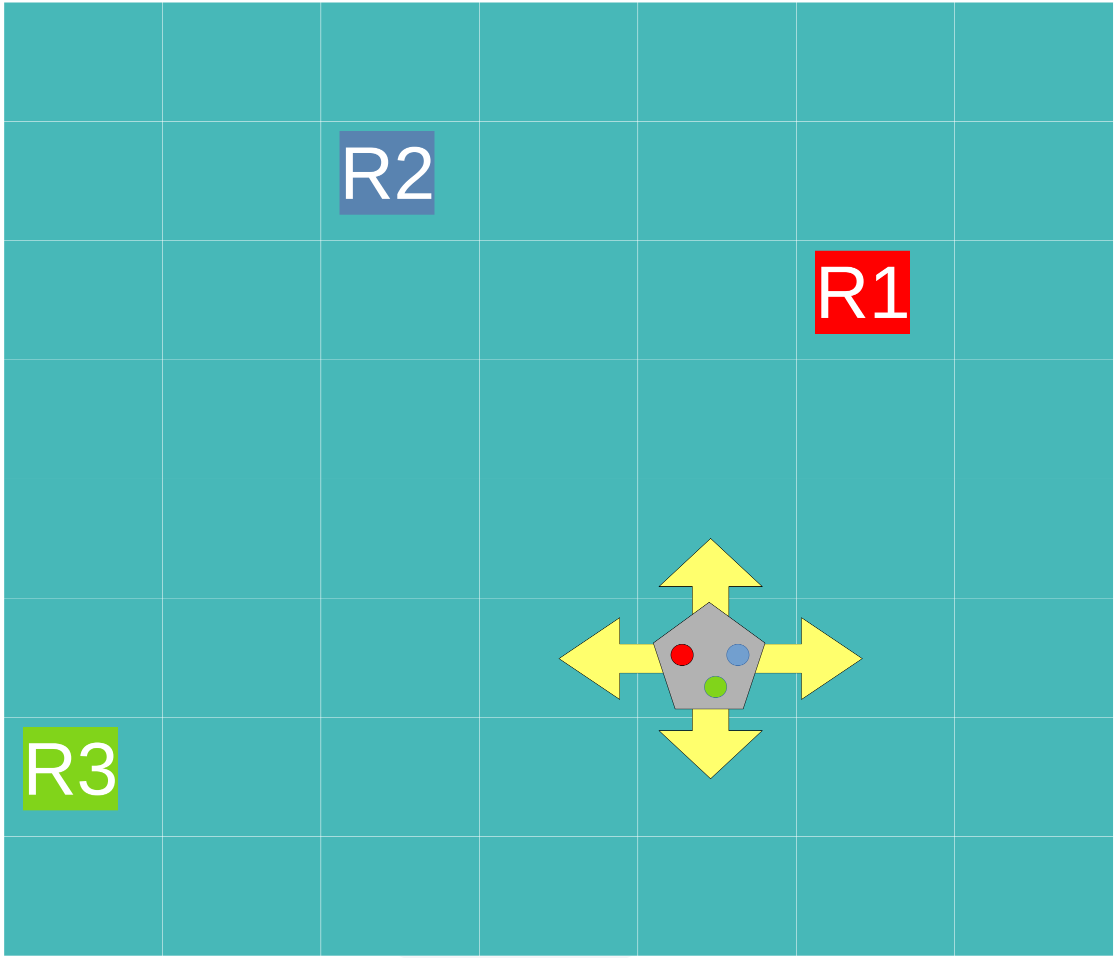
Multi-agent Democratic Gridworld

An anthological perspective

Andrea Longo
Reinforcement learning

"In a grid world a ship is guided by the democratic vote of some agents within it, each agent is intent on reaching a specific cell"



R2
R1
R3

Getting off

When an agent reach its goal it will not gain further rewards and cannot vote

Playoff

In case multiple directions gains the same number of votes the direction taken will be chosen randomly among those

Win away

The first cell for the ship cannot contain any reward

Goal

Learn a policy for each agent to make it reach its goal (hopefully quickly)

Evaluation

For each episode the optimality (total steps/steps of shortest path) will be the key measurement

Setting alternative

Tests are made either with random initial position or fixed at the middle

Restriction to spice it up

Use of single-agent reinforcement learning techniques readjusted to the multi-agent context

Keeping track

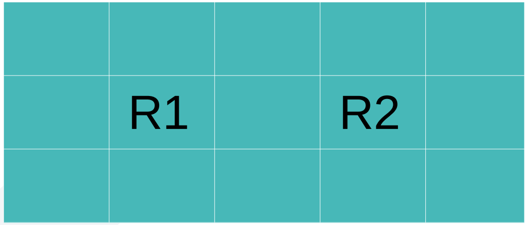
The way in which cooperative or competitive dynamics are established

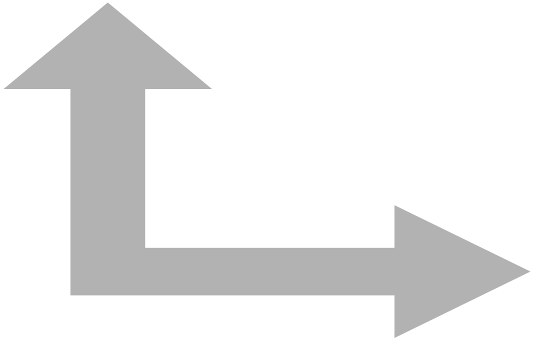
Setting extensions

- Multiple reward cells for each agent
- Obstacles inside the grid
- Competition between different agents
- Agreement incentive - move 2 steps if all vote that direction

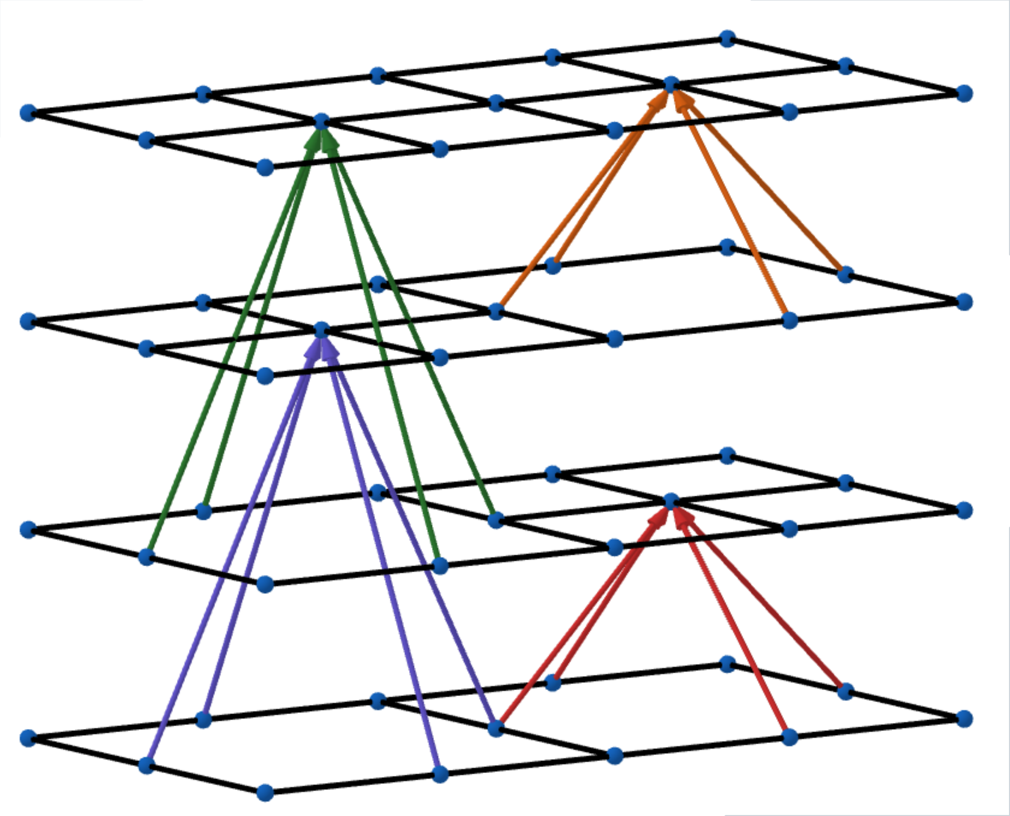
Intuition on state space

$|S| = X \cdot Y$





$|S| = (X \cdot Y) \cdot 2^N$



Approaches

Collaborative

Planning + Value contamination

$|S| = (X \cdot Y) \cdot 2^N$

Parametrized Value Iteration

Naive

Votes as Transition Probability

$|S| = (X \cdot Y) \cdot 2^N$

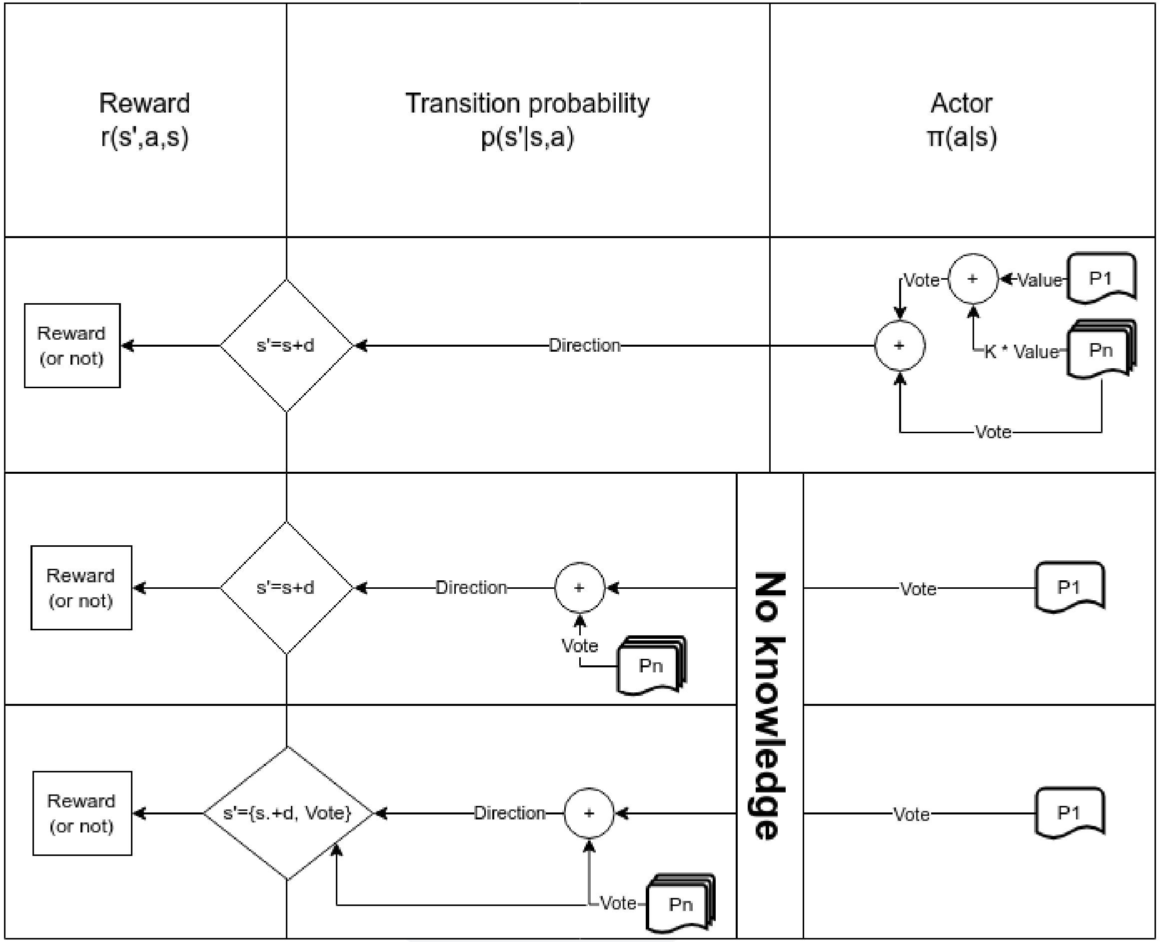
Q-learning

Unstructured

Previous votes into states

$|S| = (X \cdot Y) \cdot (2 \cdot 5)^N$

SARSA VFA with Neural Network


Reward
r(s',a,s)
Transition probability
p(s'|s,a)
Actor
π(a|s)
Vote
+
Value
P1
Reward
(or not)
s'=s+d
Direction
+
K * Value
Pn
Vote
Reward
(or not)
s'=s+d
Direction
+
Vote
Pn
No knowledge
Vote
P1
Reward
(or not)
s'={s.+d, Vote}
Direction
+
Vote
Pn
Vote
P1

Configurations

Environment

- Random initial state
- 3 Agents
- 10x10 Grid
- Rewards
  - Goal : 100
  - Wall : -10
  - Empty : -1

For all agents

- Discount : 0.95
- Evaluation episodes : 500

Collaborative Agents

- V initialization : Empty reward

Naive agents

- Epsilon : 0.1
- Alpha : 0.1
- Training episodes : 2000
- Q initialization : Empty reward

Unstructured agents

- Epsilon : 0.1
- Alpha : 0.0001
- Training episodes : 2000

Collaborative approach

Parametrized Value Iteration

- **Individual Value function construction is indipendent**

  simulating actions as direction taken, not votes

- **Contaminated Value function $CV_i = K_i \cdot sum_{j != i}(V_j)$**

  with K ~ U(k_min, k_max)

- **Since V expresses only future rewards the adjusted policy is**

$$Policy_i(s) = argmax_a[sum_i[R_i(s,a)] + \gamma \cdot CV_i(s'(s,a))]$$

Notable attempt

Inside Value Iteration

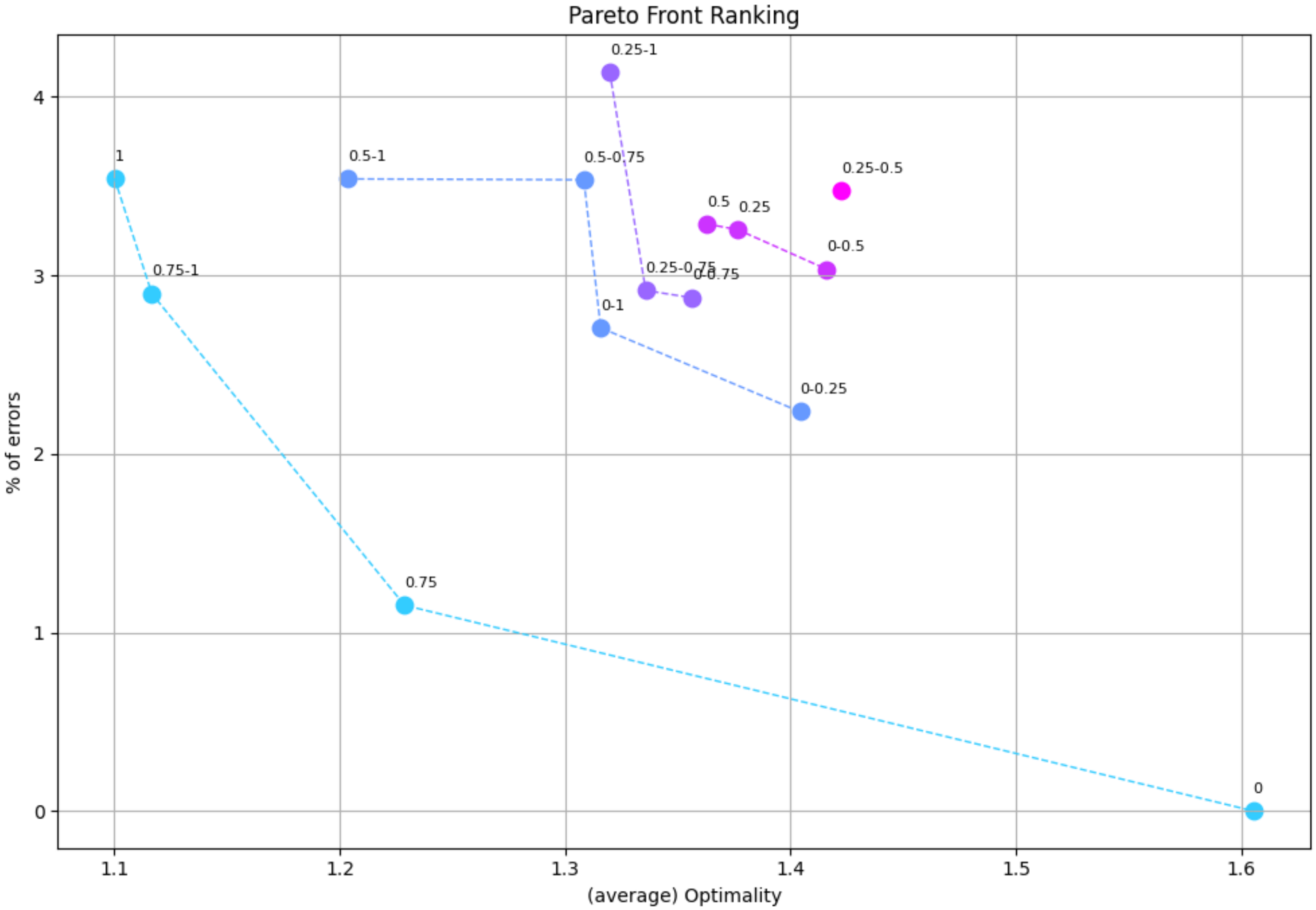
$$V_{i,t+1}(s) = \max_a[R_i(s,a) + \gamma \cdot V_{i,t}(s'(s,a))]$$

I tried with contamination during evaluation

$$V_{i,t}(s'(s,a)) \mathrel{+}= K_i \cdot \text{sum}_{j\,!=\,i}[V_{j,t}(s'(s,a))]$$

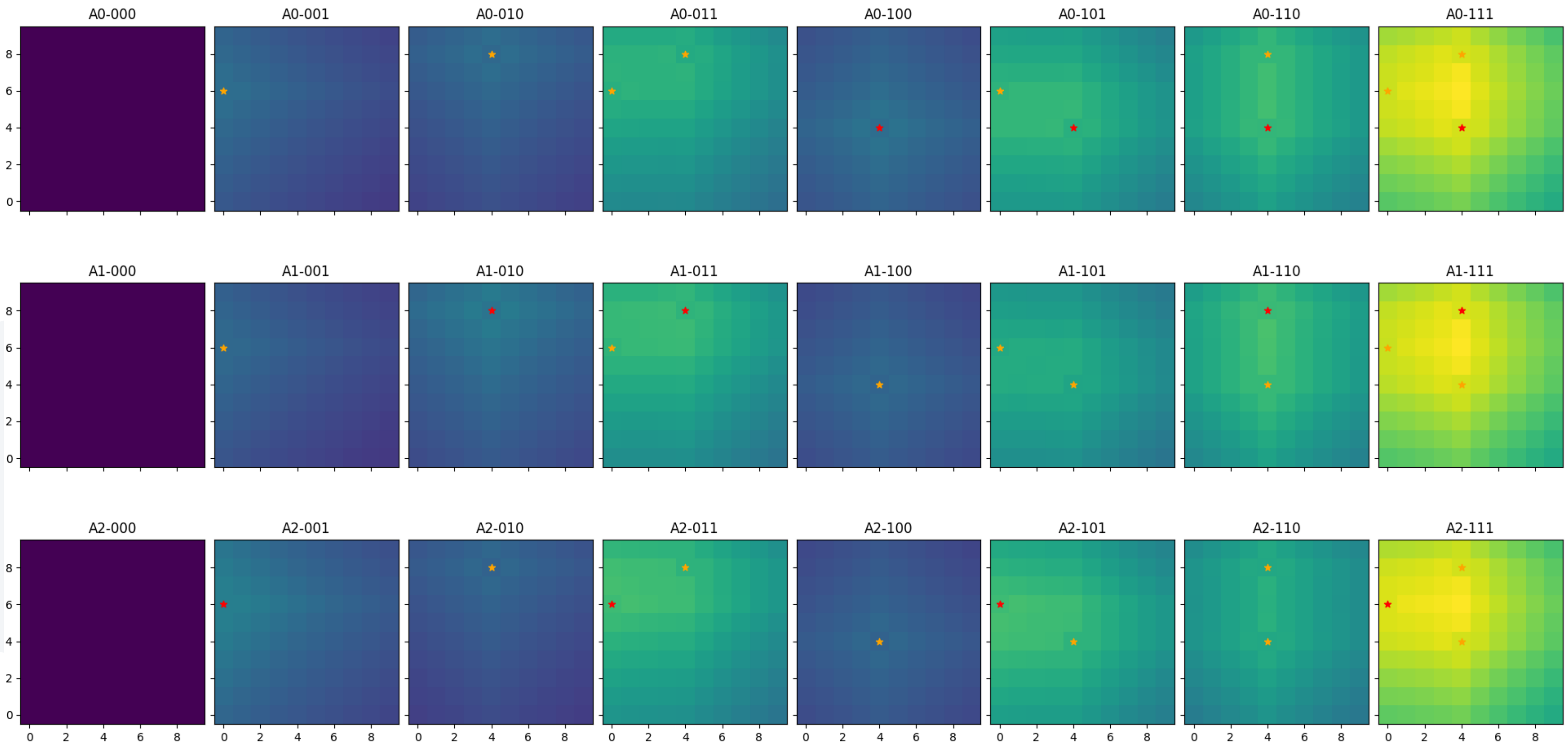
But the underline Bellman Equation rarely converged with K!=0

Best [k_min,k_max]



Results averaged over 100 test per configuration (2500 tests of 500 evaluation episodes each)

Contaminated values with K ~ U(0.75-1)



Naive Approach

Q-Learning

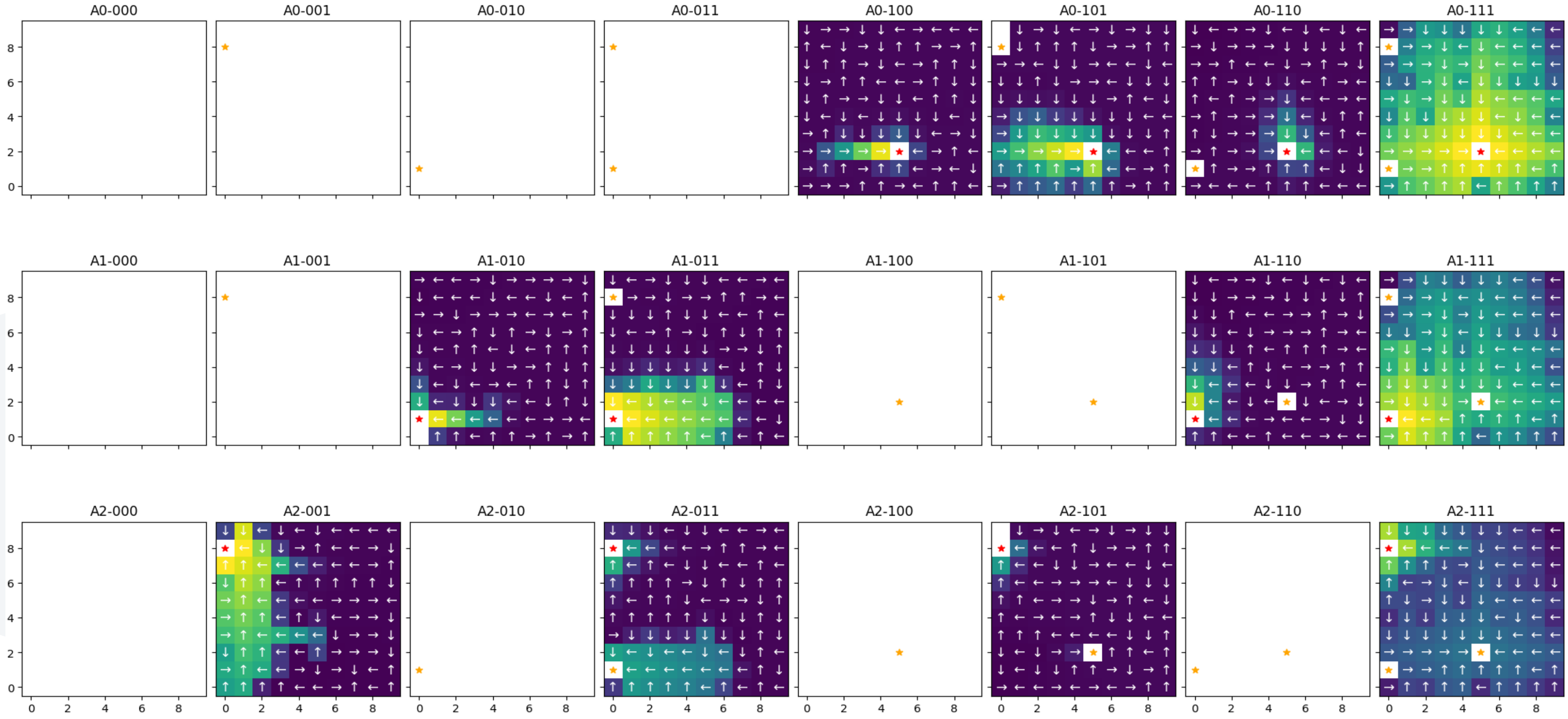
- **Agents constructs their own $Q_i(s,a)$ over the training episodes**

  after training the evaluation’s votes are greedy (no ε) and Q does not update
- **Votes are the actions**

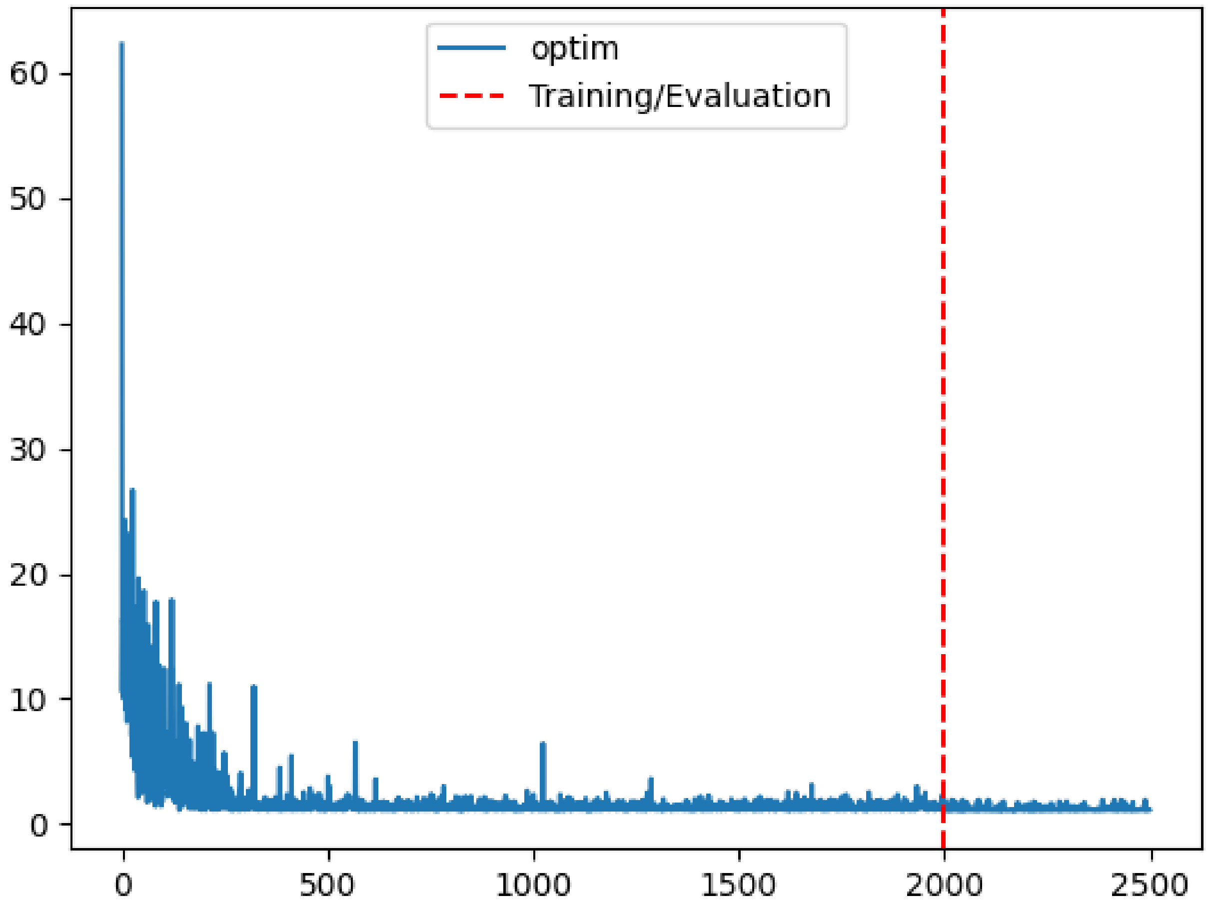
  the democratic step is integrated as Transition probability
- **Rewards are individual**

  unlike the previous case

Q Function


A0-000
A0-001
A0-010
A0-011
A0-100
A0-101
A0-110
A0-111
A1-000
A1-001
A1-010
A1-011
A1-100
A1-101
A1-110
A1-111
A2-000
A2-001
A2-010
A2-011
A2-100
A2-101
A2-110
A2-111
0
2
4
6
8

Optimality over episodes



Unstructured Approach

SARSA VFA with Neural Network

- **Features from states, previous votes, actions**

  $feat_i \rightarrow$ o-h(X) + o-h(Y) + o-h($alives_{j\,!=\,i}$) + lastVotes(actions) [+ o-h(actions)]

  |feat| = width + height + (n_agents – 1) + 4 [+ 4]

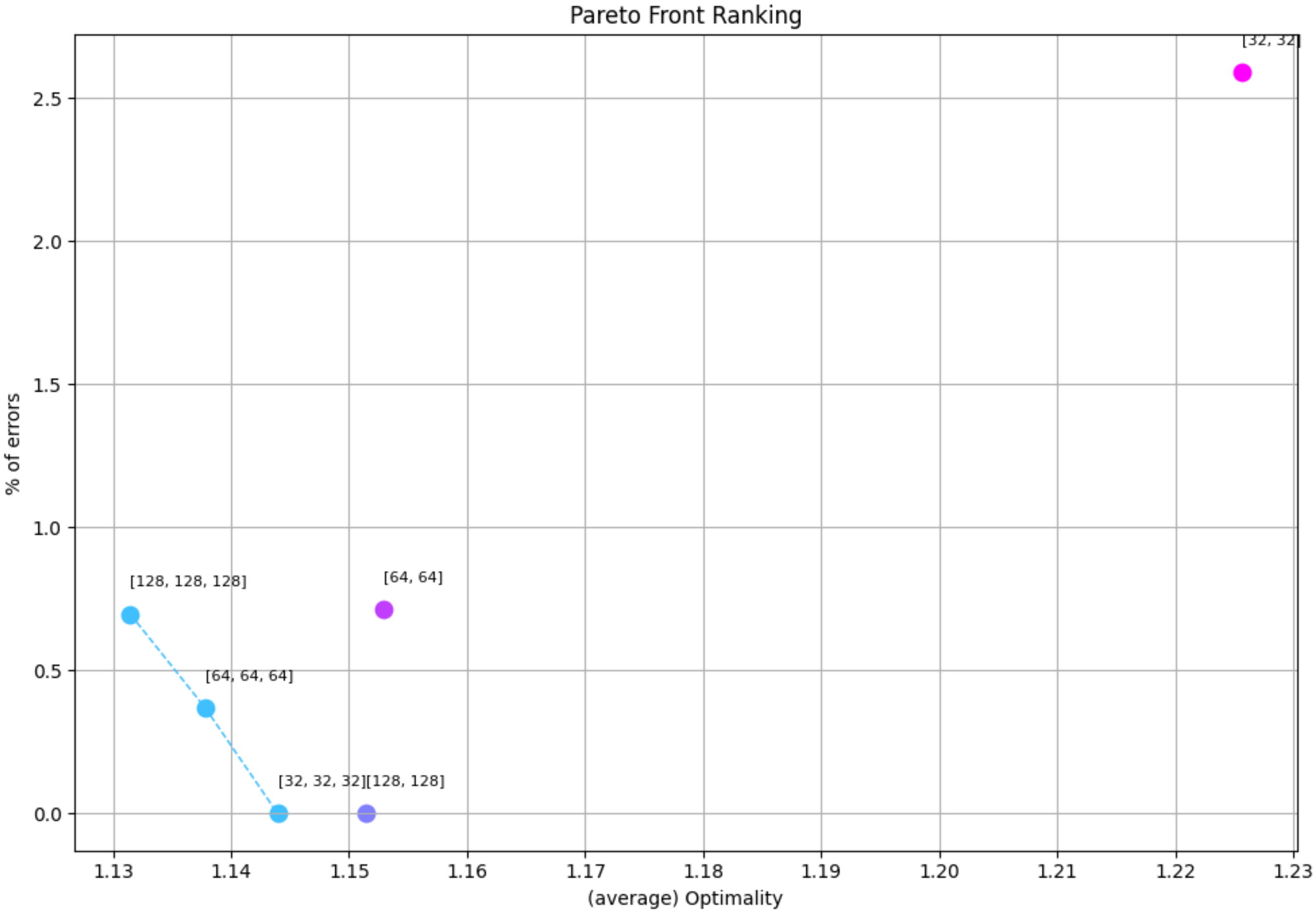
- **NN from feat to o-h(actions)**

  α = 1e-4 | loss = MSE | optim = SGD

- **SARSA was choosen as an on-policy algorithm**

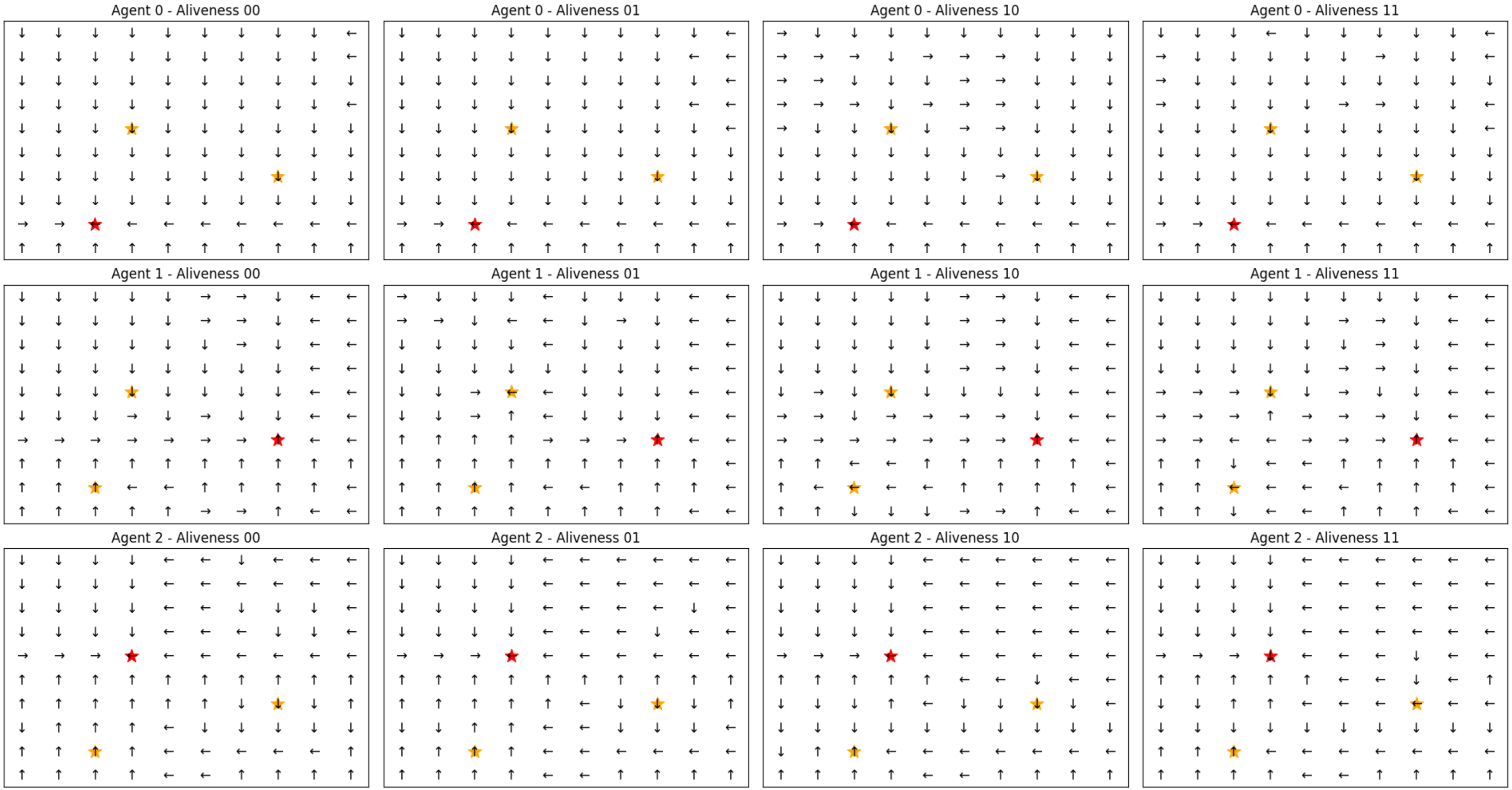
  to prevent the deadly triad

Best neural network shapes

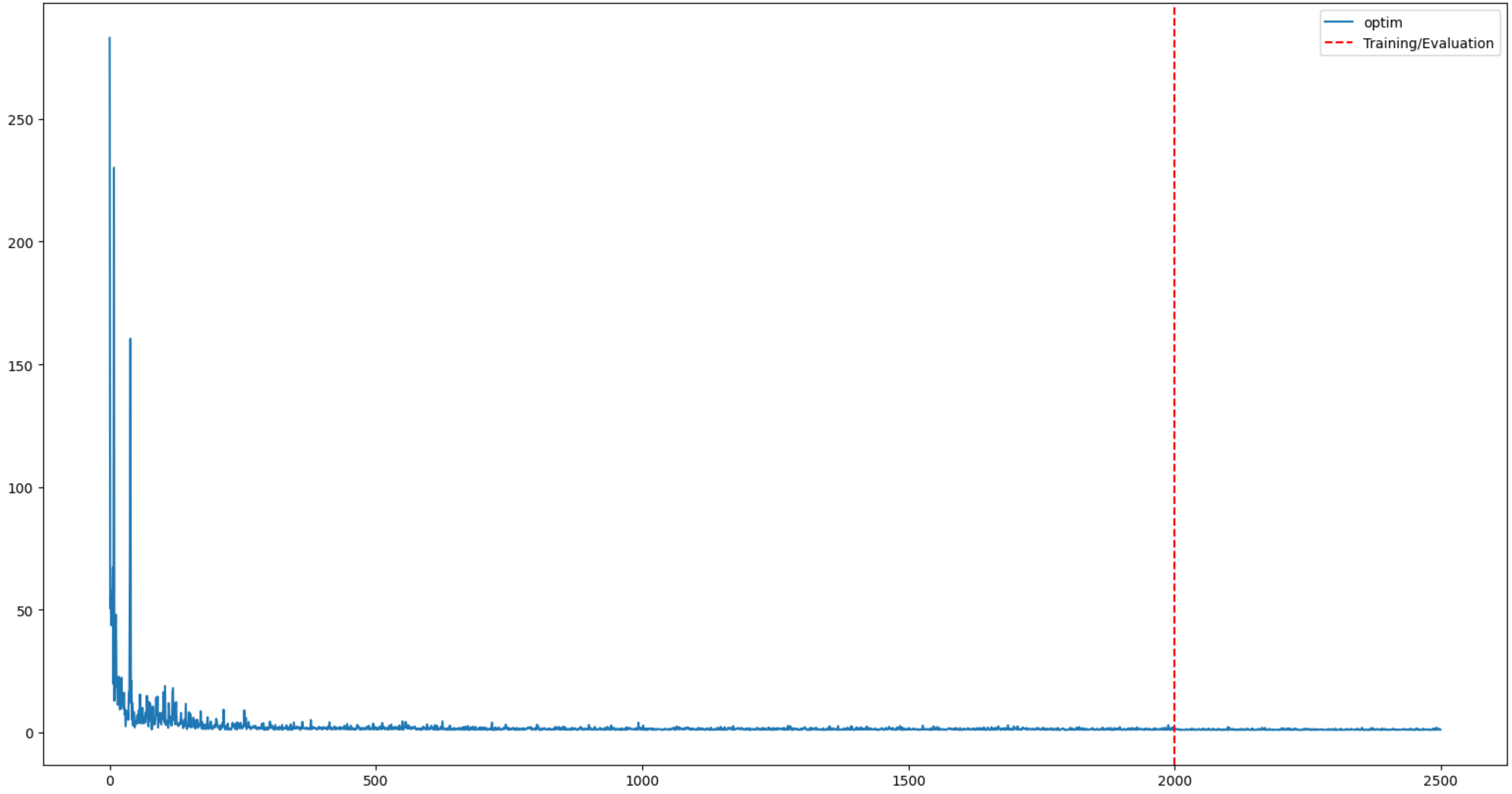


Results averaged over 25 test per configuration (150 tests of 2000 training and 500 evaluation episodes each)

NN directions
(assuming average previous votes)



Optimality over episodes



Conclusions