*AAMAS 2003*

*17th July 2003, Melbourne Australia*

David Hales & Bruce Edmonds

http://www.davidhales.com
http://bruce.edmonds.name

*Centre for Policy Modelling (CPM),*
*Manchester Metropolitan University.*
*http://cfpm.org*

# Evolving Social Rationality for MAS using "Tags"

*Trying to "make things work" by applying results gained from Agent-Based Social Simulation*

# Why study cooperation?

- Many hard to explain cooperative interactions in human societies (*not explained* by traditional economics models)
- We want to engineer large-scale open artificial agent based systems
- More generally, how low level (often selfish entities) may come to form internally cooperative higher level entities

# What is "Social Rationality" ?

Hogg and Jennings (1997) define it as:

***"Principle of Social Rationality*: If a socially rational agent can perform an action whose joint benefit is greater than its joint loss, then it may select that action."**

Kalenka and Jennings (1999) compare hard-coded "individually rational" and "socially rational" agents in a simulated warehouse unloading scenario (where simulated robots must decide if to give help to others or not). We re-implement a version of this with tags – we will look at this later.

# Some problems in implementing social rationality

- "Social rationality" is introduced as a way of engineering out (hard-coding away) selfish individualistic behaviour.
- But this means agents would be susceptible to cheating (selfish individualistic) agents that did not follow the socially rationality principle.
- It also means that agent programmers need to know *in advance* about the specific task domains the agent will be involved - difficult if not impossible with complex systems + goes against a major "vision" about what MAS are supposed to be about (i.e. run-time self-organisation)
- if it's hard to find utility functions for selfish behaviour just think how hard it is to find utility functions for socially rational behaviour!

# What are "tags"

- Tags are observable labels or social cues
- Agents can observe the tags of others
- Tags evolve in the same way that behavioral traits evolve
- Agents may evolve behavioral traits that discriminate based on tags
- John Holland (1992) discussed tags as powerful "symmetry breaking" mechanism which could be useful for understanding complex "social-like" processes
- In a biological (GA-type) interpretation, tags can be thought of as "visible" parts of the genome which have no other function than to be observable.

# Recent tag models

- Tags may be bit strings signifying some observable cultural cue Sugarscape model (Epstein&Axtell), Mabs1998, Mabs2000, Mabs2002 (Hales)
- Tags may be a single real number (Riolo, Cohen, Axelrod Nature2001).
- Earlier work by Riolo (1997) showed how tags could improve cooperation between agents playing the Iterated Prisoner’s Dilemma.

# Brief outline

- **Firstly** we present results from a simulation model showing how tags evolve altruistic behavior between strangers in the single round PD => this is a kind of "social rationality" *for free*.
- **Then we will show** how these results can be applied to the (more complex and realistic) robot scenario that was presented by Kalenka + Jennings.
- No time to describe our model of specialisation using tags – but this is in the paper.

# Tags and the Single-Round Prisoner's Dilemma

"How to get social rationality for free"

# A quick note on methodology

- The model to be presented was found by searching (automatically) a large ($10^{17}$) space of possible models.
- Automated intelligent searching of the space was implemented.
- Machine Learning tools were used to identify the characteristics of models which produced desirable results (high cooperation in this case)
- Full details at www.davidhales.com/thesis
- We are presenting results from simulations not *deductive* proofs – results are always contingent and open to revision – however, some progress is being made (mainly by Bruce Edmonds) on analytical and deductive formulations of these phenomena.

# Assumptions

- Agents are greedy (change behaviour to maximise utility)
- Agents are stupid (bounded rationality)
- Agents are envious (observe if others are getting more utility than themselves)
- Agents are imitators (copy behaviour of those they envy)

# The Prisoner's Dilemma

Given: T > R > P > S and 2R > T + S

| Player 1 / Player 2 | C | D |
|---|---|---|
| C | R / R | T / S |
| D | S / T | P / P |

# A one bit agent

- An agent represented by a single bit
- A value of "1" indicates the agent will cooperate in a game interaction
- A value of "0" indicates the agent will defect in a game interaction
- The value is not visible to other agents

# An evolutionary algorithm

Initialise all agents with randomly selected strategies

LOOP some number of generations

LOOP for each agent (a) in the population

Select a game partner (b) at random from the population

Agent (a) and (b) invoke their strategies receiving the appropriate payoff

END LOOP

Reproduce agents in proportion to their average payoff with some small probability of mutation (M)

END LOOP

# The “obvious” result

- Agents quickly become all defectors
- A defector always does at least as well as his opponent and sometimes better
- This is the “*Nash Equilibrium*” for the single round PD game
- The evolutionary algorithm therefore evolves the “*rational*” strategy

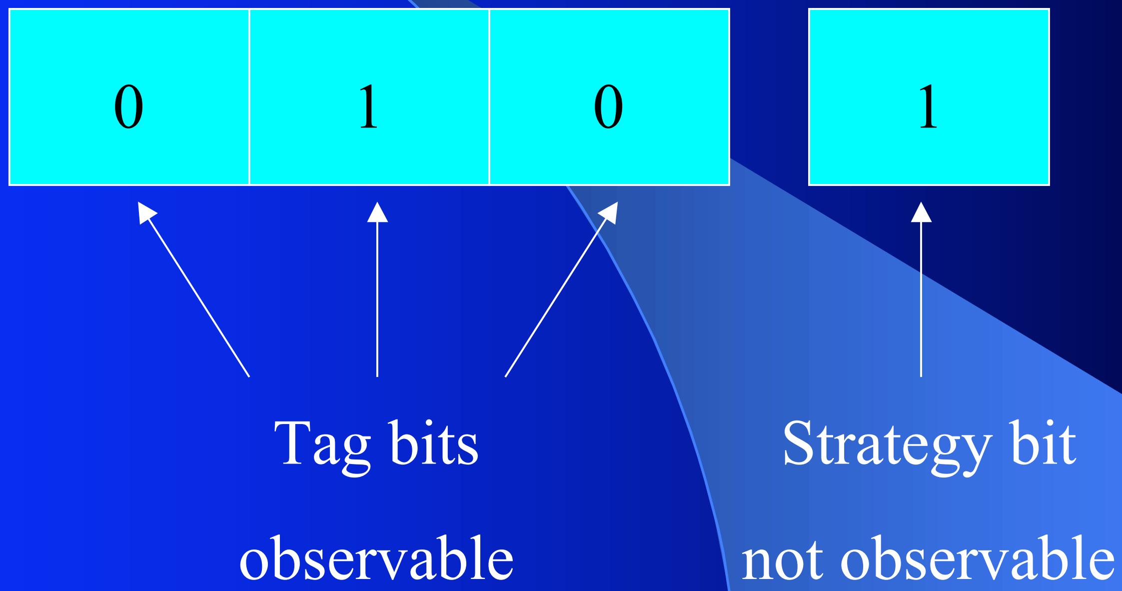
# Already well known ways that cooperation evolve

- Repeated interaction when agents remember the last strategy played by opponent (Axelrod TFT)
- Interaction restricted to spatial neighbours (Sigmund&Novak, May etc.)
- Agents observe the interactions of others before playing themselves (image and reputation – Sigmund&Novak)

*However, all these require agents with the ability to identify individuals and have many repeated interactions with them or have strict spatial structures imposed on interaction and reproduction (imitation)*

# An agent with “tags”

Take the “one bit agent” and add extra bits “tags” which have no effect on the strategy played but are observable by other agents



# Bias interaction by tag

- Change the evolutionary algorithm so agents bias their interaction towards those sharing the same tag bit pattern
- If an agent can find another agent in the population with the same tag it plays this – otherwise it selects a random partner (as before)
- During reproduction mutation is applied to both strategy bit and tag bits with same probability

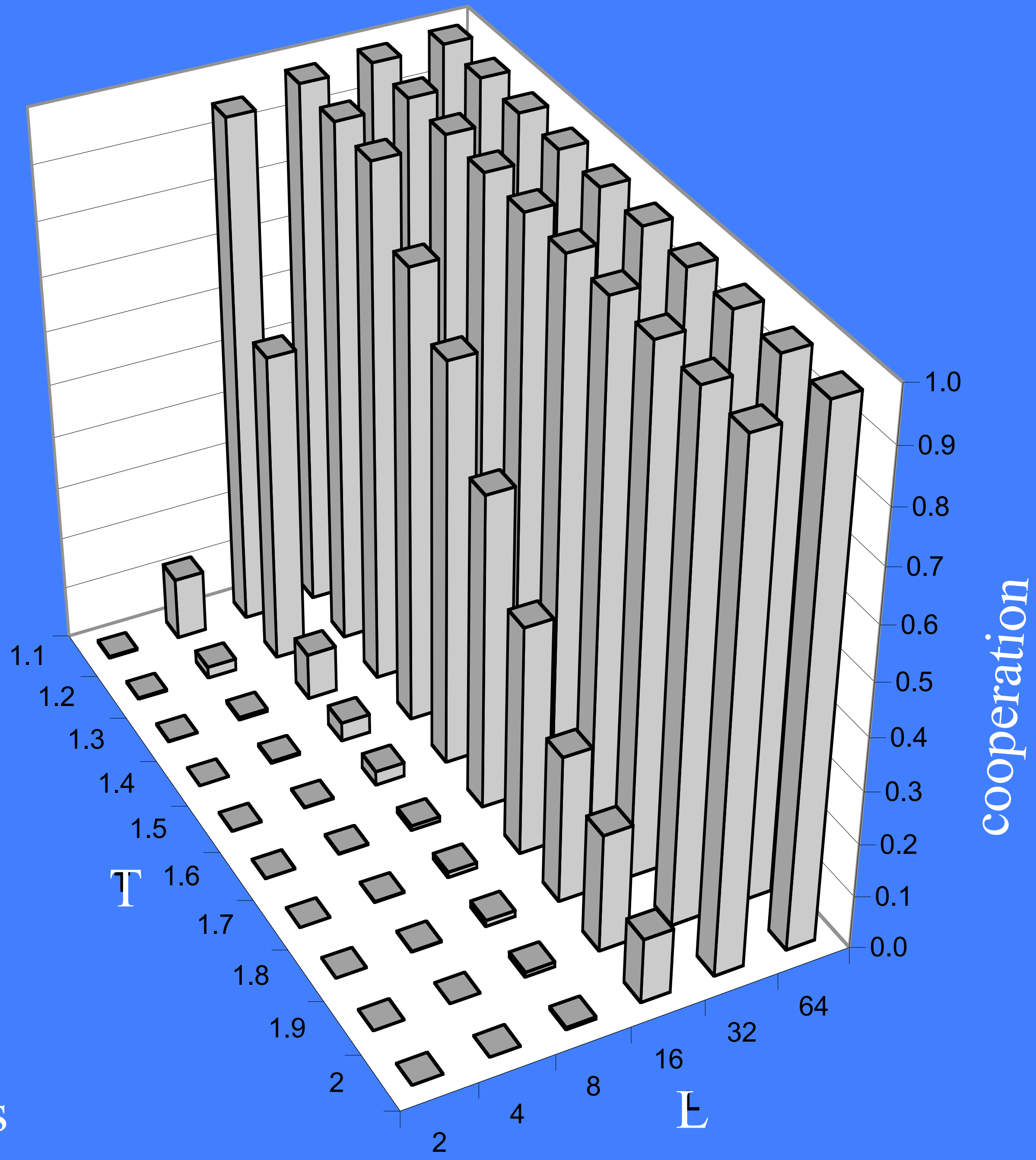
# Parameter values and measures

- Population size (N) = 100
- Length of tag (L) = [2..64] bits
- Refusals allowed (F) = 1000
- Mutation rate (M) = 0.001
- PD payoffs T = [1..2], R =1, P > S = small
- Execute algorithm for 100,000 generations
- Measure cooperation as proportion of total game interactions which are mutually cooperative

# Results

Cooperation increases:

- as T decreases
- as L increases




Each bar an average of 5 runs to 100,000 generations with different initial random number seeds

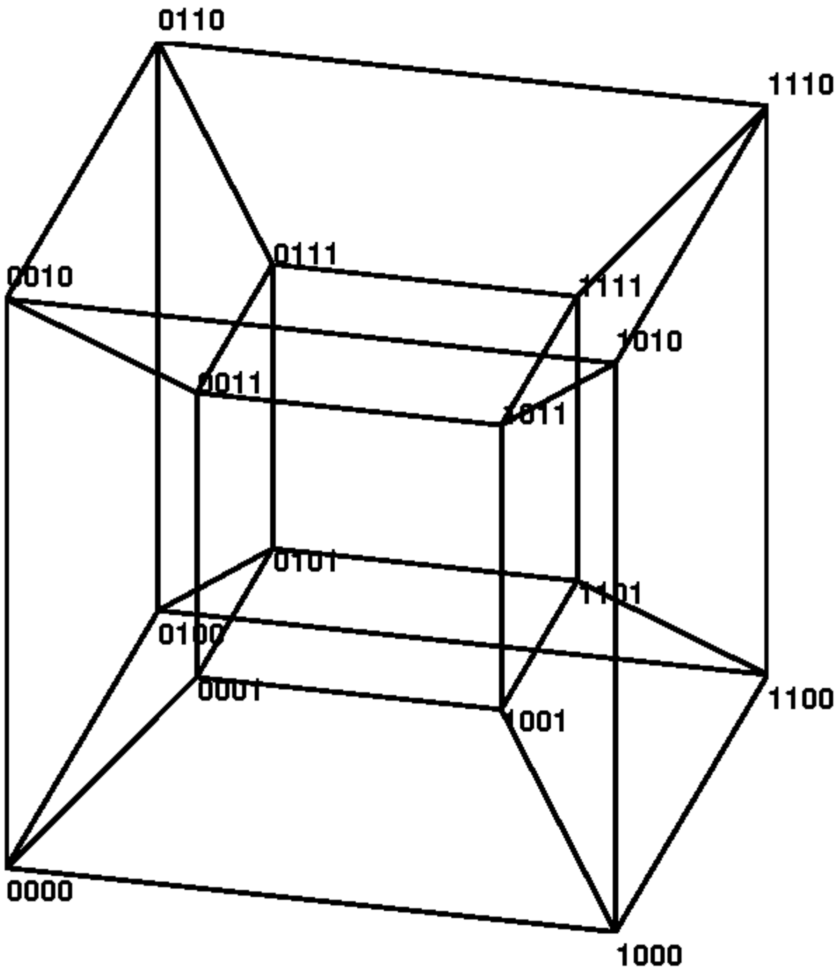
T = temptation payoff

L = length of tag in bits

# What’s happening?

- We can consider agents holding identical tags to be sharing the corner of a hyper-cube
- Interaction is limited to agents sharing a corner (identical tag bits)
- Therefore cooperative “groups” are emerging in these corners

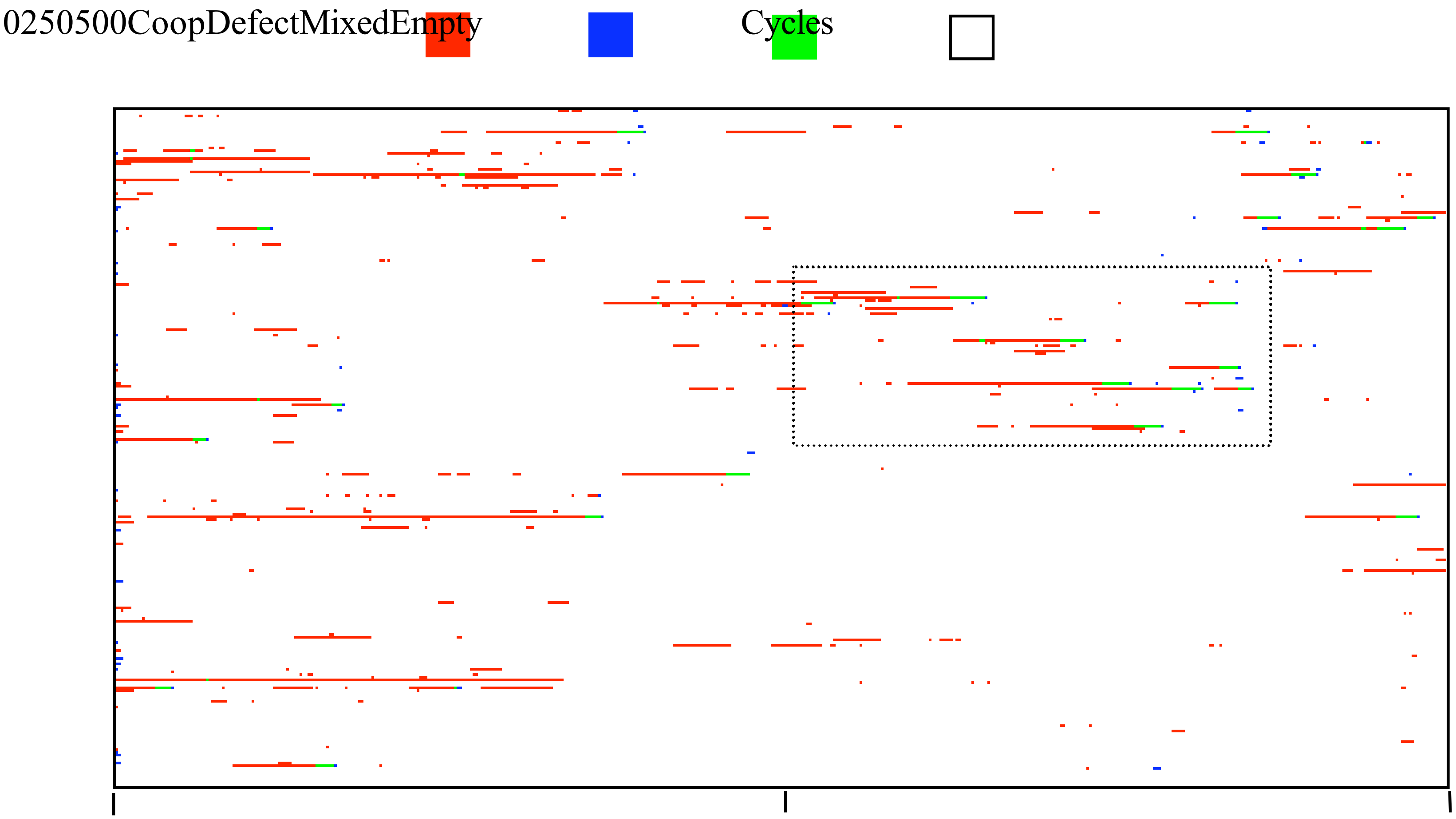
# A hypercube for 4 bit tags



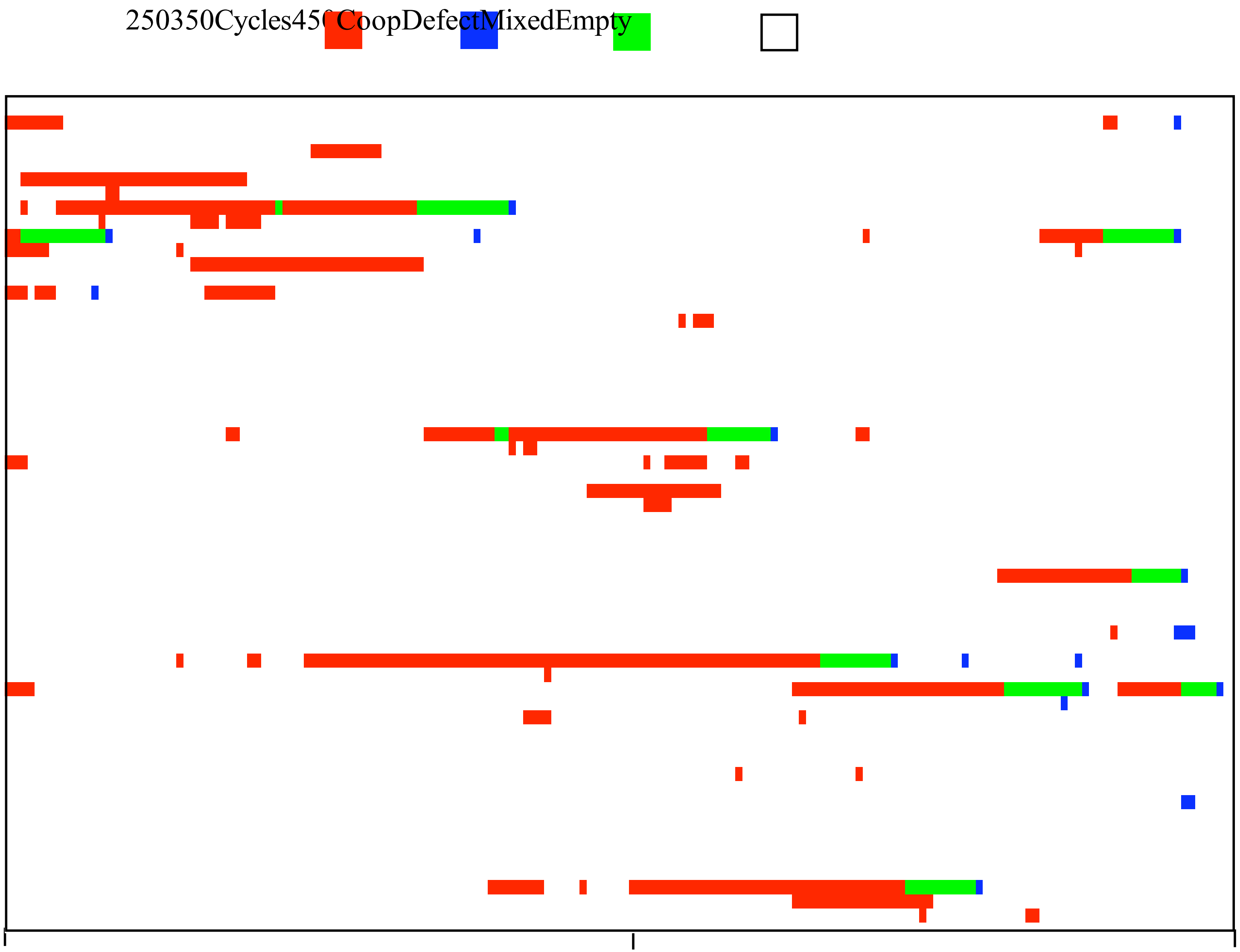

To visualise the process in time we produce a graph in which each horizontal line represents a single unique corner of the hypercube (set of unique tag bits)

We colour each line to indicate if it is occupied by all cooperative, all defective, mixed or no agents

# Visualising the process



# Visualising the process



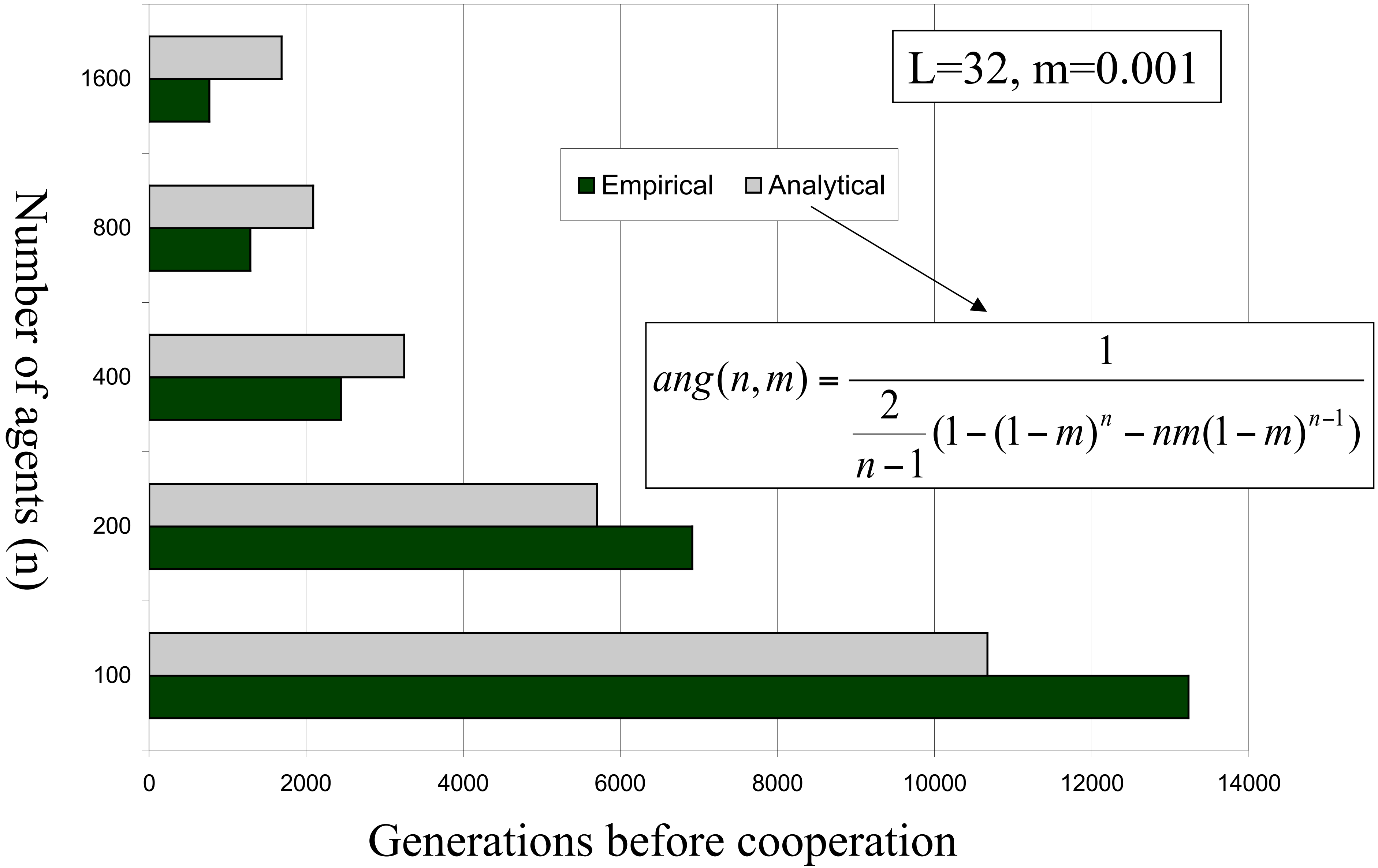
# What's happening?

- Defectors only do better than cooperators if they are in a mixed group (have cooperators to exploit)
- But by exploiting those cooperators they turn the group into all defectors quickly
- Agents in an "all defective group" do worse than agents in an "all cooperative group"
- So long as an all cooperative group exists the agents within it will out perform an all defective group, thus reproducing the group – mutation of tag bits spreads the cooperative group to neigbouring corners of the hypercube

# Cooperation from total defection

- If we start the run such that all strategy bits are set to defection, does cooperation evolve?
- Yes, from observation of the runs, cooperation emerges as soon as two agents sharing tag bits cooperate
- We can produce a crude analytical model predicting how long before cooperation evolves

# Cooperation from total defection



# Some conclusions

- A very simple mechanism can produce cooperation between strangers in the single round PD game
- Culturally, the tags can be interpreted as “social cues” or “cultural markers” which identify some kind of cultural group
- The “groups” exist in an abstract “tag space” not real physical space
- The easy *movement between groups* (via mutation and imitation) but strict game *interaction within groups* is the key to producing high cooperation

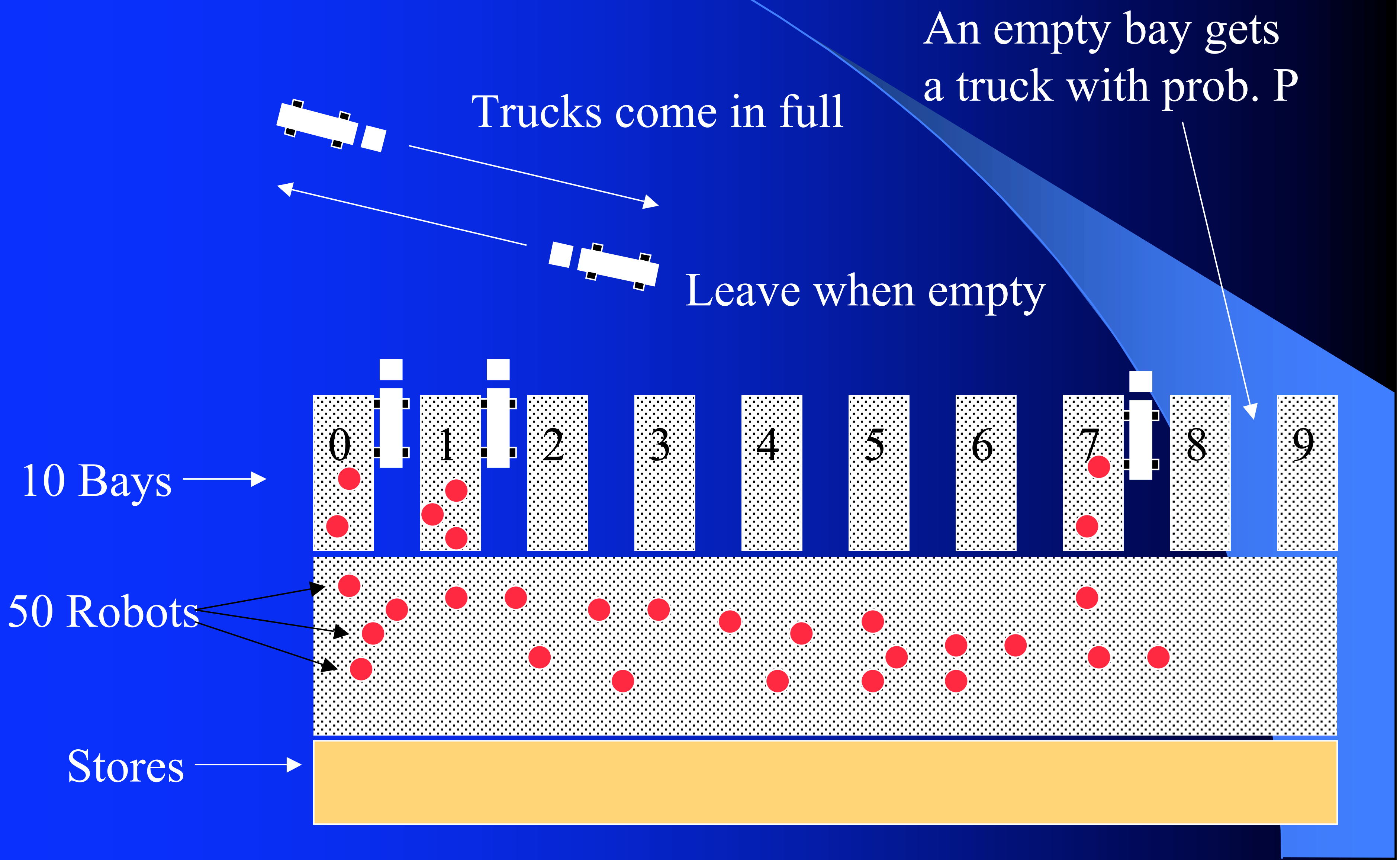
# Putting tags to work in the warehouse

- We Re-implemented a version of a warehouse scenario presented by Kalenka and Jennings (1999)
- In the original paper Selfish and Socially Rational (hard coded) robot behaviours were simulated and results compared
- The main conclusion was that socially rational agents perform better but can tolerate some degree of exploitation from selfish agents
- The problem of how socially rational behaviours could evolve was presented as an open question.

# Warehouse scenario

- 10 unloading bays – each can hold a truck
- 5 robots are assigned to each bay
- When a bay is empty trucks arrive with probability *p* and size *s (in each cycle)*
- Robots unload at a constant rate. The size *s* is proportional to the unloading time
- The time a bay remains empty is inversely proportional to *p*
- Agents (robots) are represented by triples of (*tag, L, N*) – where tag is integer [1..500] and L and N are Boolean values.

# Warehouse Scenario



# Warehouse scenario

- Robots are rewarded based on the quantity of goods *they* unload from *their* bay in each cycle
- Each bay starts empty, a truck arrives with probability *p* and leaves when fully unloaded
- In each cycle each robot can perform 5 units of unloading
- For each unit of unloading, if a robot has a truck in its bay then it asks one other robot for help in unloading – it does this by selecting a randomly selected agent with the same tag (if one exists) or a randomly selected agent from the whole population.
- If selected agent has L set and has truck in own bay then mark as potential helper
- If selected agent has N set and has no truck to unload then mark as potential helper
- For each unit of unloading, if the agent is marked as a potential helper it selected randomly and one of the agents that asked for help and helps it to unload rather than attending to it's own job.

# Outline Algorithm

```
LOOP each cycle
 LOOP 5 times
  LOOP for each robot (A)
  IF lorry in own bay THEN ask robot (B) with
    same tag (or randomly choose if no tag match)
    IF (B) has lorry in its bay THEN (B) marked as a
        potential helper with A's lorry if L is set
    ELSE (B) marked as potential helper for (A)'s lorry
        if N is set
  IF (A) marked as potential helper THEN randomly
       choose another who requested help.
  ELSE (A) unloads own lorry or sits idle
 End LOOP
 Each robot's fitness = amount unloaded in own bay
 LOOP for size of population
  Probabilistically choose a robot in proportion to fitness
  Mutate each of (tag, N, L) probability 0.1
 End LOOP
End LOOP
```

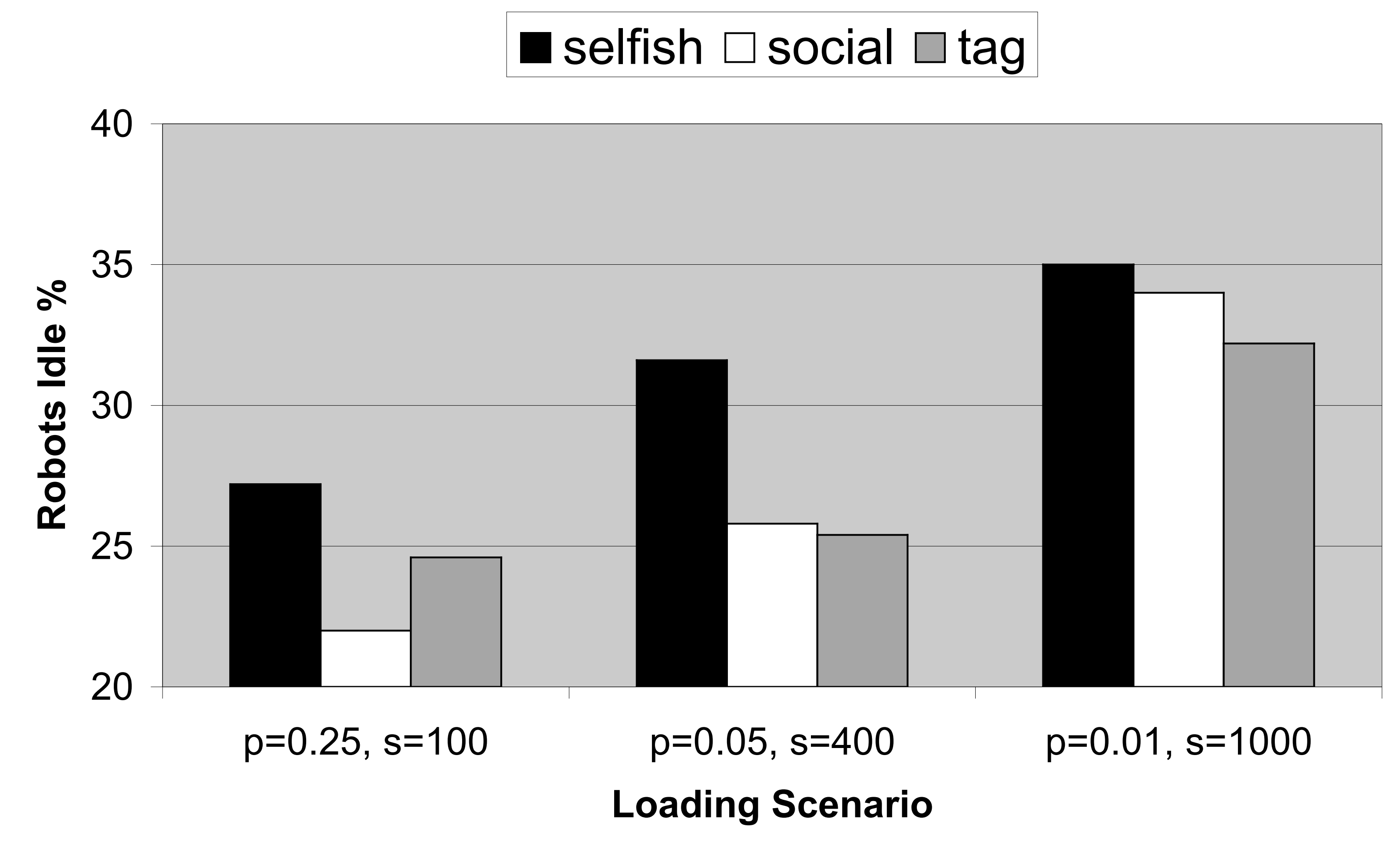
# A Comparison

- Compared this tag based algorithm to populations in which all agents were "selfish" or "social"
- Selfish agents *never* help others
- Social agents help *if they are idle* and asked
- Each simulation was run for 500 cycles (allowing each robot to unload 2500 units)
- Percentage of robot time idle was recorded
- Simulations were run over 3 different loading scenarios (values of $p$ and $s$)

# Possible Robot Strategies

| Give help when asked | Truck in own bay | Own bay empty | |
|---|---|---|---|
| | No | No | Selfish |
| | No | Yes | Social |
| | Yes | No | Altruistic |
| | Yes | Yes | Altruistic |

# Results



selfish
social
tag
40
35
30
25
20
Robots Idle %
p=0.25, s=100
p=0.05, s=400
p=0.01, s=1000
Loading Scenario

# Discussion

- The tag strategy appears to outperform the hardwired social strategy when unloading is sporadic (low *p* and high *s*)
- Speculate that the tag strategy allows (at least some) agents to abandon their own trucks when a new truck arrives in another bay
- More analysis needed to understand the dynamics and *more runs needed to increase confidence in the conclusion*
- Have the robots “self-organised” a superior solution to the hand-coded social one? How?

# Overall conclusions

- Tag models show promise but much further work required (simulation)
- Next step: Real applications need to be identified (ah hoc wi-fi – tag = frequency or protocol?).
- Current work has mainly focused on biological or social interpretations
- The "inverse scaling" and decentralised nature of tag processes – if harnessed – could produce a step-change in decentralise, adaptive applications
- But there's a lot of work to do….. See you next year ☺