Altruism “for free” using Tags

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

- My background and motivation
- The problem
- A solution: Tags and how they work
- Applying in a P2P using re-wiring rules

Background and Motivation

- Computer Science and A.I
- Agent-Based Social Simulation (ABSS)
- Interest: Emergence of Cooperation (PD)
- Sociologists and engineers - same questions!
- Now: Apply ideas from ABSS to some engineering problems (back to CS!) e.g. open P2P

The Problem

Consider a system composed of agents that are:

- Autonomous (not externally controllable)
- Selfish (maximise their own utility)
- Greedy (local hill-climb)
- Adaptive (copy other nodes and self-adapt)

How do we get the nodes to cooperative for the good of the network rather than simply free-ride?

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

Ways to get Cooperation

- 3’rd party enforcement – expensive, tends to centralisation (Thomas Hobbes 1660)
- Repeated interactions – need repeated interactions & some altruism (Axelrod 1984)
- Fixed lattice interaction – not good for dynamic networks (Nowak & May 1992)
- **Tags – scalable, single round, simple (Holland 1993, Riolo 1997, Hales 2000)**

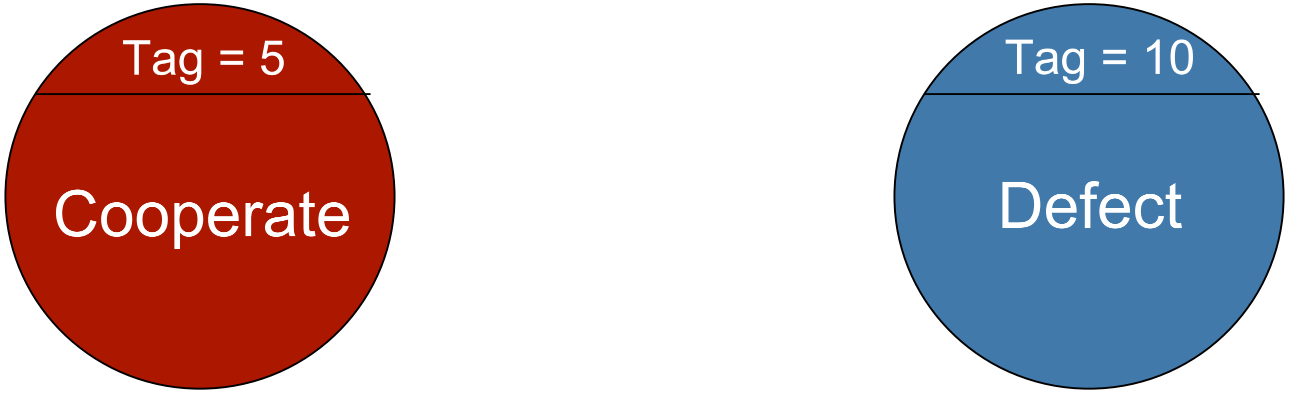
What are “tags”

- Tags are observable labels, markings, social cues
- They are attached to agents
- Agents interact preferentially with those sharing the same tag – no other function
- 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

An Evolutionary Scenario

- Agents are selfish and greedy
- Copy the behaviors of more successful
- Randomly mutate strategies
- No population structure but….
- Agents preferentially interact with those sharing the same tag

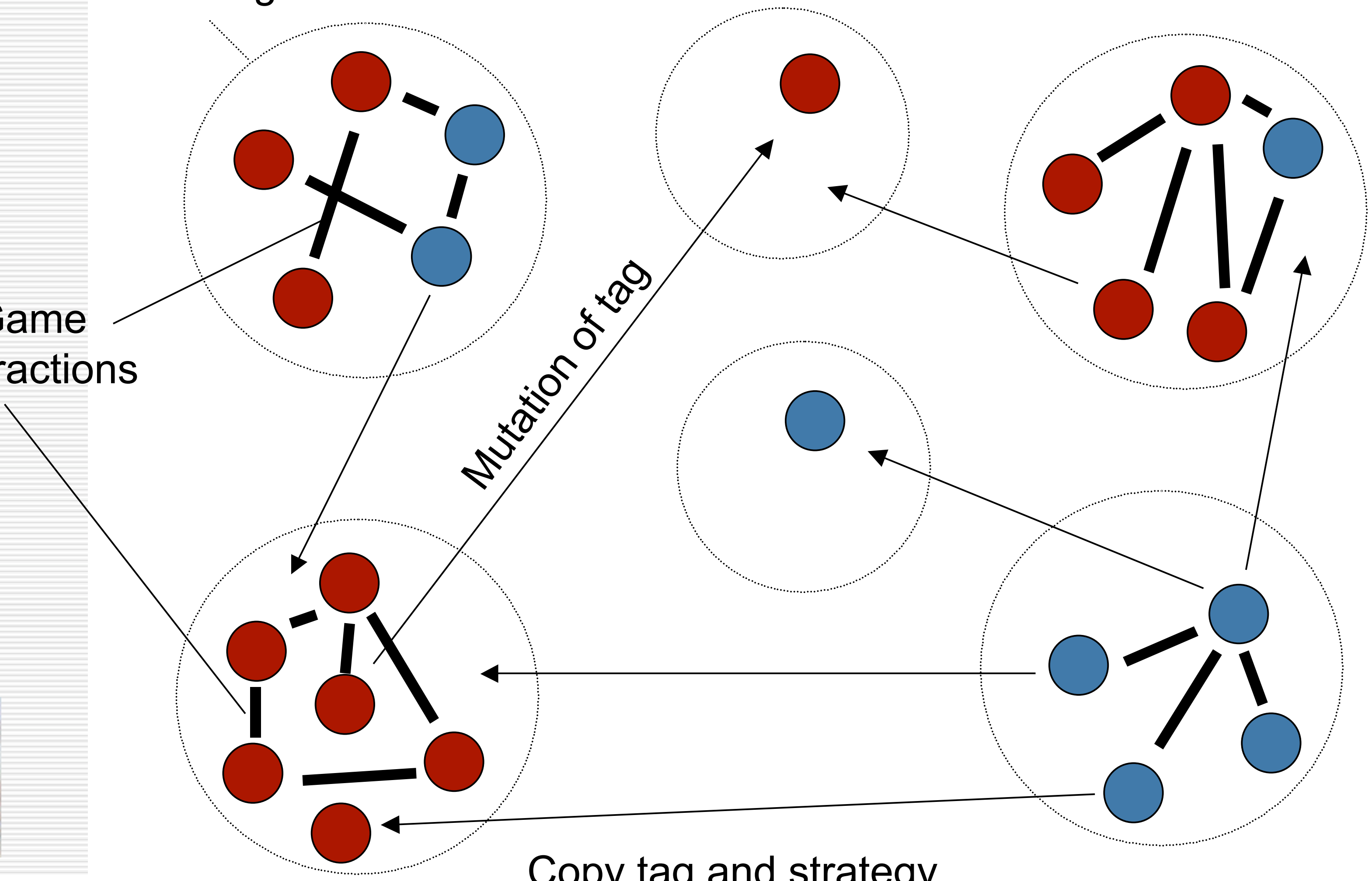
Agents - a Tag and a PD strategy



Tag = (say) Some Integer

Game interaction between those with same tag
(if possible)

How Tags Work



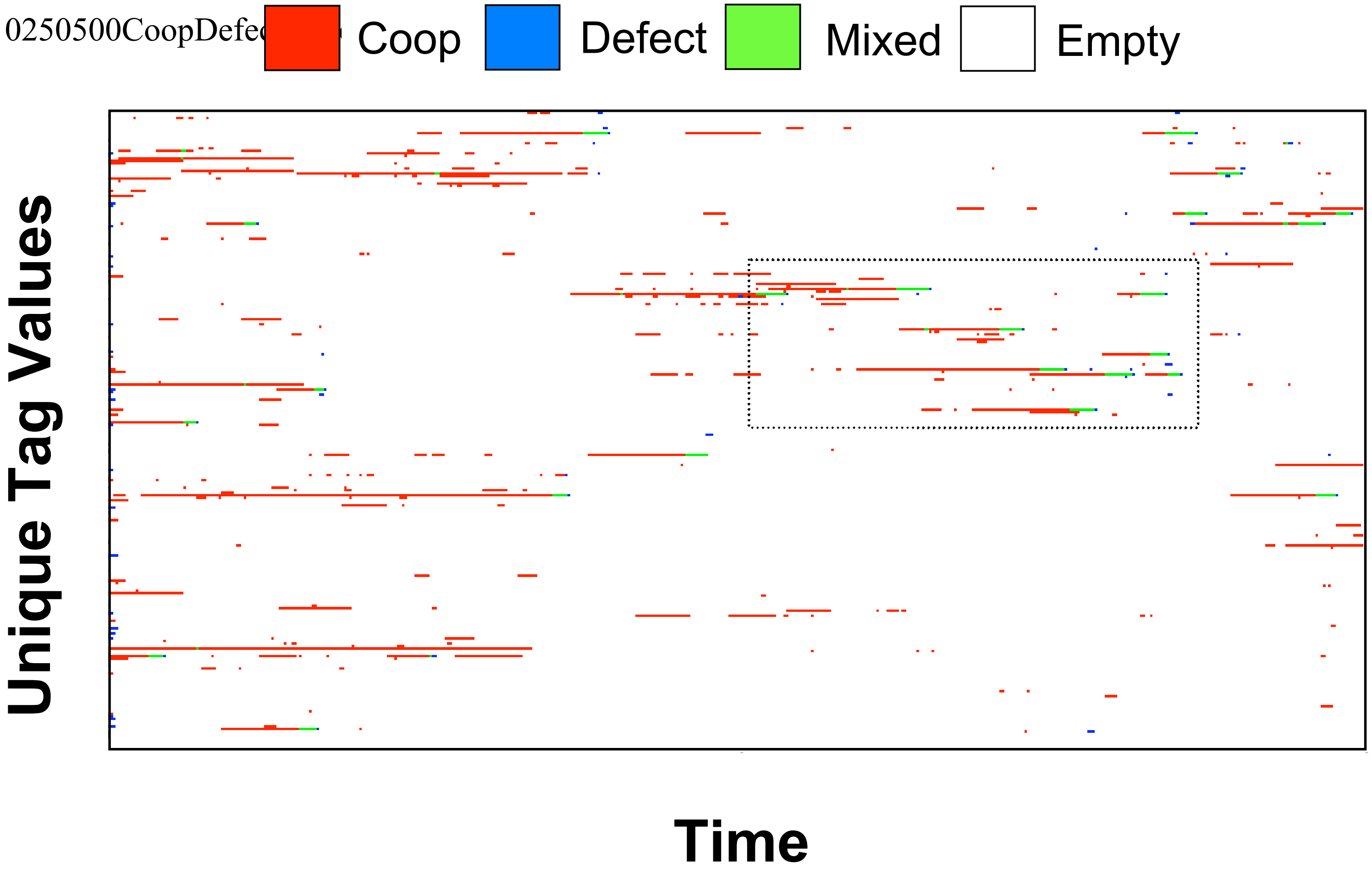
Shared tags

Game Interactions

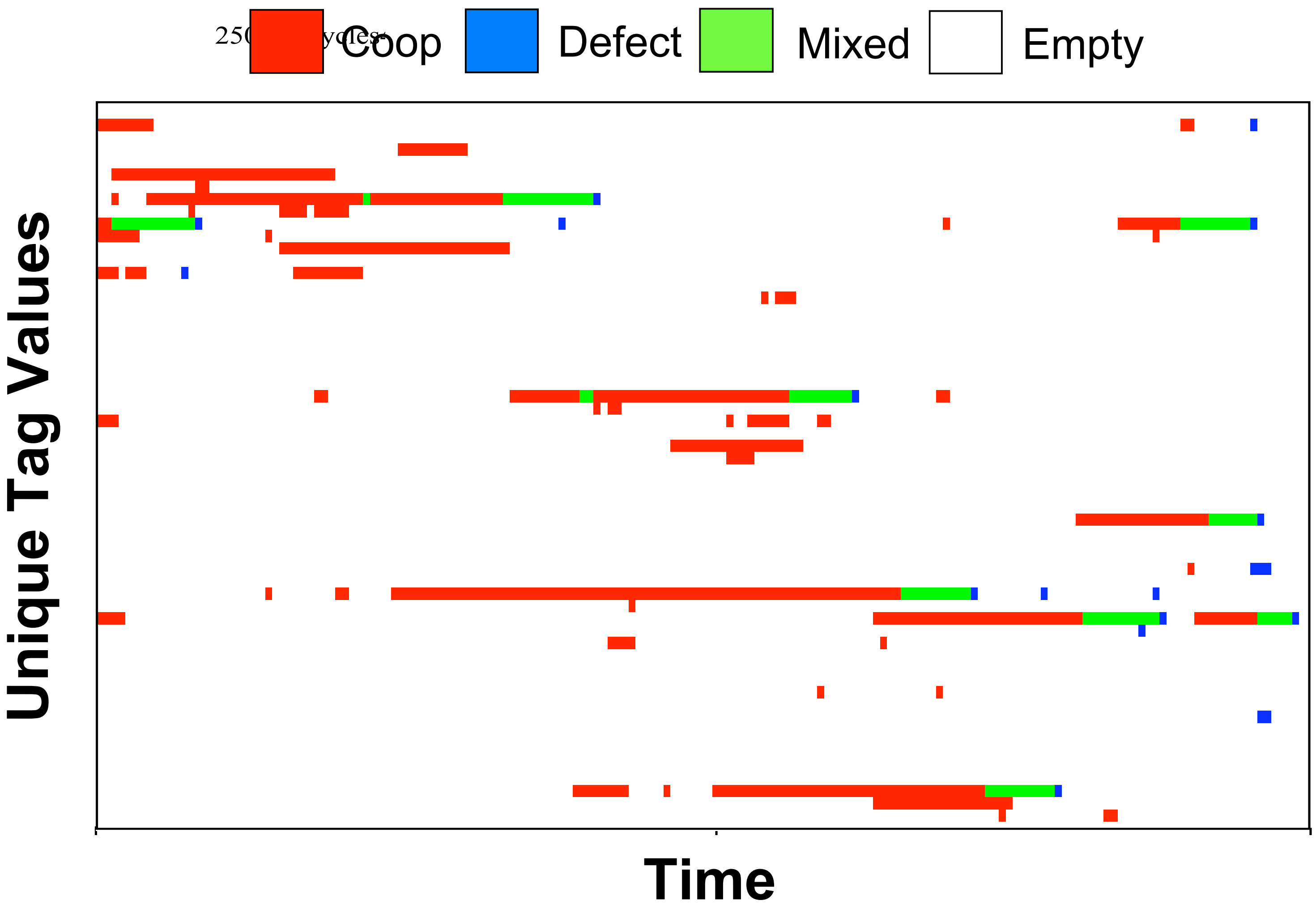
Mutation of tag

Copy tag and strategy

Visualising the Process (Hales 2000)



Visualising the Process



A P2P Scenario

Consider a P2P:

- Assume nodes maintain some max. degree
- 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

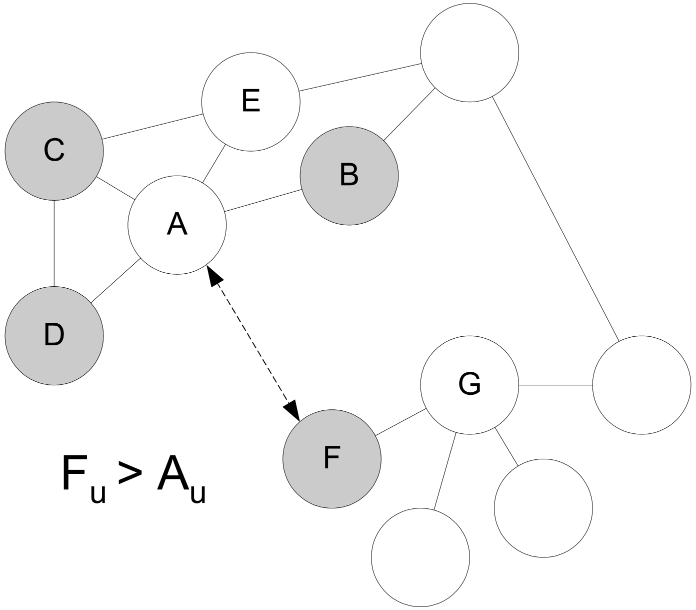
- Represent the P2P as a undirected graph
- Assume nodes are selfish and periodically:
  - Play PD with randomly selected neighbour
  - Compare performance to some randomly selected other node
  - If other node is doing better copy its neighbourhood and strategy
  - Mutate strategies and neighbourhood.

Design Decisions

- Mutation of view => replace all with single randomly chosen node
- Mutation of strategy = flip the strategy
- Node j copying a more successful node i => replace i view with j’s plus j itself
- When maximum degree of a node is exceeded throw away a randomly chosen link

Copying a more successful node

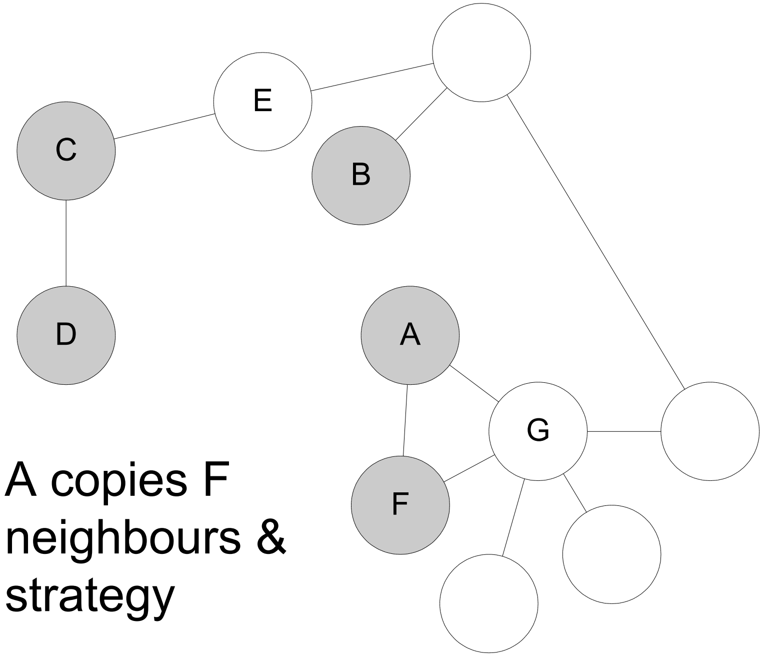
Before



$F_u > A_u$

Where A_u = average utility of node A

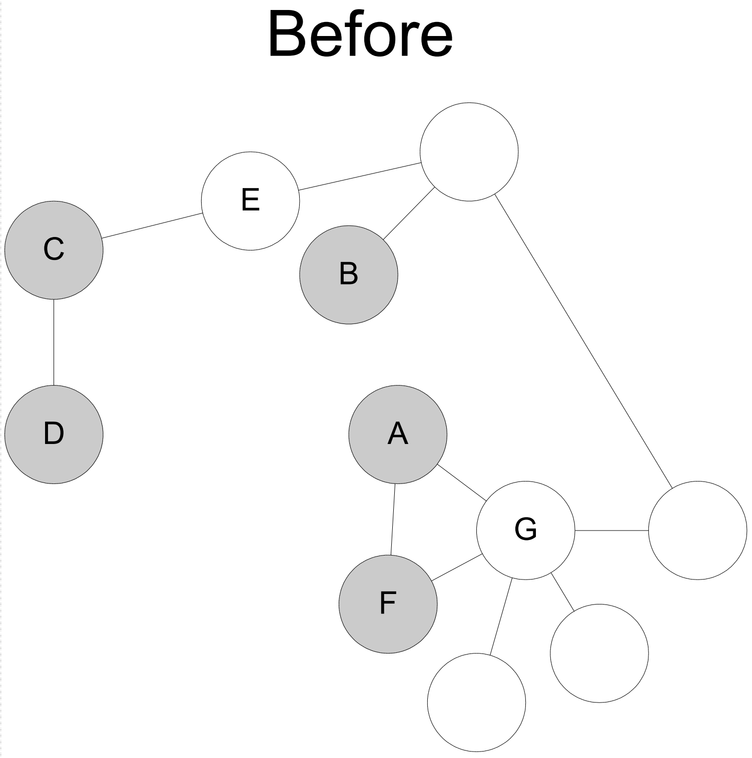
After



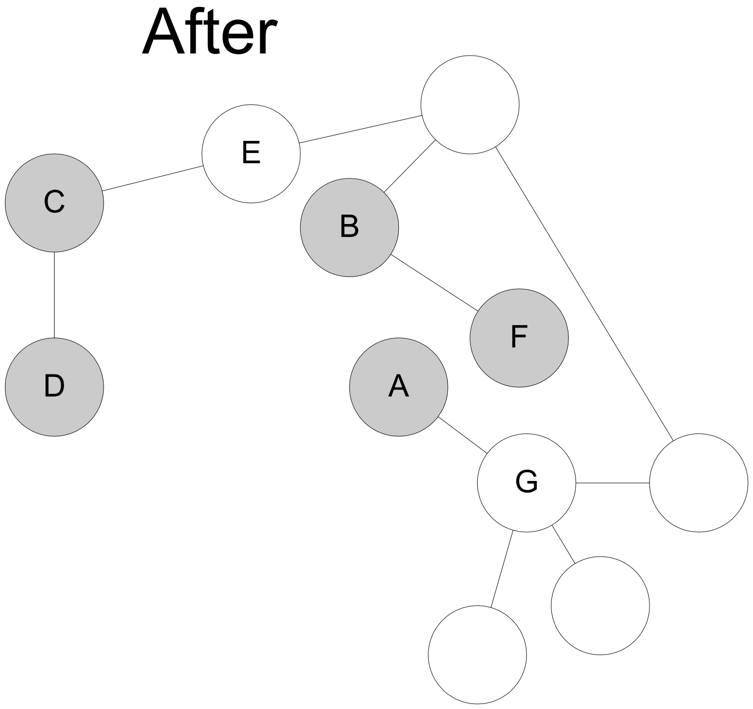
A copies F neighbours & strategy

In his case mutation has not changed anything

Random movement in the net



Mutation applied to F's neighbourhood



F is wired to a randomly selected node (B)

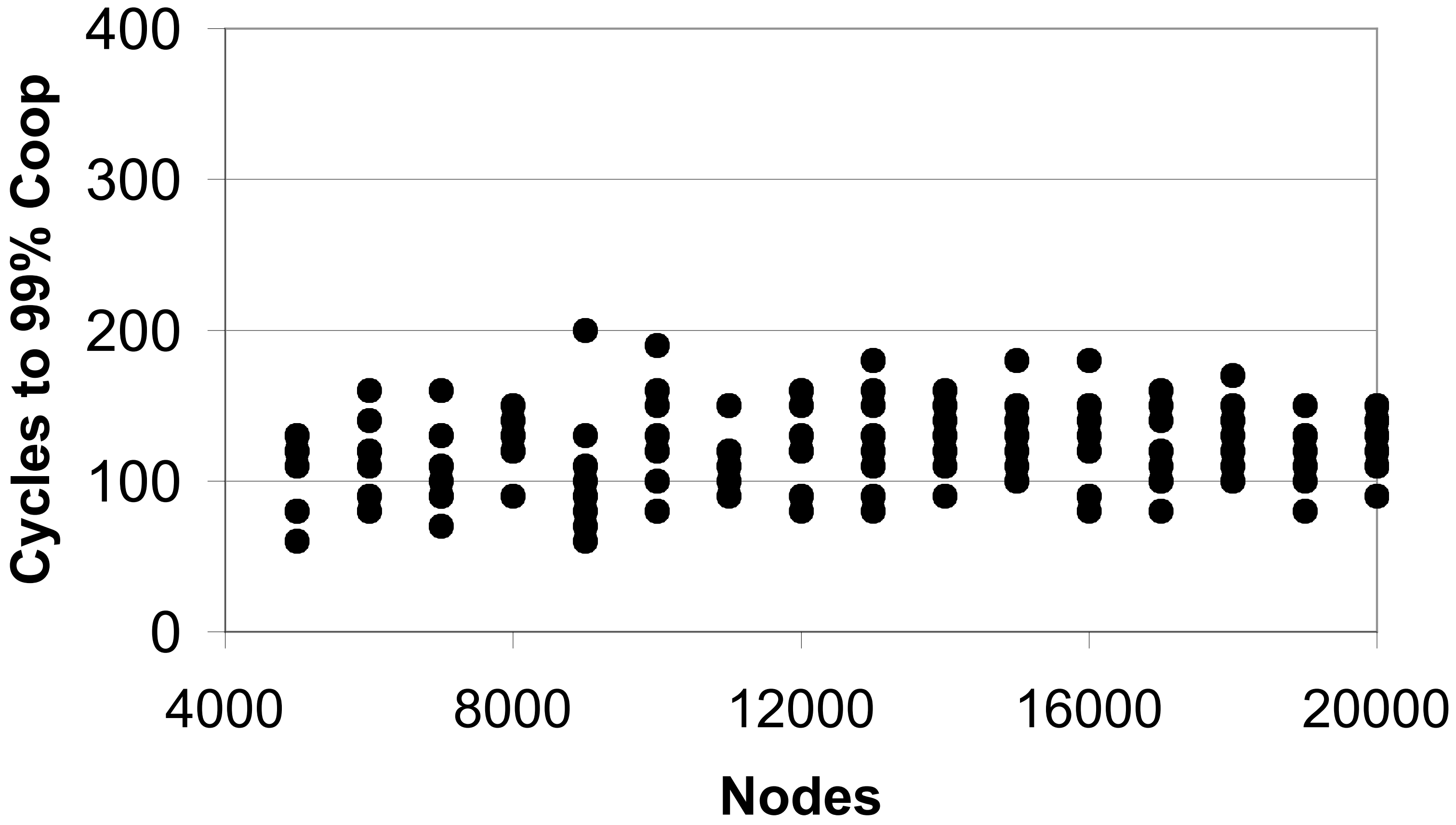
The Simulation Cycle

```
LOOP some number of generations
    LOOP for each node (i) in the population N
        Select a game partner node (j) randomly from view
        If view empty, link to random node i (mutate view)
        Agent (i) and (j) invoke their strategies and get
          appropriate payoff
    END LOOP
    Select (N / 2) random pairs of nodes (i, j) lower
      scoring node copies higher scoring node
    Apply mutation to view and strategy of each
      reproduced node with probability m
END LOOP
```

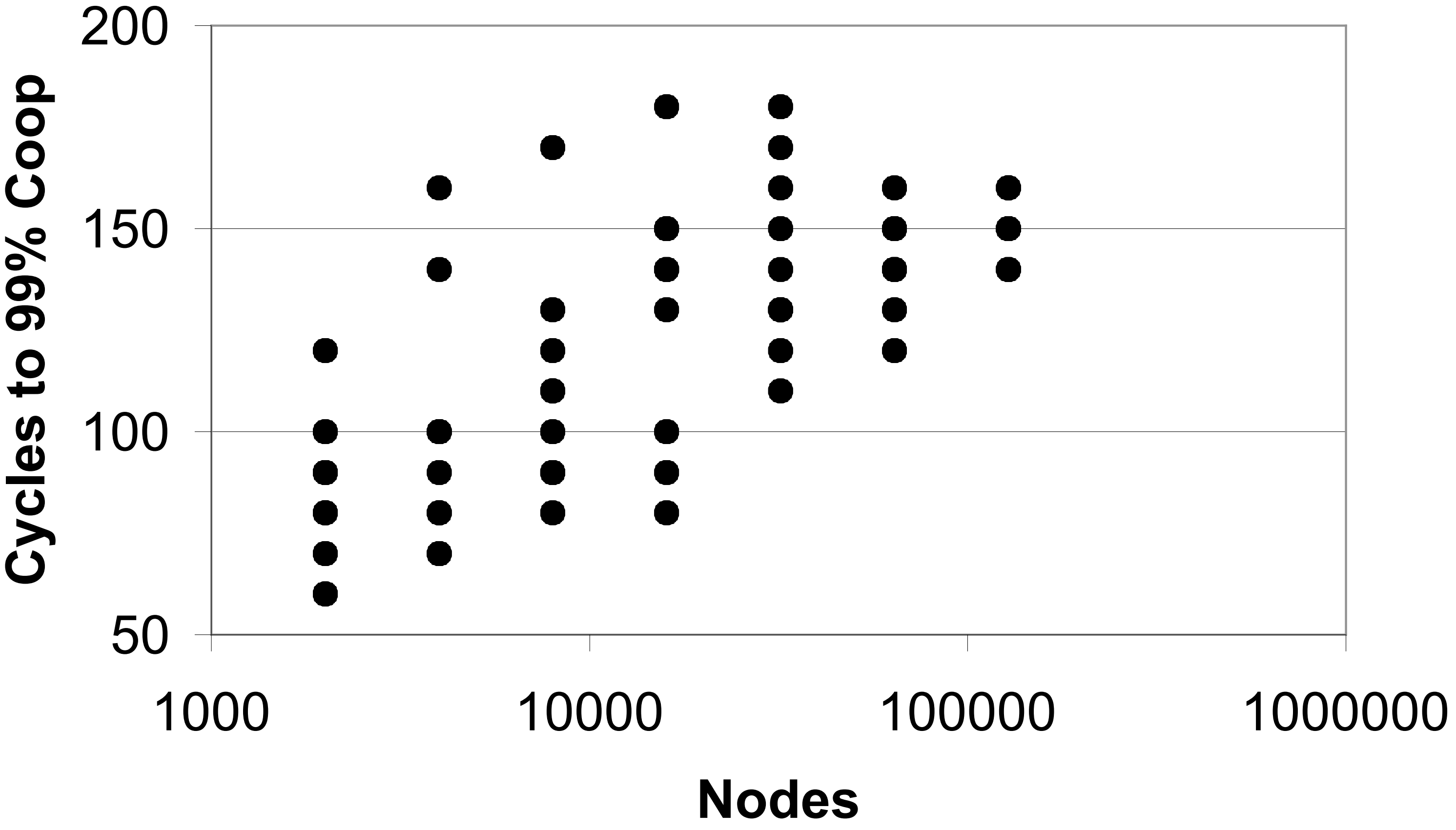
Parameters

- Vary N between 4,000..120,000
- Maximum degree 20
- Initial topology random graph (not important)
- 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)

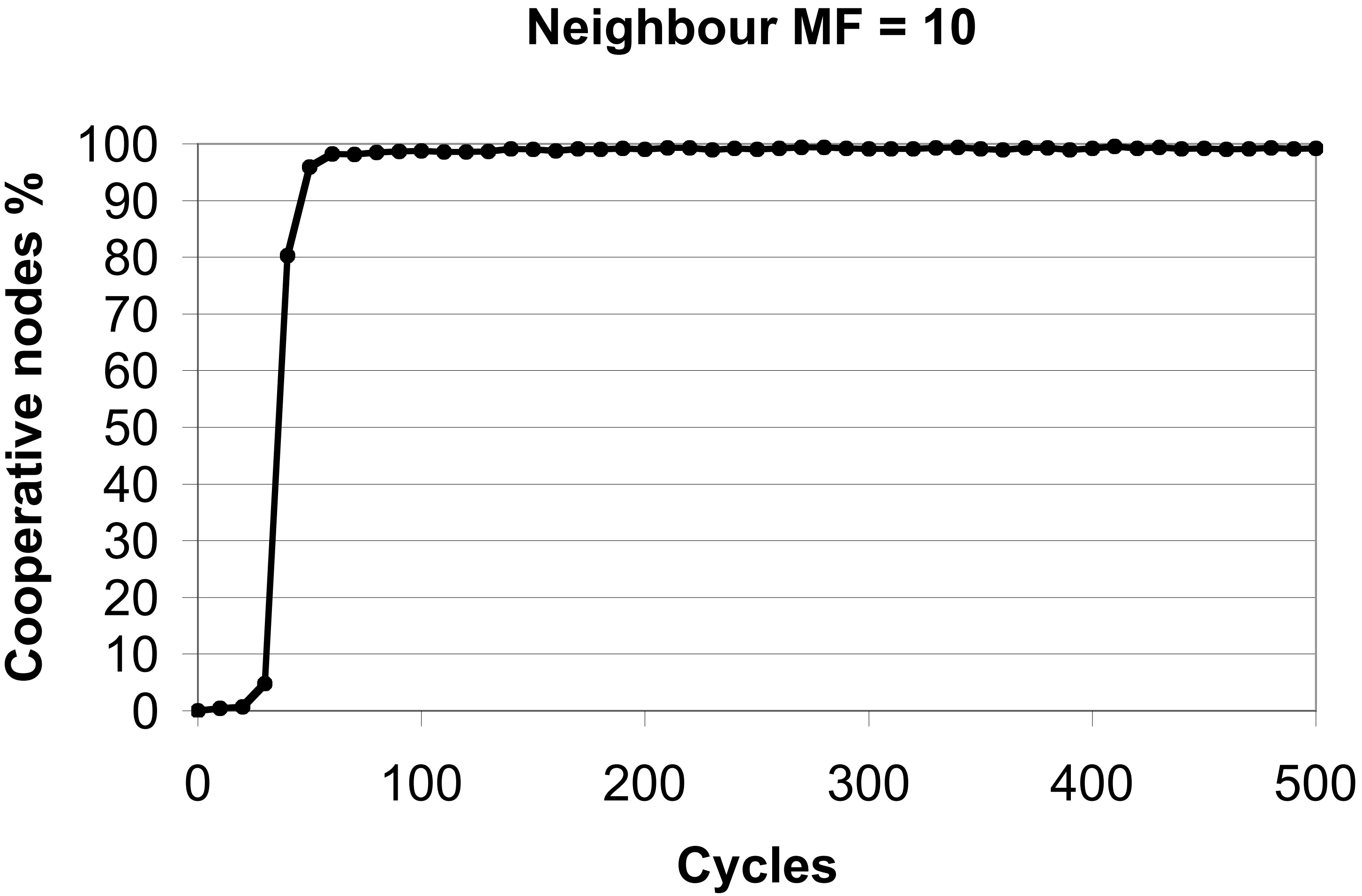
Results


400
300
200
100
0
4000
8000
12000
16000
20000
Cycles to 99% Coop
Nodes

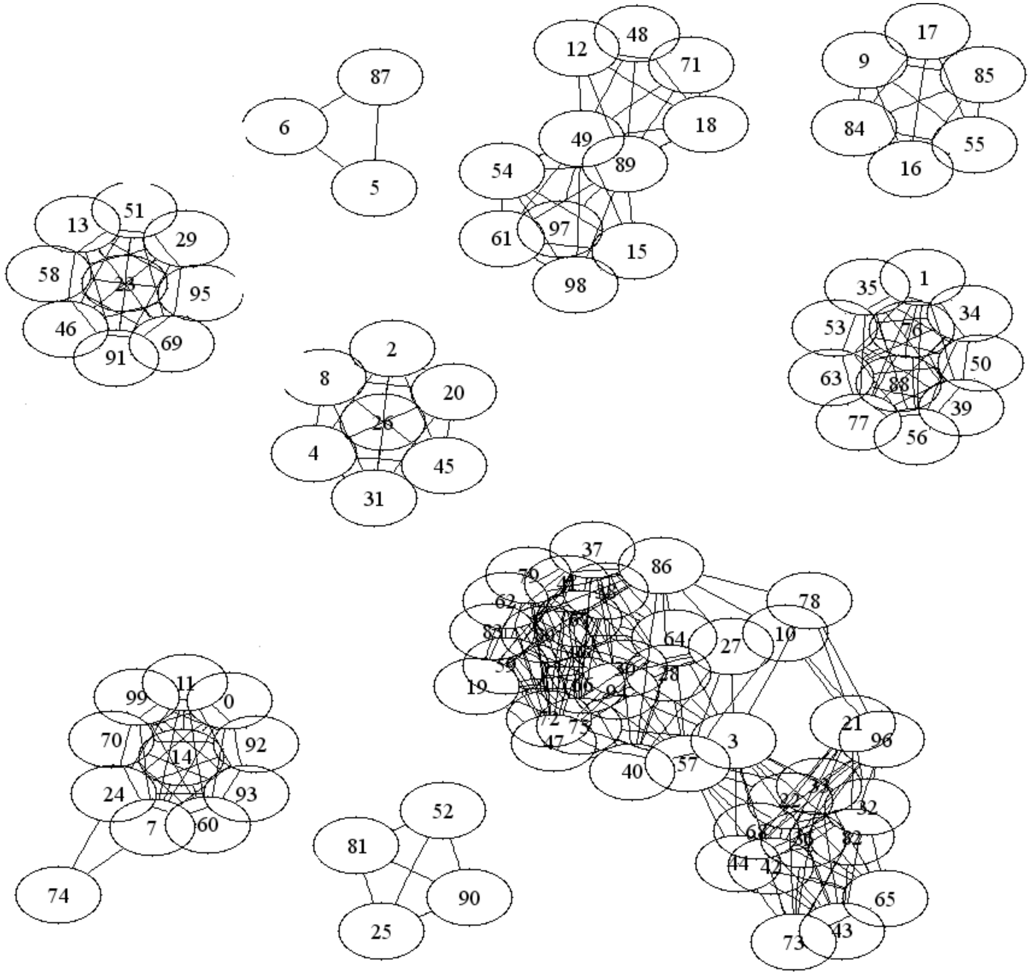
A few more nodes

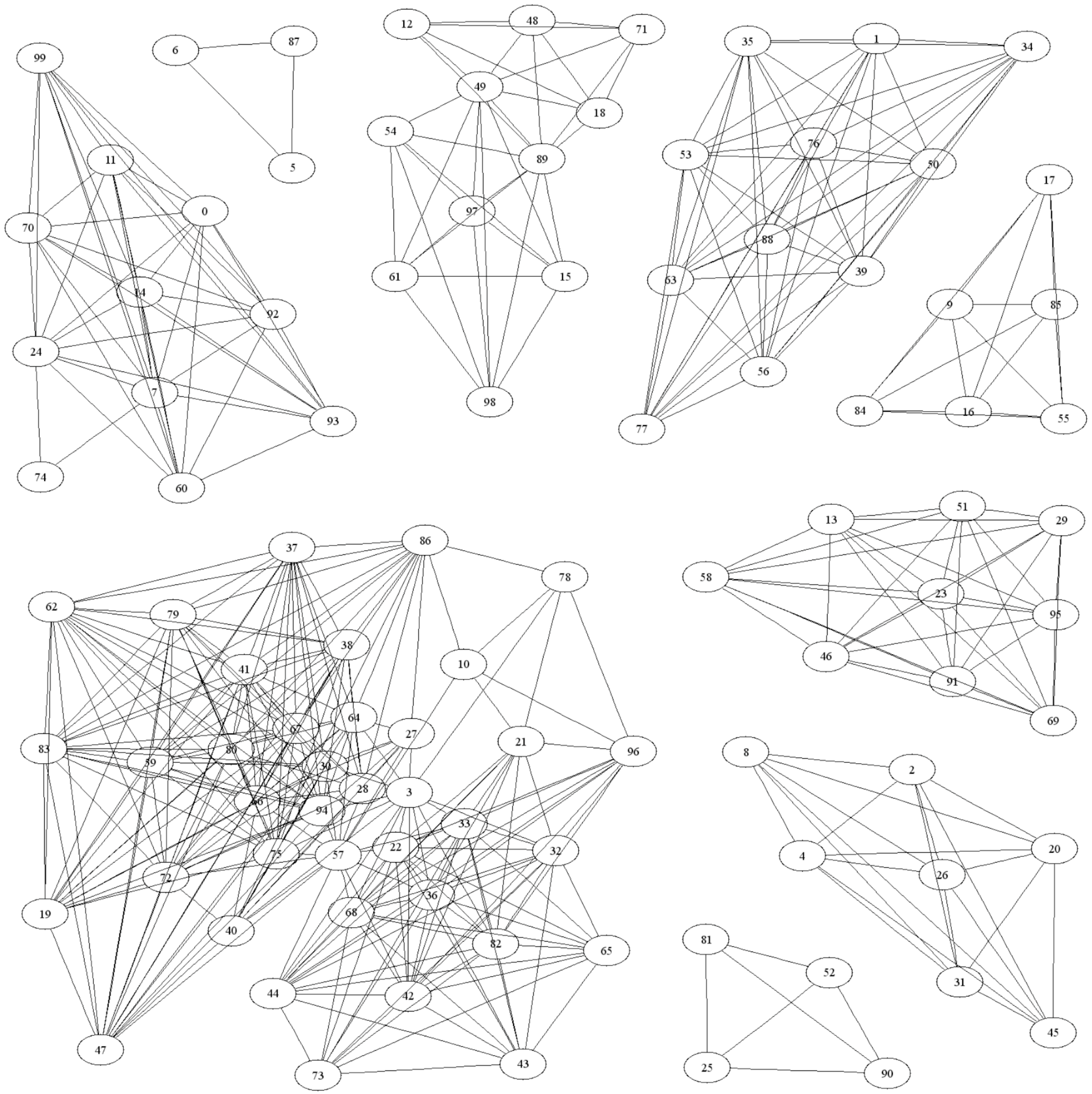


A typical run (10,000 nodes)



A 100 node example – after 500 generations

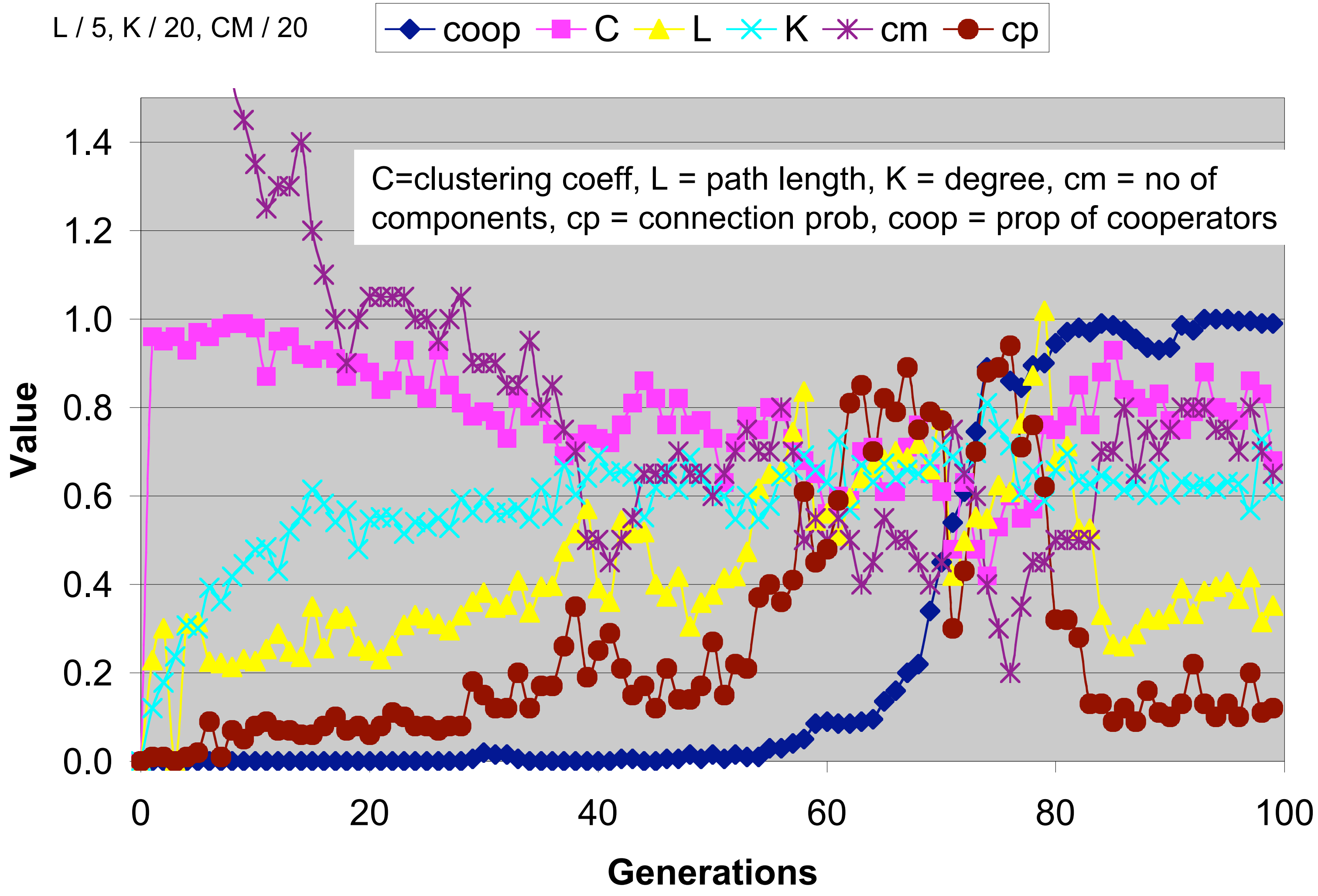



99
6
87
5
11
0
70
14
92
24
7
93
74
60
12
48
71
49
18
54
89
97
61
15
98
35
1
34
53
76
50
88
63
39
56
77
17
9
85
84
16
55
37
86
78
62
79
38
41
10
64
67
27
83
80
21
96
59
30
28
3
66
94
33
22
75
57
32
72
36
19
68
40
82
65
44
42
47
73
43
13
51
29
58
23
95
46
91
69
8
2
4
20
26
31
45
81
52
25
90

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 a set of unstable components exist - highly internally connected (L not much more than 1 and C very high)
- Constantly reforming and changing due to mutation and replication
- Rough characterisation of disconnectedness = prob. that two random nodes are connected

Typical run, 200 nodes



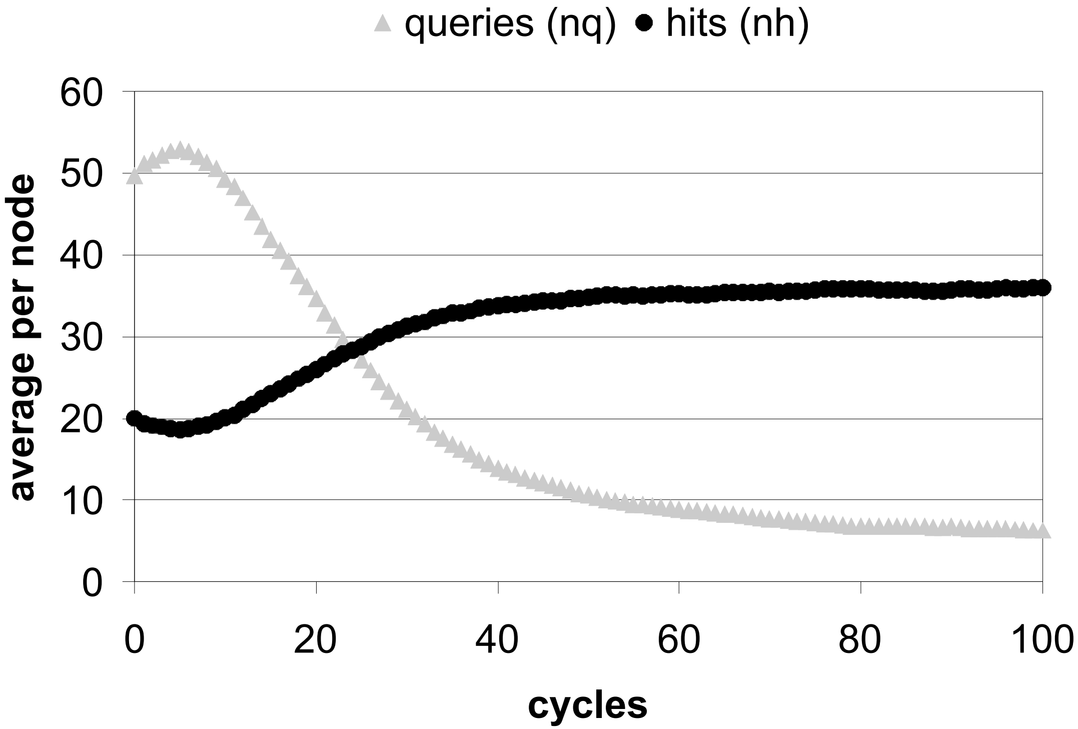
Next steps

- 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?

File Sharing Scenario

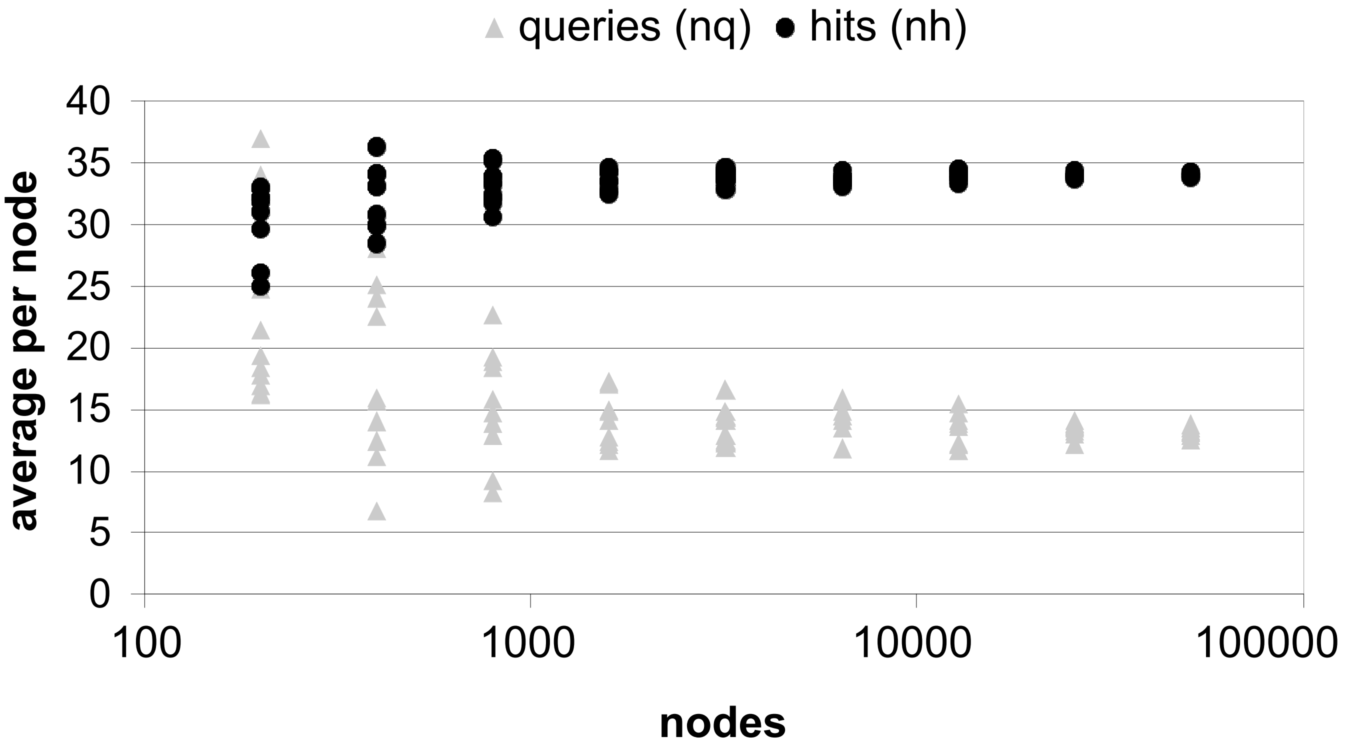
- Simplified form of that given by Q. Sun & H. Garcia-Molina 2004
- Each node has variable giving proportion of capacity (100 units) devoted to generating queries against answering them [0..1] 1=selfish, 0=altruism
- Each node has an answering power (prob. Of making a hit given any query =0.4 fixed)
- Flood fill query method, TTL's etc

Results



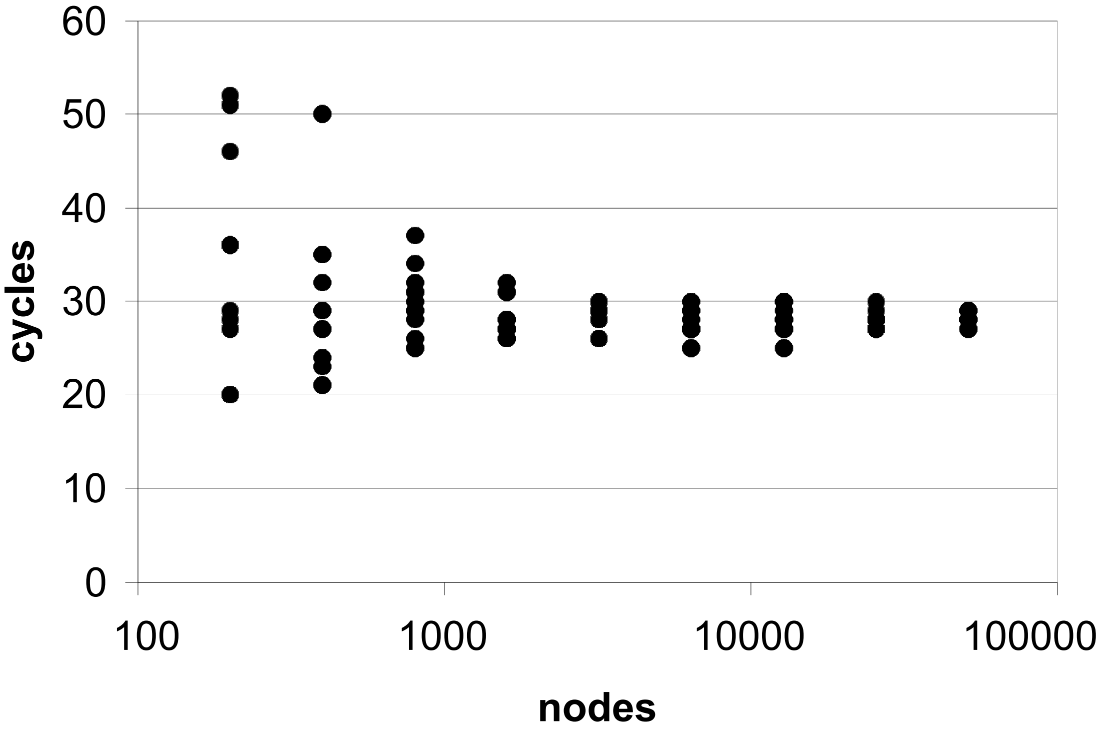
A typical run for a 10^4 node network

Results



Results showing number of queries (nq) and number of hits (nh) (averaged over cycle 40..50) for different network sizes (10 individual runs for each network size)

Results



Cycles to high hit values (number of hits nh > 30) for different network sizes (10 runs each)

What’s going on?

- A “Socially emergent incentive system” ?
- Selfish myopic behaviour causes nodes to migrate to more cooperative clusters and adopt cooperative strategies.
- Bad guys end-up alone or surrounded by other bad guys.
- *being a bad guy is not a sustainable strategy*
- However, at any given point in time a small number of bad guys are doing “better” than any good guys

Conclusion

- Tag-like dynamics using simple rewiring rules
- Free-riding low even though nodes are selfish
- No knowledge of past interaction required
- Scales well in tested domains
- But: produces many (dynamic) components
- What about whitewashers? Different churn rates? Hyper-rational or irrational behaviour? Copying links and strategies?