Empirically Evaluating Multiagent Reinforcement Learning Algorithms

Asher Lipson – alipson@cs.ubc.ca

MSc Thesis talk: 12 September 2005

Supervised by Kevin Leyton-Brown and Nando de Freitas

Road map

- Introduction
- Reinforcement Learning
- Multiagent Learning Algorithms
- Game Theory
- Existing Experimental Methods
- A Platform for Multiagent Reinforcement Learning
- Empirical Test and Results
- Questions

Introduction

- Interest in algorithms for game theoretic settings

  Focus: New Algorithms, eg. Littman [1994]; Claus and Boutilier [1997]; Singh *et al.* [2000]; Bowling and Veloso [2001]; Bowling [2004]

  Lack general understanding of strengths and weaknesses

  Different metrics used to judge performance

- This research has two contributions:

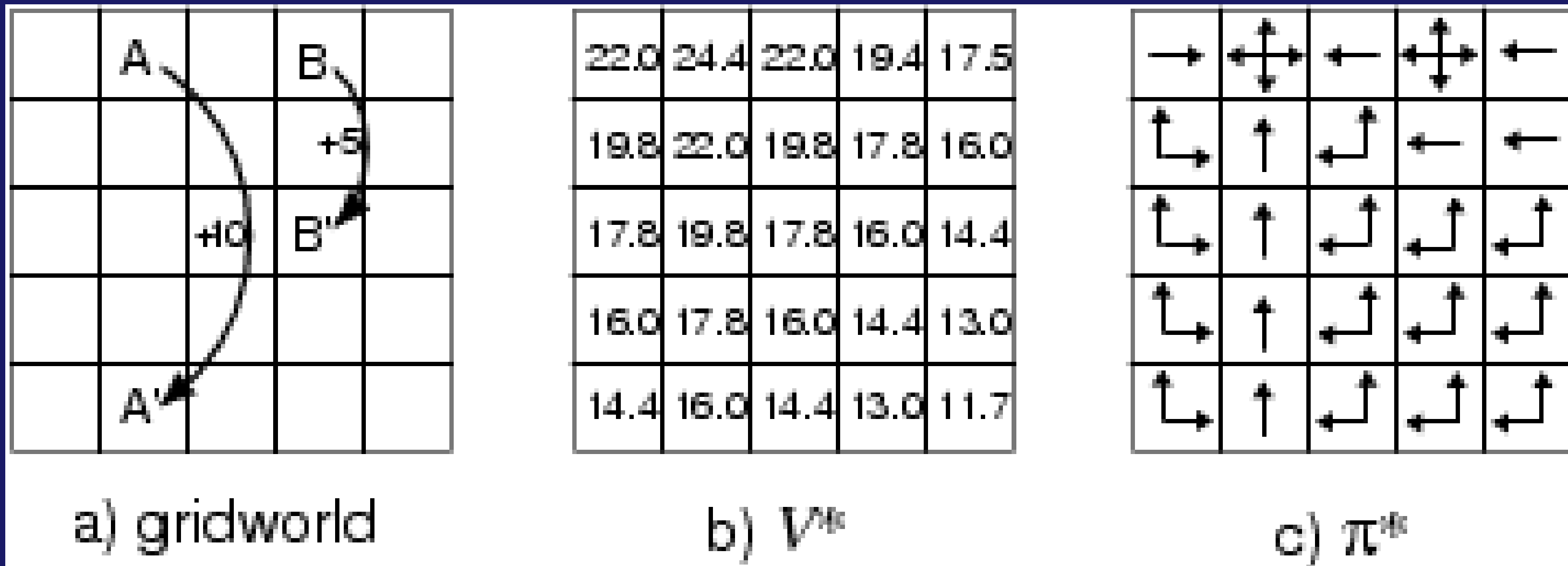
  1. A platform for experiments on MARL algorithms
  2. Analysis of an empirical test run on the platform

Reinforcement Learning

- Method to learn optimal actions in an environment
- Algorithm receives information about the state, takes an action and then receives feedback/reward
- **Reward only dependent on agent's action**
- Goal: Find optimal action in each state
- Popular RL method: Q-learning [Watkins and Dayan, 1992]
- Examples: Helicopter flying [Ng *et al.*, 2004], Single agent environments [Sutton and Barto, 1999]

(a)

(b)


A
B
+5
+10
B'
A'
22.0 24.4 22.0 19.4 17.5
19.8 22.0 19.8 17.8 16.0
17.8 19.8 17.8 16.0 14.4
16.0 17.8 16.0 14.4 13.0
14.4 16.0 14.4 13.0 11.7
a) gridworld
b) V^*
c) π^*

Multiagent Learning

- Multiple agents interacting in single environment
- Repeatedly play actions
- **BUT**

  Environment is no longer stationary

  Agent's reward dependent on EVERYONE's actions

  Notion of optimality from SARL does not exist

Game Theory

- Repeated games:

  Set of agents repeatedly play a normal form game (NFG)

  NFG: Matrix of payoffs indexed by agents' actions

- Nash equilibrium (NE):

  Every agent is best responding to every other agent

  No agent can obtain higher reward by changing strategy

- Two most common paradigms:

  Reward obtained and Convergence to a NE

MARL: Algorithms (some of them)

- Fictitious play [Brown, 1951]

  Count-based estimate, play best response

- Minimax-Q [Littman, 1994]

  Modify Q-learning; Assume the worst of the opponent

- GIGA-WoLF [Bowling, 2004]

  Estimate, Gradient, WoLF (variable step size), regret

- Global Stochastic Approximation (GSA) [Spall, 2003]

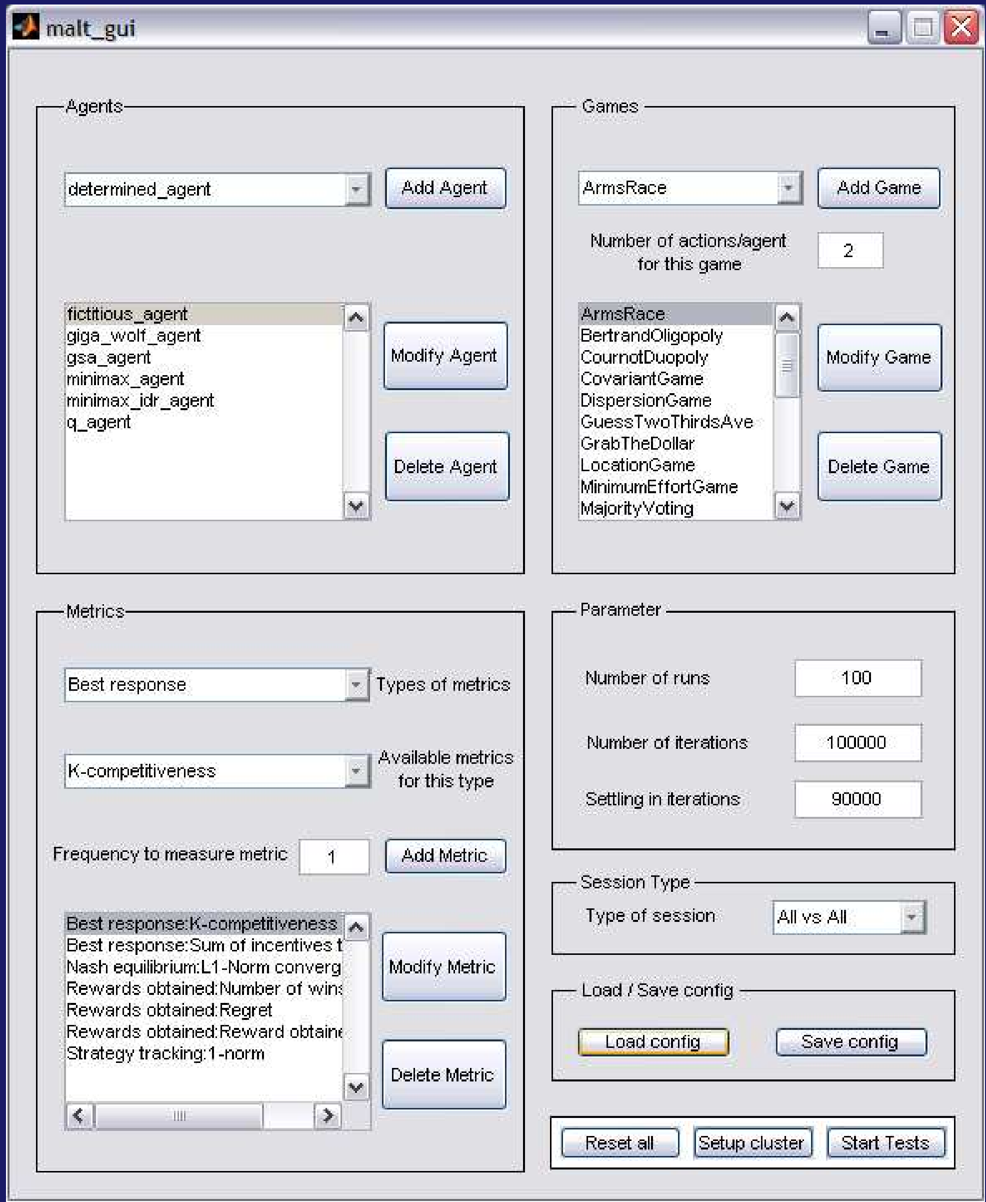
  Estimate, Annealing+Stochastic approximation, adds "jump"

Existing Experimental Methods

- Algorithms & their parameters
- Games
- Runs or trials
- Iterations per trial
- Settling vs. recording iterations

A Platform for MARL: Details

- Open, reusable platform
- Now available on the web
- Object-oriented Matlab
- All interaction through GUIs
- Currently 12 algorithms (including ones described earlier)
- Games from GAMUT software [Nudelman *et al.*, 2004]
- Game properties solved by Gambit [McKelvey *et al.*, 2004]


malt_gui
Agents
determined_agent
Add Agent
fictitious_agent
giga_wolf_agent
gsa_agent
minimax_agent
minimax_idr_agent
q_agent
Modify Agent
Delete Agent
Games
ArmsRace
Add Game
Number of actions/agent for this game
2
ArmsRace
BertrandOligopoly
CournotDuopoly
CovariantGame
DispersionGame
GuessTwoThirdsAve
GrabTheDollar
LocationGame
MinimumEffortGame
MajorityVoting
Modify Game
Delete Game
Metrics
Best response
Types of metrics
K-competitiveness
Available metrics for this type
Frequency to measure metric
1
Add Metric
Best response:K-competitiveness
Best response:Sum of incentives t
Nash equilibrium:L1-Norm converg
Rewards obtained:Number of wins
Rewards obtained:Regret
Rewards obtained:Reward obtaine
Strategy tracking:1-norm
Modify Metric
Delete Metric
Parameter
Number of runs
100
Number of iterations
100000
Settling in iterations
90000
Session Type
Type of session
All vs All
Load / Save config
Load config
Save config
Reset all
Setup cluster
Start Tests

A Platform for MARL: Metrics

- Reward-based Metrics (7)

  eg. Reward, regret, incentive to deviate, # wins

- Nash Convergence-based Metrics (2) :

  eg. Joint ℓ_1 distance to closest equilibrium

- Estimating opponent's strategy (4):

  ℓ_1 distance between estimate and actual

Visualisation

- View 4D table (algorithms, games, iterations, runs)
- User controlled in a step-by-step process
- Can visualise specific subset of data cells in table and aggregate over the rest
- eg: Average reward achieved by each agent overall; Box plot of a metric results for each algorithm pairing; Average distance to a NE in each game

Empirical Test

- Six Algorithms: GIGA-WoLF, GSA, Minimax-Q, Minimax-Q-IDR, Q-learning, Fictitious Play
- Seven metrics
- 1200 10x10 instances from 12 game generators
- 1200 2x2 instances from TwoByTwo game generator
- 100k iterations, 90k settle, 10k record
- Kolmogorov-Smirnov Z test used to test statistical similarity

High-level Observations

- 9 High-level observations, including:

1. No algorithm dominates
2. Different generators are required for accurate performance
3. No relationship between algorithm performance and the number of actions in the game
4. Large experiments are easier to run on our platform

Reducing the Size of the Space

- 21 algorithm pairs, 24 game generators, 100 instances, 10k iterations = 504 million cells in the 4D data table
- Too big to consider the results in each cell $\Rightarrow$
  1. Average over iterations
  2. Average over instances
  3. Generators split into 2x2 & 10x10 sets
  4. Algorithms kept separate
- 19 total claims/hypotheses, subset described next

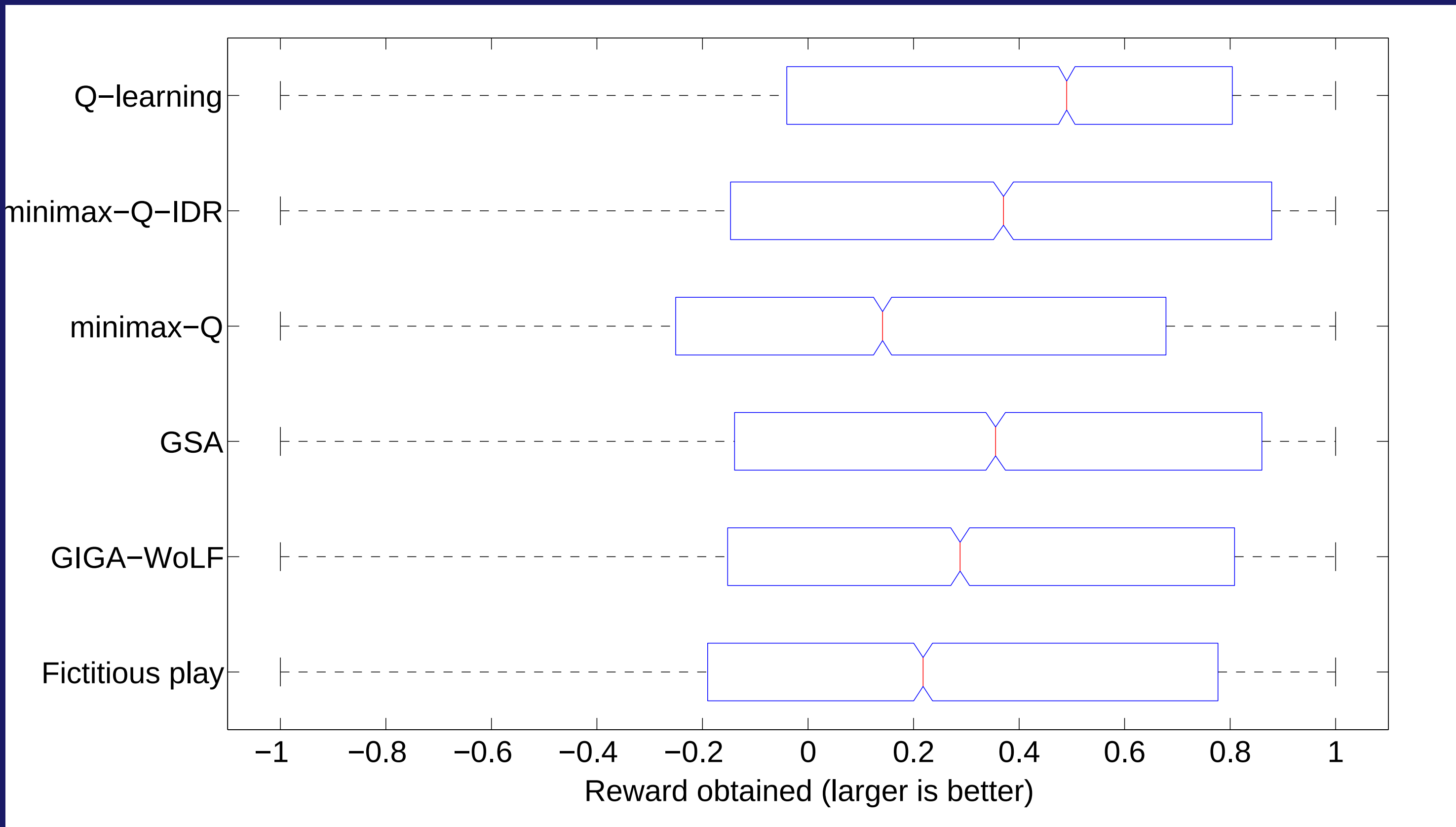
Results: Reward-based

- No algorithm obtains highest avg. reward in either 2x2 or 10x10 sets of generators.

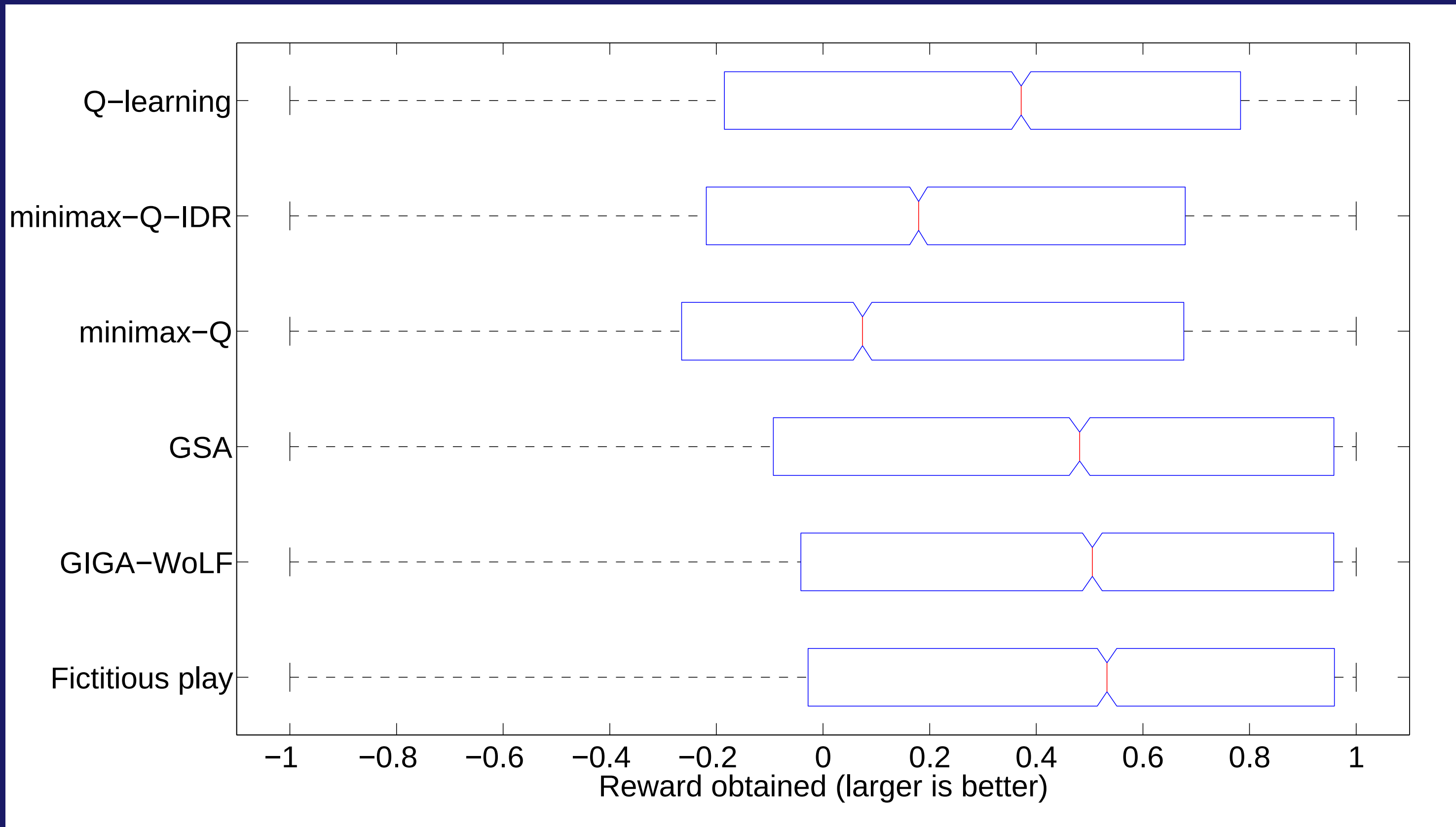
$\Rightarrow$ Average reward is opponent dependent

- Q-learning achieves highest mean and median reward in 2x2 set.

⇒ Averaged over all opponents, games



- Fictitious play obtains highest avg. mean and median reward in 10x10 set.

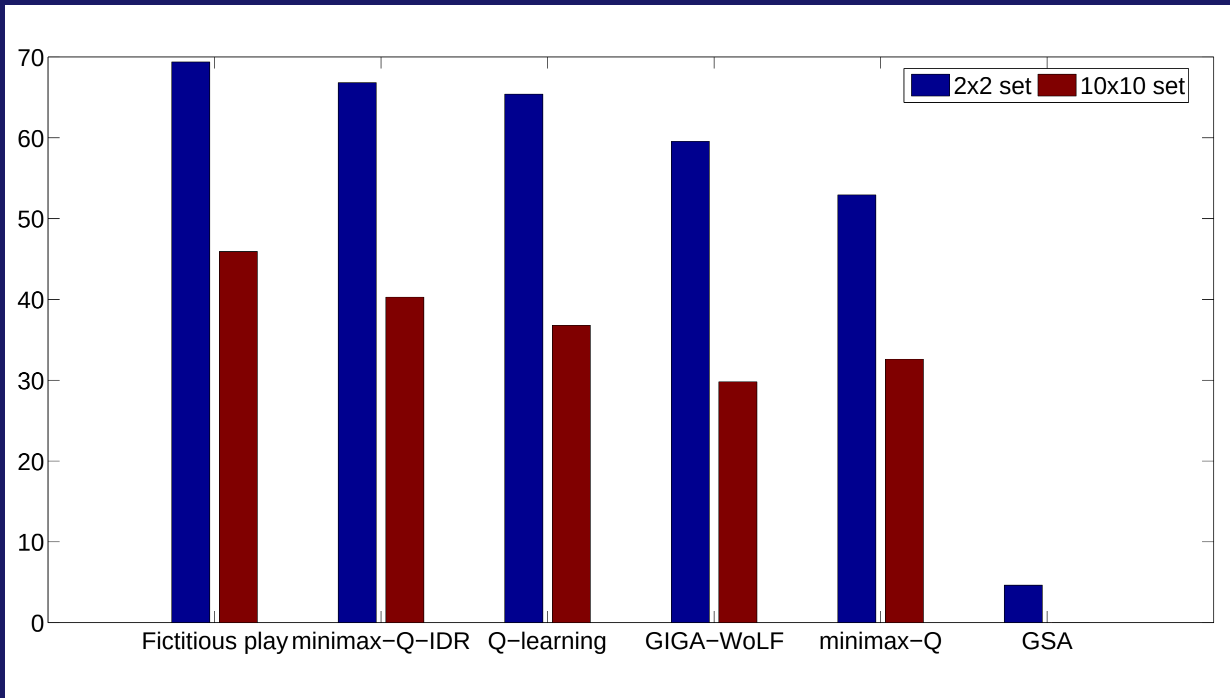


- Fictitious play obtains highest avg. mean and median reward in 10x10 set.
- GIGA-WoLF achieves lower avg regret, sometimes negative.

$\Rightarrow$ Designed with this goal in mind

Results: Nash Convergence-based

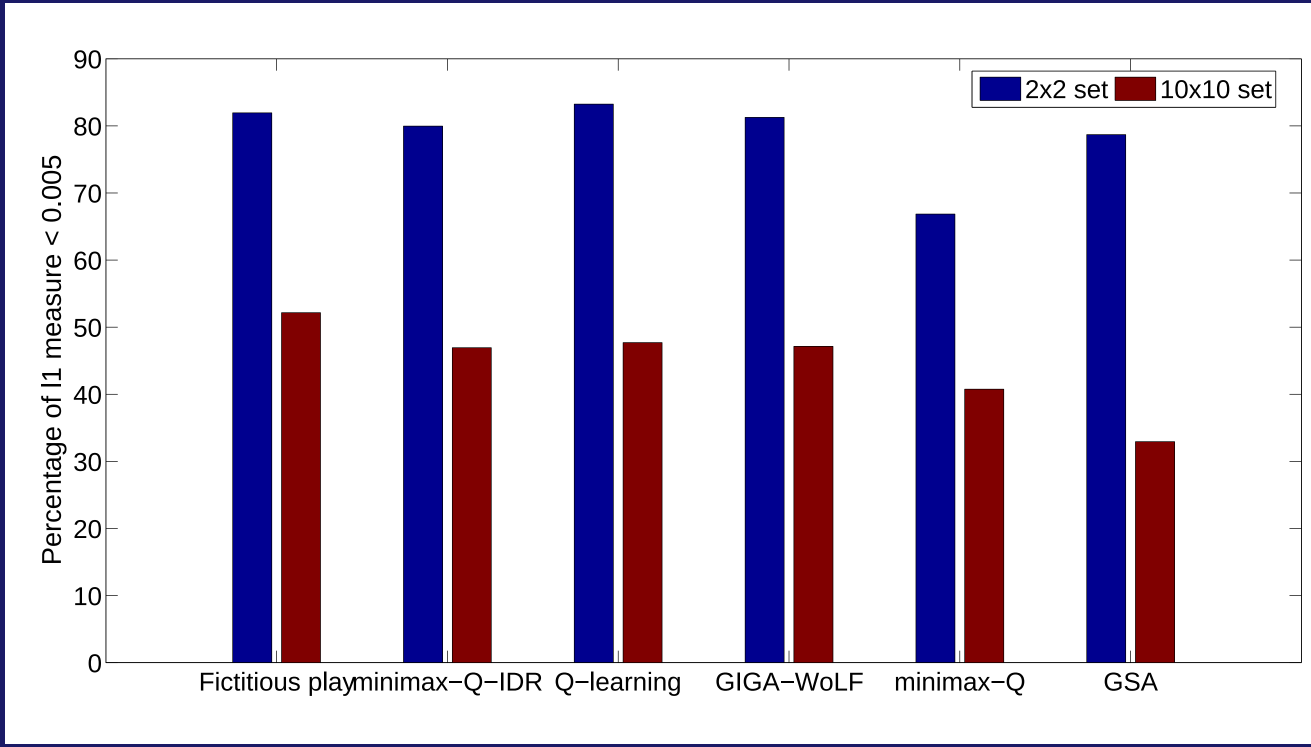
- No relationship between obtaining reward & converging to a NE.
- Algorithms often converge, but often fail to converge.



Results: Nash Convergence-based

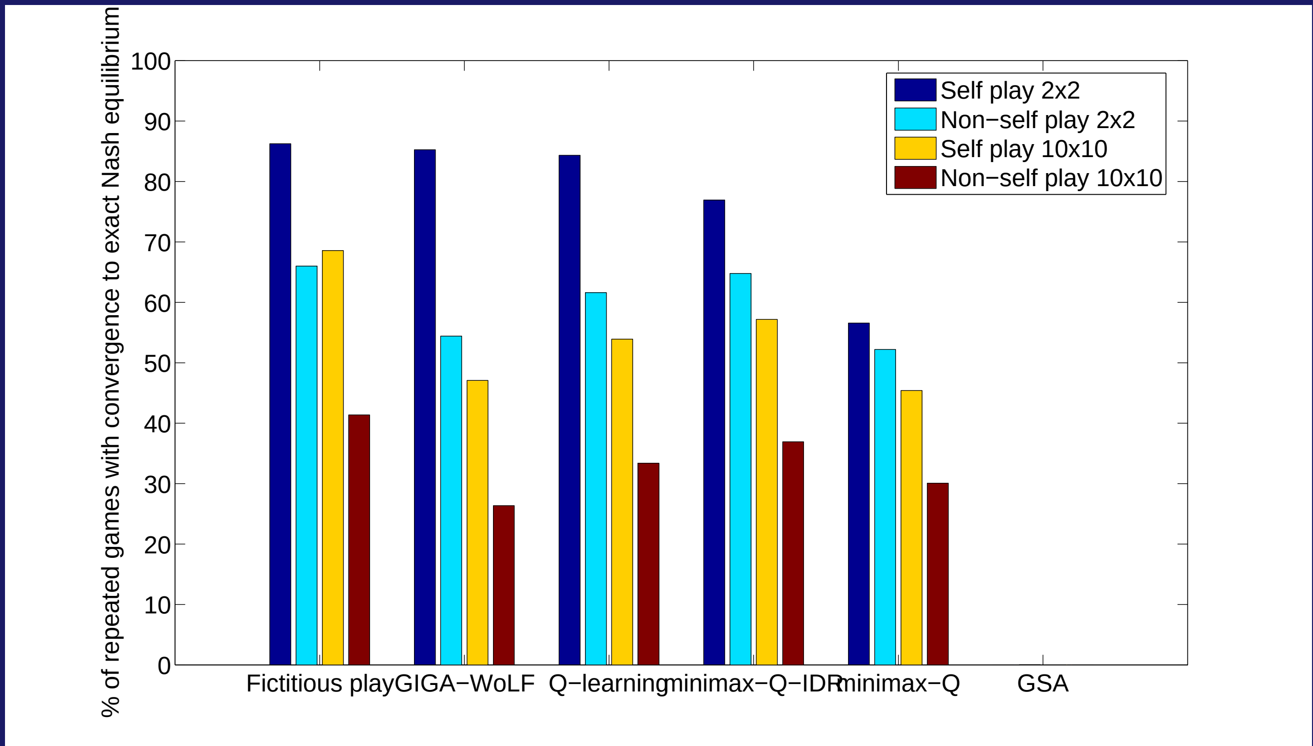
- Algorithms often converge "close" (< 0.005) to a NE.

$\Rightarrow$ 2x2: algorithms $> 70\%$; 10x10: Fictitious play $> 50\%$



Results: Nash Convergence-based

- Algorithms converge more often exactly in self play than non-self play.



Conclusion

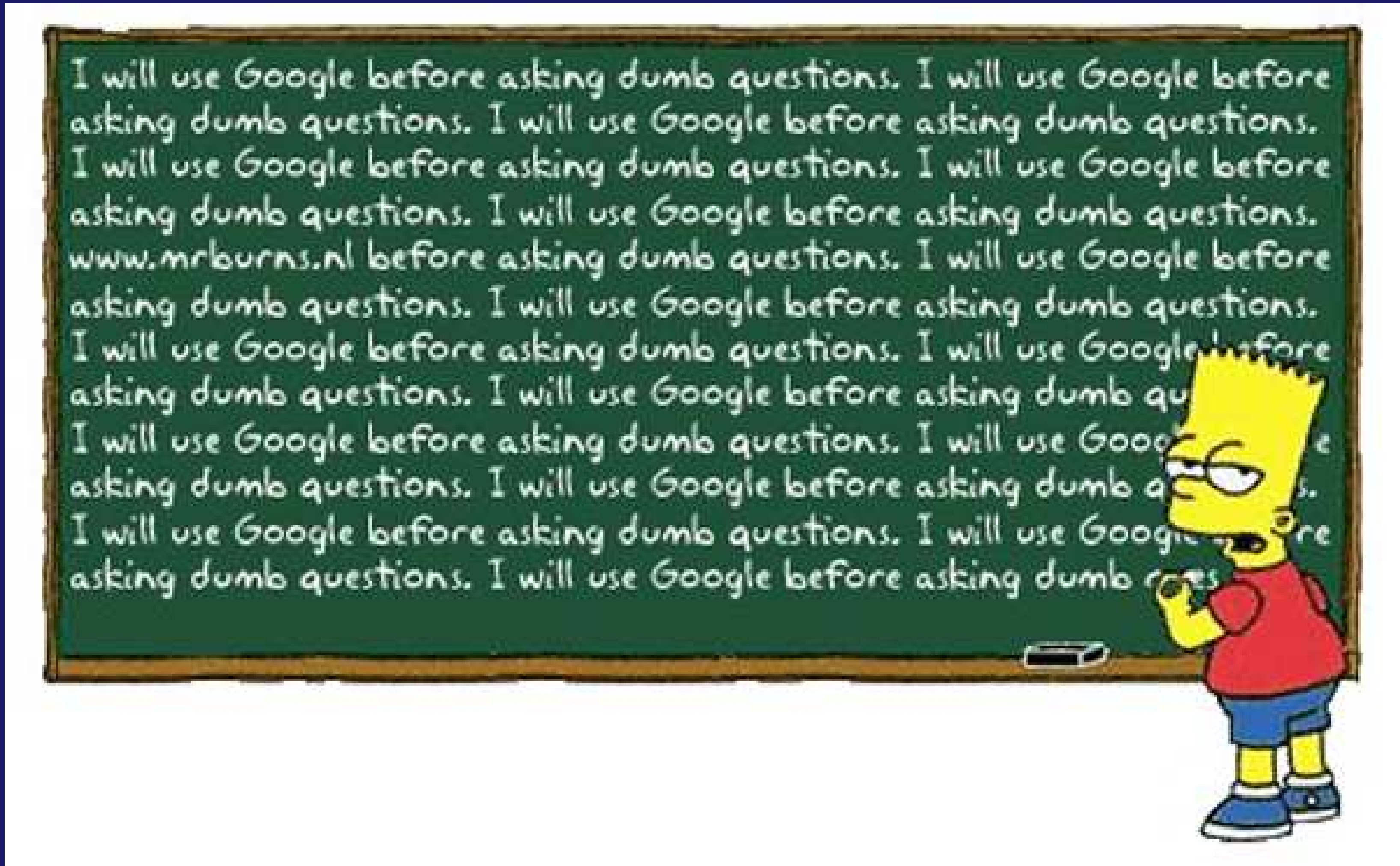
- Final analysis: 9 observations, 19 claims
- Platform proved to be extremely useful for this research

  Experiment ran for 2 CPU years on the cluster

  Survived several cluster outages
- In analysis phase:

  GUI speeded up selection of interesting parameters

  Meant we probably ran more iterations of analysis
- Configuration files made available for reproducibility


I will use Google before asking dumb questions. I will use Google before
asking dumb questions. I will use Google before asking dumb questions.
I will use Google before asking dumb questions. I will use Google before
asking dumb questions. I will use Google before asking dumb questions.
www.mrburns.nl before asking dumb questions. I will use Google before
asking dumb questions. I will use Google before asking dumb questions.
I will use Google before asking dumb questions. I will use Google
asking dumb questions. I will use Google before asking dumb
I will use Google before asking dumb questions. I will use
asking dumb questions. I will use Google before asking dumb
I will use Google before asking dumb questions. I will use
asking dumb questions. I will use Google before asking dumb

References

M. Bowling and M. Veloso. Convergence of Gradient Dynamics with a Variable Learning Rate. In *ICML 18*, June 28 – July 1 2001.

M. Bowling. Convergence and no-regret in multiagent learning. In *NIPS 17*, 2004.

G. Brown. Iterative Solution of Games by Ficticious Play. In *Activity Analysis of Production and Allocation*, New York, 1951.

C. Claus and C. Boutilier. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *AAAI 4*, pages 746 – 752, July 28 1997.

M. Littman. Markov Games as a Framework for Multi-agent Reinforcement Learning. In *ICML 11*, pages 157 – 163, 1994.

R.D. McKelvey, A.M. McLennan, and T.L. Turocy. Gambit: Software Tools for Game Theory. Version 0.97.0.6. `http://econweb.tamu.edu/gambit`, 2004.

A.Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. Inverted Autonomous Helicopter Flight via Reinforcement Learning. In *Int. Symposium on Experimental Robototics*, 2004.

E. Nudelman, J. Wortman, K. Leyton-Brown, and Y. Shoham. Run the GAMUT: A Comprehensive Approach to Evaluating Game-Theoretic Algorithms. In *AAMAS 3*, July 19 – 14 2004.

S. Singh, M. Kearns, and Y. Mansour. Nash Convergence of Gradient Dynamics in General-Sum Games. In *UAI 16*, 2000.

J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*. John Wiley & Sons, Hoboken, New Jersey, 2003.

R.S. Sutton and A.G. Barto. *Reinforcement Learning, An Introduction*. The MIT Press, Cambridge, Massachusetts, 1999.

C.H. Watkins and P. Dayan. Q–Learning: Technical Note. *Machine Learning*, 8:279–292, 1992.