DELIS

Self-Organising, Open and Cooperative P2P Societies – *From Tags to Networks*

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

Talk Overview

- Why study cooperation in P2P systems?
- The Prisoner’s Dilemma game
- Tags and how they work
- Applying in a P2P using re-wiring rules

Why study cooperation?

How can nodes (agents) do tasks involving:

- Coordination & Teamwork
- Specialisation & Self-Repair
- Emergent Functions & Adapting to Change

WITHOUT centralised supervision and in a scalable way when nodes are "peers" (autonomous)

DELIS

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

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)**

What are "tags"

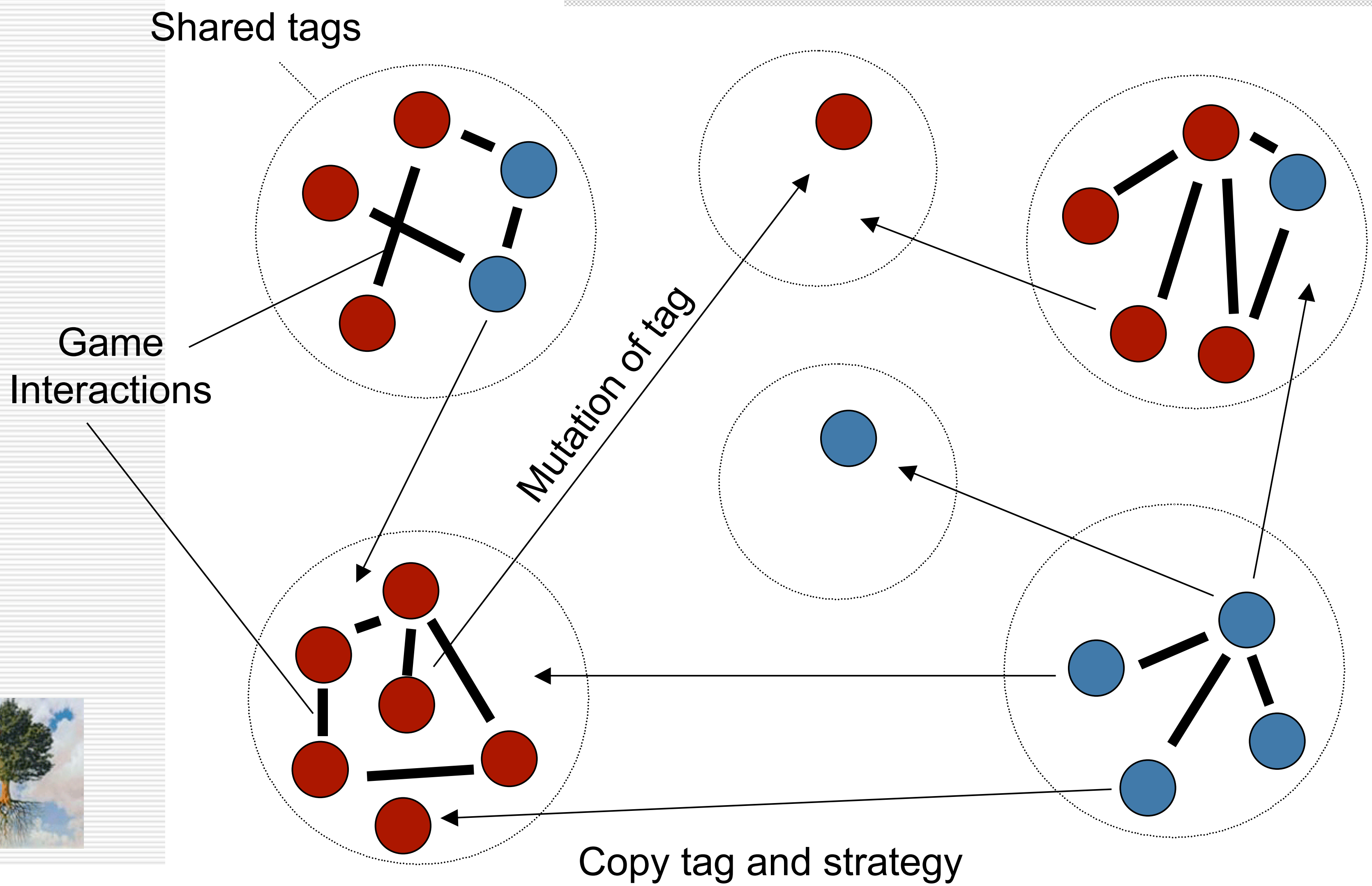
- Tags are observable labels, markings or social cues
- Agents can observe tags
- Tags evolve like any other trait (or gene)
- Agents may discriminate based on tags
- John Holland (1992) => tags powerful "symmetry breaking" function in "social-like" processes
- In GA-type interpretation, tags = parts of the genotype reflected directly in the phenotype

Agents - a Tag and a PD strategy

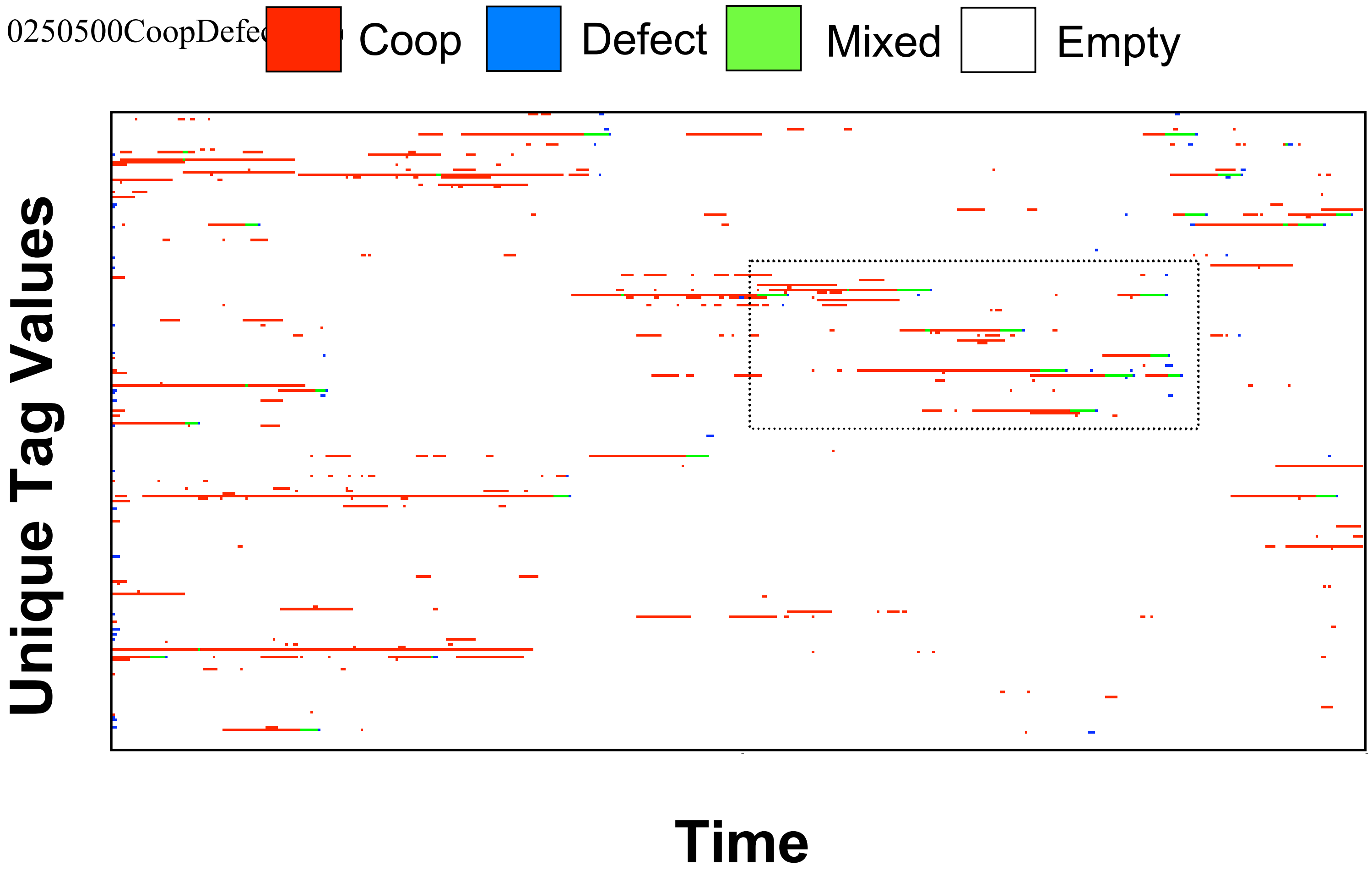
Tag = (say) Some Integer

Game interaction between those with same tag (if possible)

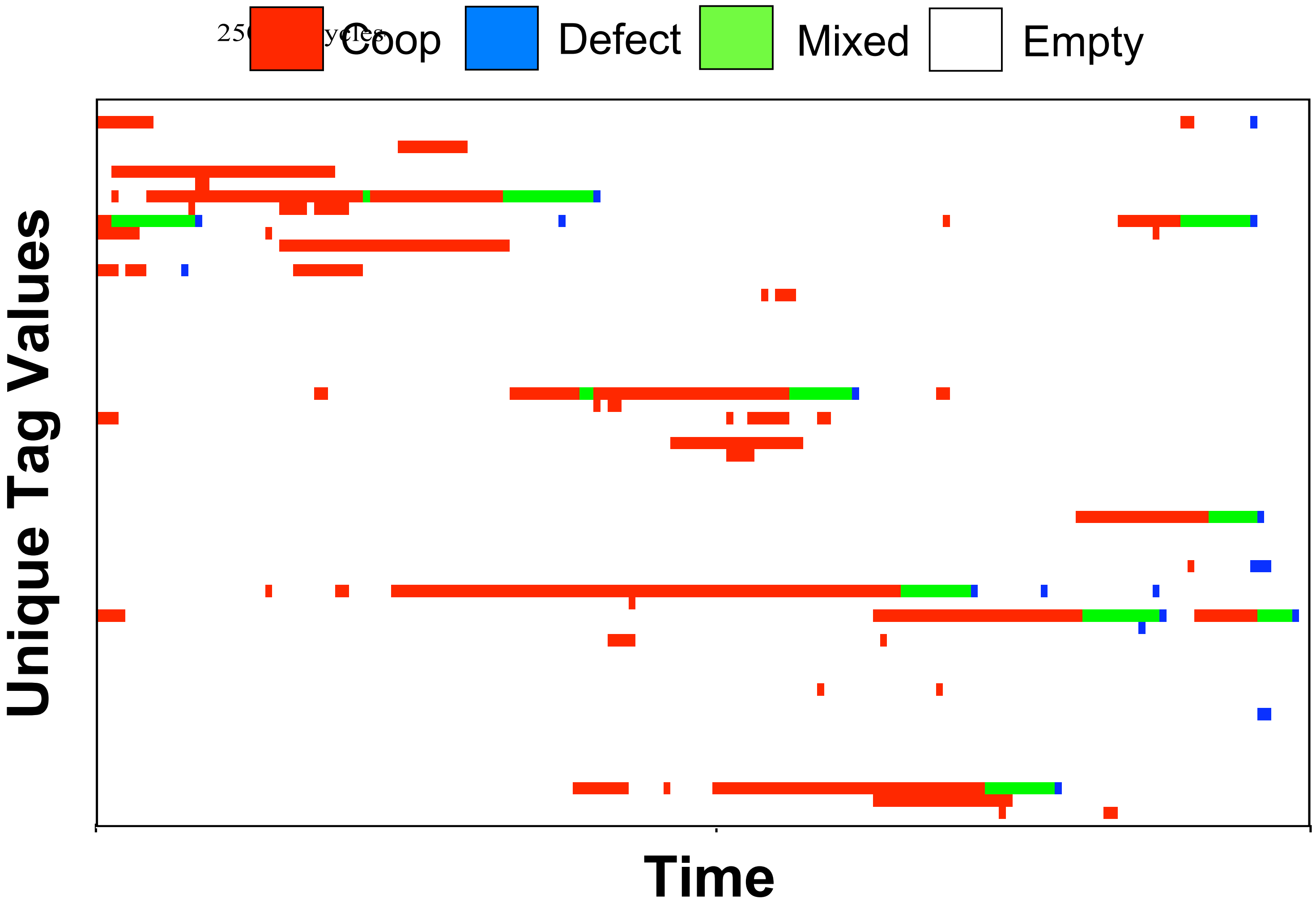
How Tags Work



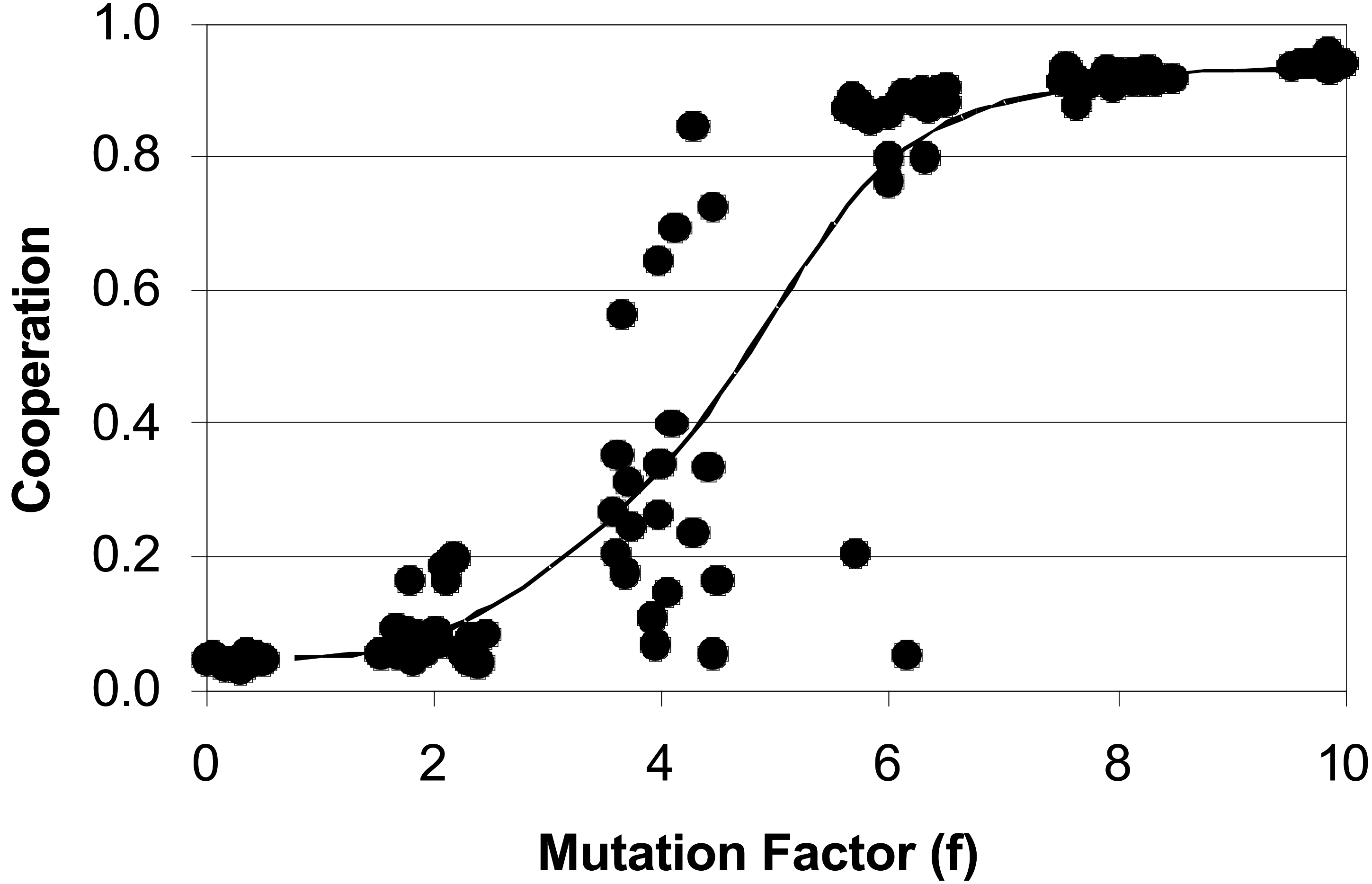
Visualising the Process (Hales 2000)



Visualising the Process



Recent finding – 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 that 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

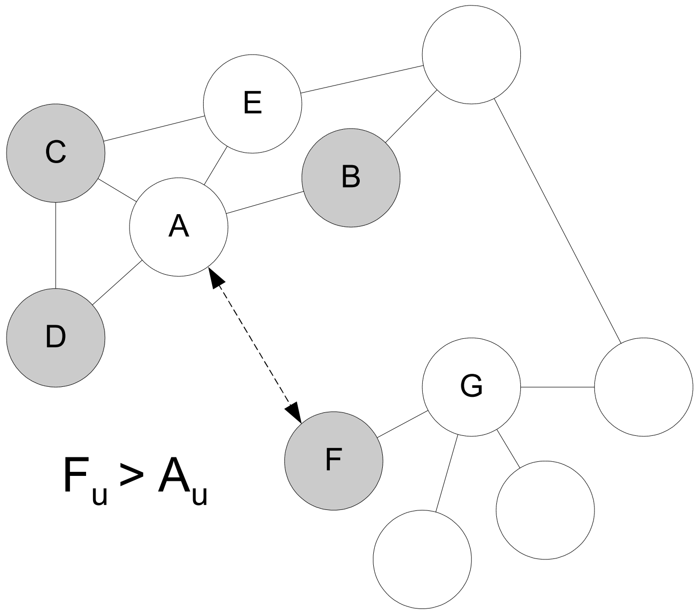
- For tag-like dynamics high clustering would appear to be required (groups required)
- Will dynamic nature of the scenario support this?
- Can cooperation be maintained without it?
- We might start simulations of the model with high clustering initially (say small world or lattice) and compare that 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 plus j itself
- When maximum degree of node is exceeded throw away a randomly chosen link

Social Climbing

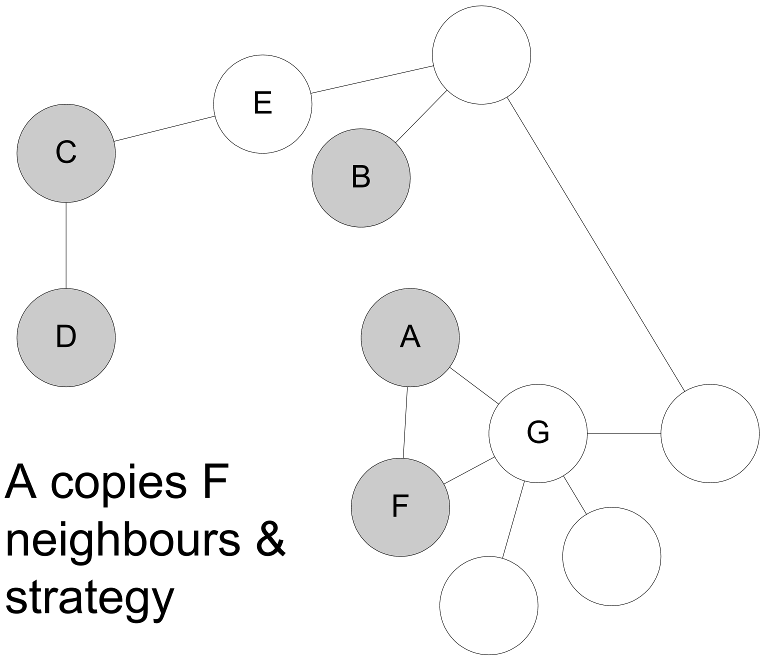
Before



$F_u > A_u$

Where A_u = average utility of node A

After

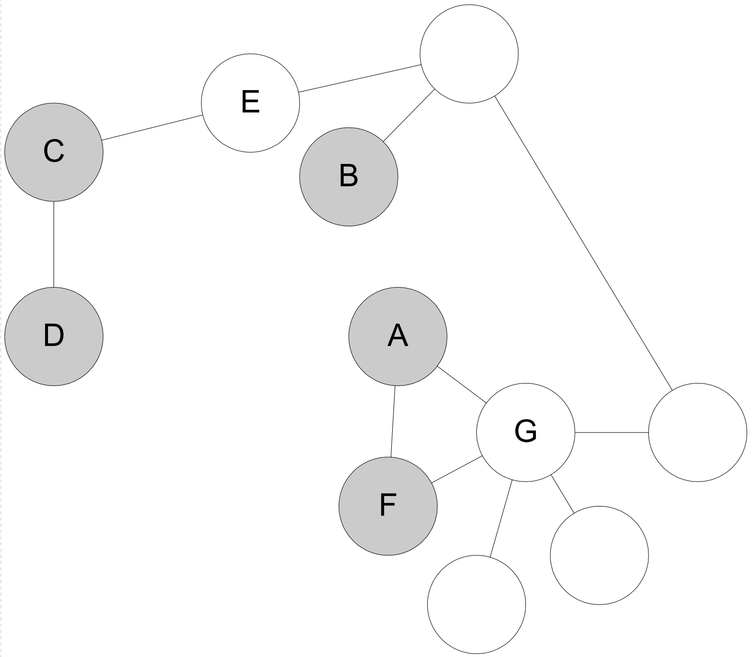


A copies F neighbours & strategy

In his case mutation has not changed anything

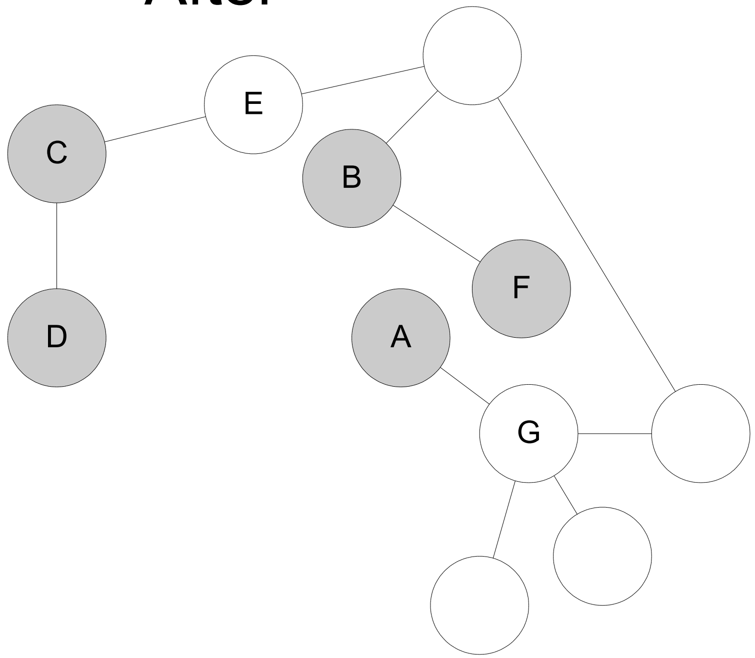
Mutation on the Neighbourhood

Before



Mutation applied to F's neighbourhood

After



F is wired to a randomly selected node (B)

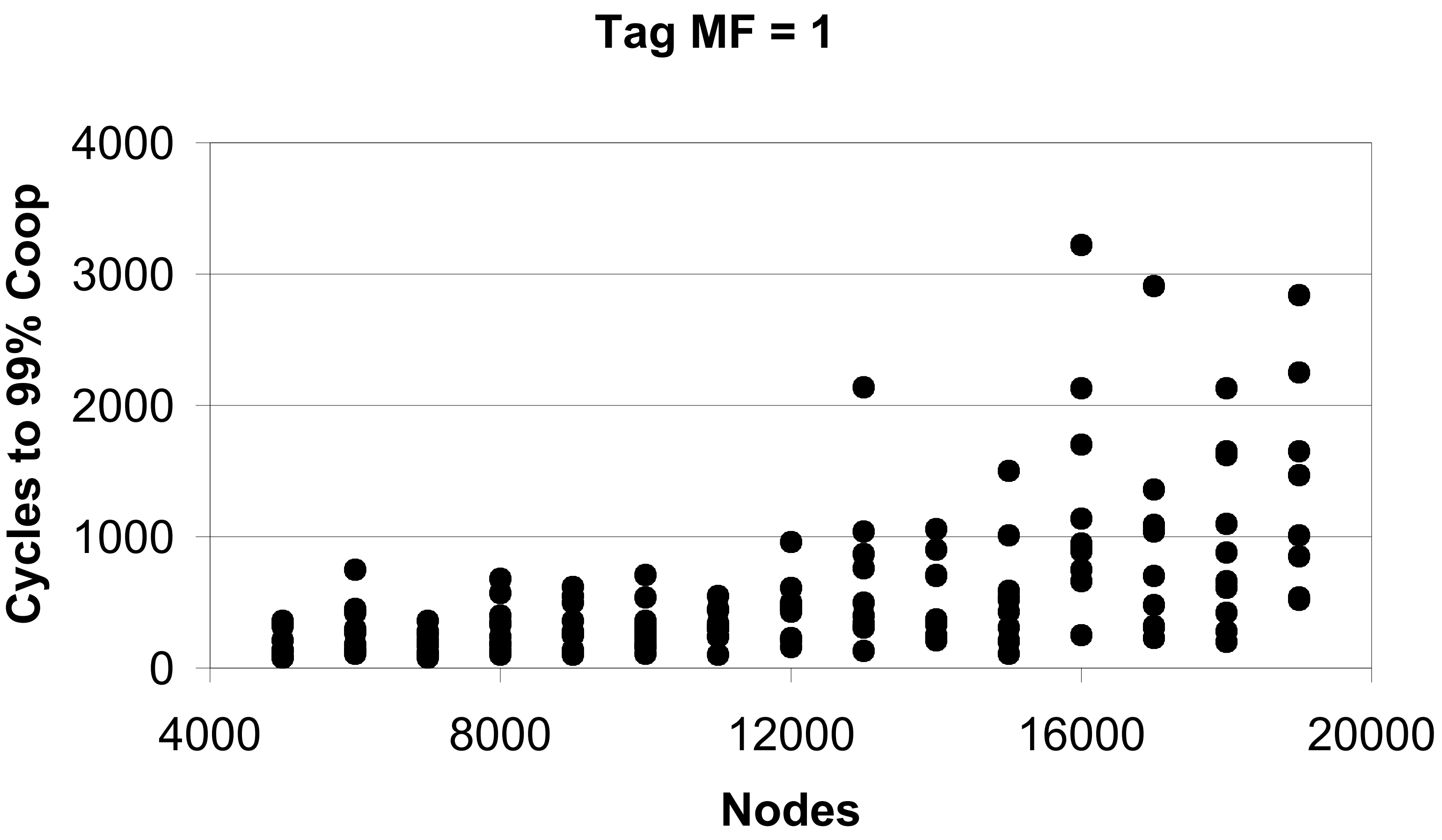
The Simulation Cycle

```
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
```

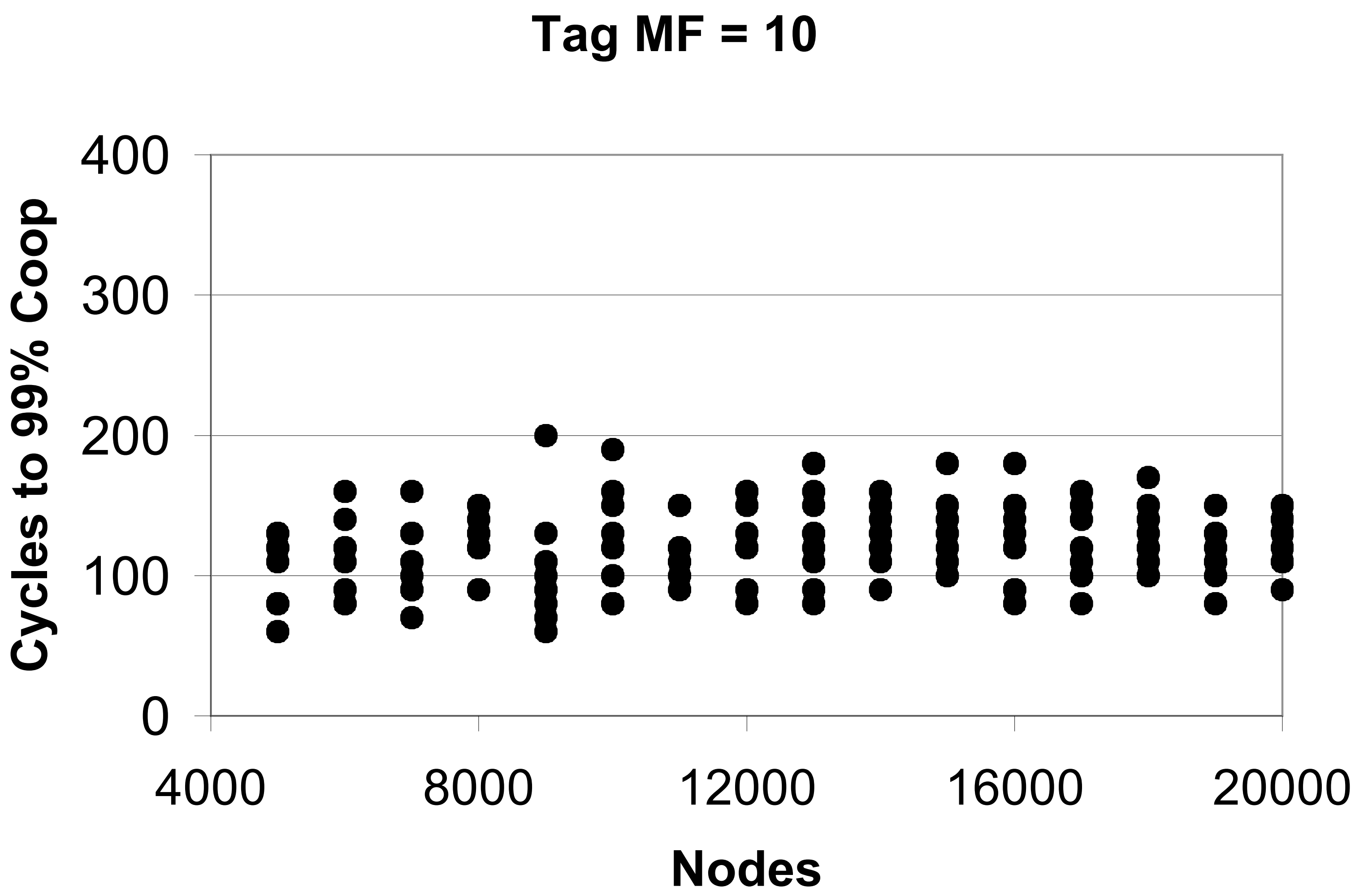
Parameters

- 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)
- PD payoffs: T=1.9, R=1, P=d, S=d
  (where d is a small value)

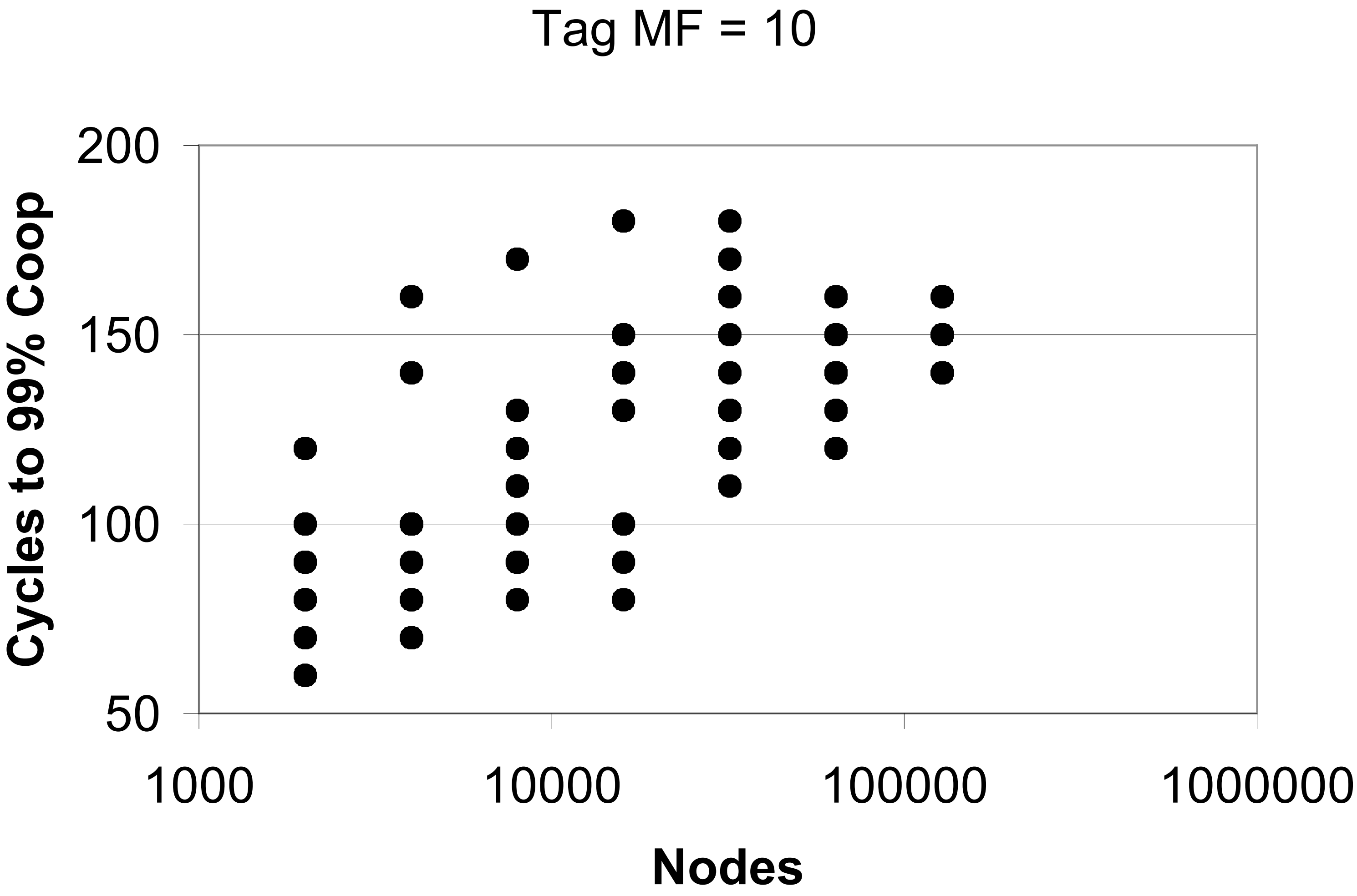
Results


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

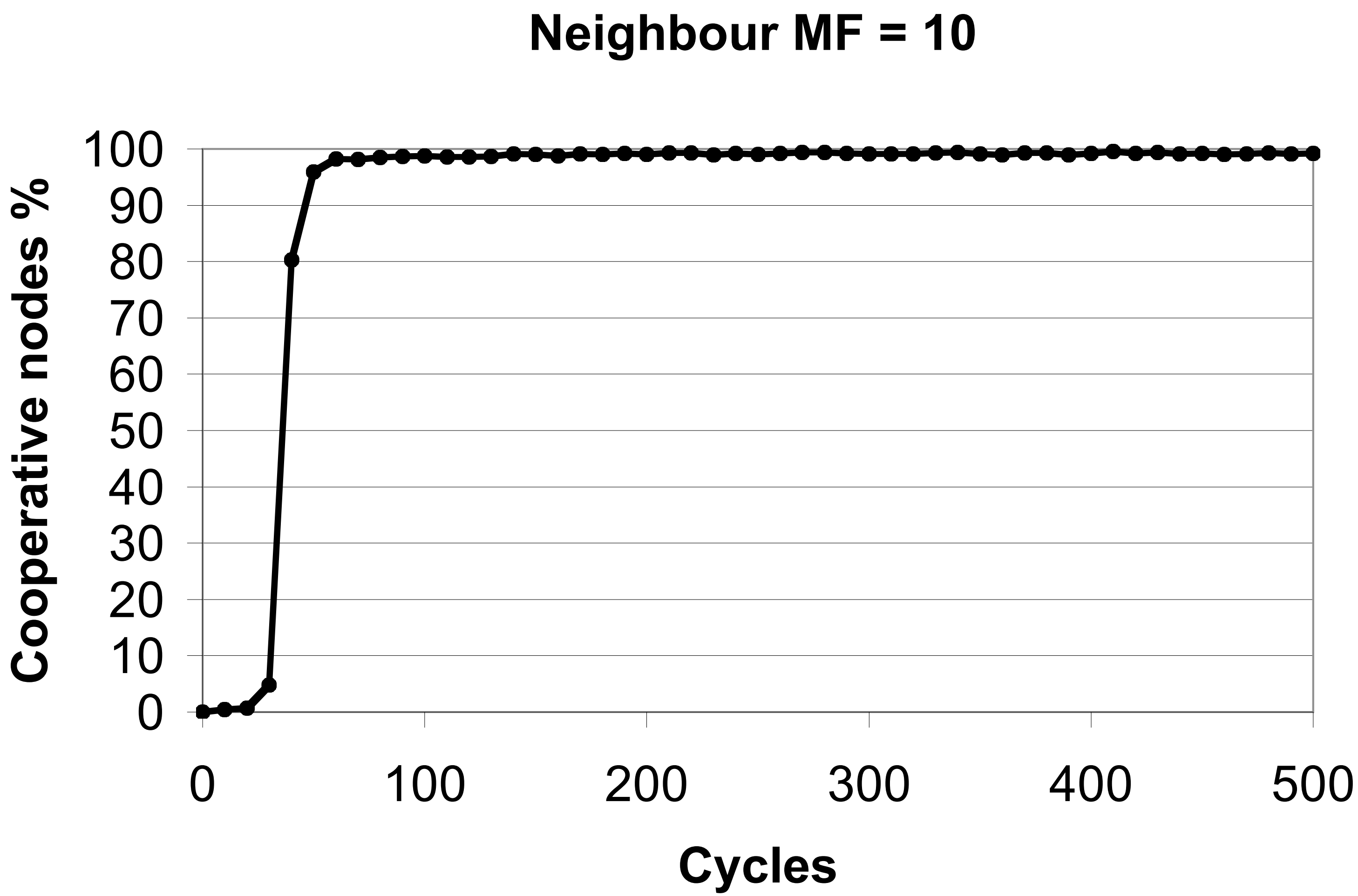
Results – increased mf=10



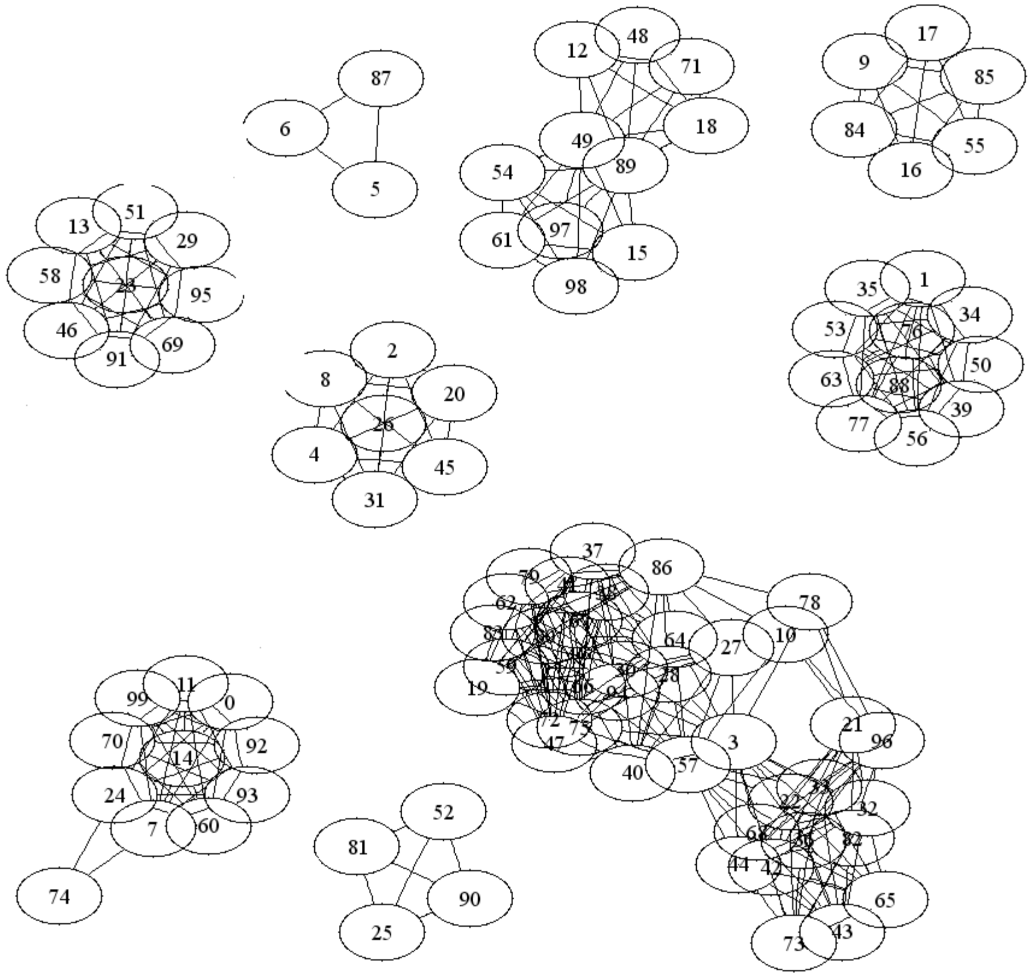
A few more nodes

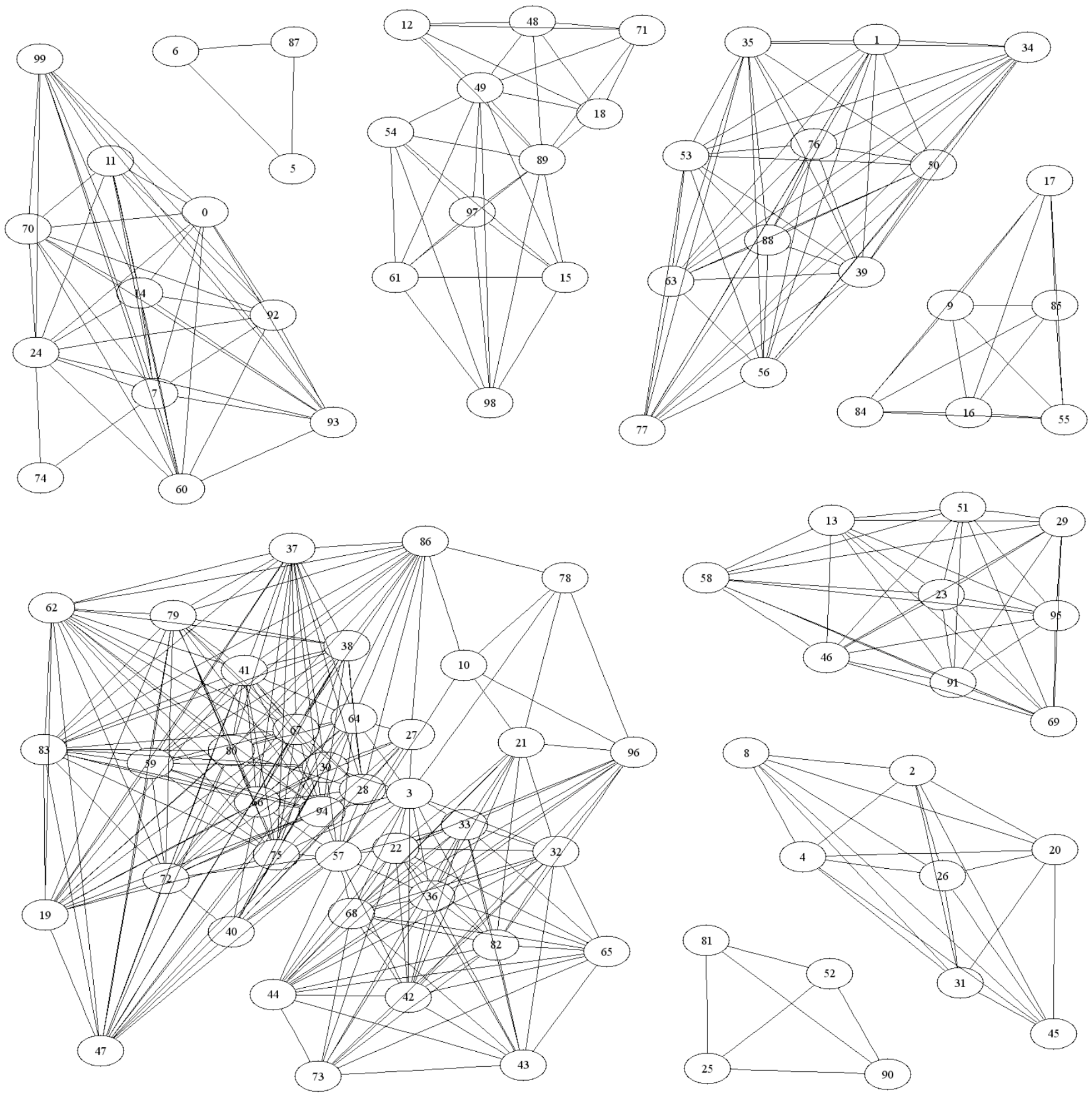


A typical run (10,000 nodes)



A 100 node example – after 500 generations



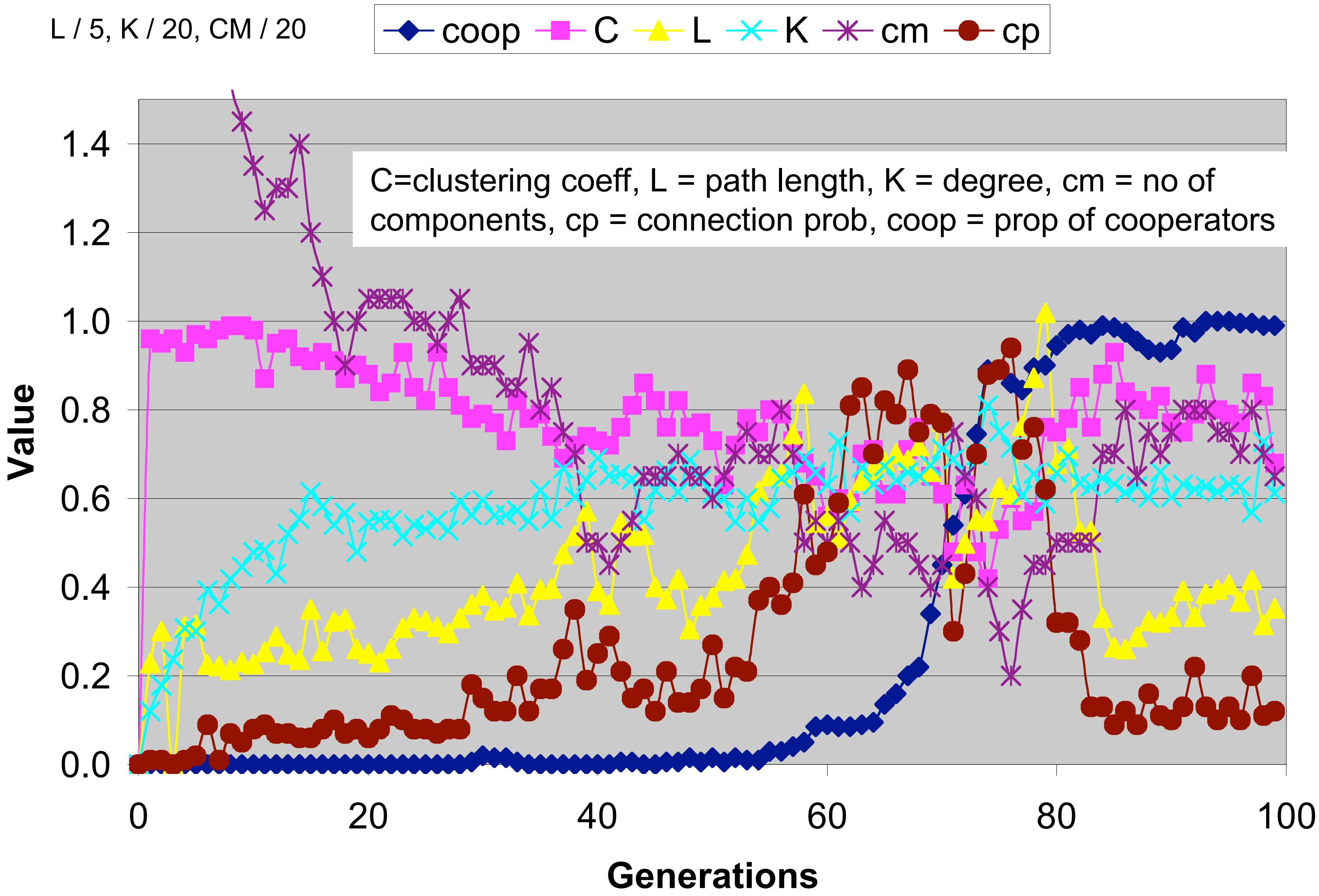


99 6 87 5 11 0 70 14 92 24 7 93 74 60
12 48 71 49 18 54 89 97 61 15 98
35 1 34 53 76 50 88 63 39 56 77
17 9 85 84 16 55
13 51 29 58 23 95 46 91 69
37 86 78 62 79 38 10 41 64 27 21 96 83 80 67 59 30 28 3 66 94 33 22 32 75 57 72 36 19 68 40 82 65 44 42 47 73 43
8 2 4 20 26 31 45
81 52 25 90

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

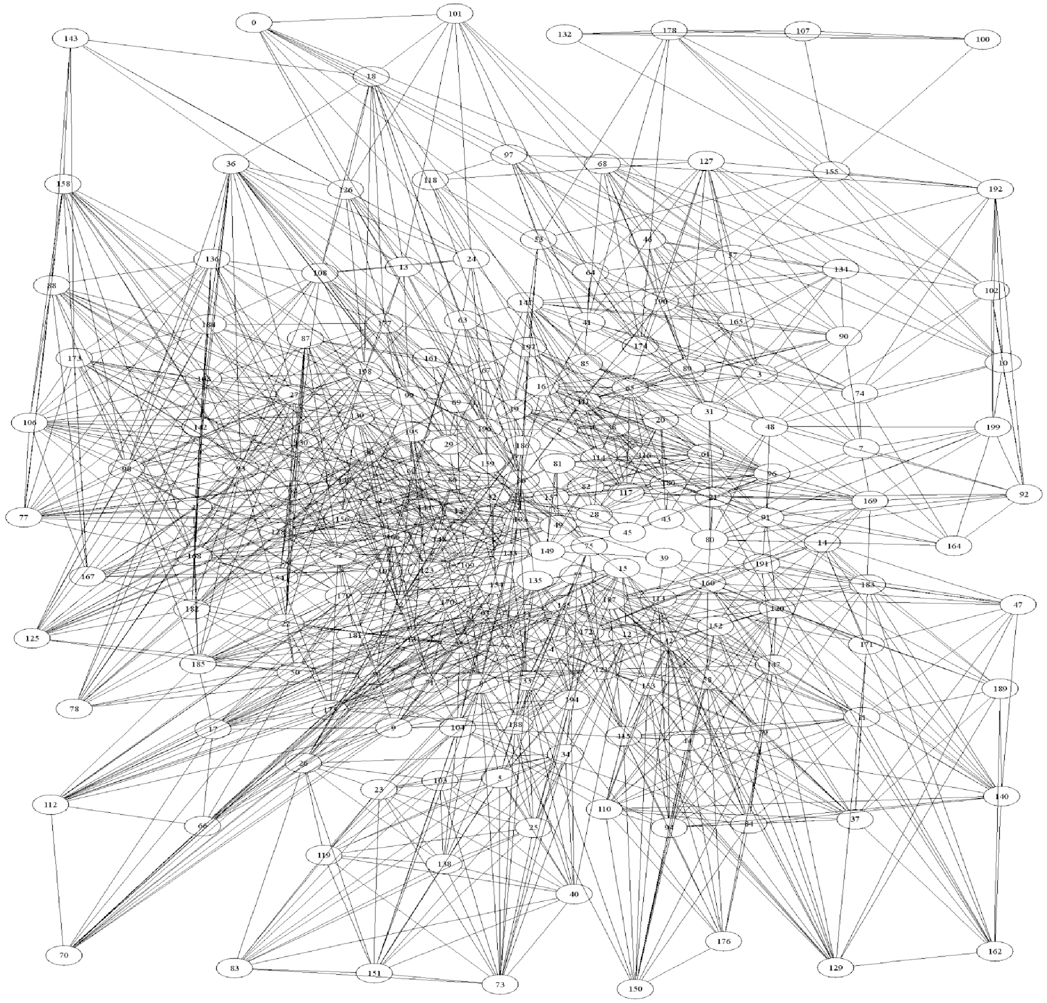
Typical run, 200 nodes



A message passing game

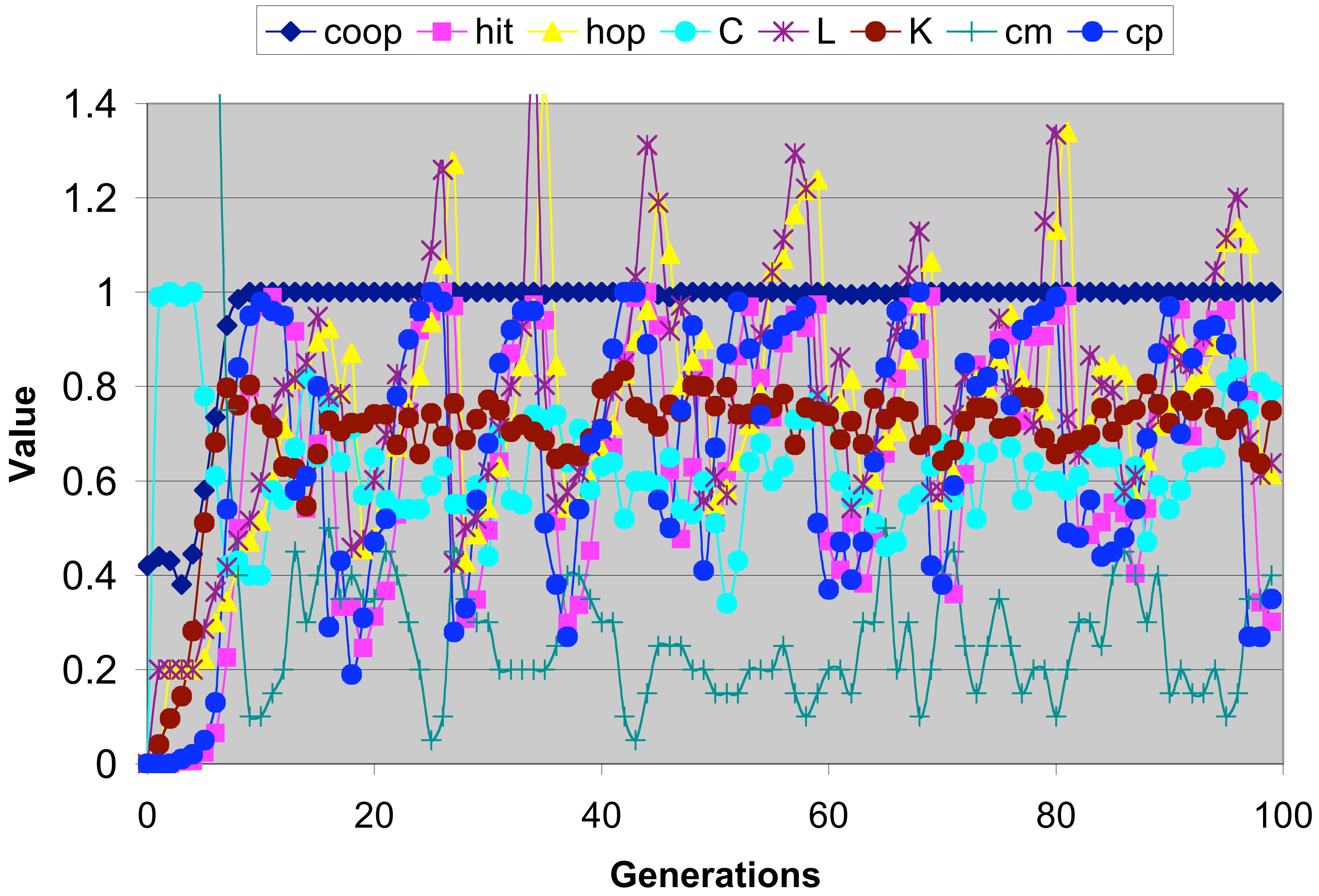
- Keep everything the same but change “game”
- A message passing game – select two nodes (i,j) randomly from G. i tries to send a message to j.
- Do a flood fill query from i to j.
- If a route of *cooperators* is found from i to j then i gets a “hit” (one point added to score)
- Only cooperators pass 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

- Currently random selections - will it work with network generated selections?
- Realistic task (file sharing) (Qixiang Sun & Hector Garcia-Molina 2004 – see Hales 2004 IEEE P2P2004)
- So far robustness tested as effect of mutation – static pop size – try various “churn rates”
- Treats node links as “one chunk” rather than selectively removing links
- Modified form might enhance BitTorrent?

Conclusion

- Tag-like dynamics using simple rewiring rules
- Appears flexible - different topologies for different tasks
- Free-riding minimised with selfish nodes and no knowledge of past interaction
- Method scales well at least in some tasks
- More analysis needs to be done
- Also links with incentive based systems - "socially emergent incentive system"