DELIS

Dynamically Evolving, Large-scale Information Systems

Simple Rewire Protocols for Cooperation in Dynamic Networks

David Hales, Stefano Arteconi, Ozalp Babaoglu

University of Bologna, Italy

Bio-Inspired Workshop, Rome, 25th July 2006

Information Society Technologies

Evolutionary Models

- Recent evolutionary models demonstrate desirable properties of cooperation and coordination
- Based on ideas coming from evolutionary / bounded rationality approaches (Simon, Arthur, Axelrod et al)
- Such models relax assumptions of “ideal” rationality
- Consider agents operate using simple heuristics
- Often collective learning via a (cultural) evolutionary approach
- The idea that (potentially random) innovations in agents are copied by others (in some way) if they improve utility (defined in some way)

Evolutionary Models

- Such models capture self-organising and emergent processes
- Argued: similar to those that occur in human or animal societies
- Computational Social Science using agent-based simulation
- Obviously controversial, rarely validated
- Yet increasingly accepted as alternative to equilibrium analysis / ideal rational approaches
- More applicable to engineering applications - noise, incomplete information, high dynamicity, heterogeneous agents etc.
- ***Side-stepping controversy and validity of such models, can we steal and adapt these ideas for "engineering" of desirable properties in distributed systems?***

Peer-to-Peer Systems

- We have translated some of these models into protocols for use in peer-to-peer (P2P) systems
- P2P are generally open systems of client programs running on user machines with no central authority or control
- Electronically mediated and semi-automated social systems
- Some general motivating questions are:
  - How can such systems come to self-organise, cooperate and coordinate to produce productive behaviour?
  - How can the negative effects of free-riding and selfish behaviour be avoided - promote social good?
  - How can such systems scale well in a robust way?
  - How can the effects of malicious behaviour be minimised?

tag systems

- Previous “tag” models offer a simple mechanism by which some of these questions can be addressed
- Both cooperation and coordination (specialisation)

Self-Organising Cooperation in Peer-to-Peer Systems

- Algorithm based on social simulation models of "tags"
  - Introduced by Holland early 1990's
  - Developed recently by Riolo; Hales; Edmonds.
- Tags are observable "markings", labels or social cues, attached to agents (e.g. hairstyle, dress, accent)
- In an evolutionary algorithm tags evolve just like any other artificial gene in the "genotype"
- They are displayed directly in the "phenotype"
- When agents bias interactions towards those with similar tags, even selfish evolution selects for cooperative and altruistic behaviour

Self-Organising Cooperation in Peer-to-Peer Systems

We translated the tag algorithm into a network

- nodes move to find “better” neighbors
- producing a kind of evolution in the network
- “bad guys” become isolated

Results in a “duplicate and re-wire” rule

- Producing a kind of “group selection” between clusters
- a functional reason for temporal structures found in the “natural” networks?

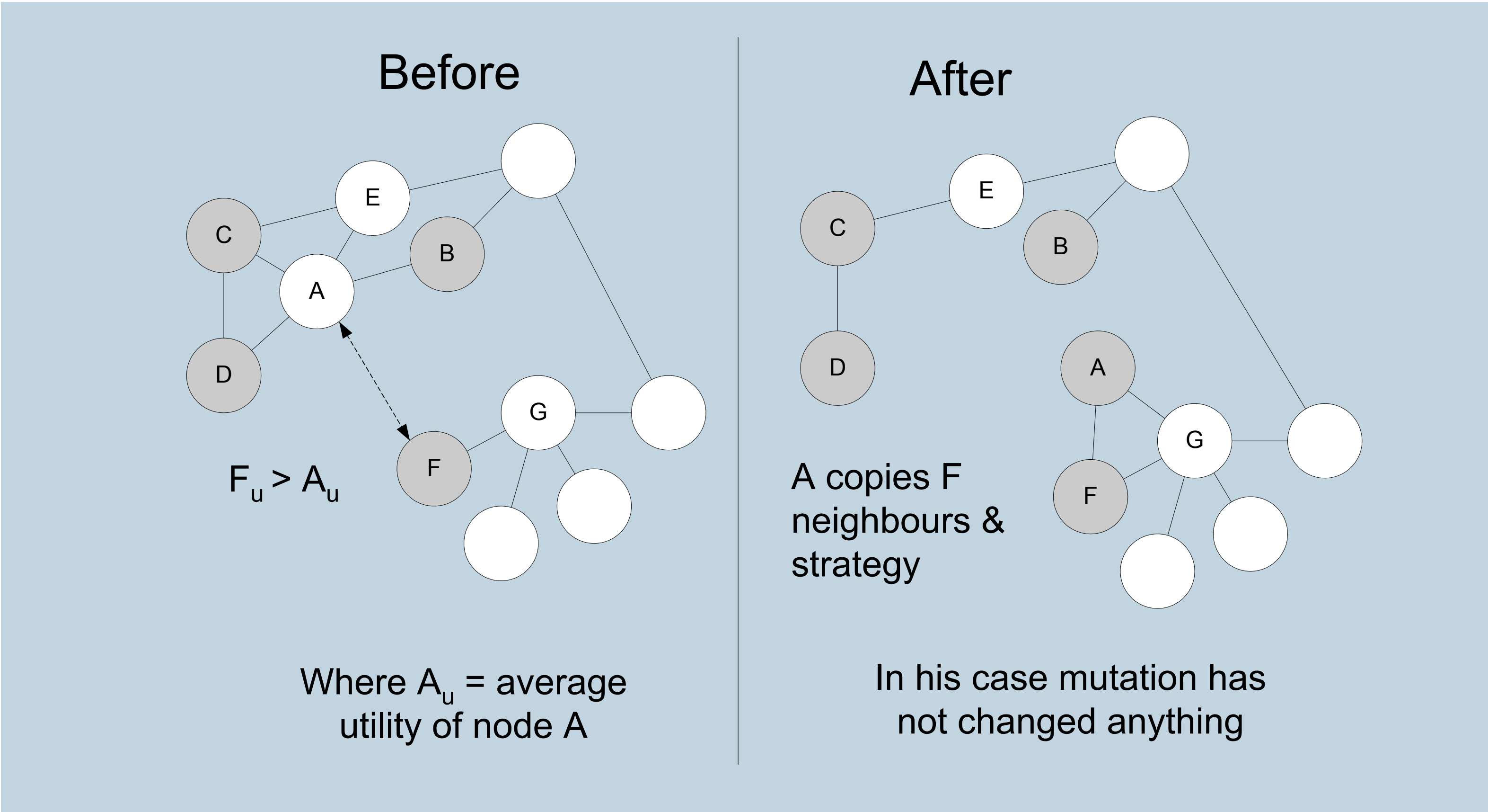
Self-Organising Cooperation in Peer-to-Peer Systems

Basic Algorithm

- Periodically **do**
  - Each node compare “utility” with a random node
  - **if** the other node has higher utility
    - copy that node’s strategy and links (reproduction)
    - mutate (with a small probability):
      change strategy (behavior)
      change neighborhood (links)
  - **fi**
- **od**

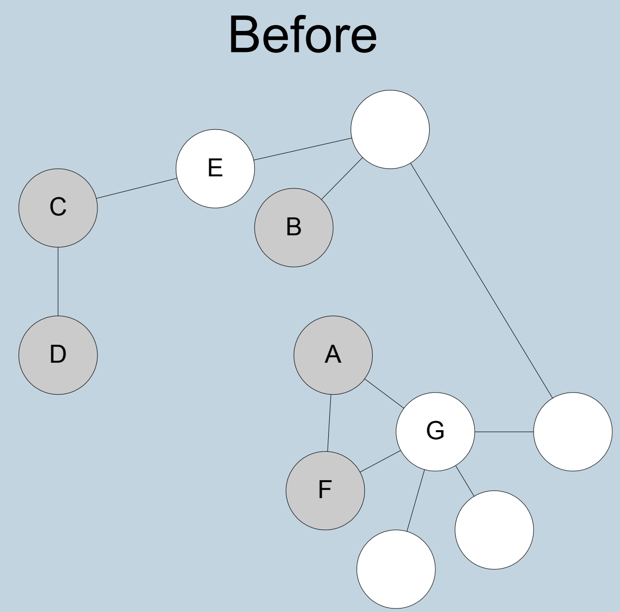
Self-Organising Cooperation in Peer-to-Peer Systems

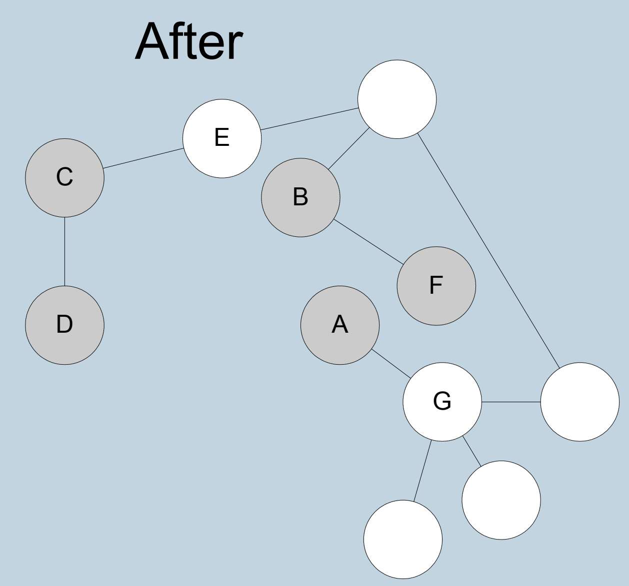
“Reproduction” = copying a more successful node



Self-Organising Cooperation in Peer-to-Peer Systems

"Mutation of the neighbourhood" = random movement in the net





Mutation applied to F's neighbourhood

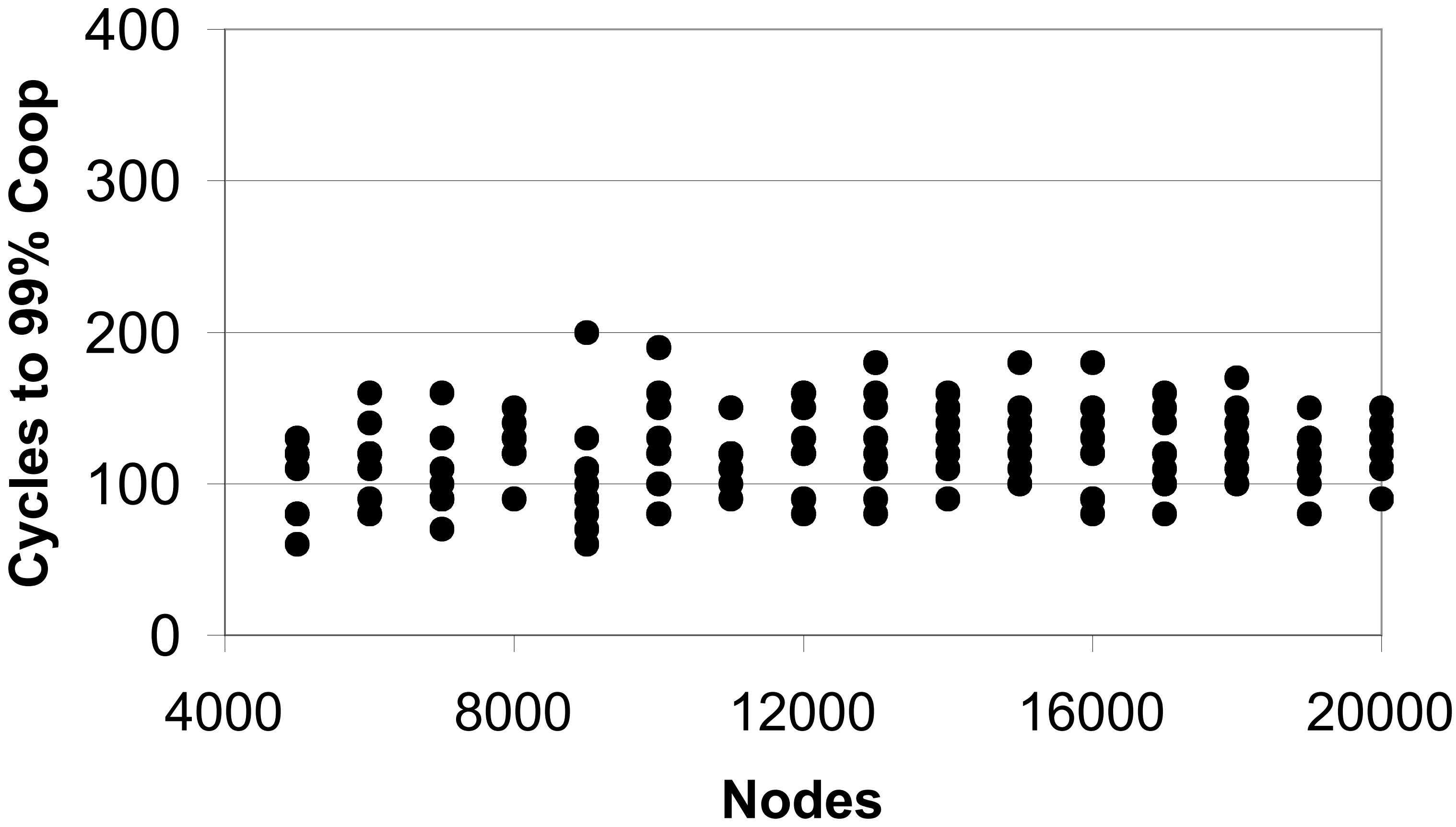
F is wired to a randomly selected node (B)

Self-Organising Cooperation in Peer-to-Peer Systems

Applied to a simulated Prisoner's Dilemma Scenario:

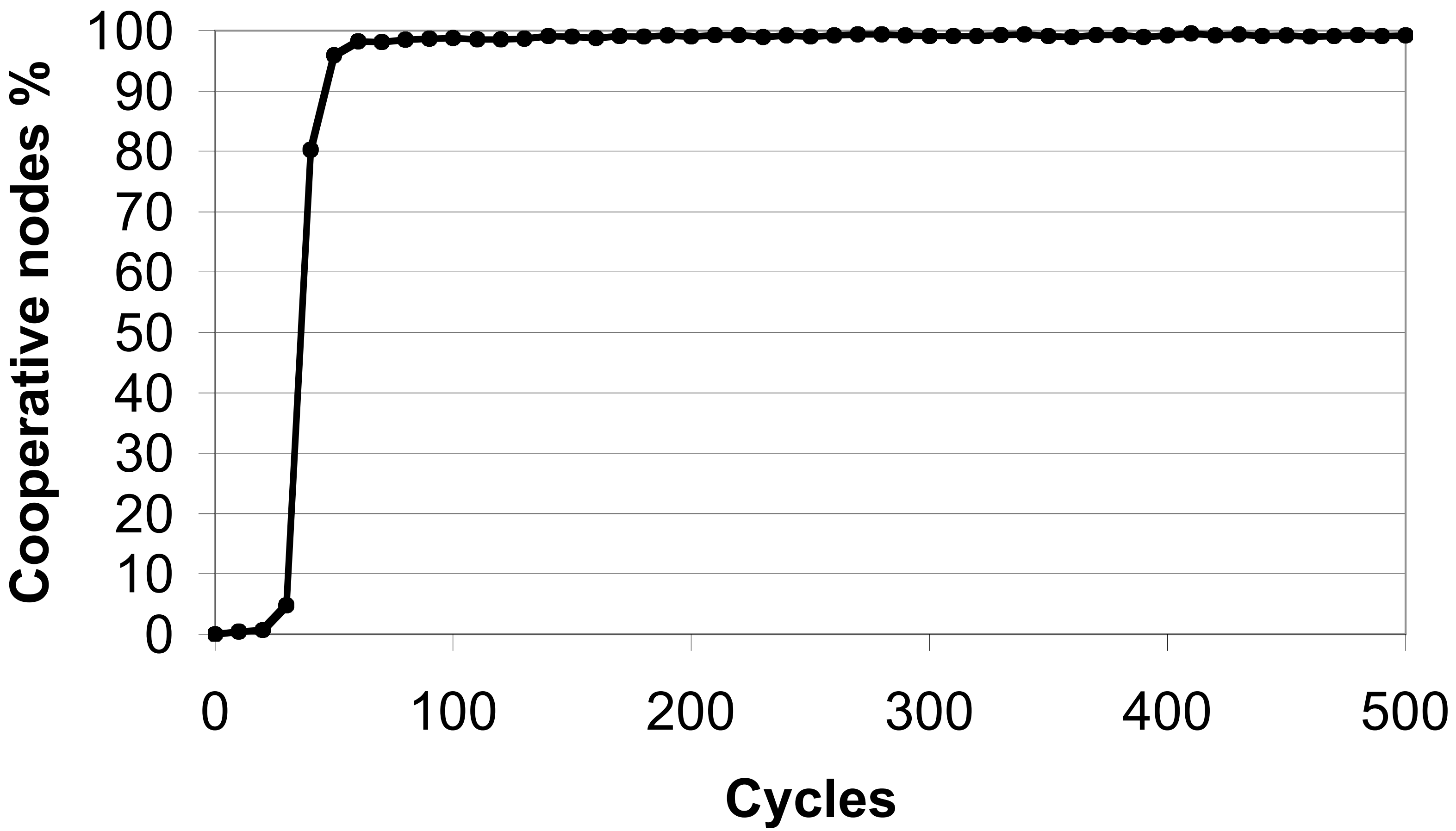
- Where selfish behavior produces poor performance – Nash Eq.
- Nodes store a pure strategy, either cooperate or defect
  - *Play the single round PD with randomly selected neighbours*
  - *Using their strategy*
- We take average payoff as the node utility
- Mutation of strategy: flip strategy
- Nodes randomly selected to play a random neighbours some number of times each period

Cycles to High Cooperation

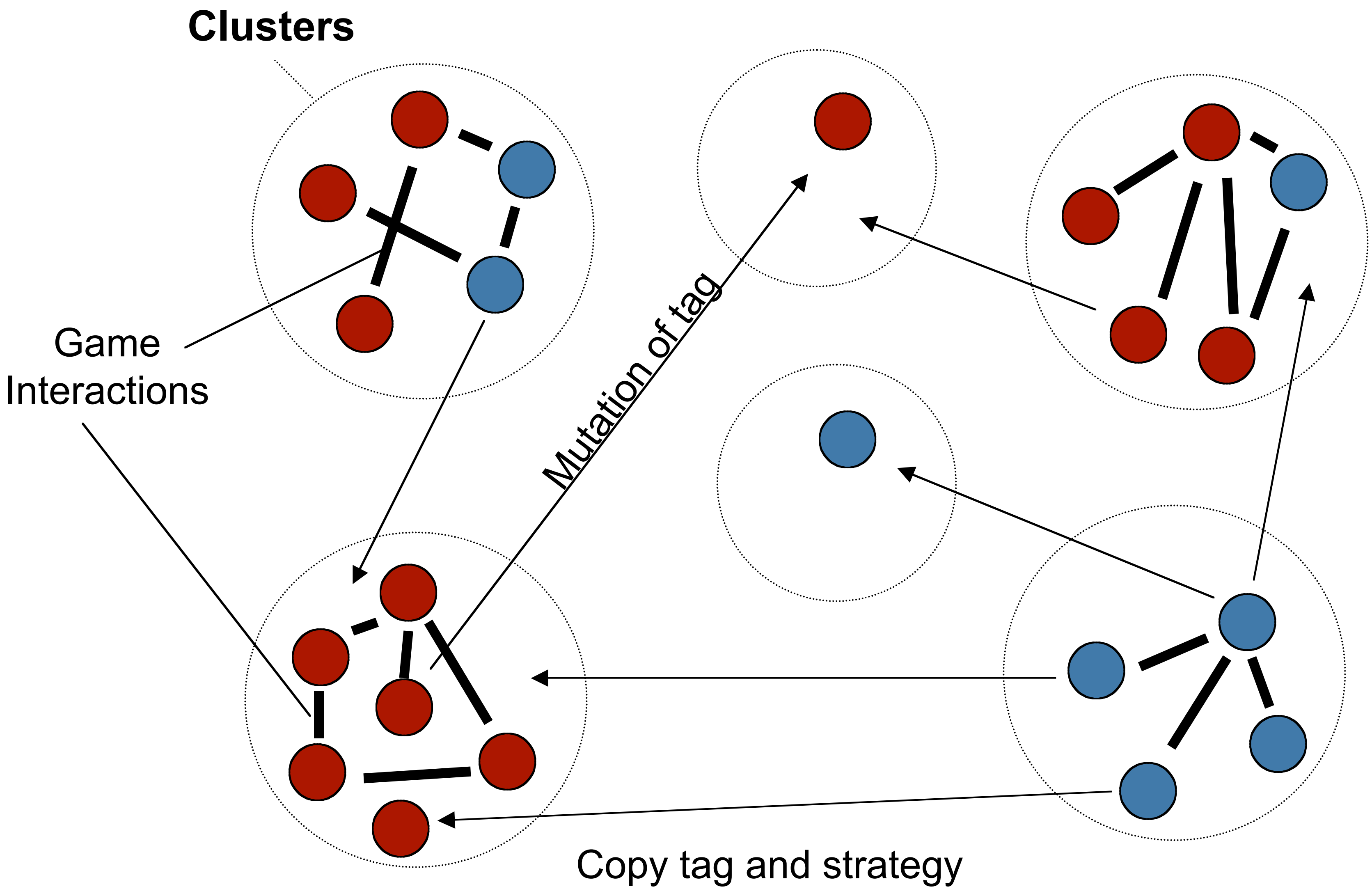


SLAC Applied to PD

Typical Individual Run



How Does SLAC Work?



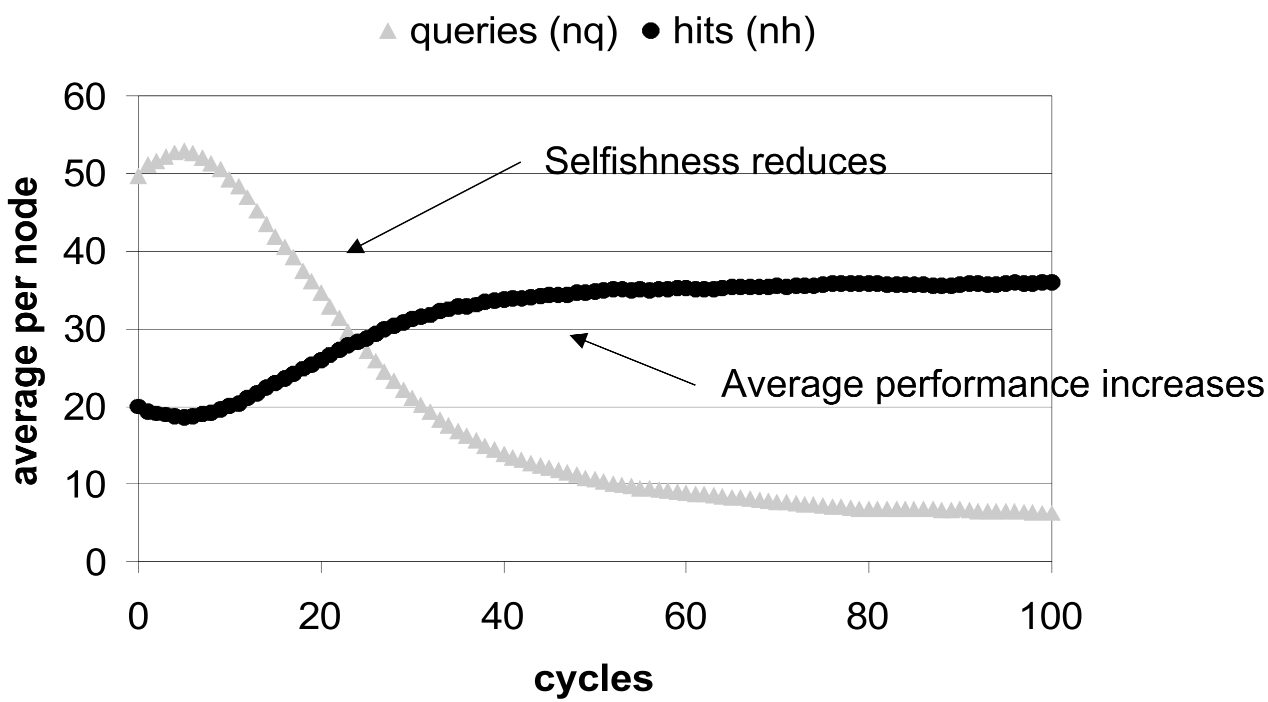
Self-Organising Cooperation in Peer-to-Peer Systems

Applied to a simulated P2P File Sharing Scenario:

- Simplified form of that given by Q. Sun & H. Garcia-Molina 2004
- Nodes control how much capacity devoted to generating or answering queries based on *P* = [0..1]
  - *P* =1.0 *selfish* (only generates queries)
  - *P* =0.0 *altruist* (only answers queries)
- We take as node utility the number of *hits*
- Mutation of strategy: change *P* randomly
- Flood fill query method, TTL's etc

Self-Organising Cooperation in Peer-to-Peer Systems

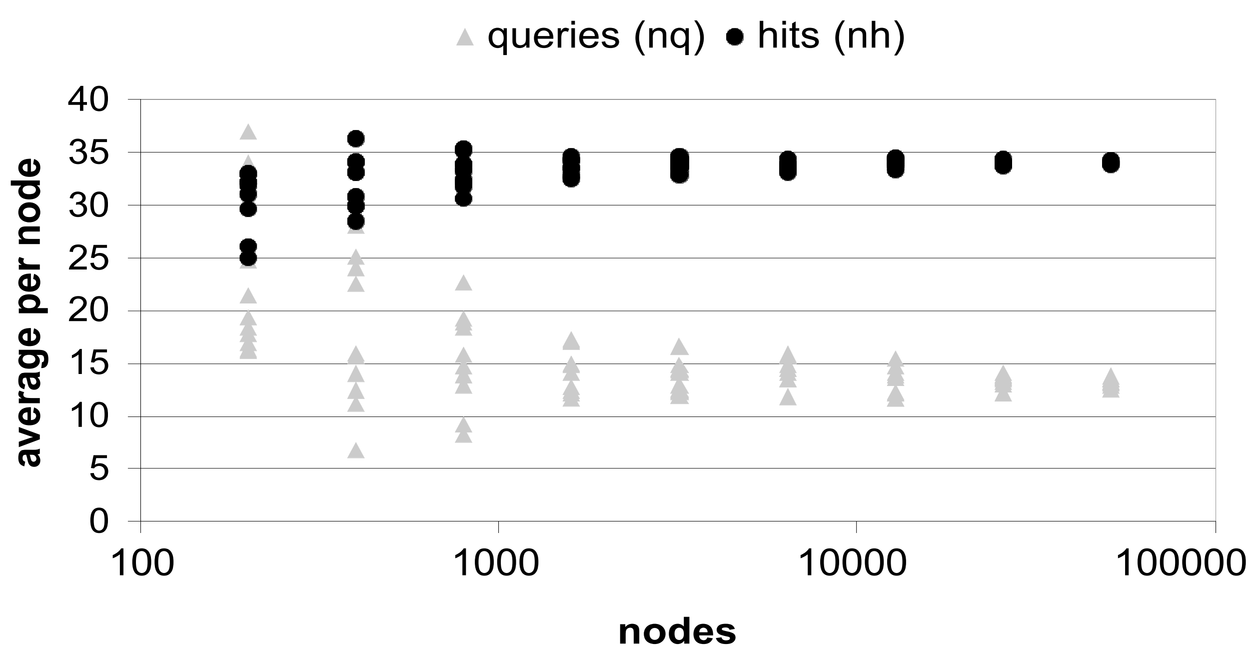
Some simulation results



A typical run for a 10^4 node network

Self-Organising Cooperation in Peer-to-Peer Systems

Some simulation results



Results showing number of queries (nq) and number of hits (nh) (averaged over cycle 40..50) for different network sizes with10 individual runs for each network size

SLAC to SLACER

- SLAC is OK for some tasks – as we have seen
- But produces disconnected components
- This is no good when we want
- An “Artificial Friendship Network” to span the network
- Connected – such that all nodes are linked with short path
- Chains of trust between all nodes – preferably short also
- To achieve this we modify SLAC and introduce SLACER

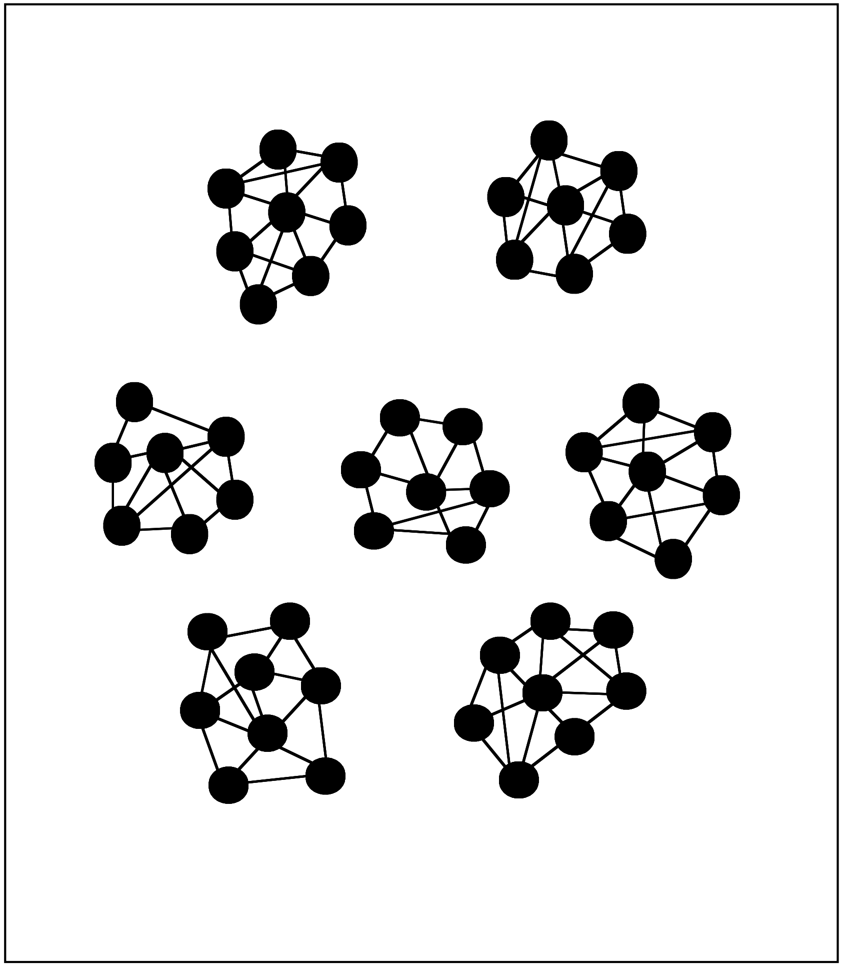
SLACER algorithm

David Hales (University of Bologna)

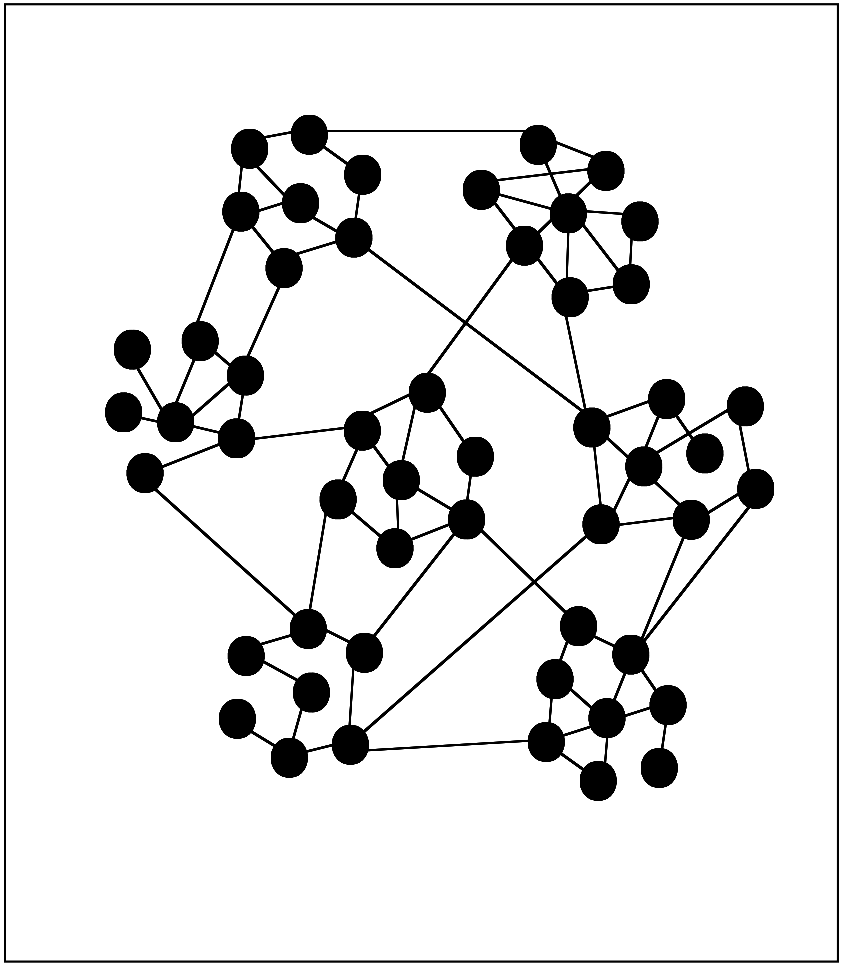
Basic Algorithm

- Periodically **do**
  - Each node compare “utility” with a random node
  - **if** the other node has higher utility
    - copy that node’s strategy and links, *probabilistically retaining some existing links*
    - mutate (with a small probability):
      change strategy (behavior)
      change neighborhood (links), *probabilistically retaining some existing links*
  - **fi**
- **od**

SLAC to SLACER

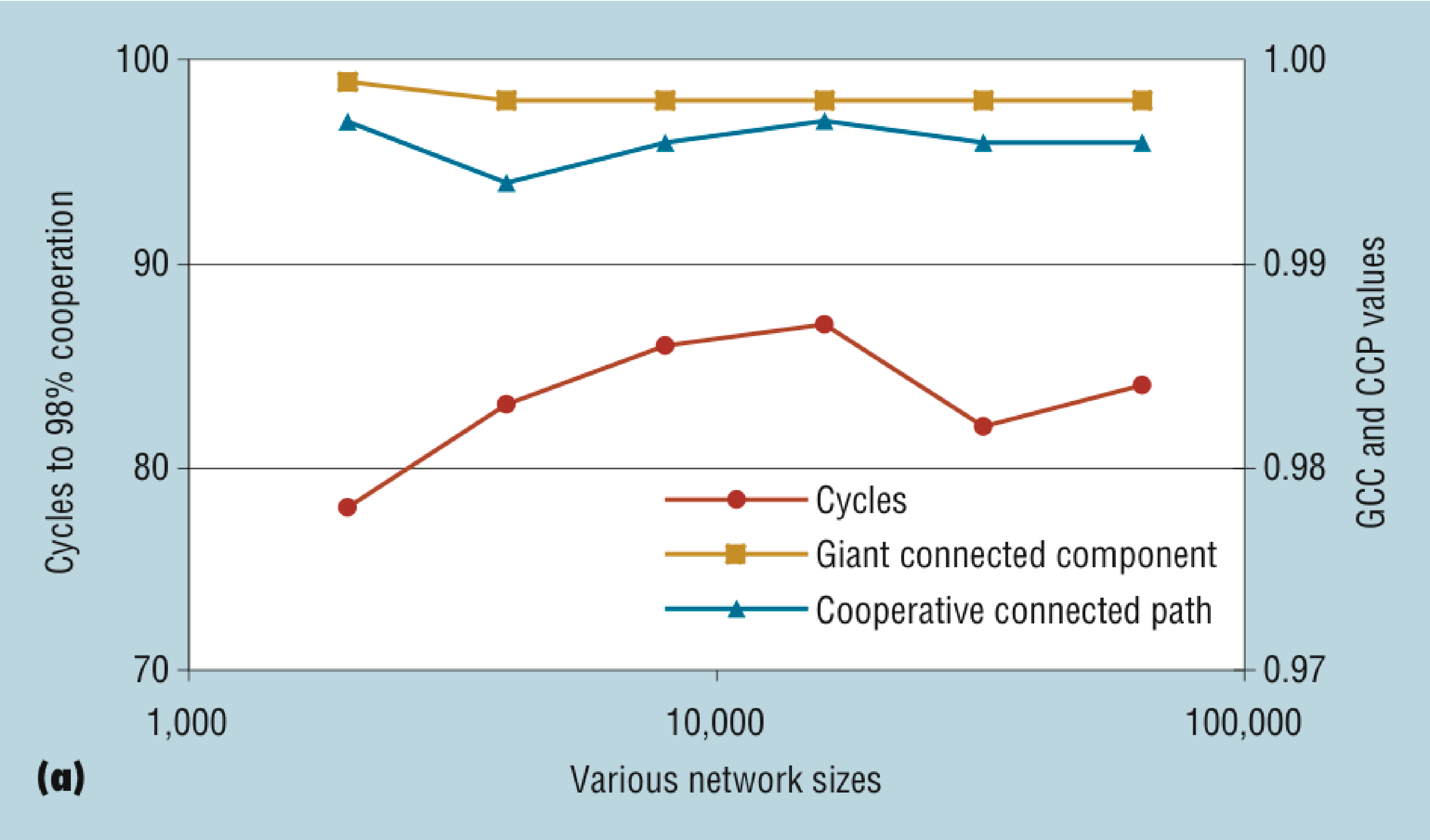


SLAC



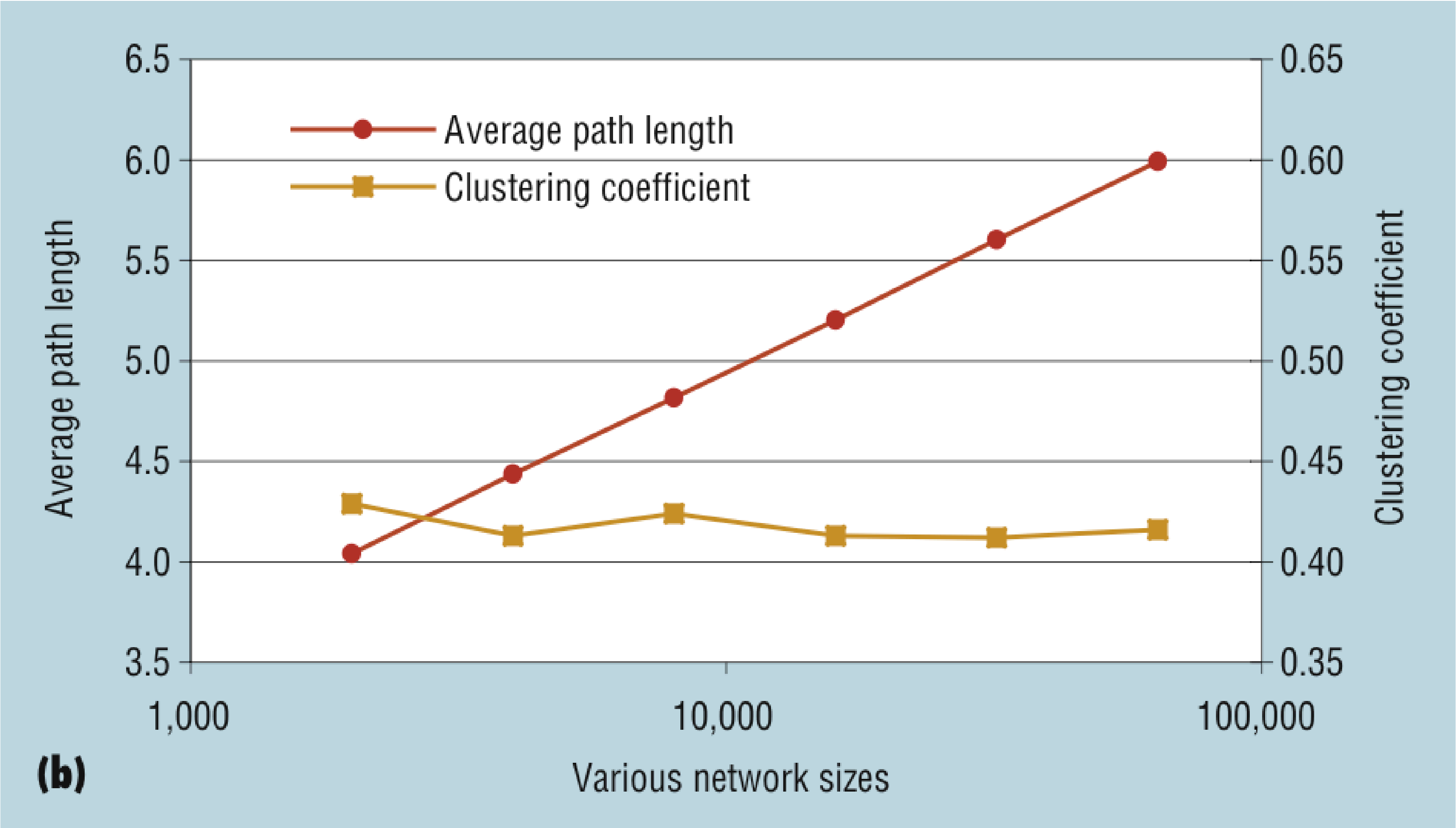
SLACER

SLACER – Some Results

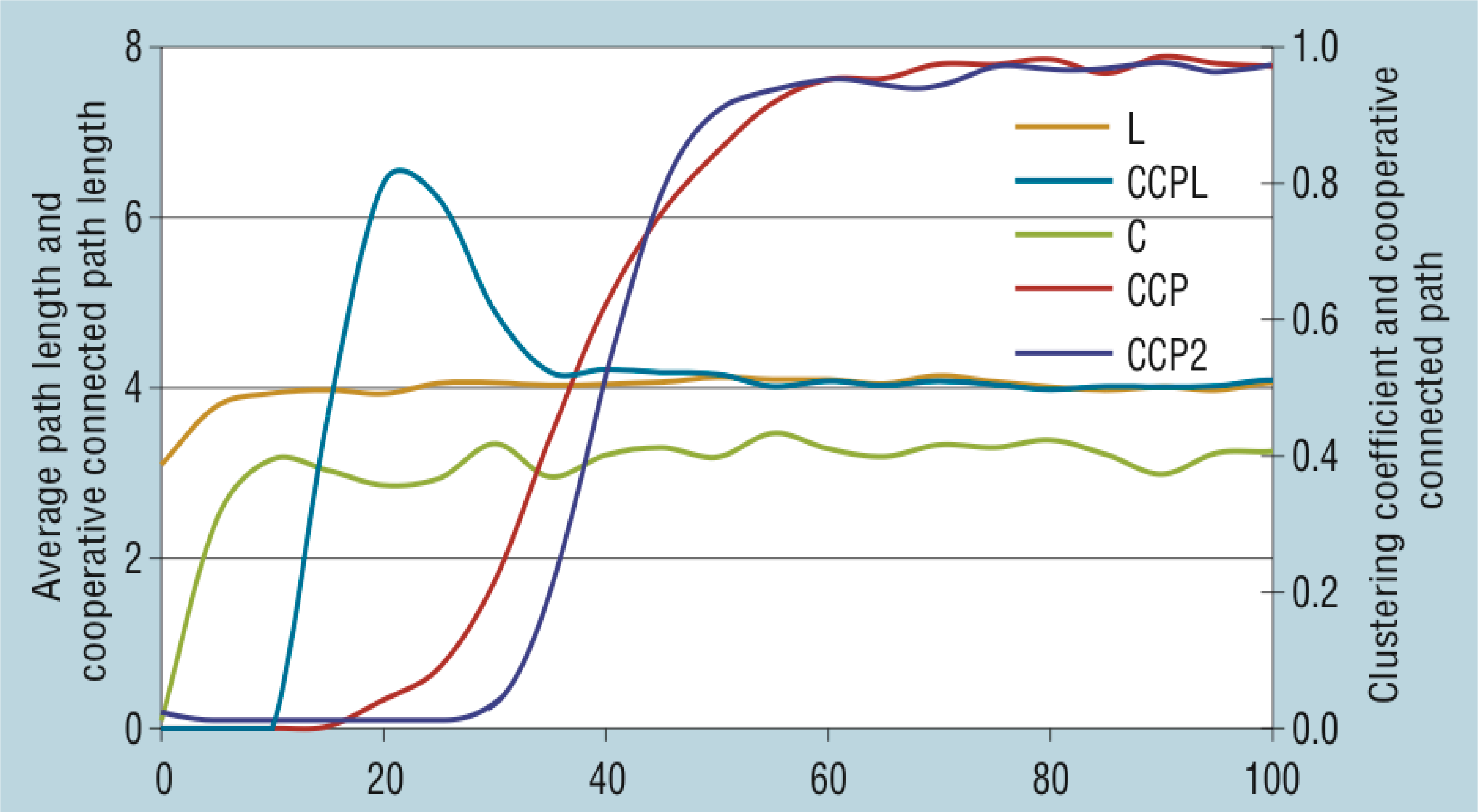


SLACER – Some Results

David Hales (University of Bologna)



SLACER - Some Results



Cycles

SLACER – Future Applications

- By establishing a fully connected "Artificial Social Network" (ASN)
- This can be used as input to existing P2P applications
- Specifically those that assume or require trusted social networks as input
- Currently harvested from e-mail contacts or "buddy lists" in chat applications
- Example: Collective spam filtering:
- *J. S. Kong, P. O. Boykin, B. Rezei, N. Sarshar, and V. Roychowdhury, "Let you cyberalter ego share information and manage spam," 2005. Available as pre-print: http://xxx.lanl.gov/abs/physics/0504026.*

Conclusion

- Simple copy and rewire algorithm
- No need for centralized trust or enforcement mechanism
- No need for knowledge of past interactions
- Process cooperative behavior even when nodes behave in an egotistical way, locally and greedy optimizing
- Works through a kind of “group selection” – “tribal selection”
- Can produce trusted and cooperative Artificial Social Networks
- Could be applied to existing protocols with minor modification
- Available on open source P2P simulation platform Peersim.

Self-Organising Cooperation in Peer-to-Peer Systems

References

- *Hales & Edmonds (2005) "Applying a socially-inspired technique (tags) to improve cooperation in P2P Networks", IEEE Transactions on Systems, Man, and Cybernetics, Part A*
- *Hales & Arteconi (2006) "SLACER: A Self-Organizing Protocol for Coordination in P2P Networks", IEEE Intelligent Systems, 21(2):29-35, March / April 2006.*

www.davidhales.com

peersim.sourceforge.net

- Fini

The End

Thank you!