

1st ESOA Workshop @ AAMAS 2003

15th July 2003, Melbourne

David Hales

<http://www.davidhales.com>

dave@davidhales.ac.uk

*Centre for Policy Modelling (CPM),
Manchester Metropolitan University.*

<http://cfpm.org>

Can Tags Build Working Systems? *From MABS to ESOA*

*Attempting to apply results gained from Multi-Agent-
Based Social Simulation (MABSS) to Engineering
Self-Organising Applications (ESOA)*

Talk Overview – 3 parts

- Part 1: A simple tag model producing cooperation in the single-round PD
- Part 2: A simple tag model demonstrating in-group specialisation
- Part 3: A tentative application of tags to a simulated “warehouse unloading” problem

What are “tags”

- Holland (1992) discussed tags as powerful “symmetry breaking” mechanism which could be useful for understanding complex “social-like” processes
- 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

Recent tag models

- Tags may be bit strings signifying some observable cultural cue (Sugarscape model, Hales MABS1998, Hales Mabs2000)
- Tags may be a single real number (Riolo, Cohen, Axelrod Nature2001)
- Earlier work by Riolo showed how tags could improve cooperation between agents playing the IPD.
- More recent work focused on how, even without memory of past interactions, tags can cause seemingly altruistic behavior between strangers

Recent tag models

- In Hales (Mabs2000) high levels of cooperation evolved using tag game biasing in the single round PD.
- In Riolo et al (Nature2001) high levels of altruistic donation evolved using a tag toleration mechanism.
- However, in both these models the agents effectively either “cooperate” or “defect”.
- In both, groups of agents sharing the same tag form cooperative groups.
- There is a dynamic formation and dissolution of such groups – groups break down when agents invade them that do not cooperate and exploit them

Tags and the Single-Round Prisoner's Dilemma

Cooperation with strangers without
reciprocity

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

Why study cooperation?

- Many hard to explain cooperative interactions in human societies
- Production of large-scale open artificial agent based systems
- More generally, how low level entities can come to form internally cooperative higher level entities

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 |
| Player 2 | C | R | T |
| | D | T | P |

Payoff values

- Temptation $T > 1$ (say, 1.5)
- Reward $R = 1$
- Punishment (P) and Sucker (S) set to small values (say, 0.0001 and 0.0002)
- Hence $T > R > P > S$ and $2R > T + S$

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

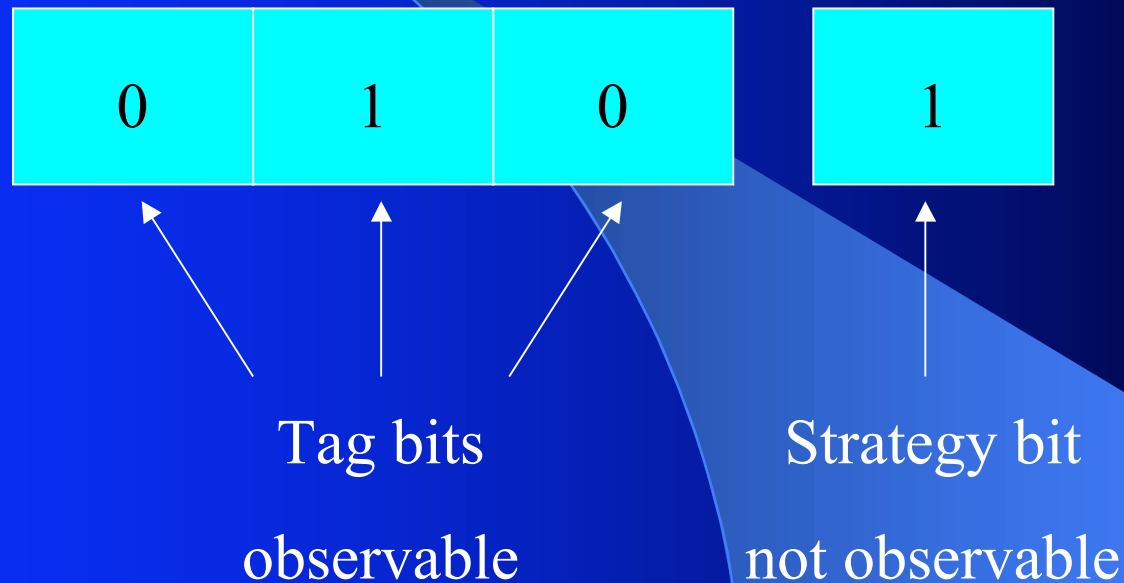
How can cooperation evolve?

- Repeated interaction when agents remember the last strategy played by opponent
- Interaction restricted to spatial neighbours
- Agents observe the interactions of others before playing themselves (image and reputation)

However, these require agents with the ability to identify individuals or have strict spatial structures imposed on interaction

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 $T = [1..2]$, $R = 1$, $P > S = \text{small}$
- Execute algorithm for 100,000 generations
- Measure cooperation as proportion of total game interactions which are mutually cooperative

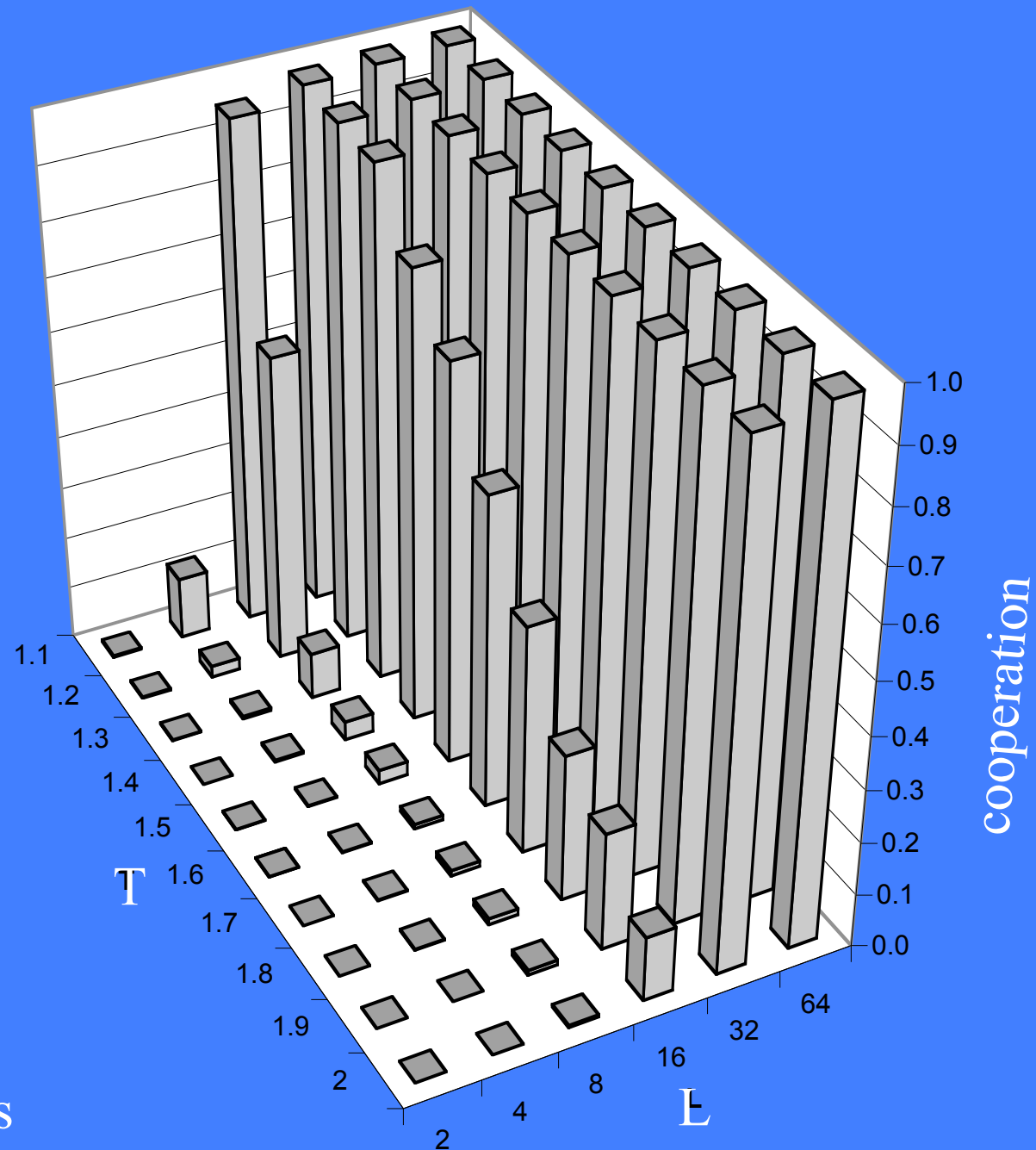
Results

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

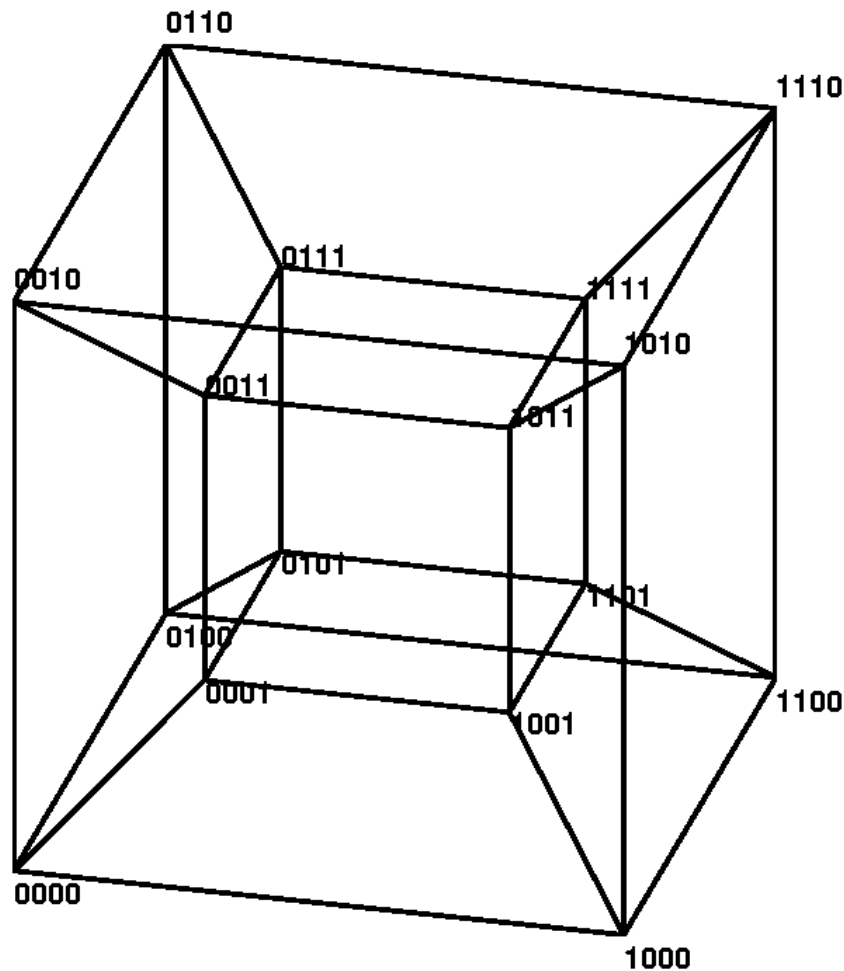
T = temptation payoff
 L = length of tag in bits



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)

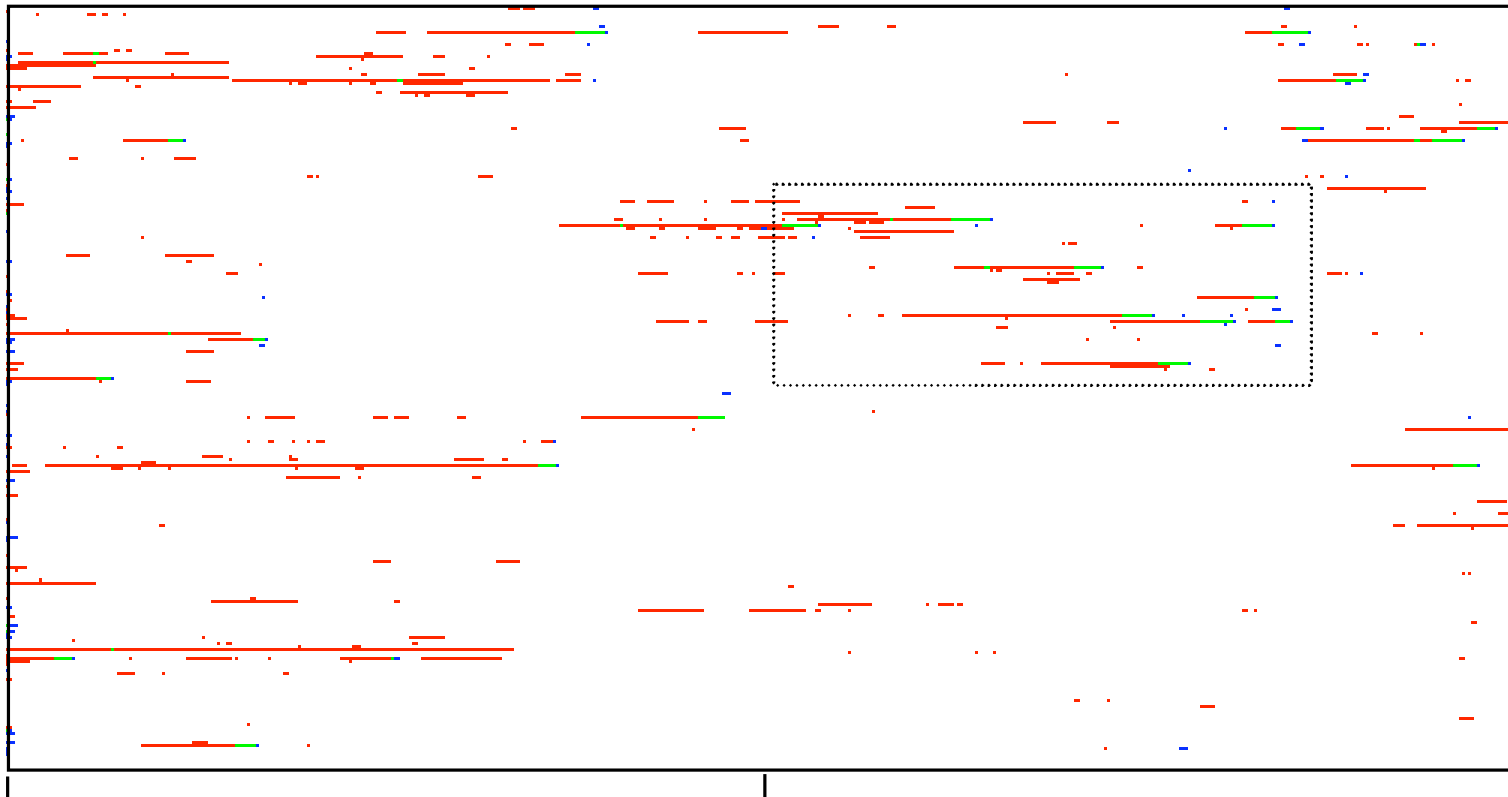
We colour each line to indicate if it is occupied by all cooperative, all defective, mixed or no agents

Visualising the process

0250500CoopDefectMixedEmpty

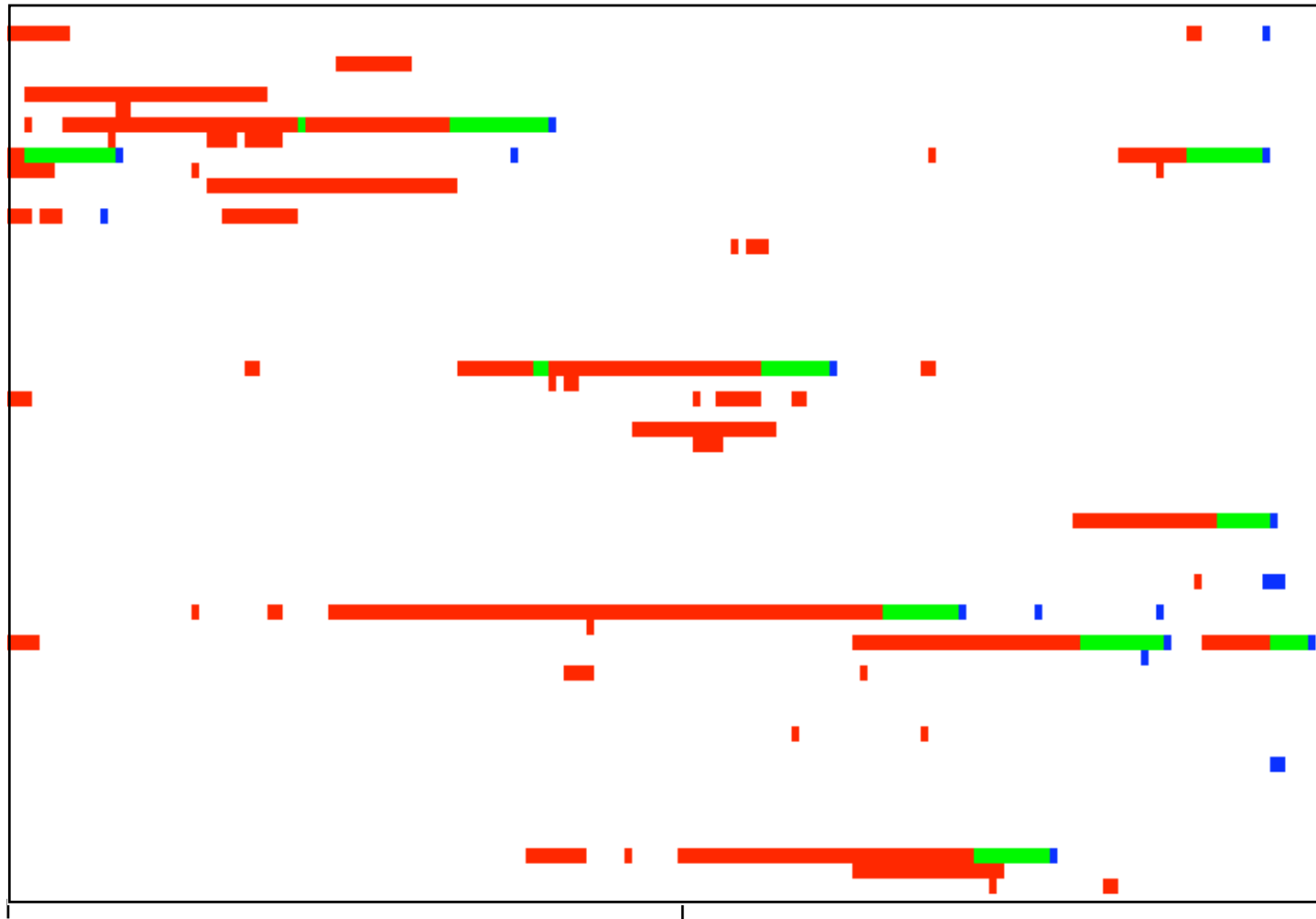


Cycles



Visualising the process

250350Cycles45 CoopDefectMixedEmpty



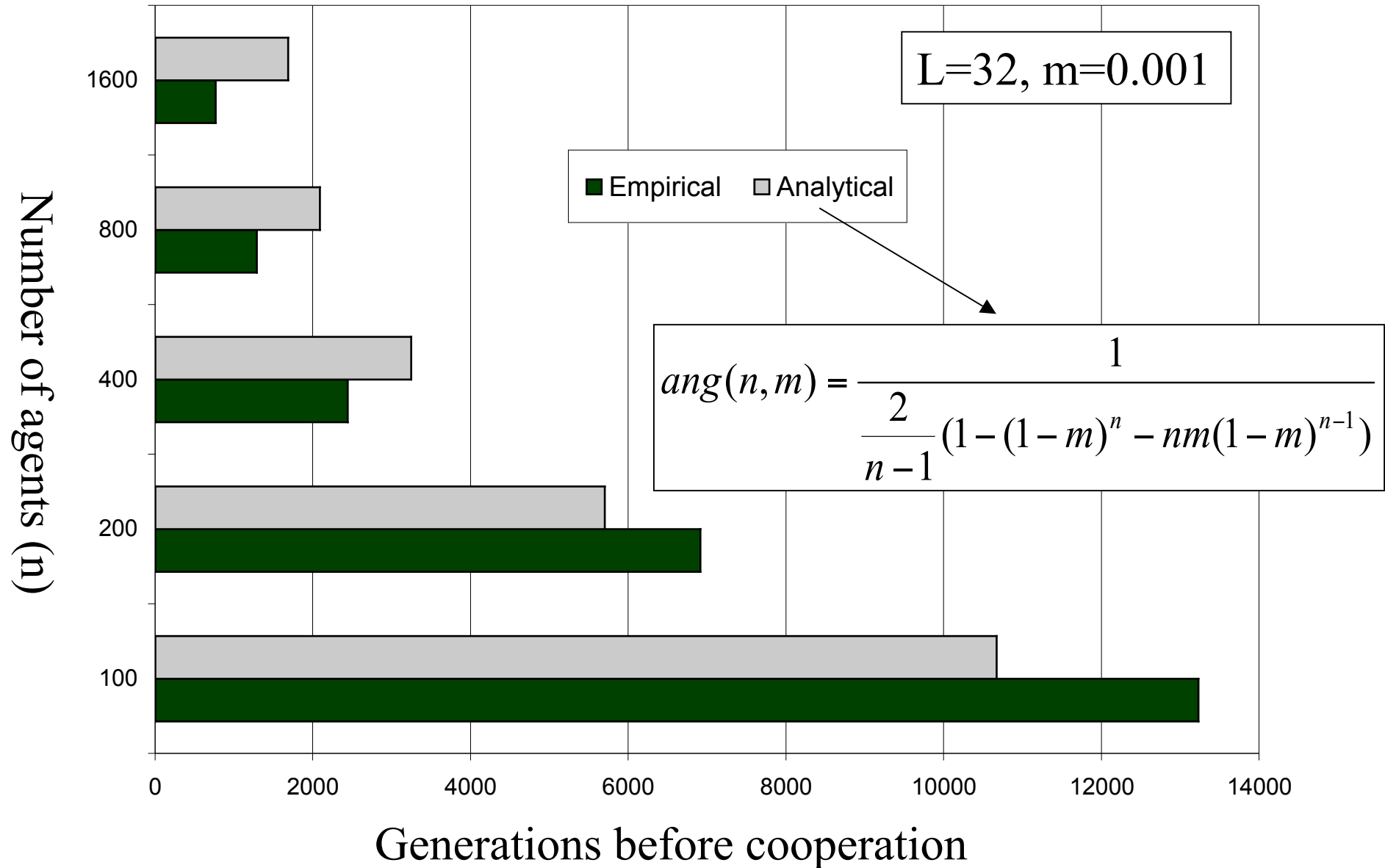
What's happening?

- 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 neighbouring 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

Part 2: Evolving Specialisation, Using Tags

Towards a kind of “group selection”

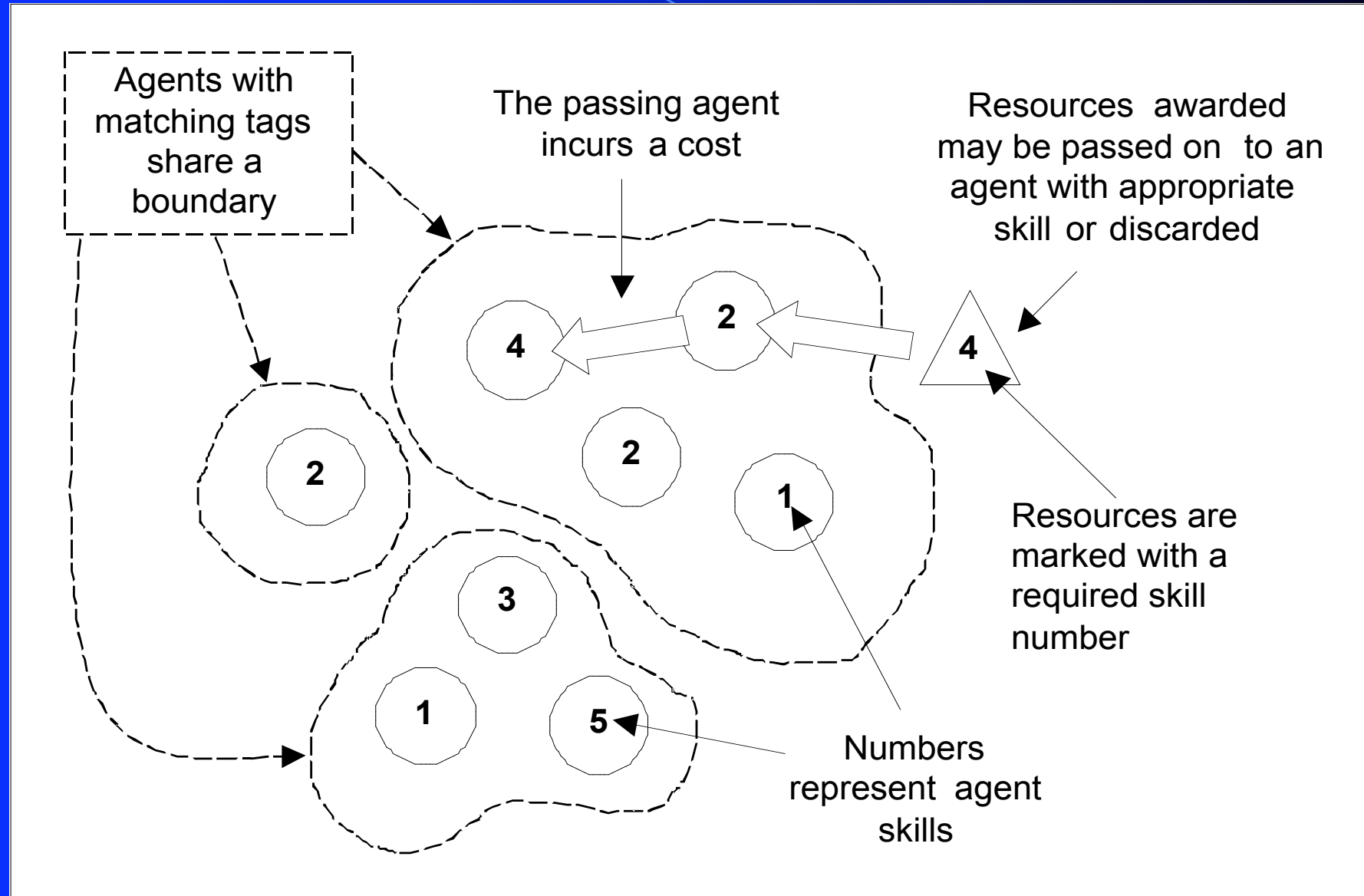
What else can tags do?

- These previous models show that cooperation can evolve in groups with tags – overcoming commons dilemmas
- But, can tags support the formation of groups in which agents perform specialised functions – supporting each other to exploit the environment as a “team” or “productive unit”
- We extended the Riolo et al model to test this

The model

- Agents consist of a tag (real number), a tolerance (real number) and a skill (integer)
- Each agent is awarded some of resources in each cycle.
- Resources associated with randomly selected skill
- An agent can only “harvest” a resource matching it’s own skill
- If it can not harvest the resource, it may donate the resource to another agent (if it can find one) that matches its tag

The model



The model

- An agent is considered to “match” the tags of another if the difference between the tag values is no more than the tolerance value
- So a high tolerance means “donate to any agent” and a low tolerance means “only donate to those with similar tag value”
- When an agent attempts to make a donation it selects another agent from the population compares tags for a match and then passes the resource if the receiving agents has the required skill value

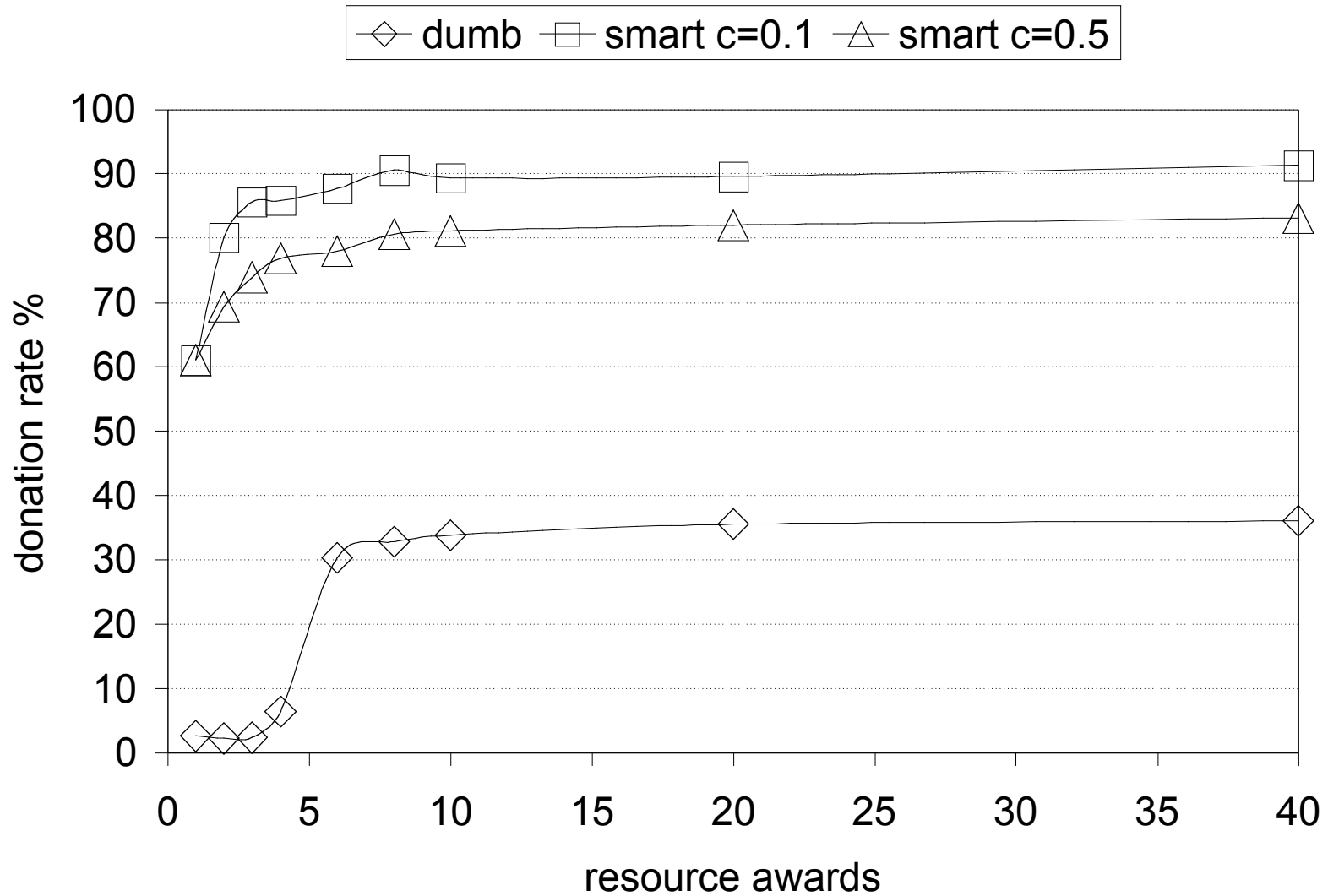
The model

- In the initial model, there are 2-skills, 100 agents, partner selection involves a single random selection from the population
- When agents make a successful donation they incur an energy cost (0.1)
- When an agent successfully harvests a resource it gets a unit of energy (1)
- After each cycle a tournament selection process based on energy, increases the number of successful agents (high energy) over those with low energy
- When successful agents are copied, mutation is applied to both tag, tolerance and skill

What will the results tell us?

- *If the donation rate (over time) is non-zero, then we can conclude that:*
- Agents are forming tag groups with a diversity of skills
- Agents are behaving altruistically, since donation produces immediate costs but does to produce immediate returns
- Therefore agents (from a myopic individual bounded rationality) form internally specialised altruistic teams

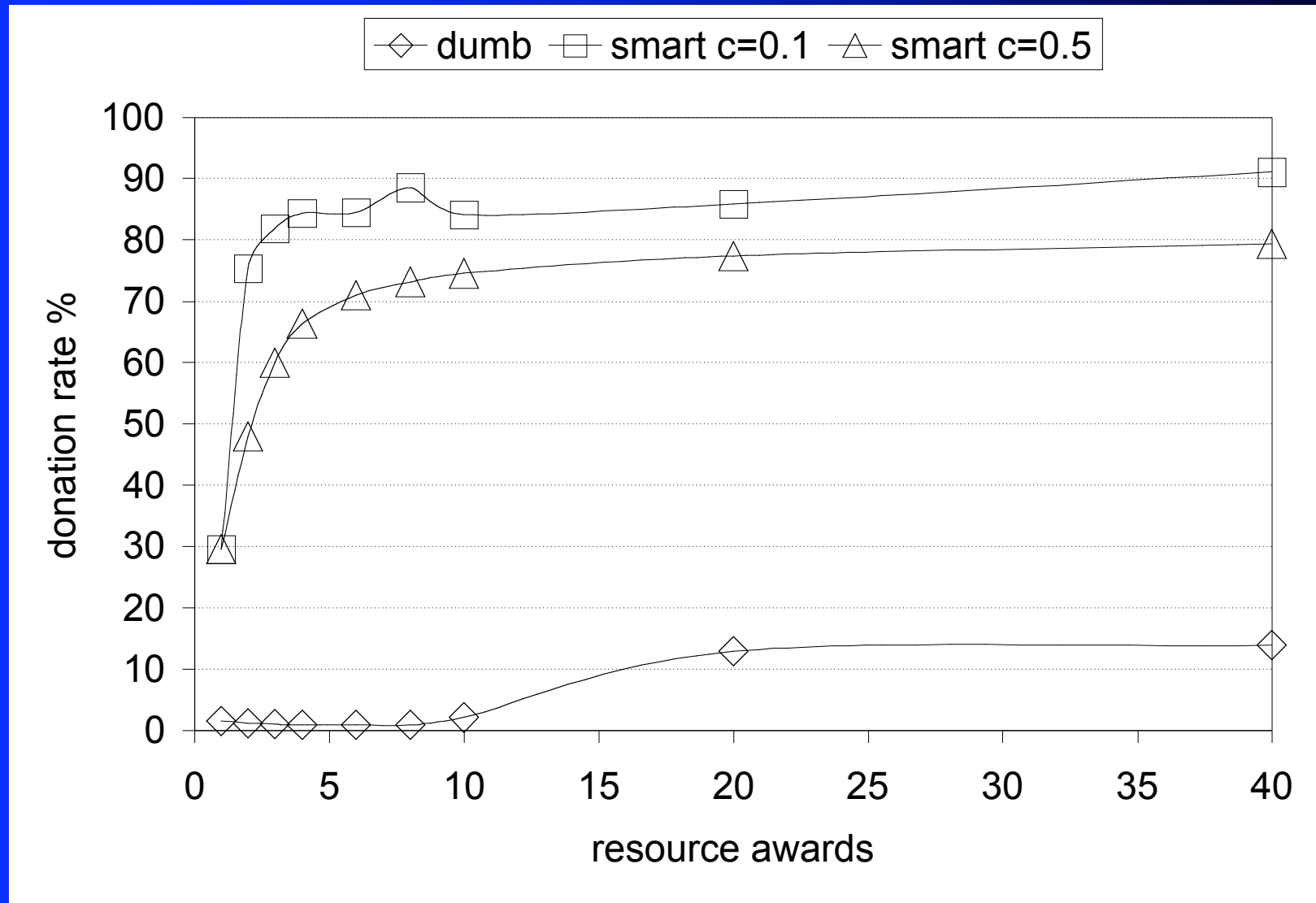
2-skills, averages of 30 runs to 30,000 generations



Results – what does it mean?

- A significant level of donation – confirming specialisation and altruism (of a sort!)
- But not so high, if we instead of selecting potential donation partners at random we use a “smart” matching method then significant increases in the donation rate are seen (previous slide)
- This smart matching can even support higher donor costs

5-skills, averages of 30 runs to 30,000 generations



Results – what does it mean?

- The random (or dumb) matching goes lower
- The smart matching goes lower too but still stays high and recovers quickly as the number of resource awards increases
- Hence, it would seem that to support a higher degree of specialisation (more skills) smart matching is required

Conclusions

- Agents form groups based on tag similarity, containing diverse skills, donating resources to between each other, to efficiently exploit the environment – for the good of the group
- This happens even though individuals are selected on the basis of their individual utility
- Can such models help us to understand how early social groups formed with specialised roles?
- Group distinguishing abilities (smart searching) would appear to be important
- Future work: does smart searching evolve (see Hales 2002 - yes)? What about putting agents in social networks = smart is cheap?
- The Tag Clone issue! What are we really seeing here (see Hales 2003 – forthcoming JASSS special issue)?

Part 3: Evolving “Social Rationality” in MAS using Tags

Tentative application to a simulated
MAS

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 “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).

Warehouse scenario

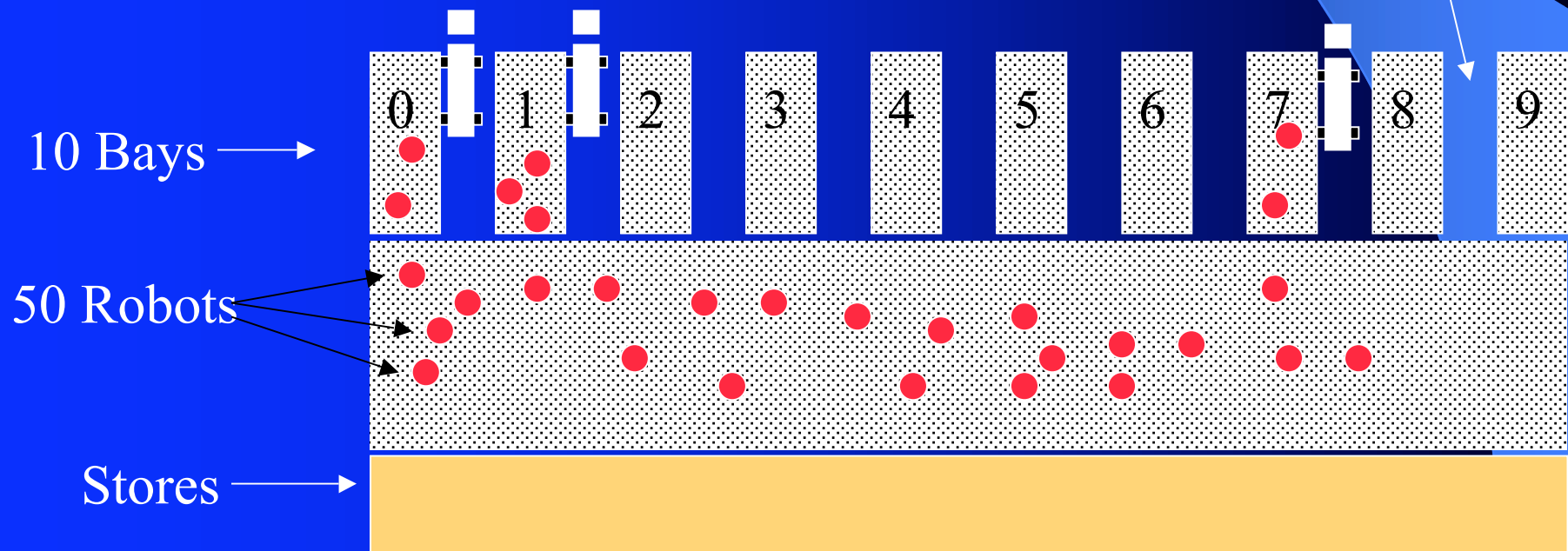
- 10 unloading bays – each can hold a truck
- 5 robots are assigned to each bay
- When a bay is empty trucks arrive with probability p and size s (*in each cycle*)
- Robots unload at a constant rate. The size s is proportional to the unloading time
- The time a bay remains empty is inversely proportional 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

An empty bay gets a truck with prob. P

Trucks come in full

Leave when empty



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)

Robot personalities

Give help
when asked

Truck in
own bay

Own bay
empty

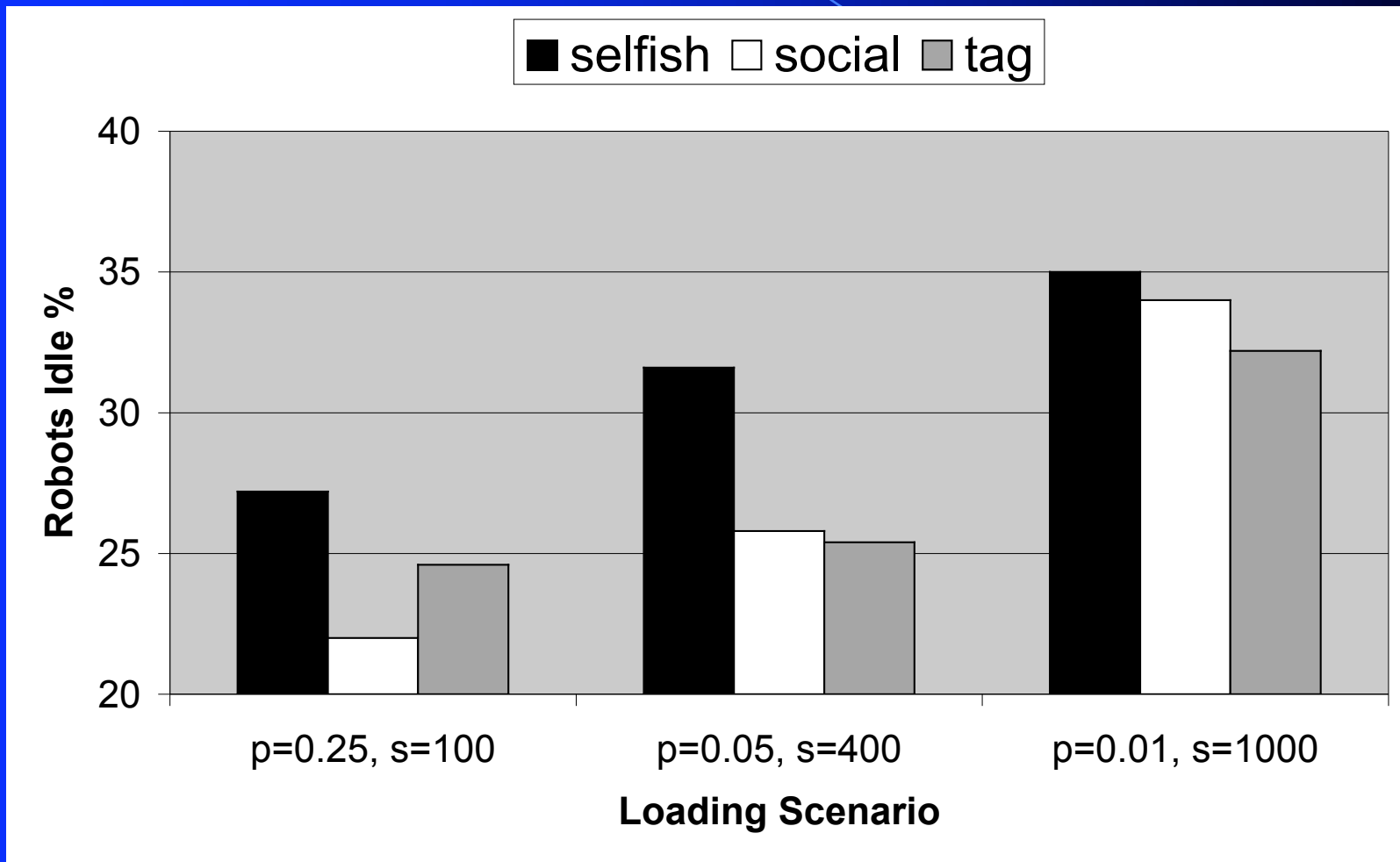
| | |
|-----|-----|
| No | No |
| No | Yes |
| Yes | No |
| Yes | Yes |

Selfish

Social

Altruistic

Results



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 – which could help
- More analysis needed to understand the dynamics and more runs needed to confirm the conclusion
- Have the robots “self-organised” a superior solution to the hand-coded social one?

Overall conclusions

- Tag models show promise but much further work required (simulation)
- Network applications need to be identified
- 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.....