Motifs in Evolving Cooperative Networks Look Like Protein Structure Networks^{*}

David Hales and Stefano Arteconi

University of Bologna, Dept. of Computer Science, Mura Anteo Zamboni 7, 40127 Bologna, Italy. {hales@cs.unibo.it | arteconi@cs.unibo.it}

Summary. The structure of networks can be characterized by the frequency of different subnetwork patterns found within them. Where these frequencies deviate from what would be expected in random networks they are termed "motifs" of the network. Interestingly, it is often found that, networks performing similar functions evidence similar motif frequencies. We present results from a motif analysis of networks produced by peer-to-peer protocols that support cooperation between selfish nodes. We were surprised to find that their motif profiles match closely protein structure networks. It is currently an open issue as to why this is.

1 Introduction

In previous works we presented a copy and re-wire peer-to-peer (P2P) protocol that promoted cooperation between connected nodes even when they had incentives to behave selfishly [5]. The protocol structures the population into competing 'tribes' that, through a group-like selection process, lead to socially beneficial behaviour even when the individual nodes behave in an essentially selfish way - copying the behaviours and links of other nodes that outperform them.

We tested the protocol by having nodes play the Prisoner's Dilemma game (PD) - a canonical game for exploring situations in which collective and individual interests diverge. We called this protocol SLAC (Selfish Link Adaptation for Cooperation) and applied it to a simulated file-sharing scenario demonstrating it had the ability to control the outbreak of selfish behaviour by nodes (downloading without uploading, so called "leeching") [4].

Although the SLAC algorithm performed well for certain task domains it produces networks with many disconnected components. Certain kinds of task require fully connected networks, for example a broadcast task that requires a single node to send a message to all nodes in the network, collective spam

^{*} Partially supported by the EU within the 6th Framework Programme under contract 001907 "Dynamically Evolving, Large Scale Information Systems" (DELIS).

filtering [8] or improving distributed hash table (DHT) performance [9]. In order to address this limitation of SLAC we modified the protocol such that the copying and re-wiring of links follows a probabilistic rule. In this way there is a probability that old links are retained when nodes re-wire (move) within the network. We called this new protocol SLACER (Selfish Link Adaptation Excluding Rewiring) [2].

In order to examine the structure of the networks produced by the SLAC and SLACER protocols we applied a motif analysis technique which produces subgraph ratio profiles (SRP) [10]. This approach allows for the classification of networks into similar functional groups - so called "superfamilies" of networks. We found that both the SLAC and SLACER protocols produce networks which fit within a superfamily of networks derived from protein structure networks.

It is not currently understood why networks with similar functions have similar SRPs. In this paper we do not attempt to answer why our P2P protocols have similar features to protein structure networks although this could be an interesting topic for further study.

We introduce the SLAC and SLACER protocols in section 2. We describe the PD cooperation game in section 3. The technique of motif analysis using SRPs is presented in 4. Finally we show the results of the analysis and give some brief observations and conclusions.

2 SLAC and SLACER protocols

The SLACER protocol follows a link based incentive approach. That is, nodes make and break links in the network in order to minimise the effect of selfish behaviour. Hence the topology itself reflects a network of cooperation.

Figure 1 shows the pseudocode. Over time, nodes engage in some application task and generate some measure of utility U. This utility is a numeric value that each node needs to calculate based on the specifics of the particular application domain. For example, this might be number of files downloaded, jobs processed or an inverse measure of spyware infections over some period. The higher the value of U the better the node believes it is performing in its target domain.

Periodically, each node *i* compares its performance against another node j, randomly selected from the population. If $Ui \leq Uj$ then *i* drops each of its current links to other nodes with high probability W, and copies all *j*'s links and adds a link to *j* itself. Additionally *i* then copies *j*'s strategy - the strategy codes some application level behaviour. After such a copy operation has occurred, then, with low probability M, node *i* adapts its strategy and with probability MR adapts its links. Adaptation involves the application of a "mutation" operation. Mutation of the links involves removing each existing link with probability W and adding a single link to a node randomly drawn from the network. Mutation of the strategy involves applying some form of

3

```
Periodically for node i:
select a random node j from the population
if utility Ui \le utility Uj then
copy strategy from j
drop each link from i with prob(W)
copy each link from j
link to j
with prob(M) mutate strategy of i
with prob(MR) mutate links of i
```

Fig. 1. The SLACER protocol pseudocode. Note that when W = 1 SLACER collapses into the SLAC protocol. For an overview of the protocol see the text but for more detail see [2].

change in application behaviour with probability M - the specifics of strategy mutation are dictated by the application domain (see later). After the periodic utility comparison, the node resets its utility to zero. Previous "tag" models, on which SLACER is based [4] have indicated that the rate of mutation applied to the links needs to be significantly higher than that applied to the strategy by about one order of magnitude hence MR >> M.

Each node is limited to a maximum number of links or neighbours (view size). If any SLACER operation causes a node to require an additional neighbour above this limit then a randomly selected existing link is removed to make space for the new link. Links are always undirected, hence symmetrical, so that if node i links to node j, then j must also maintain a link to node i and conversely if node i breaks a link to node j then node j also breaks its link to node i. In the work presented here each node has a maximum view size of 20 links.

When applied in a suitably large population the algorithm follows an evolutionary process in which nodes with high utility replace nodes with low utility. However, as will be seen, this does not lead to the dominance of selfish behaviour, as might be intuitively expected, because a form of social incentive mechanism results from the emergent network topology. This means that high utility but anti-social strategies, even though favoured by the individual nodes, do not dominate the population. The topology therefore guides the adaptation of the strategy away from anti-social selfish behaviours.

When the link drop probability W = 1 (see figure 1) SLACER collapses to the SLAC protocol producing highly cooperative yet disconnected networks. However, when W is slightly reduced, SLACER produces networks in which almost all nodes are members of a giant connected and cooperative component. SLACER networks are also small-world, with a low average path length between nodes growing logarithmically and a high clustering coefficient remaining constant with respect to the size of the network. Both SLAC and SLACER networks promote cooperation between nodes in a number of tested scenarios. In addition they have desirable properties of scalability, self-organisation and robustness. This means that if nodes are inserted or removed, or links broken, the network quickly readjusts back into a cooperative state.

Although we have tested the protocols in P2P-like scenarios, such as file sharing [4] and cooperative job sharing [3] we evaluated them initially with an abstract cooperation game called the Prisoner's Dilemma. This is a canonical test for the emergence of cooperation between selfish individuals.

3 Prisoner's Dilemma

The two players single-round Prisoners Dilemma (PD) game captures a situation in which there is a contradiction between collective and individual selfinterest. Two players interact by selecting one of two choices: to "cooperate" (C) or "defect" (D). For the four possible outcomes of the game, players receive specified payoffs. Both players receive a reward (R) or a punishment (P)payoff for mutual cooperation and defection respectively. However, when individuals select different moves, different payoffs of temptation (T) and sucker (S) are awarded to the defector and the cooperator respectively (see figure 2a). Assuming that neither player can know in advance which move the other will make and wishes to maximise her own payoff, the dilemma is evident in the ranking of payoffs: T > R > P > S and the constraint that 2R > T + S. Although both players would prefer T, because its the highest payoff, only one can attain it in a single game. No player wants S because its the lowest payoff. No matter what the other player does, by selecting a D move a player always gets a higher score than it would have obtained if it had selected C. D is therefore the dominant strategy, hence an ideally rational player would always choose D, and the only Nash equilibrium in the game is given by the pair of strategies (D, D)

Therefore, the dilemma is that if both players select a cooperative (C) move they are jointly better off (getting R each) than if they both select D, but selfish players will select mutual defection, getting only P each, because of the individual incentive to select defection. We select this game as a minimal test that captures a range of possible application tasks in which nodes need to establish cooperation and trust with their neighbours but without central authority or external mechanisms that enforce it.

In order to apply the game to test the SLAC and SLACER P2P protocols we set application level behaviour to each node playing the PD with randomly selected network neighbours. A node can only choose one out of the two pure strategies: cooperate or defect. The utility value required by each node in the protocols is set to the average payoff the node received from recent game interactions. The SLACER algorithm then adapts the links and strategy of the nodes as discussed previously. The specific payoff values chosen for T, R, P

⁴ Motifs in Cooperative Networks



(c) Coop. component breaks apart

(d) Cooperative groups are formed

Fig. 2. Evolution of a SLAC network with nodes playing the Prisoner's Dilemma. From an initially random topology composed of all nodes playing the defect strategy (dark shaded nodes), components quickly evolve, still containing all defect nodes (a). Then a large cooperative component emerges in which all nodes cooperate (b). Subsequently the large component begins to break apart as defect nodes invade the large cooperative component and make it less desirable for cooperative nodes (c). Finally an ecology of cooperative components dynamically persists as new components form and old components die (d). Note: the cooperative status of a node is indicated by a light shade.

and S do not significantly effect performance as long as they meet the PD constraints discussed above. Figure 2(d) shows the typical evolution of a SLAC network over time.

4 Motifs and Subgraph Ratio Profiles

Networks are often characterised using average global measures, such as average path length and clustering coefficient. Although valuable such measures rarely give a picture of the detailed structure of the networks. This means that networks with different topologies on a local scale, can have identical global average measurements. Hence, in order to further understand and classify natural and artificial networks new methods have been proposed.

Recently, researchers working with complex networks (both natural and artificial) have begun to analyse and characterise them using more sophisticated topological techniques and one of these approaches is so called "motif analysis" [11].

By breaking a network down into all possible n-node subgraph patterns and counting them it is possible to compare those counts against randomly generated networks with the same characteristics (number of nodes and in / out degree links). Then, where certain n-node subgraph patterns are significally more prevalent than in the random case, these are considered mofits of the network. Additionally, although less discussed in the literature, n-node subgraph patterns that are under-represented in the network have been termed anti-motifs [10] and are of equal value in characterising network structure.

Obviously, for large subgraph sizes the number of possible motifs becomes large but for smaller sizes (3 and 4 nodes) it is computationally feasible to search, even large, networks for all occurrences. Figure 3 shows all possible three node subgraphs for directed graphs. Note, that for non-directed threenode subgraphs there would be only two possible subgraphs (shown as id78and id238 in figure 3).



Fig. 3. All thirteen possible three-node directed subgraphs (taken from [12]). The id is obtained by representing the subgraph as an adjacency matrix structured as a binary integer extracted by concatenation of the rows of the matrix. In this way any size of subgraph can be given a unique id which specifies the structure completely.

The P2P networks produced by SLAC and SLACER are undirected in the sense that all links are bidirectional. So for the purposes of analysis we search for all undirected four-node subgraphs (tetrads). Figure 4 shows the six possible undirected tetrads. We analysed the networks using the subgraph ratio profile (SRP) method [10]. This approach is particularly useful for undirected networks since traditional motif analysis methods using z-scores are not network size invariant for tetrads and this makes comparison with networks of different sizes difficult. For a given network N the SRP is a normalised vector of Δ_i values:

$$SRP_i = \frac{\Delta_i}{\sqrt{\sum\limits_i \Delta_i^2}} \tag{1}$$

The vector elements, one for each of the six tetrads, are calculated based on the abundance of each tetrad *i* relative to randomly generated networks, To avoid large values as an artefact of very small occurrences of tetrads in both the real and random networks the value $\varepsilon = 4$ is added to the the denominator.:

$$\Delta_i = \frac{Nreal_i - \langle Nrand_i \rangle}{Nreal_i + \langle Nrand_i \rangle + \varepsilon}$$
(2)



Fig. 4. All six possible four-node undirected subgraphs (tetrads). Nodes are not shown but should be assumed at the end of each straight line. The adjacency matrix derived id's are not shown but the tetrads are ordered by ascending value of them.

A given SRP can be graphed producing a curve which characterises the tetrad motifs and anti-motifs visually.

5 Results of Analysis

Figures 5 and 6 show SRP's for both SLAC and SLACER P2P networks at different stages in their evolution ². In each case three time ordered network snapshots are shown. The application task that generated the node level utility required for the protocols was to periodically play the Prisoner's Dilemma (PD) game with a randomly chosen neighbour and accumulate average payoff. The period of game playing was one order of magnitude higher than the period used by the SLACER protocol as shown in figure 1. This means that on average ten games of PD would be played between SLACER invocations.

As previously discussed, the difference between SLAC and SLACER is that the link drop probability (W) is set to a value lower than 1 (in this case we used W = 0.9 for SLACER). All nodes in the network stored a pure PD strategy (either to cooperate or defect) - representing the application behaviour. The mutation probabilities were set to M = 0.001 and MR = 0.01. We fixed the maximum in / out degree of each node to 20 links. All links are undirected and hence symmetric. The protocol perseveres the symmetry of links between nodes at all times.

The networks were initialised with random topologies and all nodes as non-cooperative (defect strategy). The first snapshot is taken immediately before any cooperation has been attained, the second during the rapid outbreak of cooperation and the final snapshot is taken after stable cooperation is attained. We do not show the SRP for the initial network topology since this would be a flat line along the x-axis (being a random network). However, we have found that from *any* initial topology the network evolves to the same topology immediately before cooperation breaks-out [2] so we would expect the shown results to look the same no matter what the starting topology.

Notice that the curves in figures 5 and 6 follow a similar time evolution for both SLAC and SLACER: immediately before cooperation (snapshot 1) the curve already has a very similar shape to the final curve, during the outbreak of cooperation (snapshot 2) the curve tends to move upward slightly (less anti-motifs 1 and 2, but more motifs 3 to 6), then, after stable cooperation is attained (snapshot 3) the curve tends to flatten (with all points moving towards the x-axis). Motifs 1 and 2 are under represented (anti-motifs) and motifs 3 to 6 are over represented (motifs) but with a large dip for motif 4 almost close to zero (identical to the random occurrence) when cooperation has stabilised (snapshot 3).

² We used the freely available "mfinder" [13] software tool for identifying the subgraphs in the P2P networks, the P2P networks themselves were implemented in the open source Peersim environment. The code for the SLACER protocol, in addition to Peersim itself and tutorial materials are available from the Peersim website [14]. The random sampling service required by SLACER is provided by the Newscast protocol [7]. Peersim was initially developed within the BISON project [16]

9



Fig. 5. The subgraph ratio profiles (SRP) of SLAC P2P networks at different stages in their evolution to cooperation - network size N = 500, edges $E \approx 3500$. In each case three time ordered network snapshots are shown. The networks were initialised with random topologies and all nodes as non-cooperative. The first snapshot is taken immediately before any cooperation has been attained, the second during the rapid outbreak of cooperation and the final snapshot is taken after stable cooperation is attained. The results show a similarity to the protein-structure superfamily (PSA). PSA shows the average SRP of the protein structure results given in [10]. These results are from networks of secondary-structure elements adjacency for several large proteins [structure based on the PDB database (www. rcsb.org/pdb/); the proteins (and their PDB ID) were 1A4J, an immunoglobulin (PROTEINSTRUCTURE-1 N = 95, E = 213); 1EAW, a serine protease inhibitor (PROTEIN-STRUCTURE-2 N = 53, E = 123); and 1AOR, an oxidoreductase (PROTEINSTRUCTURE-3 N = 99, E = 212)]



Fig. 6. The subgraph ratio profiles (SRP) of the SLACER P2P networks at different stages in their evolution to cooperation compared to the protein structure averages (PSA) - results for SLACER were obtained in the same was as described for figure 5

10 Motifs in Cooperative Networks

The SRP curves shown in figures 5 and 6 were directly compared to those given in[10]. There a number of curves were given for different networks. Similar curves were clustered into similar groups (or superfamilies). We have copied the average of the protein-structure curves onto our figures (labeled as PSA). The PSA line shows a characteristic anti-motif representation in 1 and 2 and dip at 4. These characteristics are reproduced in the P2P networks. However, notice that motif 6 dips slightly in the protein-structure networks (PSA) but this is not reflected in the P2P networks. Here the P2P networks look more like the BA models [1] or the power-grid networks shown given in [10]. Notice also that the early snapshots of SLAC look most like the protein-structure networks - interestingly this is before cooperation has stabilised. The snapshot 1 networks are taken before *any* cooperation has formed in the networks and can therefore only be the result of random copy and rewire since all utilities are equal, at this stage, because all nodes use the defect strategy in the PD.

This suggests some rather intriguing new possibilities *and* reinforces existing findings. Firstly, that the motifs and anti-motifs evident in the P2P SRP's initially result from a randomised process rather than the specific function of cooperation *but* that this randomised process is an artefact of an algorithm designed *for* cooperation.

SLACER appears to work *because* a randomised re-wiring process draws the network towards a topology that eventually supports cooperation. We have discussed in detail this multi-stage process elsewhere [2]. What we see is a complex interplay of function and topology formation leading to a phase transition in cooperation with negative scaling properties (i.e. the bigger the network the quicker it happens).

Another interesting result is that the different snapshots show significant differences in the SRP, such that for a given (P2P network) SRP it would be be possible to predict if it was in a stable cooperative mode or not - without considering the strategies stored in the nodes. This could possibly be very useful, since it indicates a way to detect if the network is functioning cooperatively or which stage in the evolution to cooperation has been reached based purely on structural characteristics. Since we have observed SLACER to support other kinds of cooperative tasks (not just playing the PD) and produce networks with similar topologies we speculate that potentially the final snapshot 3 SRP curve could characterise many possible application domains that were functioning under SLACER cooperatively (correctly), even if the specifics of the applications themselves were complex or unknown or unknowable from a snapshot. If this were the case, then a SRP curve could provide a very powerful fingerprint of the global collective functioning of a SLACER supported application. A deviation from the cooperative fingerprint could indicate malfunctioning or network attacks. This could be a topic for future work along with the design of a mechanism to perform motif analysis in a distributed fashion requiring all nodes to interact locally and aggregate their knowledges on local structures.

Additionally, as stated previously, the only difference between the SLAC and SLACER protocols is the link drop probability (W). In SLAC W = 1 and in SLACER W = 0.9. The lower value of W produces a larger dip in motif 4 - nearer to the protean-structure networks. It would be interesting to calculate SRP's for different values of W to see how this changes the curve.

5.1 Conclusions

We have briefly summarised some existing P2P protocols that we have designed to promote cooperation - system level utility - when nodes behave selfishly with local information. We performed a motif analysis by calculating and plotting subgraph ratio profiles as curves. We compared this to existing work showing curves for various natural and engineered networks. We did not expect to find the P2P networks to match any of the existing graphs but were intrigued to find that the P2P networks appeared very similar to a family of protein-structure curves. However, should we be surprised? The P2P algorithms were inspired by an evolutionary tagging algorithm [6] that produces robust and desirable life-like properties such as scalability, robustness and self-repair. It therefore may not be so surprising that the resulting topologies come to resemble naturally occurring networks with similar properties.

Rather than design an algorithm for constructing a particular network form [1] we have designed the P2P protocols for a specific function - to suppress the individual selfish behaviour of nodes for the collective good of the network without the need for complex or sophisticated reputation systems, repeated reciprocal strategies or centralised control. Perhaps this function has some kind of universal applicability - if so, could this throw any light on protein networks and their role in cell level computation? Is there really any linkage at all or just superficial similarity? It may be the case that given the low fidelity of this kind of motif analysis (i.e. there are only six possible 4-node undirected subgraphs) almost any kind of "copy and rewire" operation would lead to similar profiles.

One interesting idea that this work suggests is that although networks may appear to be constructed from a random process of copy and re-wire this could be an *artefact of an underlying functional process* - which in certain phases reduces to a random process but nevertheless harnesses the functionality of the properties produced by that process. In some sense, this is what evolution is. If we were to make a grandiose conjecture we might say that the kind of SRP profiles we have found could be considered as evidence for (though not proof of) an evolutionary process in a network.

Another potentially very useful line of work (for P2P engineering) would be to explore our hunch that the cooperative fingerprint shown in the SRP will hold over many possible application domains, hence providing a method for monitoring the network for serious malfunctions or malicious attacks that degrade collective performance, generically, without having the know the application specifics. This would be consistent with the thesis of [10] which is that

12 Motifs in Cooperative Networks

perhaps the super-families identify some very general underlying functional characteristics even though we might not know the specifics.

Our simple analysis has raised a number of interesting questions which could be an interesting topic of future work.

Acknowledgements

The P2P protocol simulations were implemented in Peersim [14]. The motif analysis was carried out using the Mfinder software [13] and the graph visualisations in figure 2 were produced using Graphviz [15]. We would like to thank Sergi Valverde and Ricard Solé for pointing us towards motif analysis. Also we would like to thank attendees of a number of workshops, at which we presented this work, for their comments, interest and encouragement.

References

- 1. Barabási, Albert-László and Albert, Reka (1999) Emergence of Scaling in Random Networks, *Science 286, 509*
- Hales, D. and Arteconi, S. (2006) SLACER: A Self-Organising Protocol for Coordination in P2P Networks. *IEEE Intelligent Systems* 21(2):29-35.
- Hales, D. (2006) Emergent Group-Level Selection in a Peer-to-Peer Network. Complexus 2006;3.: 108-118.
- Hales, D. and Edmonds, B. (2005) Applying a socially-inspired technique (tags) to improve cooperation in P2P Networks. *IEEE Transactions in Systems, Man* and Cybernetics - Part A: Systems and Humans, 35(3):385-395.
- Hales, D. (2004) From Selfish Nodes to Cooperative Networks Emergent Linkbased Incentives in Peer-to-Peer Networks. In proceedings of The Fourth IEEE International Conference on Peer-to-Peer Computing (p2p2004), 25-27 August 2004, Zurich, Switzerland. IEEE Computer Society Press.
- Hales, D. (2000) Cooperation without Space or Memory: Tags, Groups and the Prisoner's Dilemma. In Moss, S., Davidsson, P. (Eds.) Multi-Agent-Based Simulation. Lecture Notes in Artificial Intelligence 1979. Berlin: Springer-Verlag. Available
- Jelasity, M., Montresor, A. and Babaoglu, O. (2005) Gossip-based aggregation in large dynamic networks. ACM Trans. Comput. Syst., 23(1):219-252
- Kong, J. S., Boykin, P. O, Rezei, B., Sarshar, N. and Roychowdhury, V. (2005) Let your cyberalter ego share information and manage spam. Available as preprint: http://xxx.lanl.gov/abs/physics/0504026.
- Marti, S., Ganesan, P. and Garcia-Molina, H. (2005) DHT Routing Using Social Links In Peer-to-Peer Systems III: IPTPS 2004, Revised Selected Papers, LNCS 3279 Springer.
- Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M. & Alon, U. (2004) Superfamilies of designed and evolved networks *Science*, 303:1538-42.

- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D. & Alon, U. (2002) Network Motifs: Simple Building Blocks of Complex Networks *Science*, 298:824-827
- 12. Motif Dictionary,
- available at: http://www.weizmann.ac.il/mcb/UriAlon/groupNetworkMotifSW.html 13. mfinder network motif detection tool software,
- available at: http://www.weizmann.ac.il/mcb/UriAlon/groupNetworkMotifSW.html 14. Peersim Peer-to-Peer Simulator, http://peersim.sf.net
- 15. Graphviz graph visualisation software, http://www.graphviz.org/
- 16. The BISON Project, http://www.cs.unibo.it/bison

Index

paragraph, 1

16 Index