DELIS
Dynamically Evolving, Large-scale Information Systems

Recent directions in DELIS / Overview of on-going work

David Hales
www.davidhales.com

Department of Computer Science
University of Bologna
Italy

Project funded by the Future and Emerging Technologies arm of the IST Programme

What's DELIS?

- Dynamically Evolving Large Scale Information Systems (DELIS)
- A four year EU funded Integrated Project (IP) of Framework Program 6 (FP6) within the Future and Emerging Technologies area (FET)
- 19 Partners across EU
- Bologna: Biologically and Socially inspired mechanisms (self-healing, scalable, robust)
- Running for 1 year now

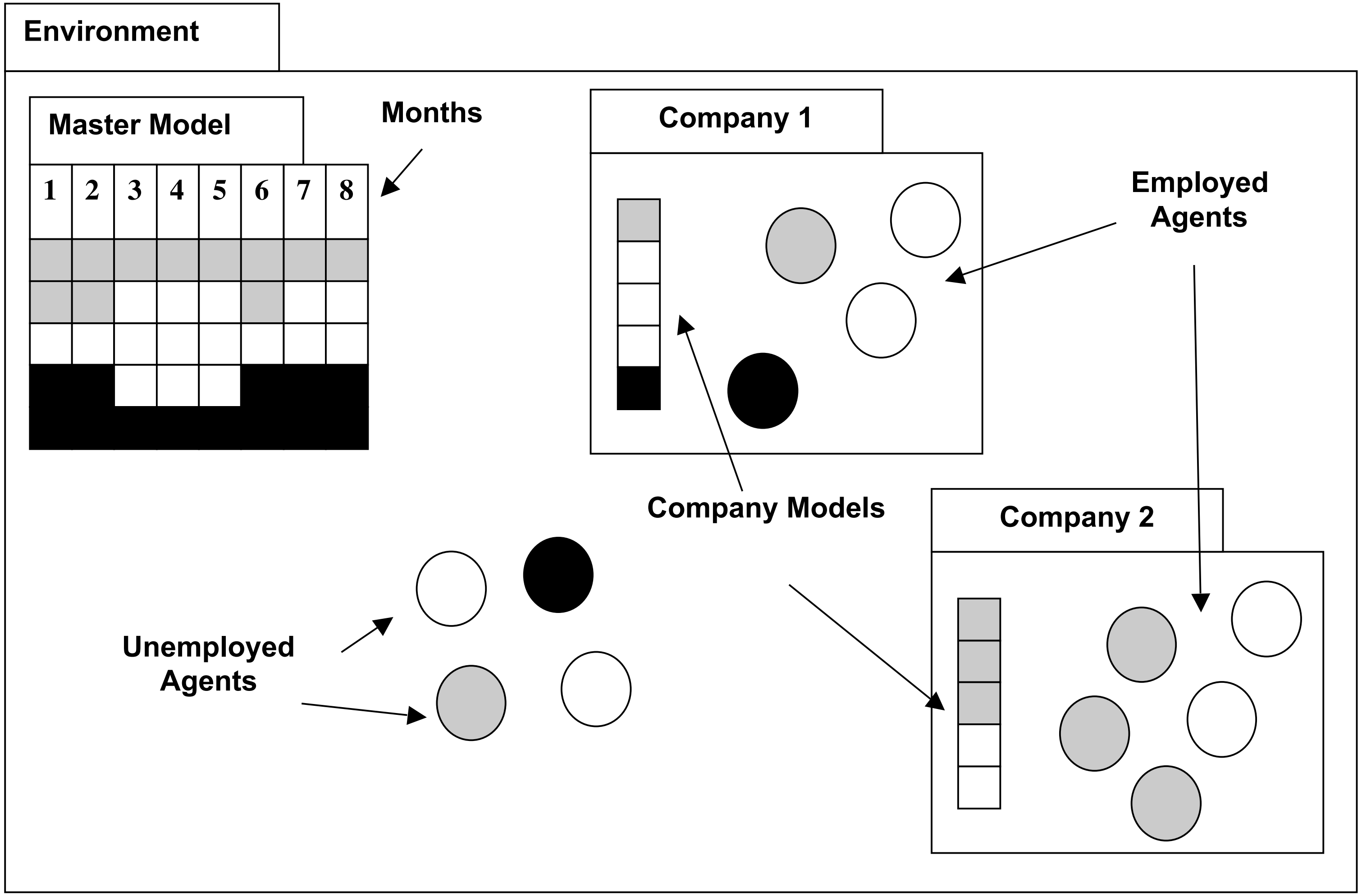
Recent directions in DELIS

- In collaboration with UPF (Sole et al)…
- Analysis of “natural” or “found” networks:
  - Biological, software, neural
- Interestingly, found “duplicate and rewire” algorithms that reproduce distributions
- “motif analysis” of functional artificial networks such as Newscast, ERA, SLAC
  - Statically and dynamically
- Potential to link empirical / scientific with engineering / functional approaches

Recent directions in DELIS

- Working with Edoardo Mollona at Bologna
- Agent-based computational economics (ACE)
- An artificial economy where “firms” recruit “workers” (with various skills) and compete in a market
- Successful firms (high-profit) copied by unsuccessful firms (evolutionary process)
- Conventional, classical, economics very limited treatments – e.g. learning, different skills,
- Agent-based model -> new ideas
- For me, implications for distributed systems design (e.g. “firms” as nodes, “workers” as agents with certain skills)

FirmWorld

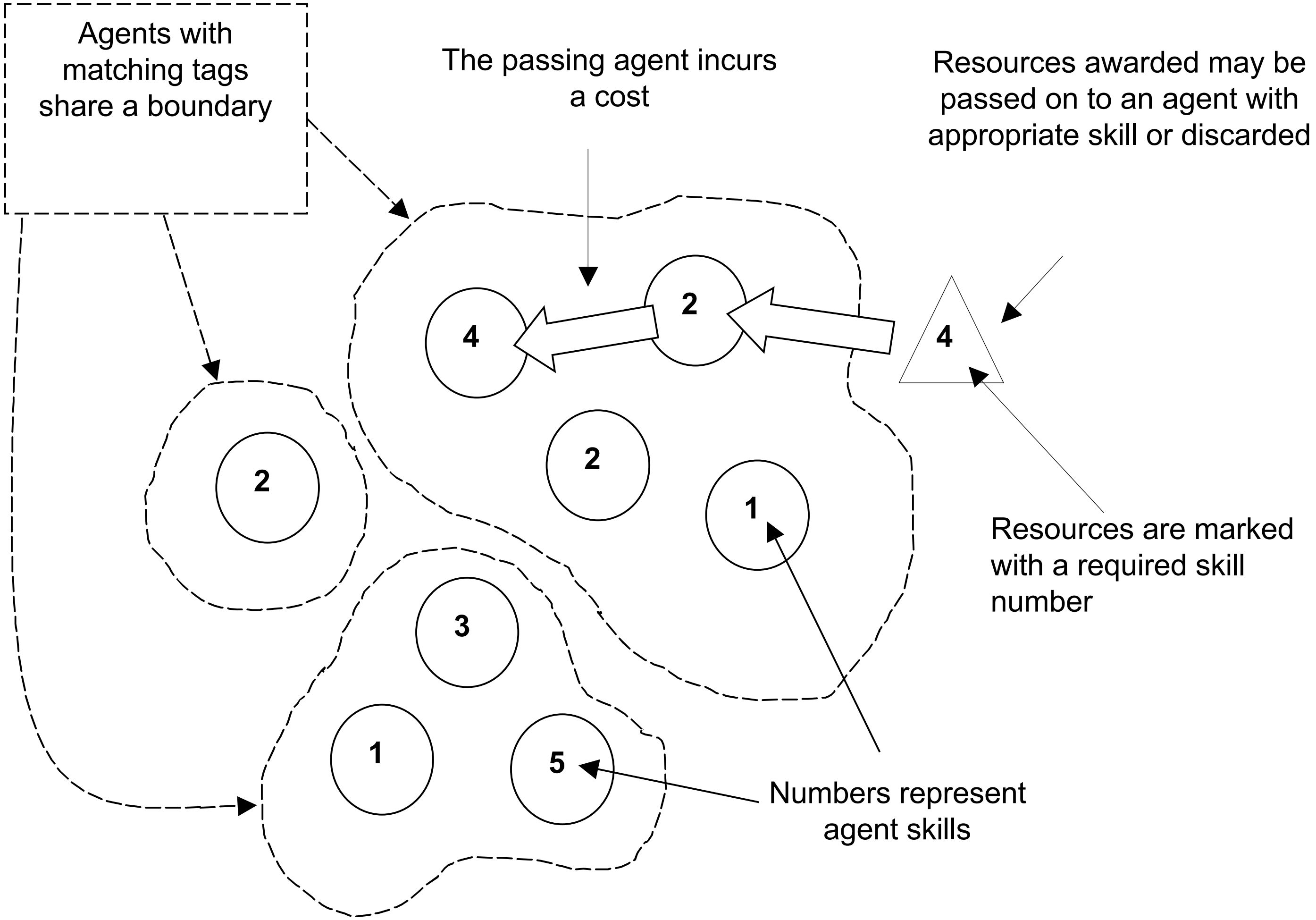


Environment
Master Model
1
2
3
4
5
6
7
8
Months
Company 1
Employed
Agents
Company Models
Company 2
Unemployed
Agents

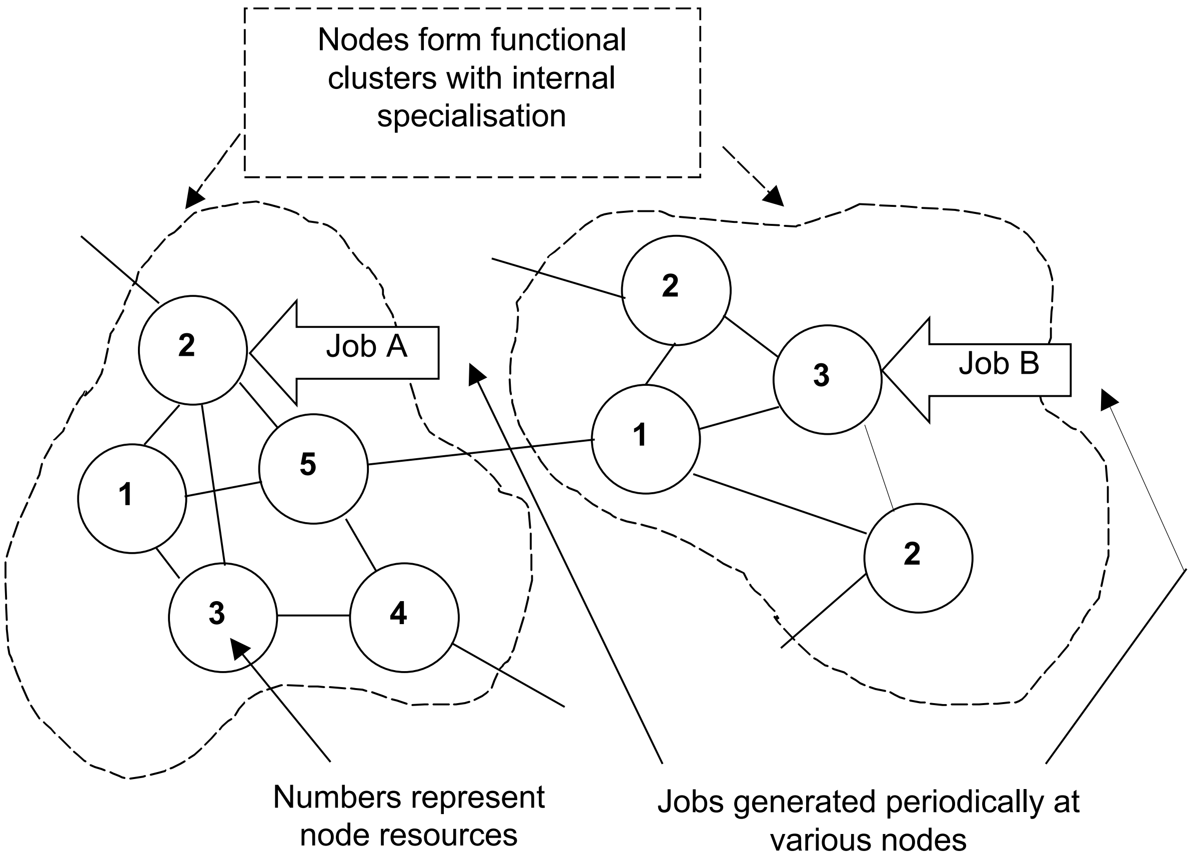
Specialisation

- Often, node specialisation in clusters produces more optimal behaviour (e.g. supernodes for example – see Alberto's papers)
- Are there "general" mechanisms that can generate this based on "*self-interest*" and "*local-behaviour*" of nodes – dynamically.
- Previous (mean-field) type tag based models exist (from social simulation – "tribes")
- Are they translatable into P2P type networks?

Specialisation in “tribal” model



Specialisation in a network



Evolutionary Re-wiring Algorithm

- Periodically:
- Compare utility with another node
- If other node has higher utility
  - copy its behaviour and links

Andrea Marcozzi – has put this “on top” of Newscast and can reproduce high cooperation in a PD game

The Prisoner's Dilemma

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

Player 2 \ Player 1	C	D
C	R, R	T, S
D	S, T	P, P

The Prisoner's Dilemma

- This is a "minimal form" of a "Commons Tragedy" (Hardin 1968).
- The "rational" game theoretic solution (the "Nash" equilibrium – is to defect)
- Selfish adaptive / evolutionary units would also tend to Nash
- It is desirable for "societies" to maintain at least some level of cooperation in such situations and many seem to. But how?

Maintaining Cooperation in the PD

- Binding Agreements (3’rd party enforcement) – expensive, complex, tends to centralisation (Thomas Hobbes 1660)
- Repeated Interactions so can punish defectors – requires enough repeated interactions and “good guys” at the start (Axelrod 1984)
- Fixed spatial relationships – lattice or fixed networks – not good with dynamic networks (Nowak & May 1992)
- **Tags – scalable, single round, simple (Holland 1993, Riolo 1997, Hales 2000)**

Tags – New and Novel Mechanism for Cooperation

A little detail on a previous tag model Hales (2000, 2004).

What are Tags?

- Visible and changeable markers attached to agents (e.g. dress style, accent, hair-style)
- If agents preferentially mix with those sharing same tags
- Distinct groups are formed - By excluding those without the same tags
- By changing tags agents move between groups
- Membership of some groups may be more desirable than others

Evolving Tags

If we assume (evolutionary process):

- Strategies and tags of agents obtaining high credit tend to get copied
- Periodically agents randomly mutate tag and strategy bits
- Result is all defection – since a defector never gets less credit from an interaction than its partner (ESS and Nash)

Evolving Tags

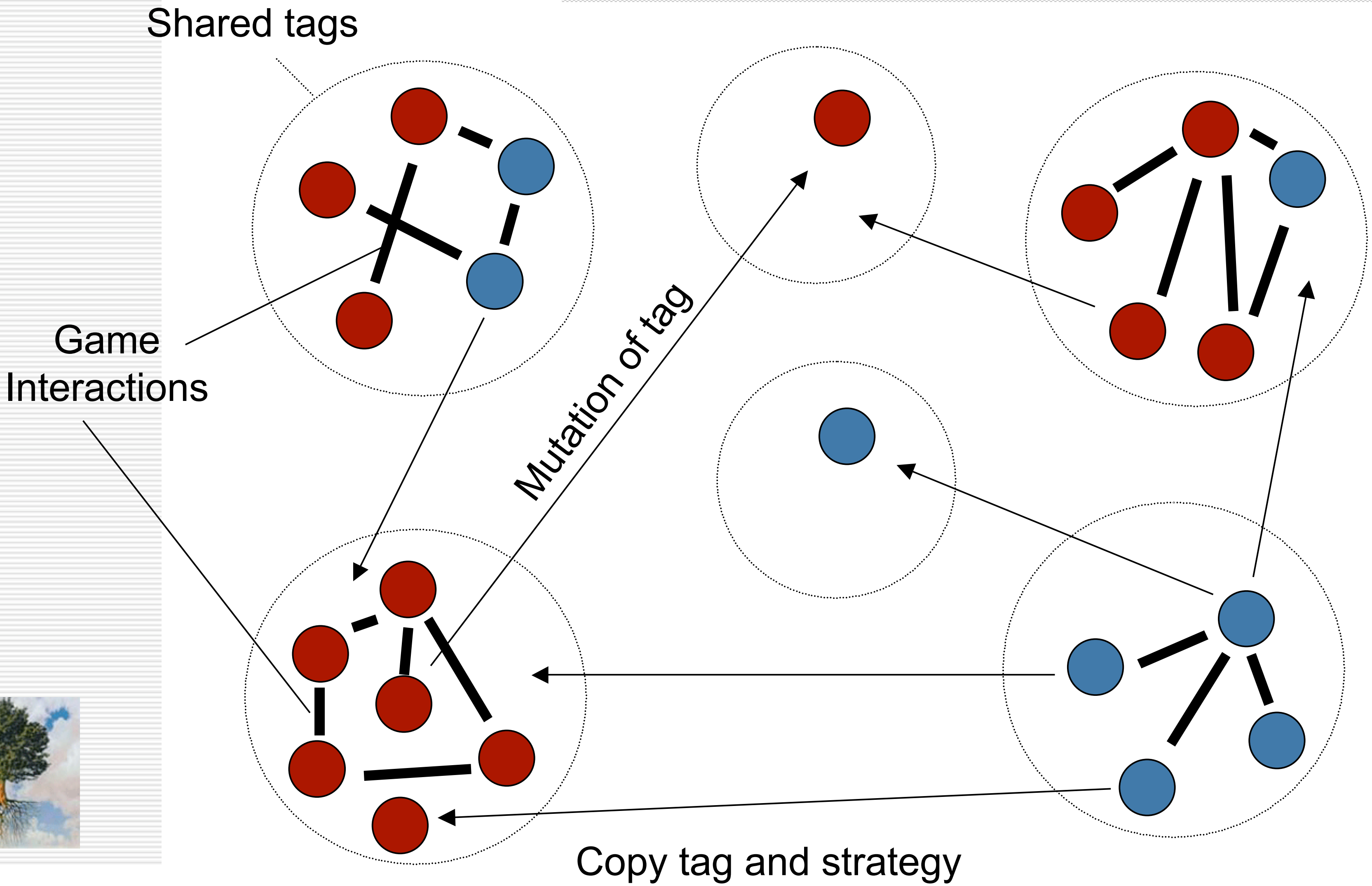
- But if we bias partner selection to those with matching tags (if any exist)
- We get unstable yet high levels of cooperation
- A dynamic group formation and dissolution process
- Tags mutate and are copied like strategies (but with a *higher mutation rate*)

Agents - a Tag and a PD strategy

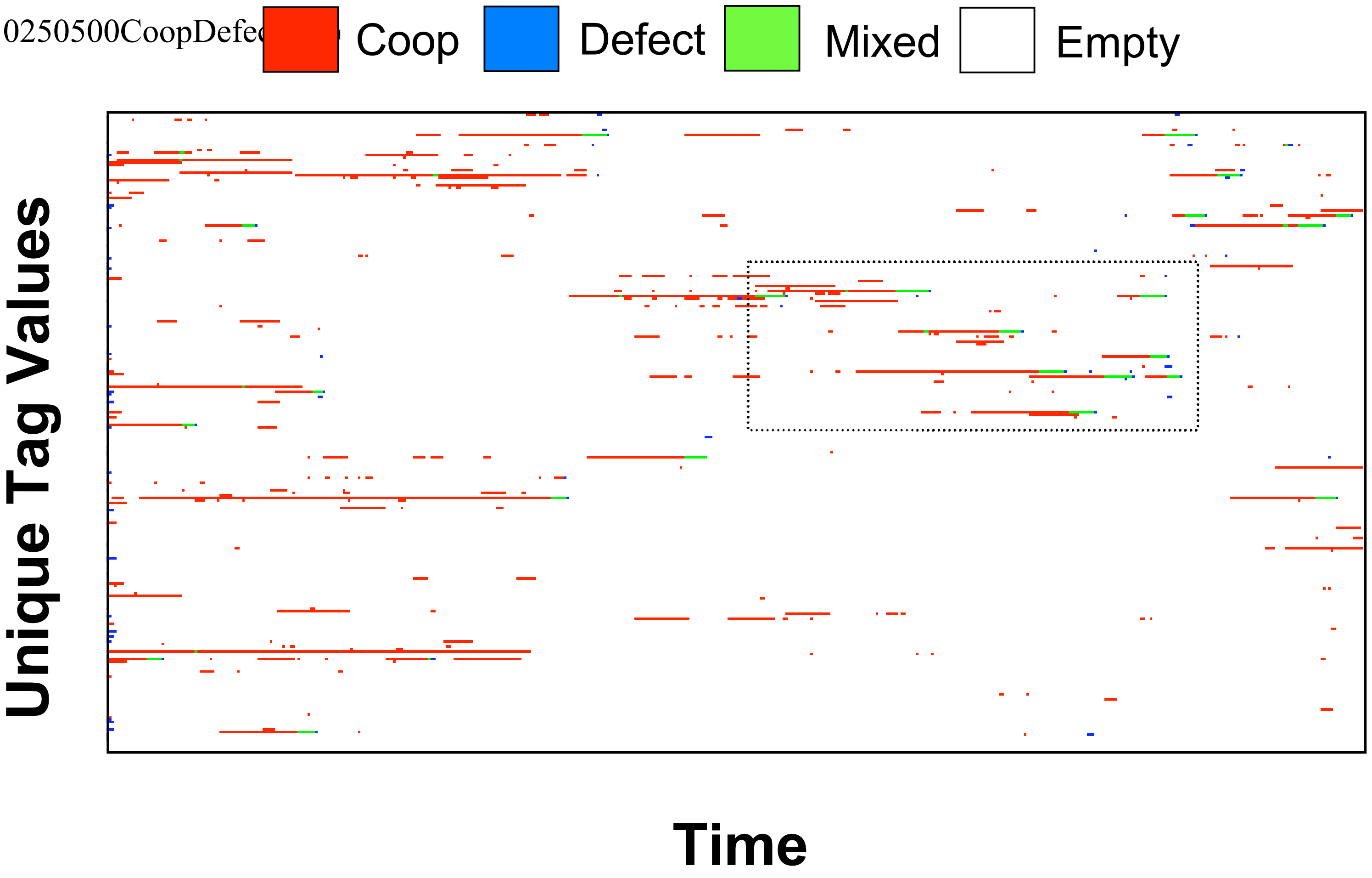
Tag = (say) Some Integer

Game interaction between those with same tag
(if possible)

How Tags Work

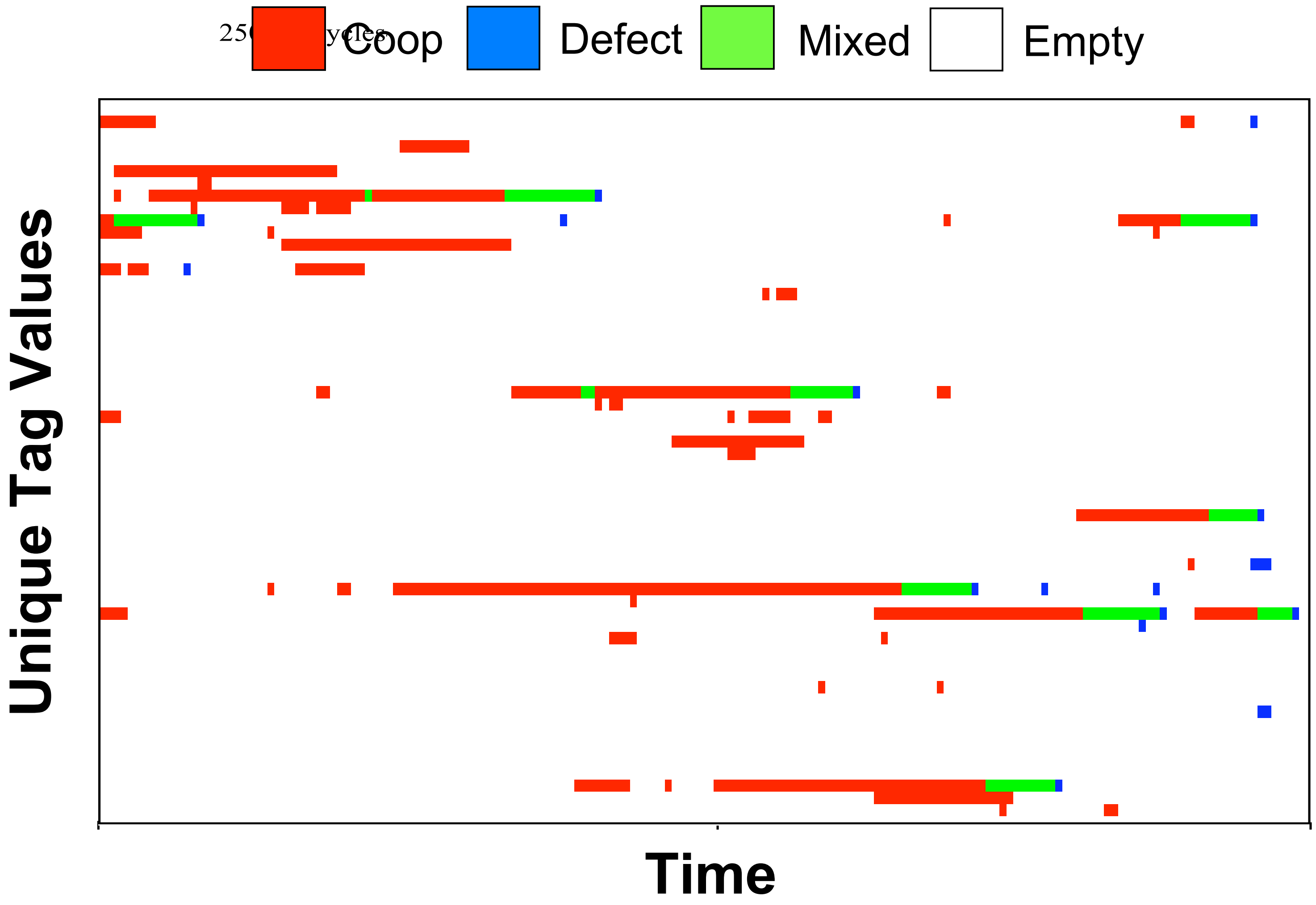


Visualising the Process (Hales 2000)



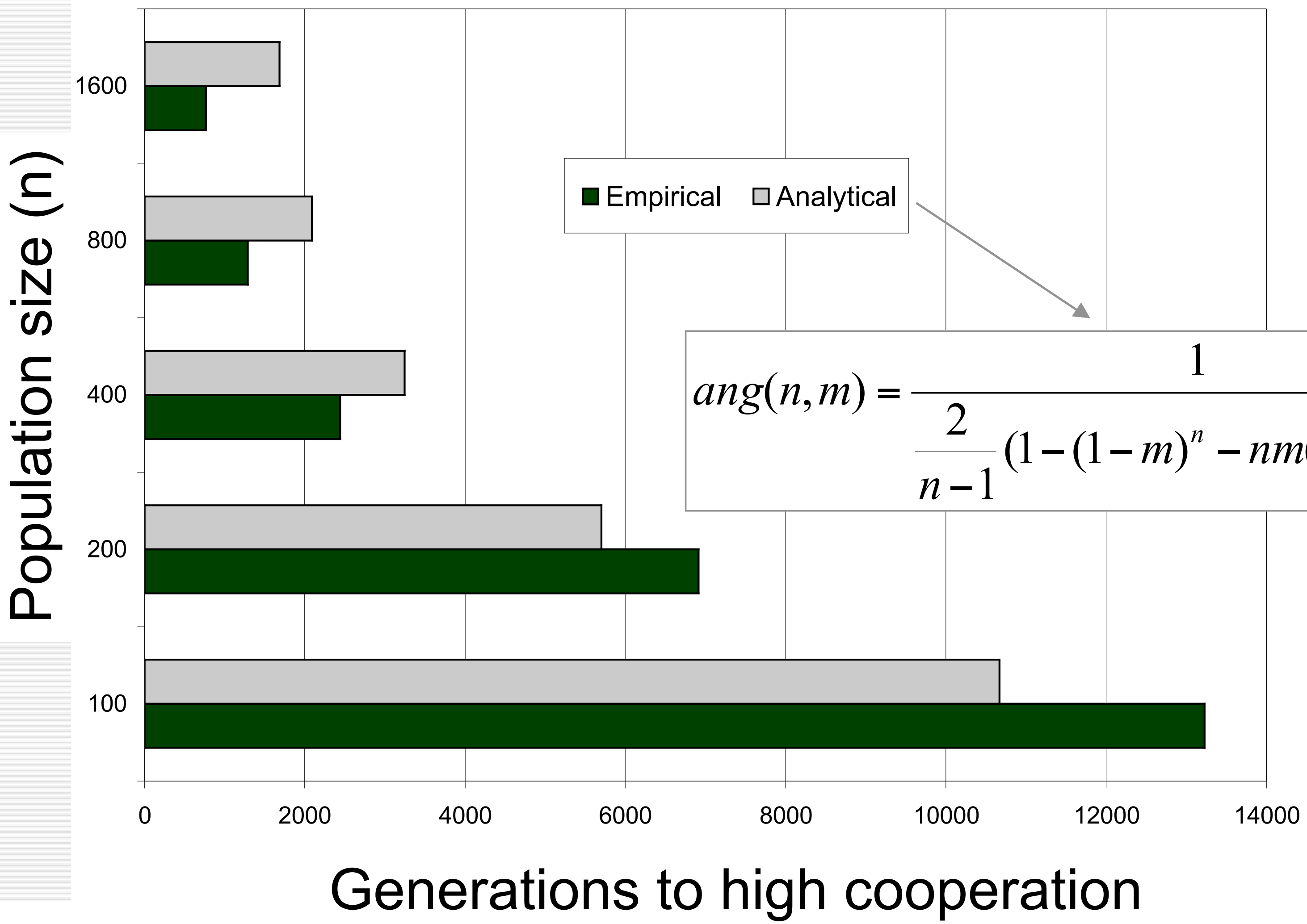
DELIS
Dynamically Evolving, Large-scale Information Systems

Visualising the Process



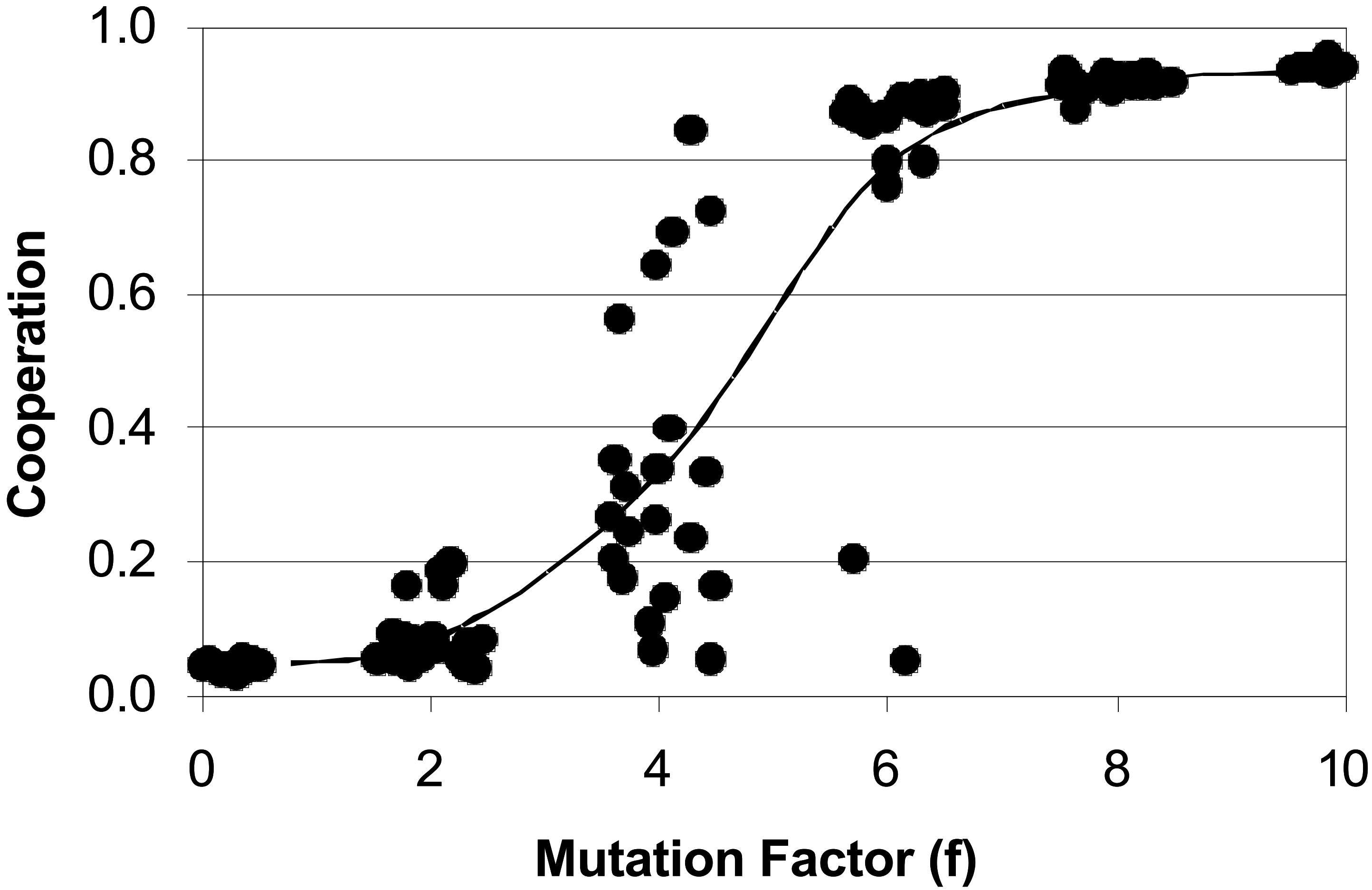
DELIS
Dynamically Evolving, Large-scale Information Systems

A Reverse Scaling Property



$$ang(n,m) = \frac{1}{\frac{2}{n-1}\left(1-(1-m)^n - nm(1-m)^{n-1}\right)}$$

Recent finding (Hales 2004) – tag mutation rate needs to be higher



Translating Tags into a P2P Scenario

All well and good, but can these previous results be applied to something like looks more like: unstructured overlay networks with limited degree and open to free riders

Project funded by the Future and Emerging Technologies arm of the IST Programme

A P2P Scenario

Consider a P2P:

- Assume nodes maintain some max. no. of links
- Node neighbours can be thought of as a group
- Nodes may be good guys, share resources with neighbours, or free-ride, using neighbours resources but not sharing theirs (PD)
- Sharing / free-riding is a Strategy
- The neighbour links (as a whole) a kind of “tag” (if clustering high enough)

A P2P Scenario

- Represent the P2P as a undirected graph
- Assume nodes are selfish and periodically:
  - Play PD with RND selected neighbour
  - Compare performance to some randomly selected other node
  - If other node is doing better copy its neighbourhood and strategy
  - Mutate strategies and neighbourhood.

Initial thoughts and questions

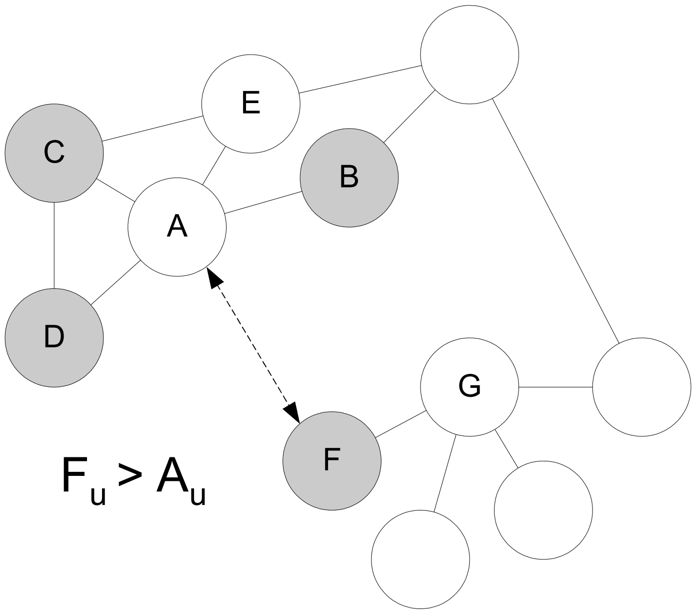
- For tag-like dynamics high clustering would appear to be required (groups required)
- Will dynamic nature of the scenario support this?
- Can cooperation be maintained without it?
- We might start simulations of the model with high clustering initially (say small world or lattice) and compare that to random networks
- Many schemes of “neighbourhood copying and mutation” are possible which to use?
- What kind of topologies emerge over time?

Design Decisions

- Mutation of neighbourhood = replace all neighbours with a single neighbour chosen at random from the population
- Mutation on strategy = flip the strategy
- Node j copying a more successful node i = replace i neighbourhood with j's U j itself
- When maximum degree of node is exceeded throw away a randomly chosen link
- Payoffs as before: T=1.9, R=1, P=d, S=d

Social Climbing, Ostracism, Replication

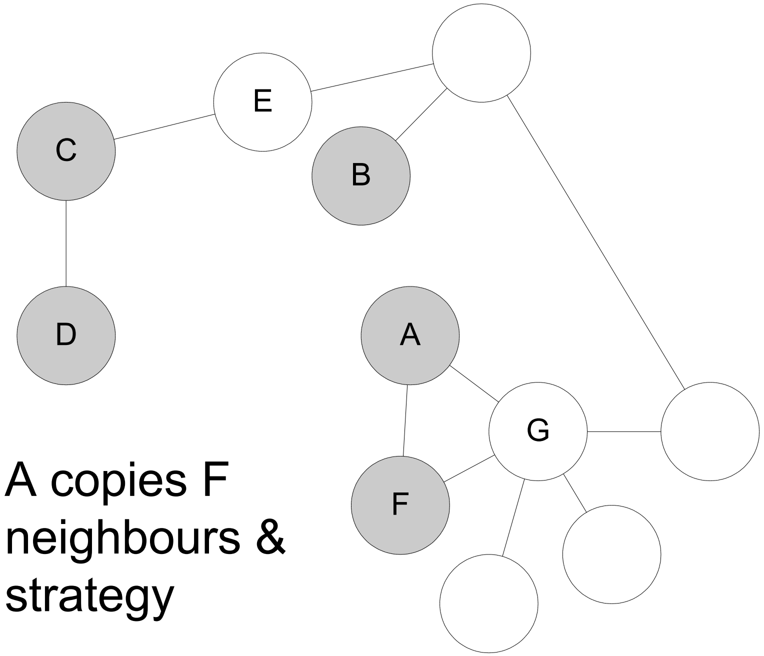
Before



$F_u > A_u$

Where A_u = average utility of node A

After

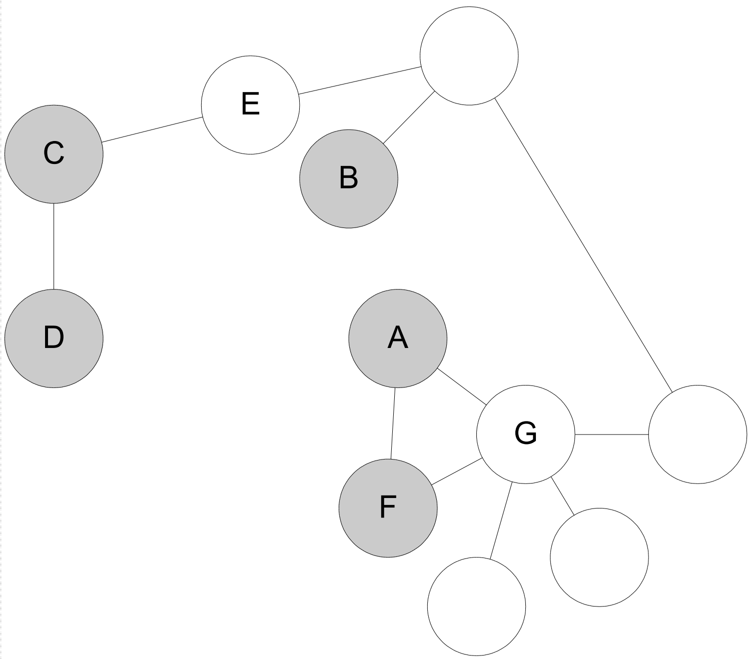


A copies F neighbours & strategy

In his case mutation has not changed anything

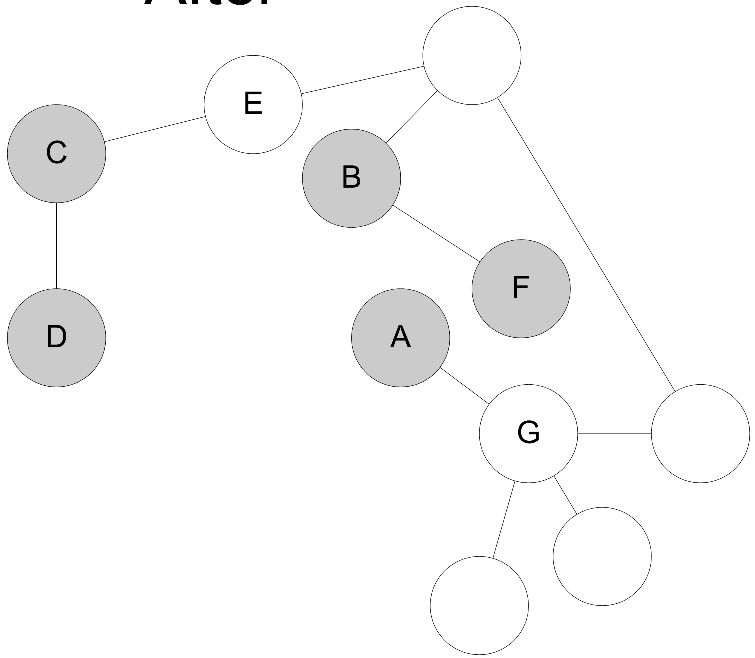
Mutation on the Neighbourhood

Before



Mutation applied to F's neighbourhood

After



F is wired to a randomly selected node (B)

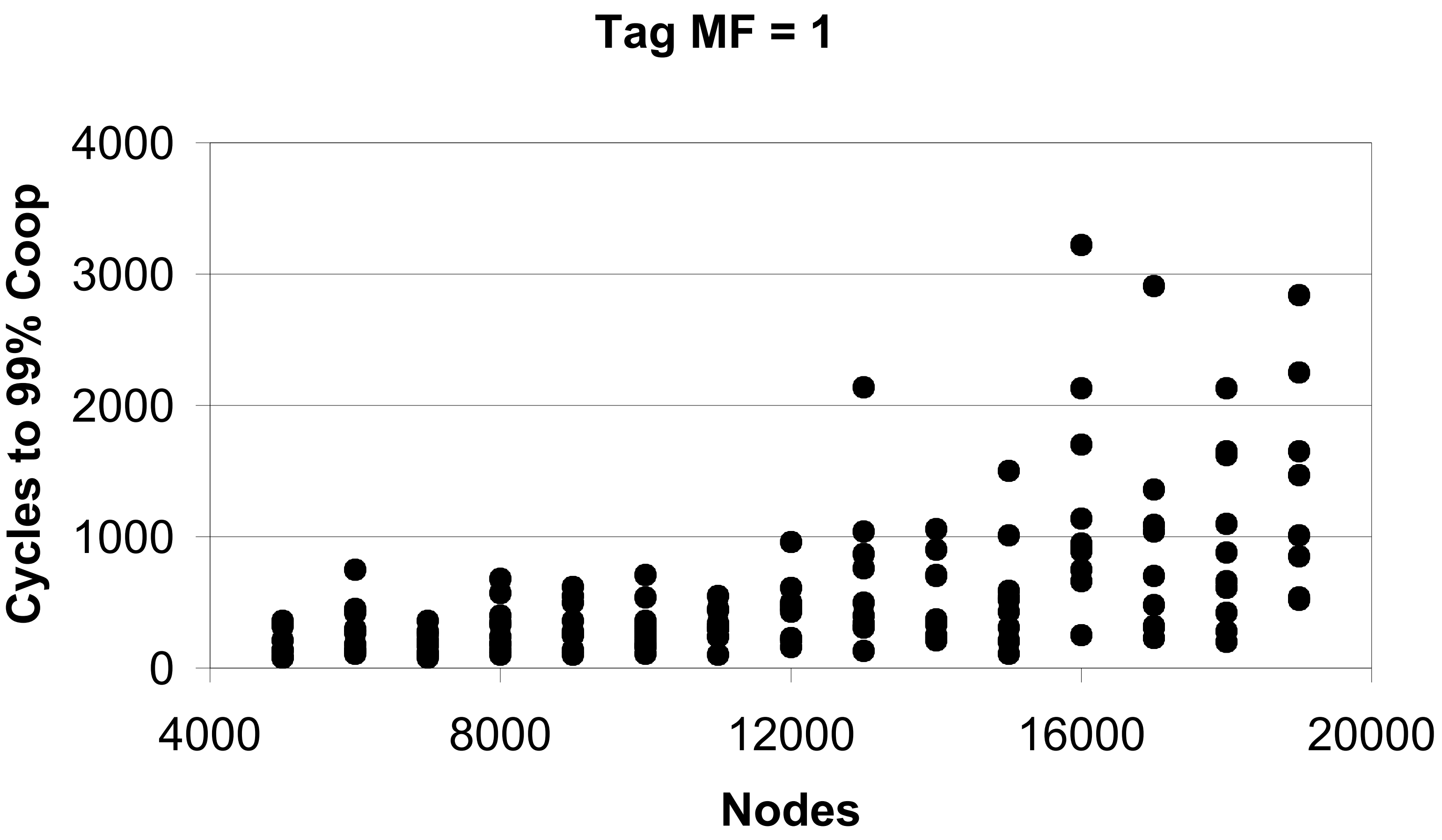
The Simulation Cycle

```
LOOP some number of generations
    LOOP for each node (i) in the population N
        Select a game partner node (j) randomly from
          neighbour list
        Agent (i) and (j) invoke their strategies and get
          appropriate payoff
    END LOOP
    Select N/2 random pairs of agents (i, j) reproduce
      higher scoring agent
    Apply mutation to neighbour list and strategy of each
      reproduced agent with probability m
END LOOP
```

Parameters

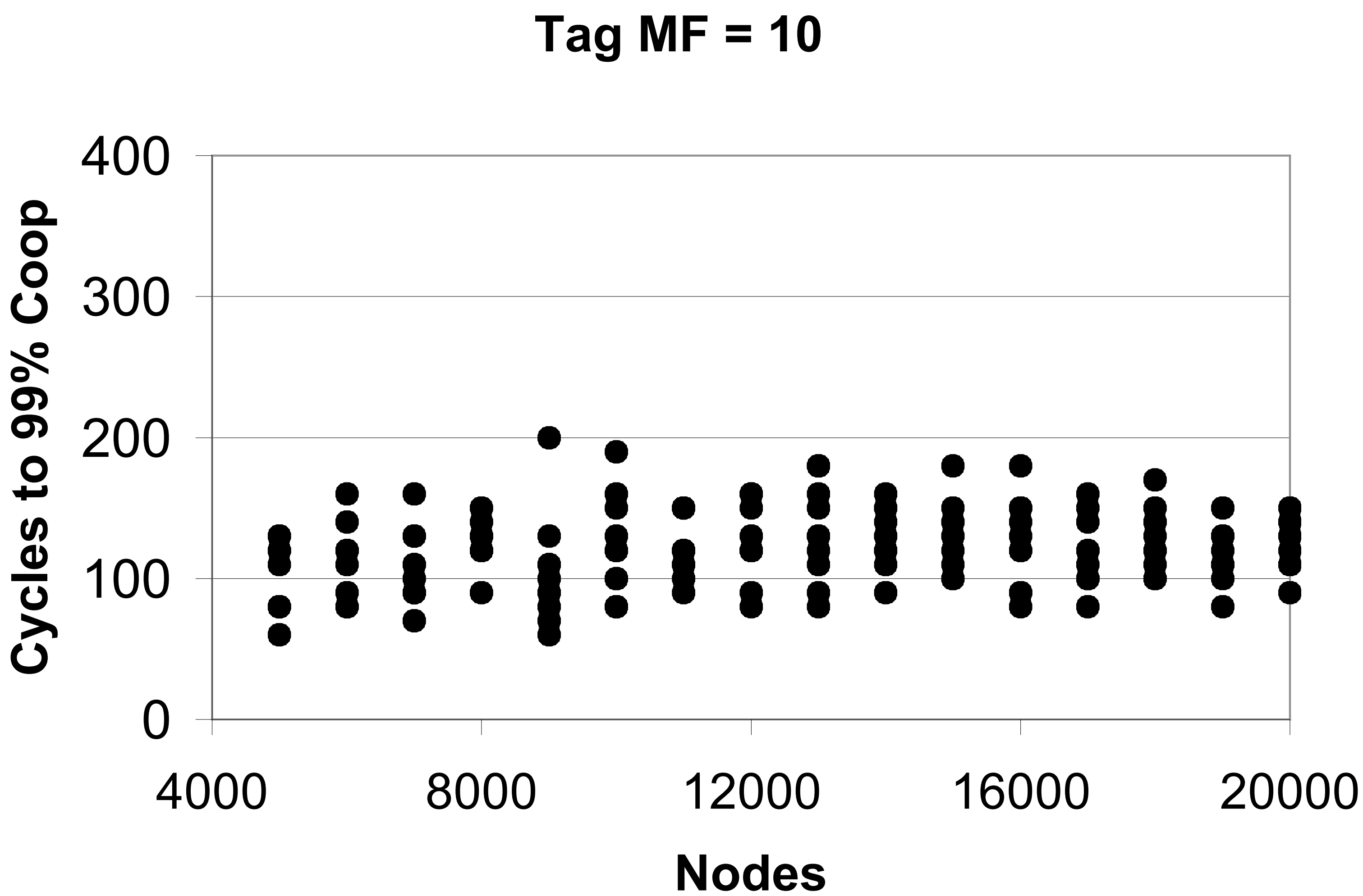
- Vary N between 4,000..120,000
- Maximum degree 20
- Initial topology random graph
- Initial strategies all defection (not random)
- Mutation rate m = 0.001 (small) a previous
- Payoffs as before: T=1.9, R=1, P=d, S=d (where d is a small value)

Results

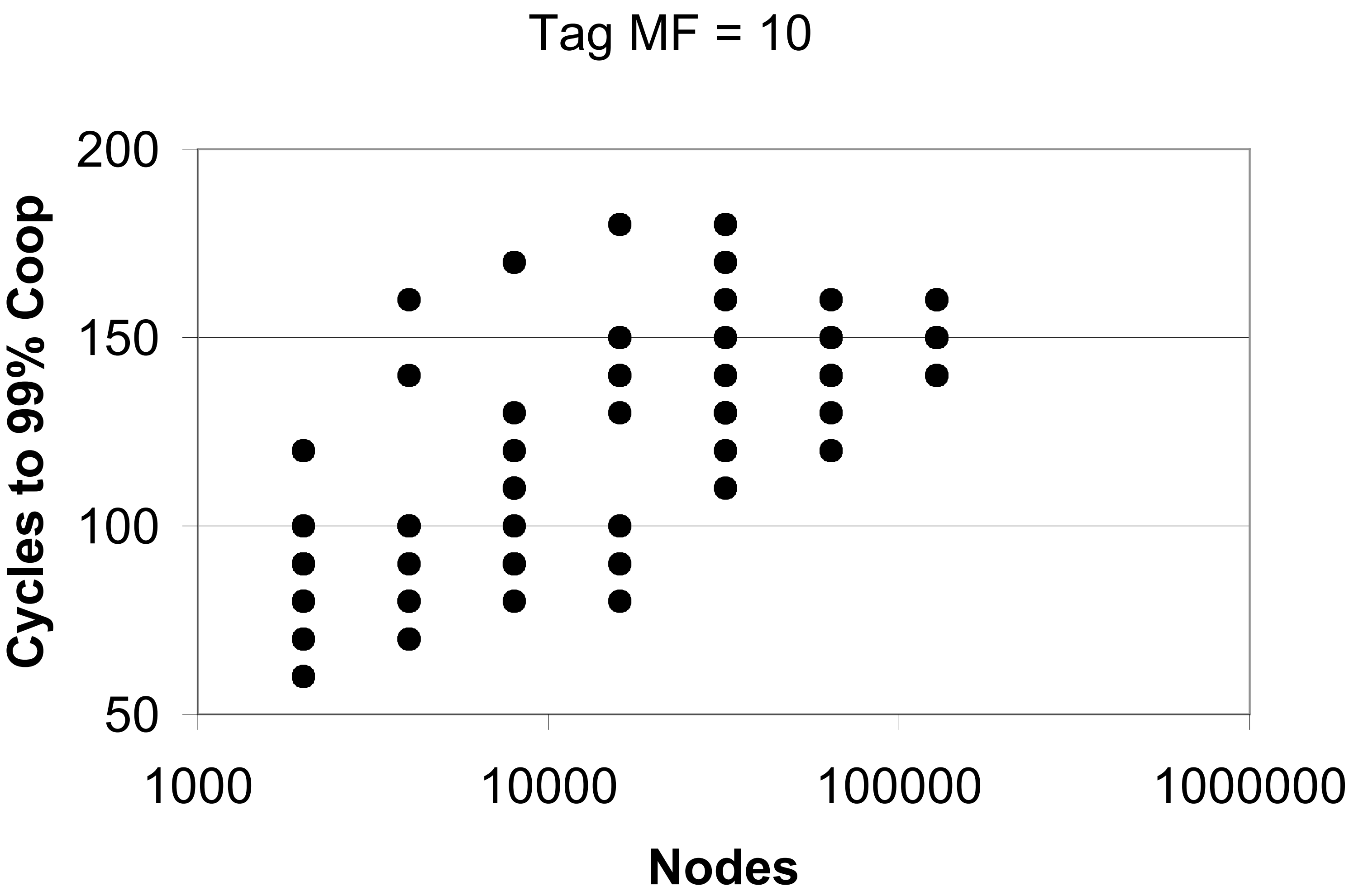


Tag MF = 1
Cycles to 99% Coop
4000
3000
2000
1000
0
4000
8000
12000
16000
20000
Nodes

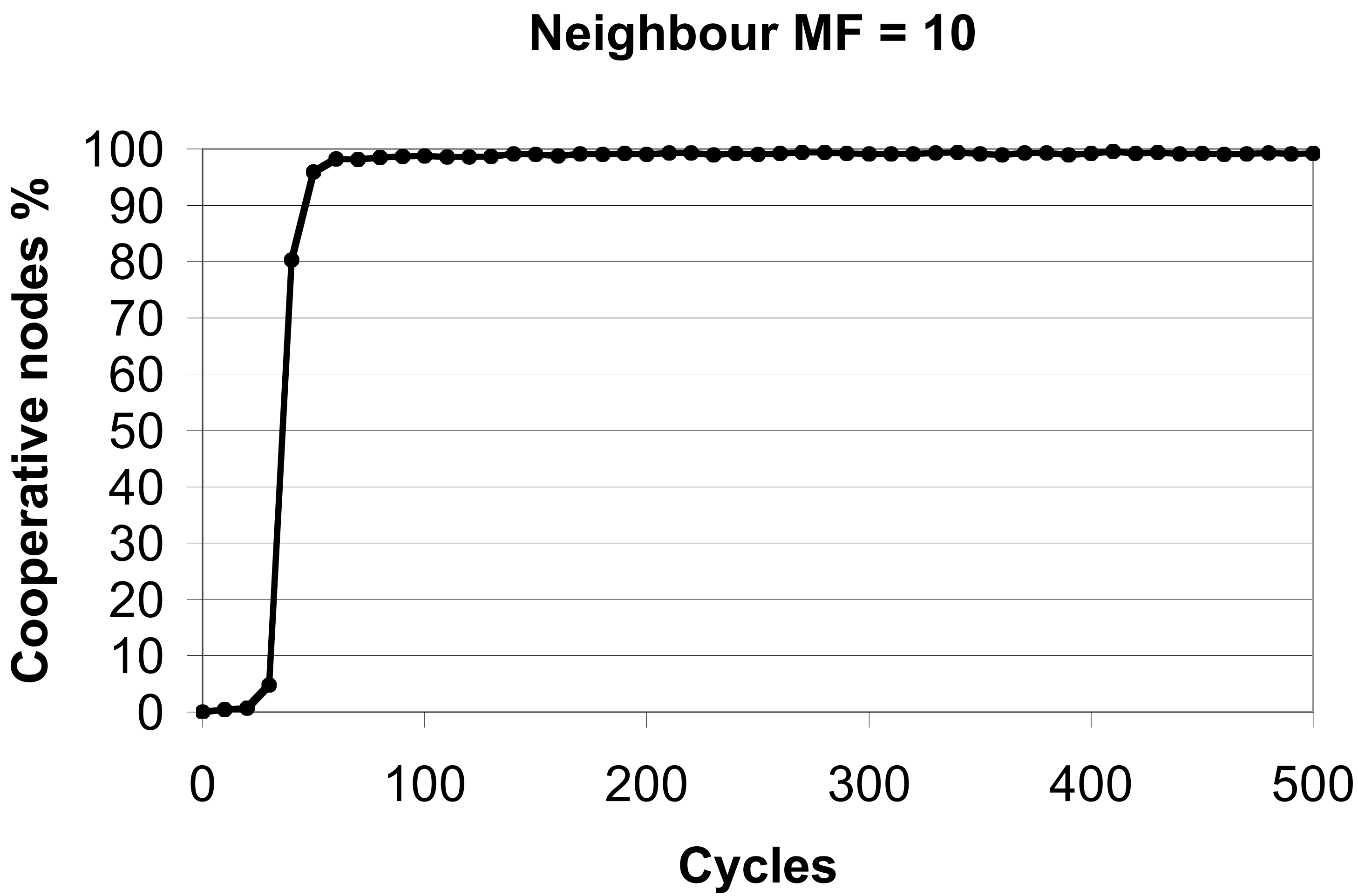
Results – increased mf=10



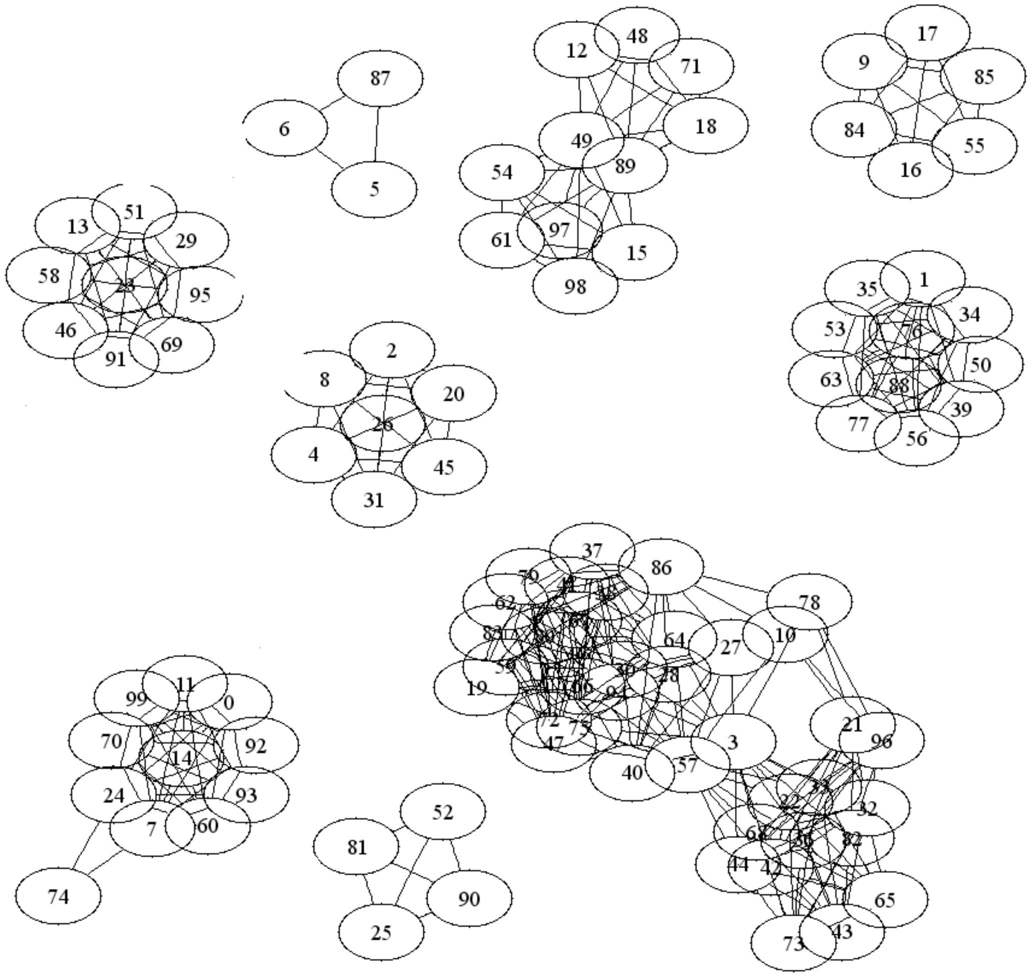
A few more nodes



A typical run (10,000 nodes)

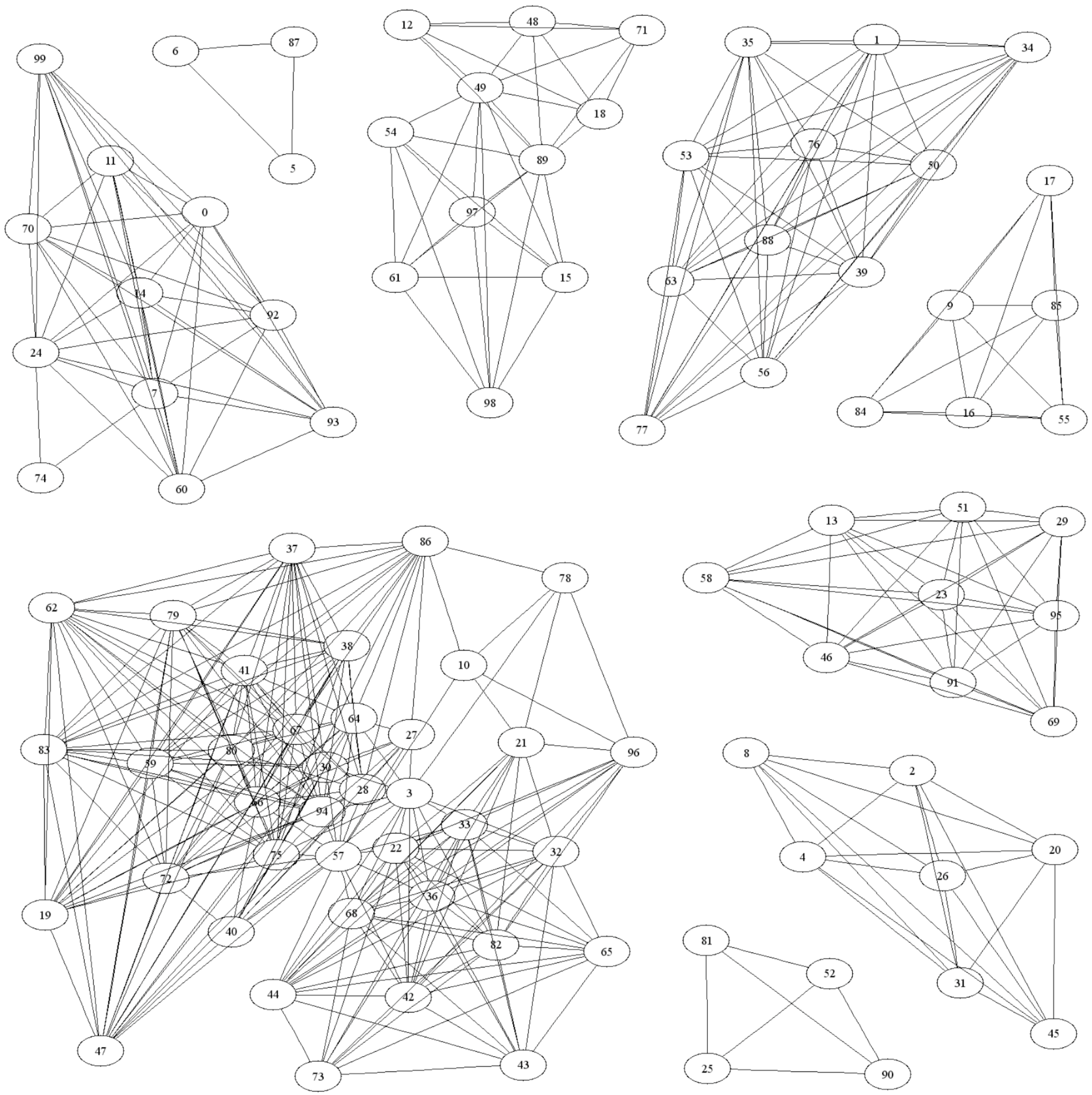


A 100 node example – after 500 generations



DELIS

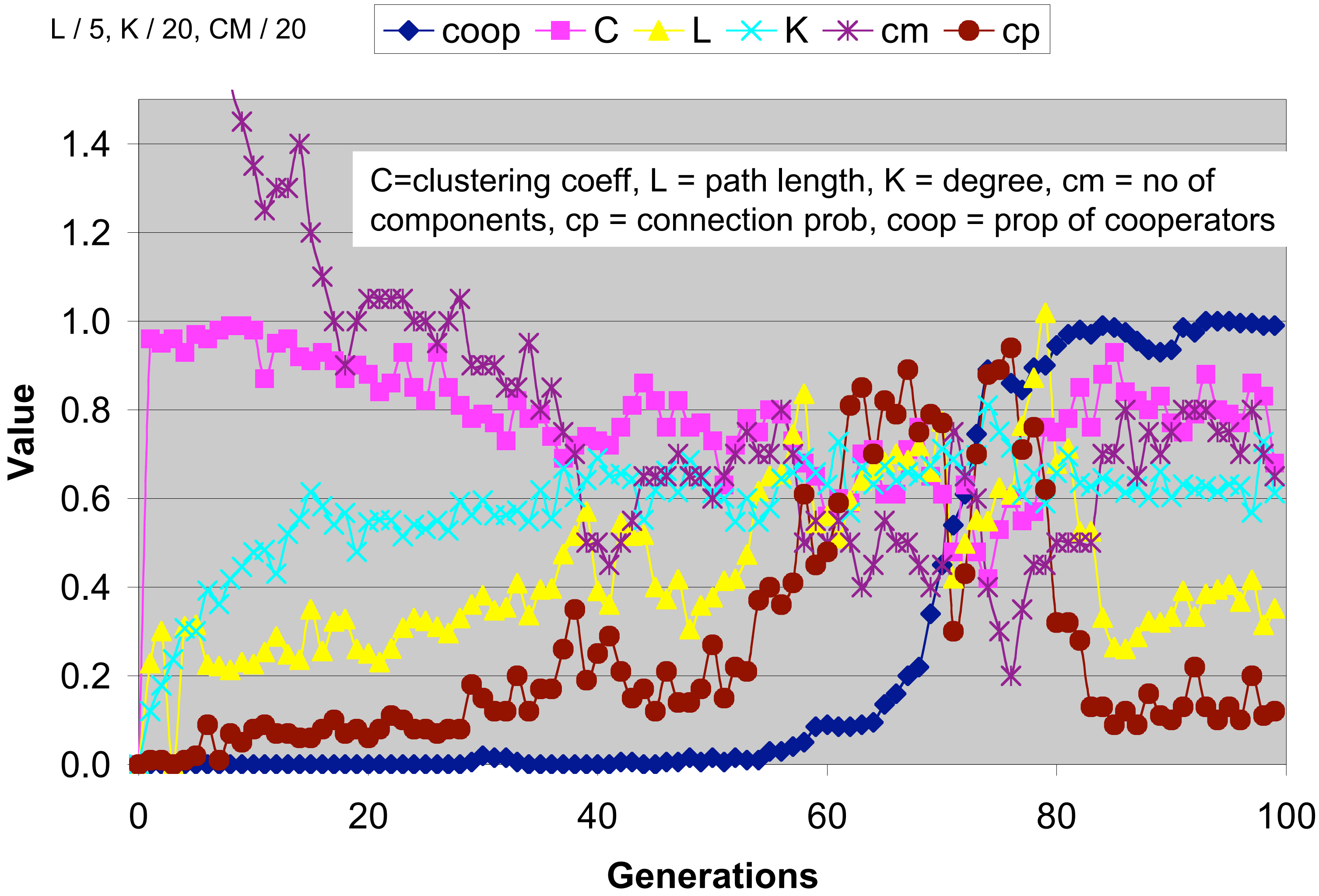
Dynamically Evolving, Large-scale Information Systems



Topology Evolution – so far it seems….

- From ANY initial starting topology / strategy mix same outcome (tried random, lattice, small world, all nodes disconnected, all defect, random, all coop)
- Typically (very approx.) a max of n/10 unstable components exist at any one time which are highly internally connected (L not much more than 1 and C very high)
- But they are not of equal size
- Constantly reforming and changing due to mutation and replication
- Rough characterisation of disconnectedness = prob. that two random nodes are connected

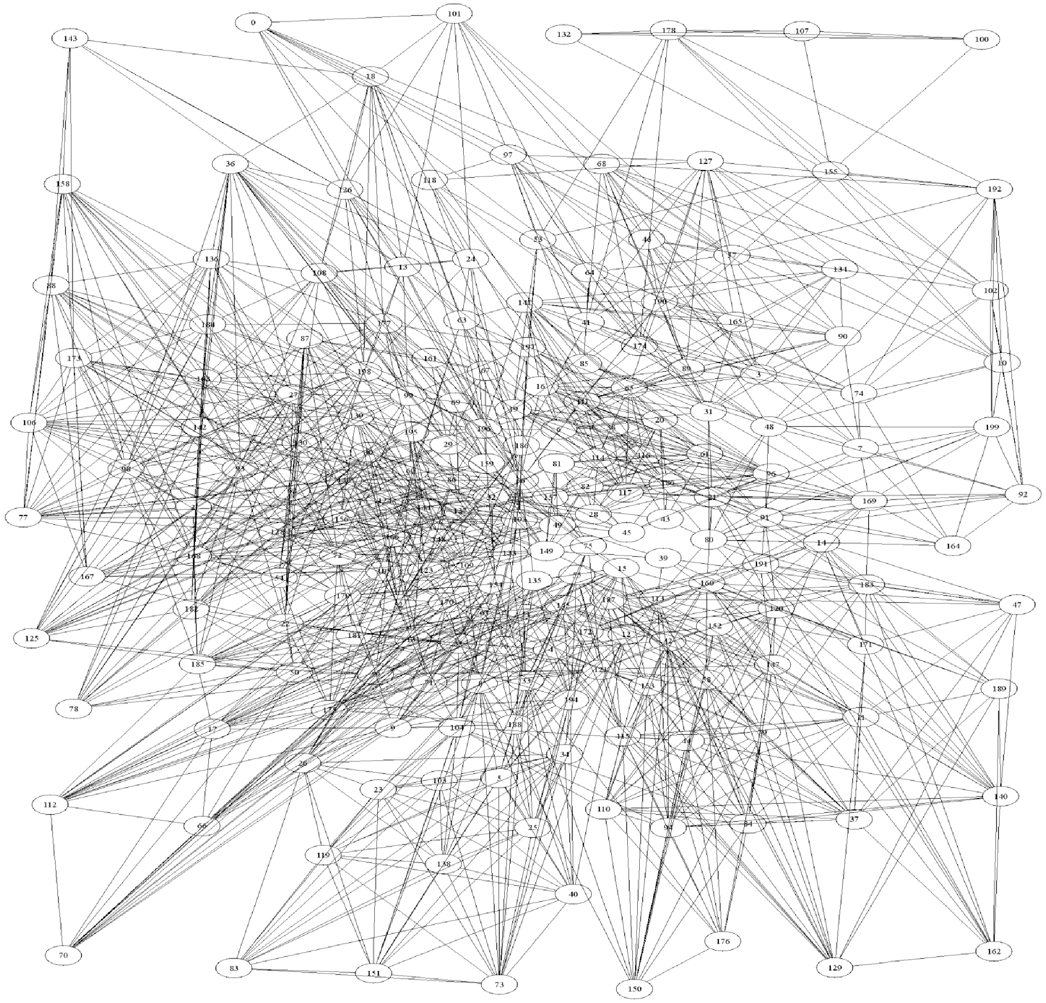
Typical run, 200 nodes



A message passing game

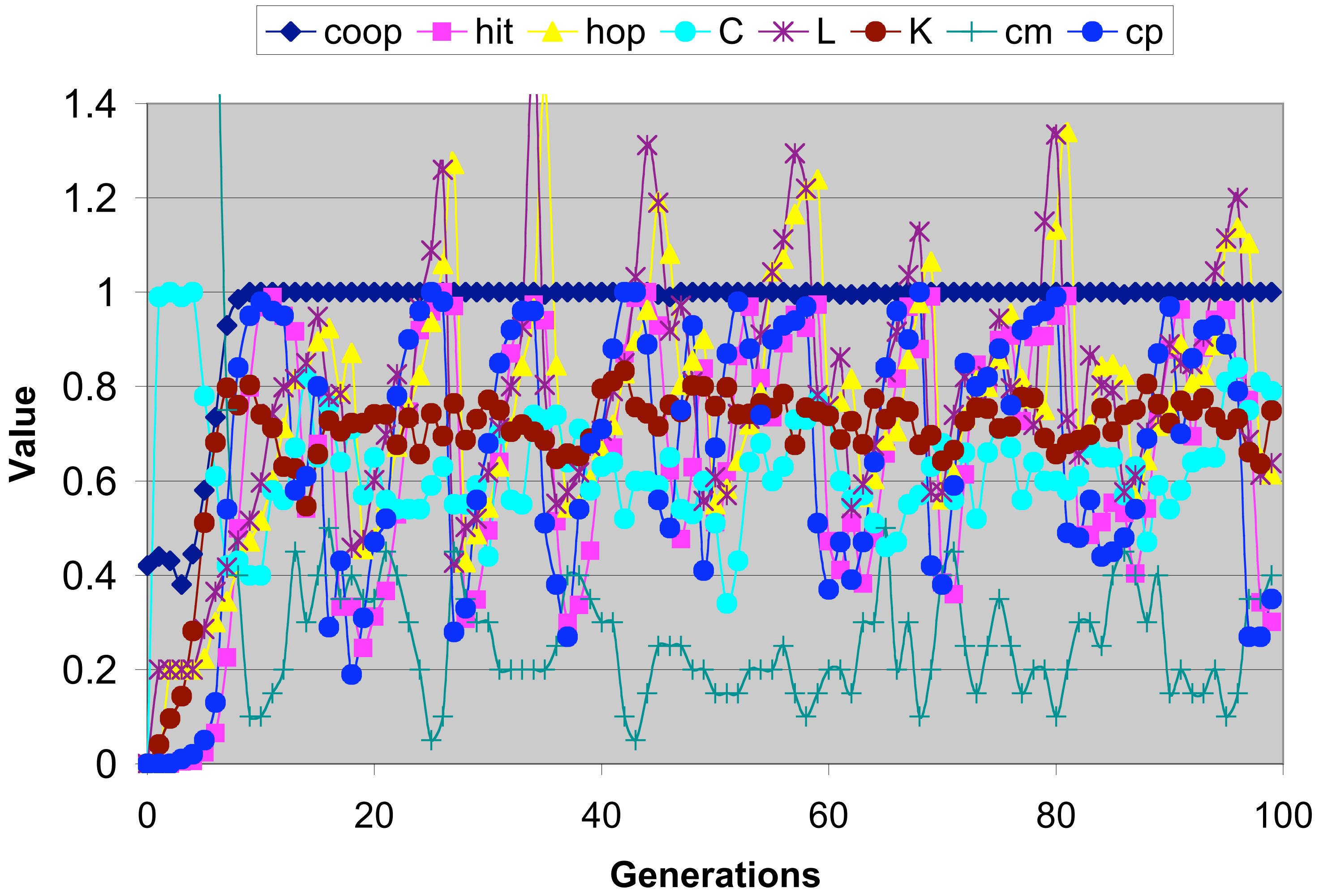
- Keep everything the same but change “game”
- A message passing game – select two nodes (i,j) randomly from G. i tries to send a message to j.
- Do a flood fill query from i to j.
- If a route of *cooperators* is found from i to j then i gets a “hit” (one point added to score)
- Only cooperators pass on a messages incurring a small cost for doing so, reducing score
- Hence defectors will do better than cooperators getting the same proportion of hits
- Tough task since need a route between specific nodes via a chain of coops only

Message Passing game - 200 nodes after 500 generations



Message passing game - 200 nodes to 100 generations

L / 5, K / 20, CM / 20



But its not as good as it seems...

- Increased games to 25n per generation
- Start with random strategies (all def. no good)
- Does not appear to scale well (oscillations)
- More work needs to be done (only a few runs)
- A very tough test for scaling on this mechanism
- On reflection - surprising it did this well
- Try “easier” and more realistic “game”

Next steps

- Assume random selections from the population (will it work with net. generated selections?)
- Try more realistic task (file sharing) (Qixiang Sun & Hector Garcia-Molina 2004)
- So far robustness tested as effect of mutation – static pop size – try drop or introduce lots of nodes at once
- Simplistically treats all neighbour links as “one chunk” rather than selectively removing links (eliminate comparison also? Vance Maverick’s idea) various schemes possible
- Translate model into PeerSim framework

Conclusion

- Tag-like dynamics can be put into a network using simple rewiring rules
- Even simple rules appear flexible, able to create and maintain different topologies for different tasks
- Free-riding is minimised, even though node behaviour selfishly and have no knowledge of past interaction
- At least for close neighbour interaction the method scales well
- But much more analysis needs to be done and more realistic kinds of p2p task domain need to be tested