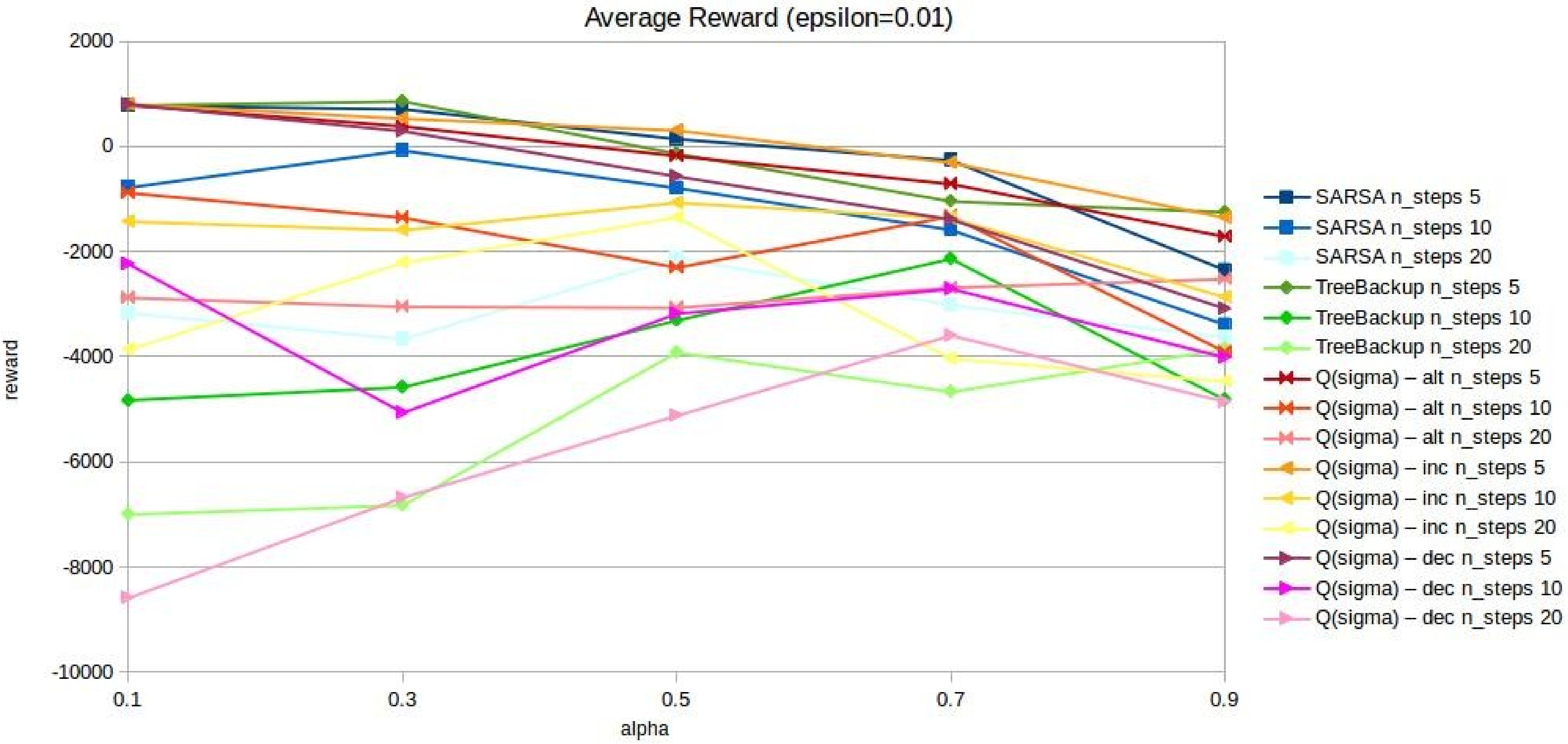
~ALWAYS



With Bonus: **V_MAX = 5**
Minumum nuber of steps: **8**

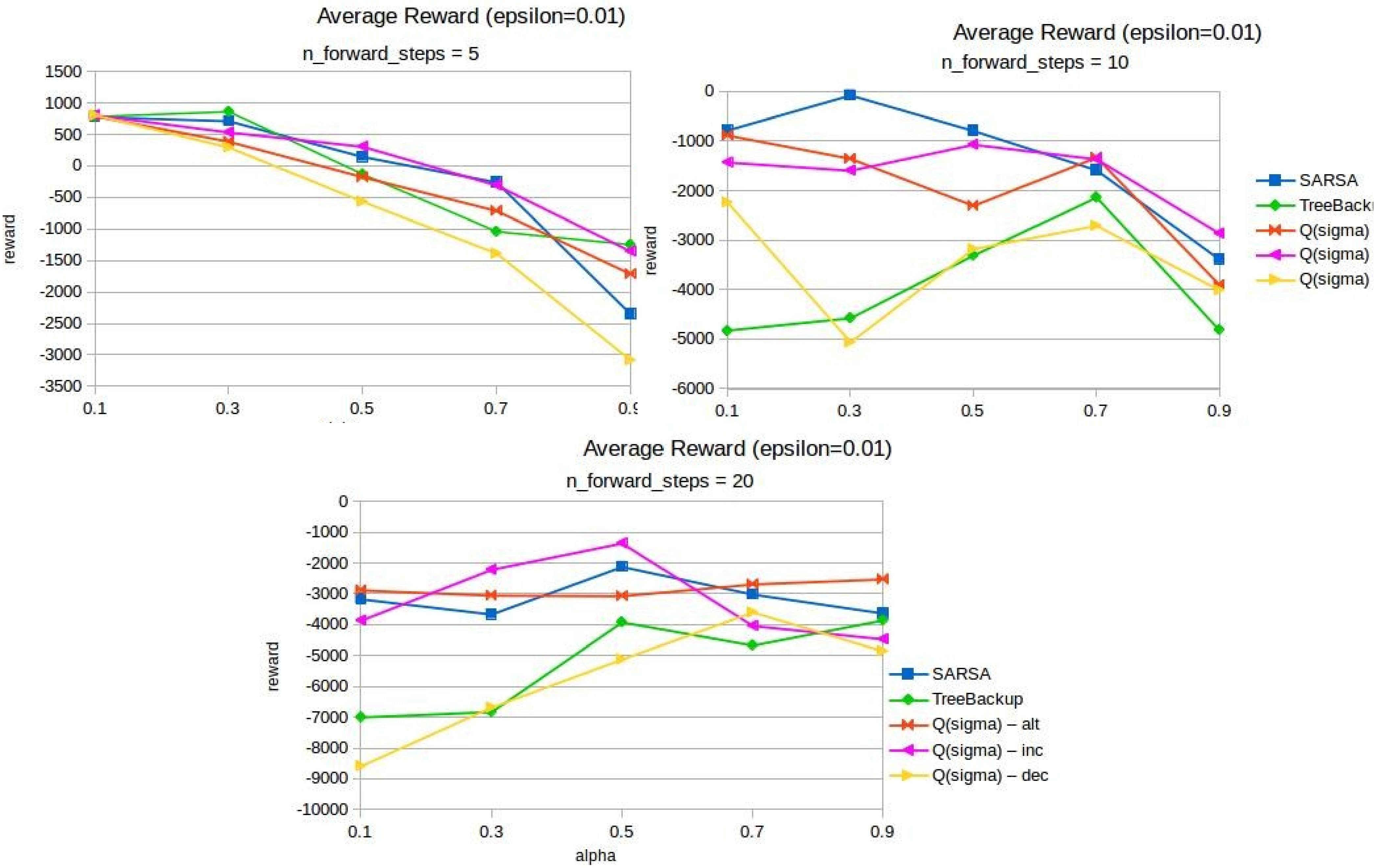
~NEVER :(

Results
(epsilon=0.01)



SARSA, TreeBackup, Qsigma–alt, Qsigma–inc, Qsigma–dec: for n_steps = 5, 10, 20

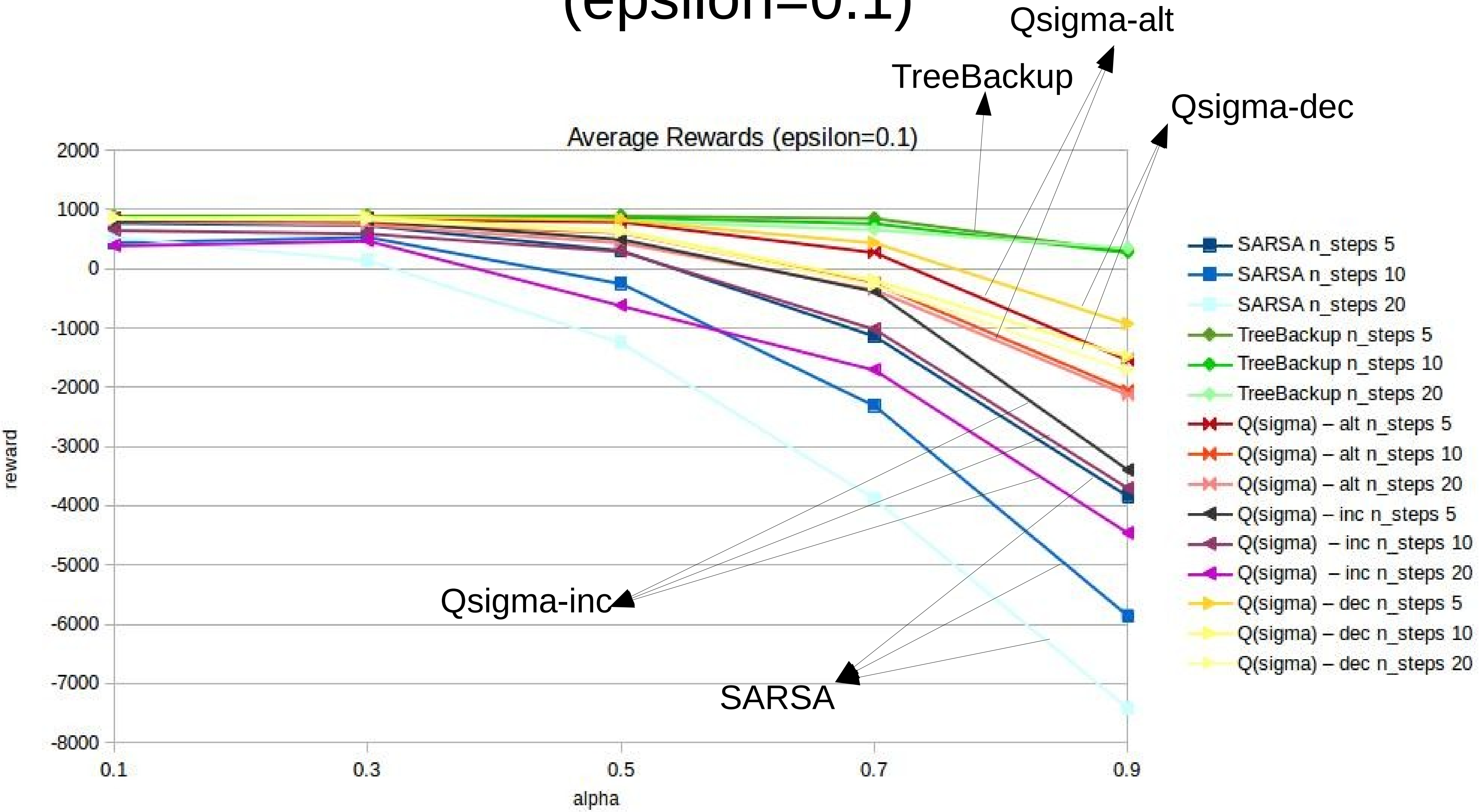
Results (epsilon=0.01)



SARSA, TreeBackup, Qsigma–alt, Qsigma–inc, Qsigma–dec: for n_steps = 5, 10, 20

Results

(epsilon=0.1)



Qsigma-alt

TreeBackup

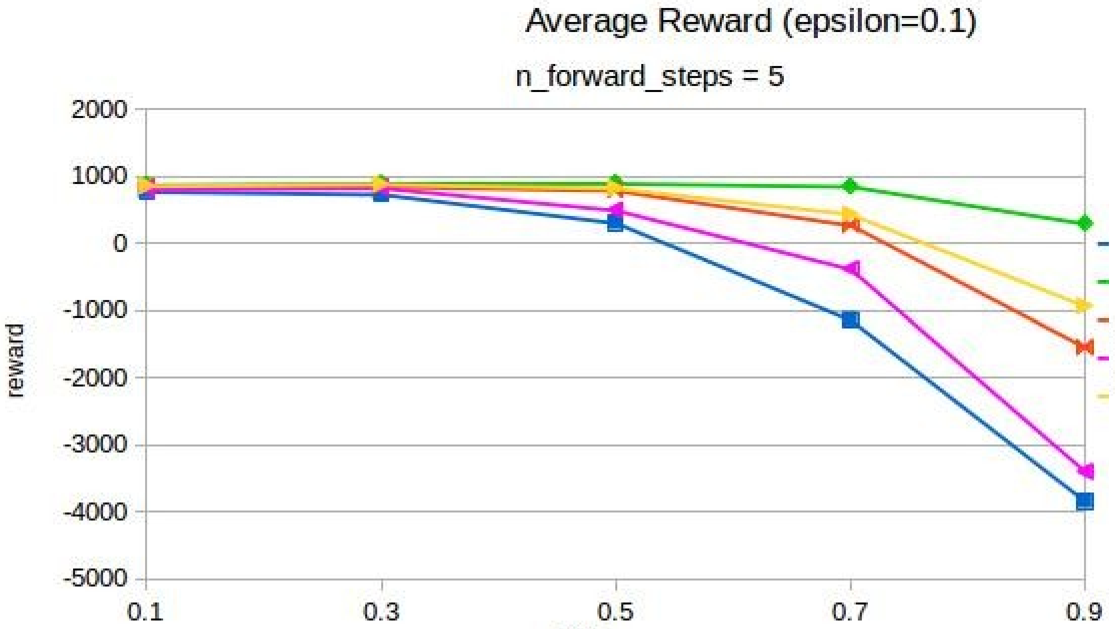
Qsigma-dec

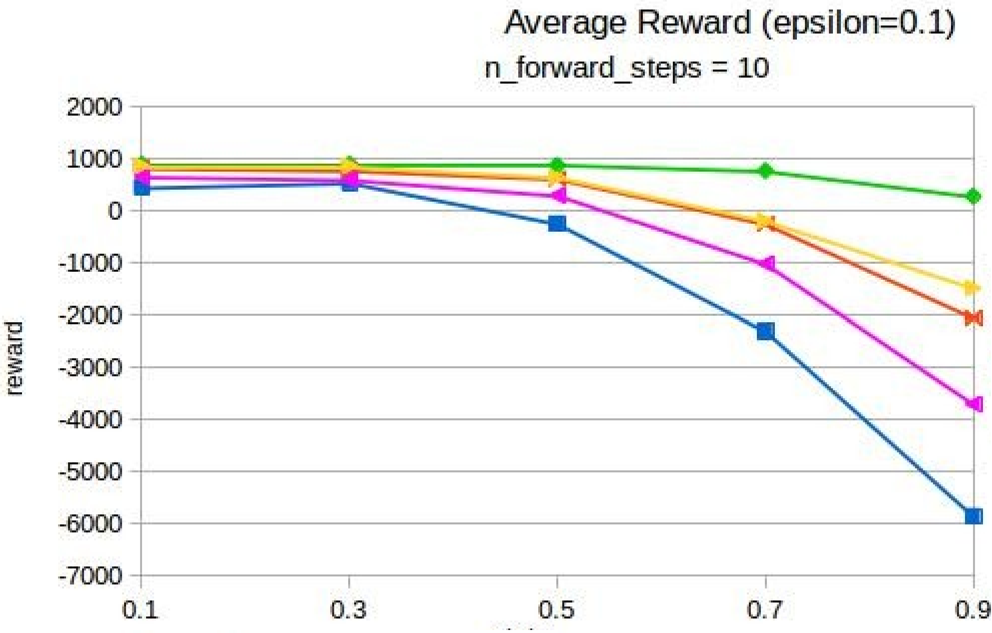
Qsigma-inc

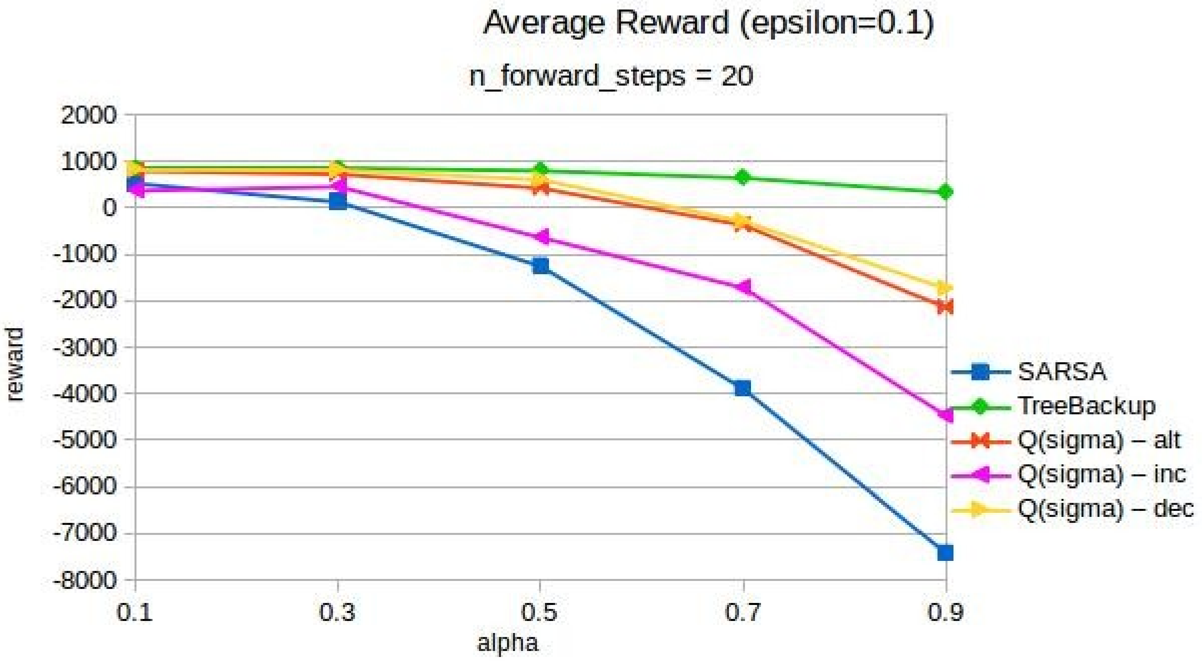
SARSA

SARSA, TreeBackup, Qsigma–alt, Qsigma–inc, Qsigma–dec: for n_steps = 5, 10, 20

Results (epsilon=0.1)



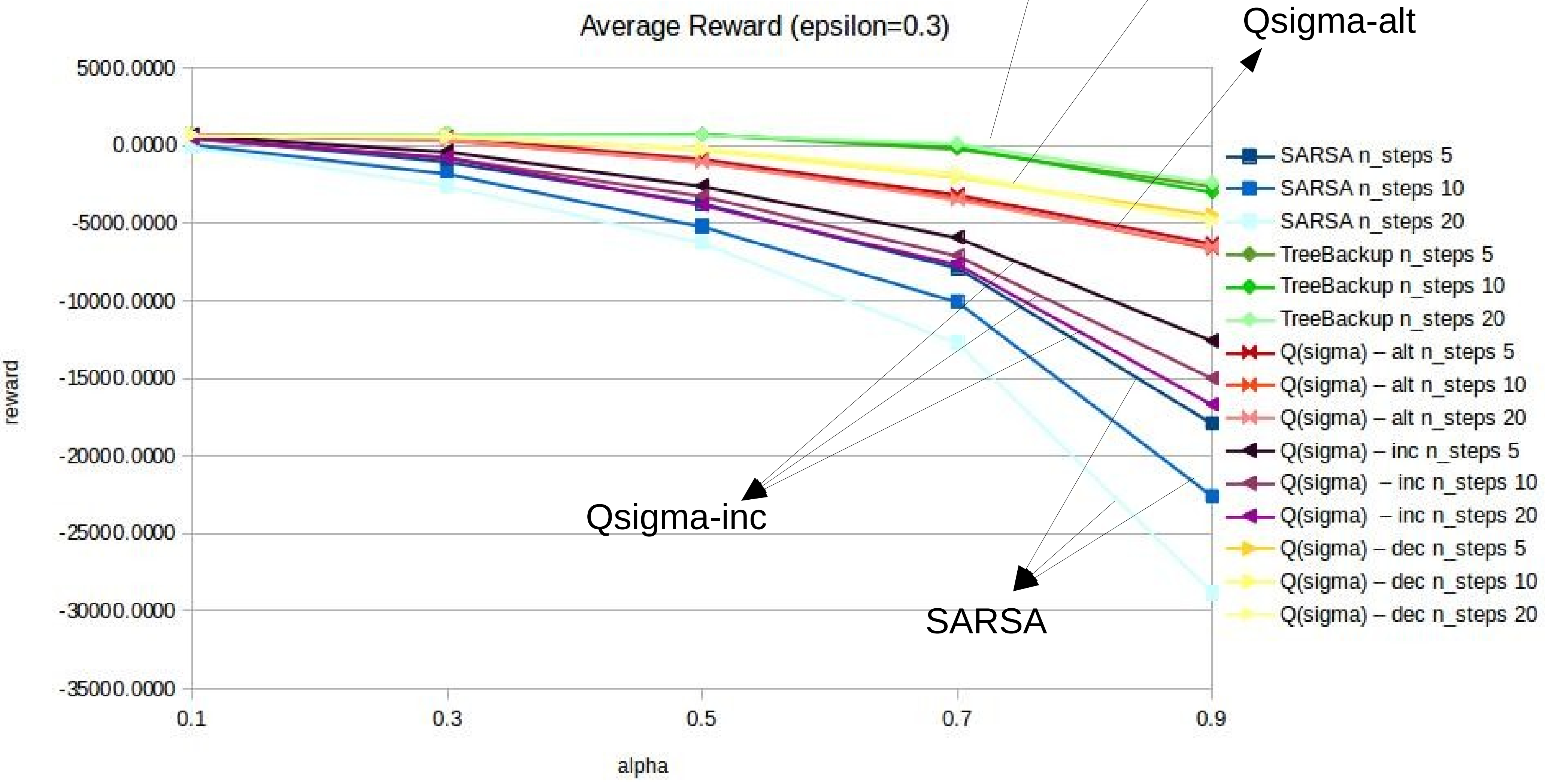




SARSA, TreeBackup, Qsigma–alt, Qsigma–inc, Qsigma–dec: for n_steps = 5, 10, 20

Results

(epsilon=0.3)



Qsigma-dec

TreeBackup

Qsigma-alt

Qsigma-inc

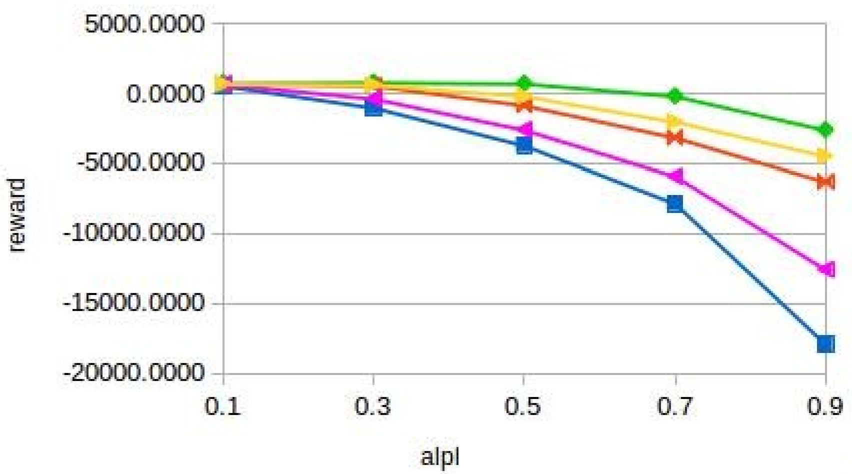
SARSA

SARSA, TreeBackup, Qsigma–alt, Qsigma–inc, Qsigma–dec: for n_steps = 5, 10, 20

Results (epsilon=0.3)

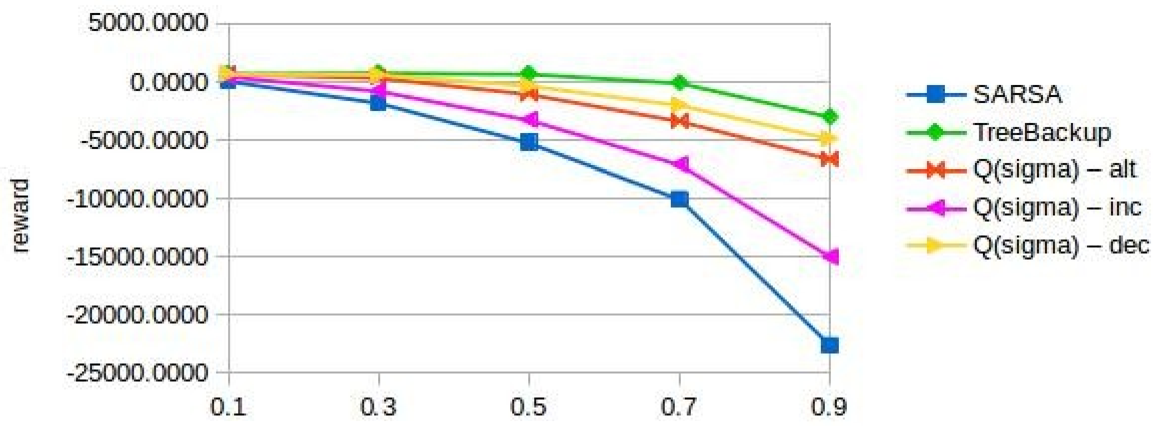
Average Reward (epsilon=0.3)

n_forward_steps = 5



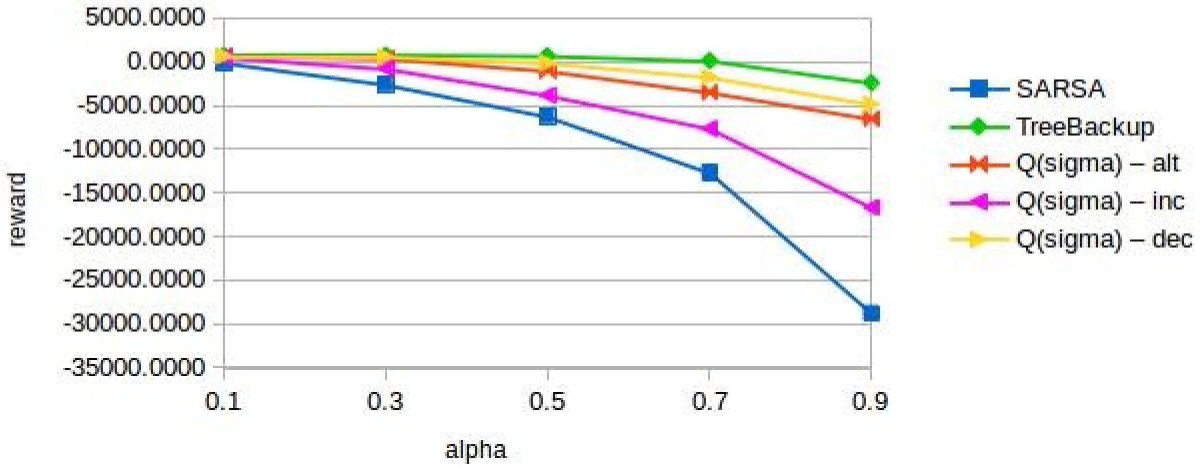
Average Reward (epsilon=0.3)

n_forward_steps = 10



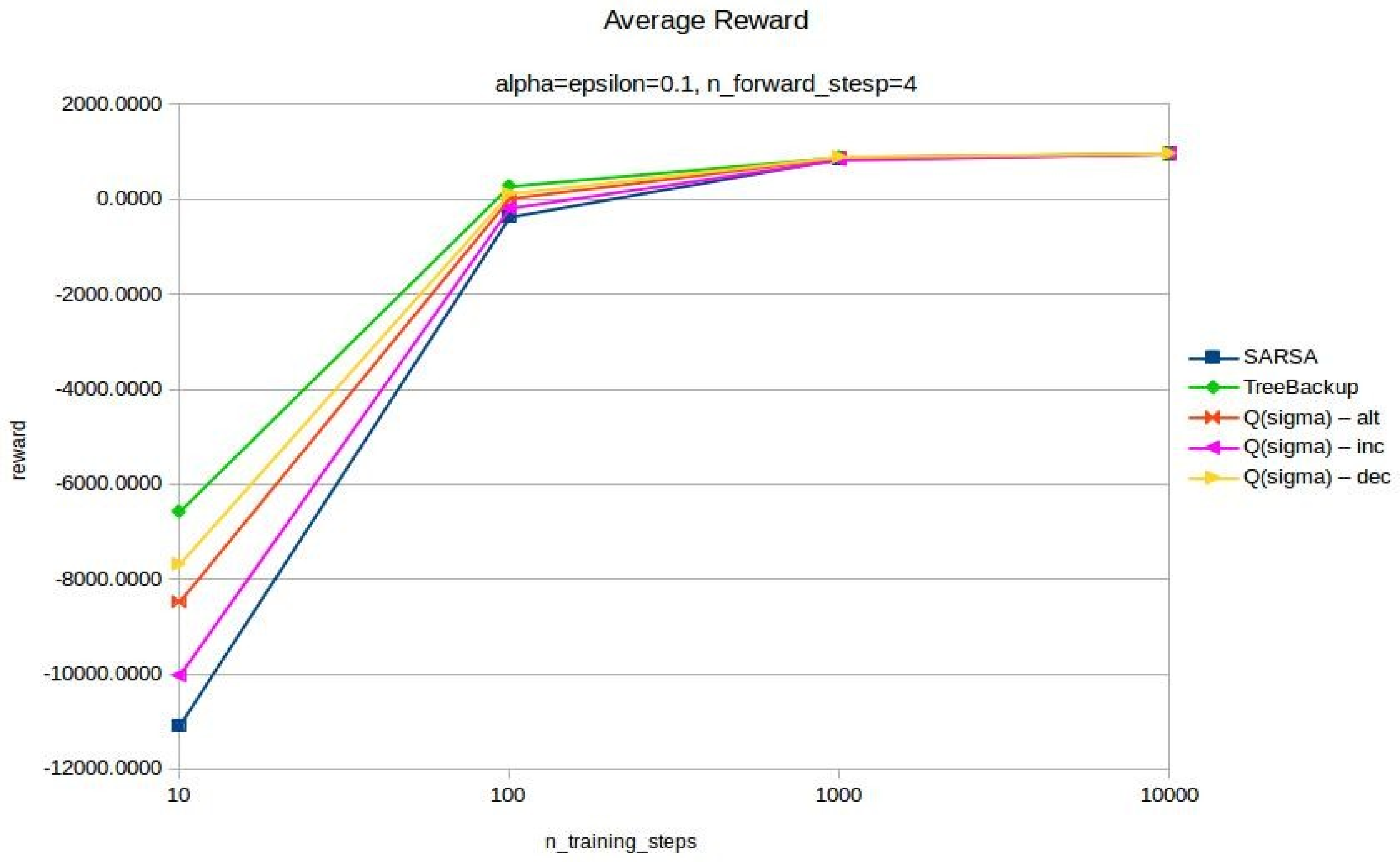
Average Reward (epsilon=0.3)

n_forward_steps = 20



SARSA, TreeBackup, Qsigma–alt, Qsigma–inc, Qsigma–dec: for n_steps = 5, 10, 20

Results
(alpha = epsilon = 0.1, n = 4)



Average Reward as we increase the number of training episodes: