Evolving Networks for Social Optima in the "Weakest Link Game" *

Giovanni Rossi, Stefano Arteconi, and David Hales

The University of Bologna, Italy
{giorossi, arteconi}@cs.unibo.it, dave@davidhales.com

Abstract. Previous models have applied evolving networks based on node-level "copy and rewire" rules to simple two player games (e.g. the Prisoner's Dilemma). It was found that such models tended to evolve toward socially optimal behavior. Here we apply a similar technique to a more tricky co-ordination game (the weakest link game) requiring interactions from several players (nodes) that may play several strategies. We define a variant of the game with several equilibria — each offering increasing social benefit. We found that the evolving network functions to select and spread more optimal equilibria while resisting invasion by lower ones. Hence the network acts as a kind of "social ratchet" selecting for increasing social benefit. Such networks have applications in peer-to-peer computing and may have implications for understanding social systems.

1 Introduction

Recent work introduced a simple "copy and rewire" algorithm (SLAC) that structures networks of nodes towards socially optimal behaviors when they interact with neighbors to generate utility [Hales(2005),Hales and Arteconi(2006)]. The algorithm demonstrates properties such as self-organisation, scalability, robustness to node turn-over and free-riding behavior without the need for central control.

These properties are highly desirable for use in self-organising applications such as peer-to-peer (P2P) overlay networks where central control is problematical and free-riding and node failure are prevalent. Hence

* This work partially supported by the EU within the 6th Framework Program under contract 001907 (DELIS).

SLAC has been proposed as a P2P protocol for a number of scenarios [Hales(2006),Hales and Edmonds(2005)].

In these previous works, the social optimum, by which we mean the maximum total sum of utility that can be generated over all nodes, required a non-equilibirum strategy. We found that SLAC produced networks close to the social optimum even though nodes behaved in a boundedly rational and myopic selfish way. The interactions between nodes were modeled as pairwise interactions between network neighbors with a focus on the elimination of free-riding and hence the one-shot, two-player Prisoner's Dilemma (PD) game was used.

In this paper we apply the same algorithm to another game called the Weakest Link (WL) which, although having a well defined social optimum strategy, has several different properties from the PD. Firstly, the WL requires simultaneous interactions between several node neighbors to produce utility, secondly, the game has several possible equilibria each offering increasing social optimality and thirdly, there is a larger space of pure strategies. This means that a higher degree of coordination is required between nodes to reach a social optimum but when the optimum is achieved it is an equilibrium (i.e. there is no incentive for an individual to free-ride). We describe the WL game in detail in section 3.

We found that SLAC produced networks that evolved in stages toward the social optimum equilibrium passing through each of the less optimal equilibria. The network stayed in each equilibria for some time before jumping to the next. This process was monotonic with the network always jumping to a higher equilibrium but never to a lower one. This process was robust even when large numbers of nodes were introduced with strategies that exploit these higher equilibria. Hence we describe SLAC as functioning rather like a "social ratchet" always selecting strategies that move to higher social optimum.

The SLAC algorithm was adapted from a model developed within computational sociology based on the "tag" concept introduced by Holland [Holland(1993)] and developed by Riolo [Riolo et al.(2001)Riolo, Cohen, and Axelrod]. The interpretation here is of a cultural evolutionary process within a society of agents forming groups based on observable markings or social cues. From a sociological perspective SLAC can be seen as an adaptation of the algorithm in which agents interact over, and construct, dynamic social networks based on social cues. We briefly discuss in the conclusion the kinds of sociological insights we might draw.

In section 2 we describe the SLAC algorithm. In section 3 we describe the Weakest Link (WL) game. In section 4 we describe how we integrated the two within a computer simulation, then in section 5 we present and discuss the results of our simulations. Finally we conclude with a discussion of what we have learned with implications for P2P protocol design and some speculative sociological interpretations.

2 Evolving Networks with SLAC

SLAC (Selfish Link-based Adaptation for Cooperation) is an evolutionary algorithm based on previous tag models [Riolo et al.(2001)Riolo, Cohen, and Axelrod,Hales(2001)]. Here the concept of the tag (a method of locating suitable partners for social interaction) is translated into nodes' neighborhoods - nodes therefore interact (in this paper, play a game) with other nodes they are directly connected to in the network.

The basic SLAC algorithm specifies how nodes should update their strategies and network neighborhood under the assumption that they are involved in some on-going game interactions with neighbors. Each node generates a *utility* measure (U) according to some interaction with its neighbors. The higher the value of U the better the node is performing.

The algorithm is executed by each node and consists in it periodically comparing its own utility (say U_i) with another node (say U_j) randomly chosen from the network. If $U_i \leq U_j$ then node i drops all of his current links and copies all j's links (adding a link to j itself) and j's strategy - see figure 1.

```
i ← this node
do periodically:
   j ← GetRandomNode()
   if Ui ≤ Uj
     i.links ← j.links ∪ j
     with low probability mutate(i)
```

Fig. 1. SLAC algorithm's pseudocode.

In SLAC all the rewiring operations are symmetric — if node i makes a link to j then node j makes a link to i and on the other hand if i drops a link to j the

link from j to i has to be dropped as well. Each node can maintain a maximum amount of links (called *viewsize*), if a new node has to be added in an already full view, a randomly selected neighbor is dropped to make space for the new link. The SLAC algorithm provides a rewiring mechanism analogous to tournament selection within evolutionary computing. Nodes act in a highly boundedly rational way hoping to selfishly improve their utility by copying better performing ones. Occasionally with low probability a node applies a "mutation" function after copying another node. This involves changing the strategy randomly and changing the links randomly.

The utility measure is provided by the game that is being played between node neighbors. In previous work, as we have discussed, we have applied SLAC to the Prisoner's Dilemma game, here we implement nodes playing the weakest link game described in section 3.

3 The Weakest Link Game

In a strategic (or non-cooperative) game there are $n \geq 2$ players each of which takes some action, and everyone's utility depends on the n-tuple of taken actions (see [Colell et al.(1995)Colell, Whinston, and Green] and [Myerson(1997)]). Formally, a game consists of a triple $\Gamma = (N, \mathcal{S}, u)$ where $N = \{1, \ldots, n\}$ is a finite player set, $\mathcal{S}$ is the n-fold product of action or strategy sets (one for each player), and $u : \mathcal{S} \to \mathcal{R}^n$ represents players' preferences: $u_i(s)$ is the utility attained by $i \in N$ when the strategy profile is $s \in \mathcal{S}$. A (pure-strategy) Nash equilibrium NE is any strategy profile $s \in \mathcal{S}$ with respect to which no player has an incentive to (unilaterally) deviate. That is, a situation where each player is playing a best response to the $n-1$-tuple of others' actions.

A main distinction is between common interest games on the one side, and conflict games on the other. In the former case, players' preferences are aligned: there is at least one outcome $s^* \in \mathcal{S}$ where each and every player's preferences are maximized, i.e., $u_i(s^*) \geq u_i(s)$ for every $s \in \mathcal{S}, i \in N$ (see [Aumann and Sorin(1989)]). Clearly, if such a profile s^* exists, then surely s^* is a NE. Additionally, in a *pure* common interest game players' preferences all induce a common ordering of strategy profiles. That is to say, for any pair $s, s' \in \mathcal{S}$ and every $i \in N$, if $u_i(s) \geq u_i(s')$, then $u_j(s) \geq u_j(s')$ for all $j \in N$ (see [Bowles(2003)]). On the other hand, a game displays some conflict if for any

strategy profile there is at least one player who strictly prefers another one (i.e., for any $s \in \mathcal{S}$, there is at least one $i \in N$ such that $u_i(s') > u_i(s)$ for some $s' \in \mathcal{S}, s' \neq s$).

Common interest games admitting several equilibria are often called coordination games. They are useful for understanding and formalizing distinct aspects of interaction within organizations (see [Camerer and Marc(1997)], [Van Huyck et al.(1992)Van Huyck, Gillette, and Battalio],
[Van Huyck et al.(1991)Van Huyck, Battalio, and Beil],
[Van Huyck et al.(1990)Van Huyck, Battalio, and Beil]). In particular, simulations based on coordination games are useful to organizational researchers because, roughly speaking, fostering coordination seems a main goal of several organizations. In fact, in coordination games the isssue is how to make players choose a n-tuple of actions resulting in a social optimum (or *Pareto-efficient* outcome).

Consider the following simple example. There are two players, 1 and 2, who both choose between 1 (or *high* or *cooperate*) and 0 (or *low* or *defeat*). Let $s_i \in \{0, 1\}$ denote the choice of player $i = 1, 2$. Payoffs for each pair $s_1, s_2 \in \{0, 1\}$, are shown in table 1.

	$s_2 = 1$	$s_2 = 0$
$s_1 = 1$	$(2, 2)$	(β, α)
$s_1 = 0$	(α, β)	$(1, 1)$

Table 1. General 2-players coordination game payoff table. The first and second number in parentheses corresponds to player 1 and 2, respectively, and $0 \leq \beta < 1 \leq \alpha$.

For $\alpha \geq 2$ this is a *prisoner dilemma* PD game. For $1 \leq \alpha < 2$ this is a cordination game. In particular, if $\beta = 0, \alpha = 1$, then this is known as a *stag hunt* or *assurance* game. More generally, if $0 \leq \beta < 1 = \alpha$, then this is a *weakest link* WL game. In fact, if $\alpha \geq 2$, then 0 is a dominant strategy (i.e., it yields a higher payoff for any choice of the other), and thus the only equilibrium is $s_1 = s_2 = 0$. On the other hand, for $\alpha < 2$ there are two equlibria, i.e., the low one $s_1 = s_2 = 0$, and the high one, i.e., $s_1 = s_2 = 1$, where this latter is the only social optimum. The issue of coordination concerns precisely how to make players select high rather than low.

Games such as the PD, where each player has a dominant strategy, are unaffected, in some respect, by strategic uncertainty. In fact, a dominant strategy yields, for any choice of the opponents, a maximum payoff (or utility). Therefore, players could not even form their own expectations about the others' choices, as a dominant strategy is a best response to any of such choices. Conversely, in coordination games the main reason why coordination may fail (thus yielding sub-optimal outcomes), is because players have strategic uncertainty, i.e., they are not sure about how others will behave. In such cases, organizations may perform precisely the task of creating *organizational (i.e., common) expectations* (see [Camerer and Marc(1997)] p. 172).

In a weakest link game each player $i \in N$ chooses an effort level s_i from some finite set $\{0, 1, \ldots, K\}$ ($K = 1$ above), and i's payoff $u_i(s) = u_i(s_1, \ldots s_n)$ depends only on the chosen effort level s_i and the minimum $\min_{j \in N} s_j$ across all players. In particular, it is increasing in s_i if $s_i = \min_{j \in N} s_j$, and decreasing in s_i if $s_i > \min_{j \in N} s_j$. In words, *each player wants to select exactly the minimum of the other players, and everyone wants the minimum to be as high as possible. But selecting high actions is risky because other players may select low actions* (see [Camerer and Marc(1997)] p. 175).

In our setting $K = 19$ and the payoff is

$$u_i(s) = \frac{1 + \min_{j \in N} s_j}{1 + s_i - \min_{j \in N} s_j} \tag{1}$$

where $s_i - \min_{j \in N} s_j \geq 0$ (of course), and thus the payoff matrix is the one given in table 2.

	$\min_{j \in N} s_j = 0$	$\min_{j \in N} s_j = 1$	$\min_{j \in N} s_j = 2$	$\cdots$	$\min_{j \in N} s_j = 19$
$s_i = 0$	$u_i(s) = 1$	$\backslash$	$\backslash$	$\cdots$	$\backslash$
$s_i = 1$	$u_i(s) = \frac{1}{2}$	$u_i(s) = 2$	$\backslash$	$\cdots$	$\backslash$
$s_i = 2$	$u_i(s) = \frac{1}{3}$	$u_i(s) = 1$	$u_i(s) = 3$	$\cdots$	$\backslash$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\backslash$
$s_i = 19$	$u_i(s) = \frac{1}{20}$	$u_i(s) = \frac{2}{19}$	$u_i(s) = \frac{3}{18}$	$\ldots$	$u_i(s) = 20$

Table 2. Weakest link payoff table with 20 possible strategy and payoff function in equation 1

It is easily seen that this game has precisely 20 equilibria, each of which attains when $s_i = k$ for every $i \in N$ as $k = 0, 1, \ldots, 19$. Furthermore, these equilibria are *Pareto-rankable*; that is to say, $s_i = 19$ for every $i \in N$ is the unique social optimum, and for $0 \leq h < k \leq 19$, each player prefers equilibrium $s_i = k$ for every $i \in N$ rather than $s_i = h$ for every $i \in N$.

A strategic game is a potential game if it admits a (ordinal) potential, i.e., a function $P : \mathcal{S} \to \mathcal{R}$ such that for every player $i \in N$, for every pair of strategies s_i, s_i' for this player, and for every $n-1$-tuple $s_{-i} = \{s_j : j \in N \backslash i\}$ of strategies for other players, the following holds:

$$P(s_i, s_{-i}) - P(s_i', s_{-i}) > 0 \Leftrightarrow u_i(s_i, s_{-i}) - u_i(s_i', s_{-i}) > 0, \text{ i.e.,} \tag{2}$$

for any unilateral deviation from a strategy profile, an improvement for the deviator must occur iff an improvement in the potential occurs (see [Monderer and Shapley(1996)]). Potential games admit at least one equilibrium, i.e., a potential-maximizer strategy profile. Put it differently, if a potential function exists, then it surely attains a maximum (of course). In fact, if $\Gamma = (N, \mathcal{S}, u)$ admits a potential P, then any equilibrium of $\Gamma_P = (N, \mathcal{S}, u^P)$, where $u_i^P = P$ for every $i \in N$, is also an equlibrium of potential game Γ. That is, the original game and the game where each player's utility is replaced with the potential itself have the same equilibria. Potential mazimizer strategies correspond to social optima (see [Monderer and Shapley(1996)] section 5).

Given the non-linearity in our specification (1) above of the WL game, this latter might fail to admit a potential (see [Monderer and Shapley(1996)] section 5). Nevertheless, a very simple function, assigning to each strategy profile the sum over all players of their utility (which also measures social welfare, in fact), displays an important feature that seems related to the "social ratchet" effect (and to condition (2) above). In practice, $P(s) = \sum_{i \in N} u_i(s)$ is monotonically increasing in unilateral deviations from lower to higher strategies that lead to an improvement for the deviator. A proof of this is briefly provided in appendix A.

4 Simulation Set-Up

To simulate nodes playing the WL game using SLAC, a peer-to-peer network simulator software package called Peersim[1] has been used. In our simulations

[1] http://peersim.sf.net

time is divided into cycles, and in each cycle each node performs specific actions described by SLAC algorithm and WL game. In a single simulation cycle every node in the network is selected once in a randomized order. When selected the node executes the SLAC algorithm with probability P_{rep} (making a utility comparison and possibly copying another node) or, if SLAC was not executed, initiates a game round[2].

The utility comparison step is performed by SLAC as described in section 2. To play the WL game a node executes the function shown in figure 2. In our implementation WL rounds are initiated locally by single nodes and only the payoff of the node which initiated the round is affected by each single game, according to equation 1.

SLAC requires a utility value to be defined for each node. This is the average utility obtained from the game played since last reproduction. Mutating the strategy consists in choosing with uniform probability a random strategy between the possible ones. Mutating the links involves removing all existing links and replacing with a single link to a randomly chosen node from the population. We mutated the strategy with probability m_s and the links with probability m_l.

```
i ← this node
for each j in i.Links
  strats ← strats ∪ j.Strategy
i.Utility ← payoff for i.Strategy given neighbor node strategies (strats)
```

Fig. 2. Weakest Link single game round pseudocode.

Our aim in these experiments is to discover how well SLAC is able to lead the network to a high social optimum. Hence we are interested in the equilibrium reached by the system, the time needed to reach it and the robustness of high social optimum equilibrium states.

[2] This captures the notion that both SLAC and the WL game rounds are occurring asynchronously and concurrently.

5 Simulation Results

Here we present simulation results and analysis. In all the following experiments the network is initialized to a random topology [3] with each node neighborhood (view size) limited to at most 20 nodes. We set reproduction probability $P_{rep} = 0.2$ and the mutation rates $m_s = 0.005$ and $m_l = 0.01$. These values were imported from previous work, which indicated the need for a higher mutation rate applied to links than to strategies [Hales and Edmonds(2005)]. We found that the same insight held in this new scenario. However, in further experiments not shown here we observed that similar results were obtain when we varied the mutation rates within an order of magnitude both higher and lower. In addition we found that varying the replication rate did not significantly effect results (although changed time required to reach higher equilibrium) up to around 0.5, at which point optimal outcomes were not obtained. The possible strategies for each player are in the set $s = \{0, 1, ..., 19\}$ and at the beginning of the simulation are uniformly distributed in the range $s_{init} = \{0, 1, ..., 9\}$ (the lower half of possible strategies) among network nodes.

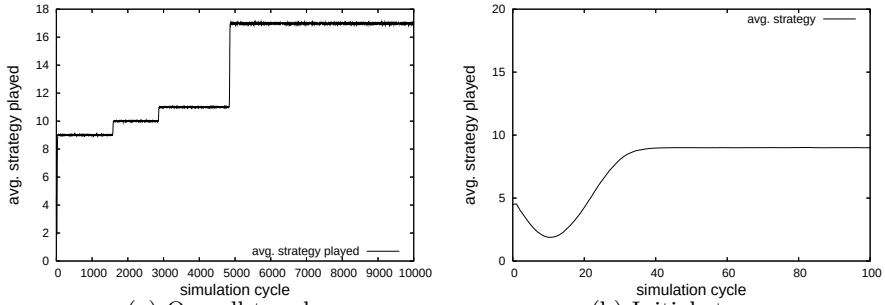
To test SLAC scalability with the WL game different network sizes ranging from 1000 to 16000 nodes have been evaluated. Given the stochastic nature of the process (initialization, random comparison) for each parameter setting 10 runs with different random seeds have been done to have high confidence on the obtained result.

5.1 A Typical Run

Figure 3 shows a single run performed on a 4000 node network showing the average strategy value over all nodes at each cycle (in general the higher the strategy the higher the social utility). The figure shows three different phases of behaviour over time:

- Initial transient where strategies have a low dip (figure 3(b))).
- Stability phase with the whole network adopting the same strategy.
- Sudden increases in average played strategy (staircase pattern in figure 3(a)).

[3] We found that starting the network with *any* initial topology did not effect our results significantly



(a) Overall trend (b) Initial stage

Fig. 3. A typical single run showing how the average strategy value changes in the network by cycle. The network has a fixed size of 4000 nodes. (a) shows the entire run to 10000 cycles, (b) shows the same run but over the first few cycles showing detail not visible in (a).

The initial transient shown in figure 3(b) is due to the random initialization of both topology and nodes' strategy. In this phase nodes form clusters of homogenous strategies through rewiring and strategy copying defined by SLAC. Once the clusters are formed the network will eventually stabilize (system-wide) on a given strategy value which is determined by the random initialization (different runs stabilize on different strategies initially).

The network does not attain complete stability (small amounts of noise are just visible in figure 3) due to node mutation and rewiring, yet we never observe the network falling to a lower strategy (system-wide) after it has stabilized on a higher one.

In neighborhoods where strategies have stabilized, i.e., where a NE attains, if a node mutates to a lower strategy it will bring its neighborhood to a lower social utility level, lowering both its own payoff and that of its neighbors. Because of the underlying SLAC algorithm such nodes will eventually compare utility with random nodes from groups where everybody plays the same (high) strategy and copy them. Hence mutation to lower strategy does not bring the whole network to a lower strategy.

On the other hand a node mutating to a high strategy value will lower its own utility without affecting its neighbors' ones. Similarly to the previous case a high strategy node will then eventually compare with a node playing the actual system-wide strategy and copy it.

More interesting things happen when an entire neighborhood mutates, rather than a single node. When in a mutated neighborhood at least one node mutates to a low strategy, the entire neighborhood will have a low utility and rewire to other groups in the network in an analogous way to the single nodes case. On the other hand if all the nodes mutate to strategies higher than the system-wide one, at least one of them (the one with lowest strategy between the mutated nodes) will have higher utility than the rest of the network and will be eventually copied by others, bringing the network to a higher strategy value.

To sum up, through strategy mutation the system is able to eventually reach a higher strategy than a lower system-wide one, but the system will never switch back to a lower strategy even if occasionally nodes (or entire neighborhood) mutate to low strategies. The system-wide strategy level is therefore monotonically increasing over time.

These "ratchet" effects apply even further to those neighborhoods where no NE attains, in which case if a node gains by mutating to a higher strategy, then there is at least another node in that neighborhood who also gains (see section 3 and the appendix). This applies, of course, to all neighborhoods affected by some noise (see above).

However, since for an increasing strategy to spread requires a whole neighborhood to mutate to a higher strategy, the time needed to pass from one strategy to a higher one is dependent on coincidental random mutations — they can assume very different values in different runs with different random generator seeds.

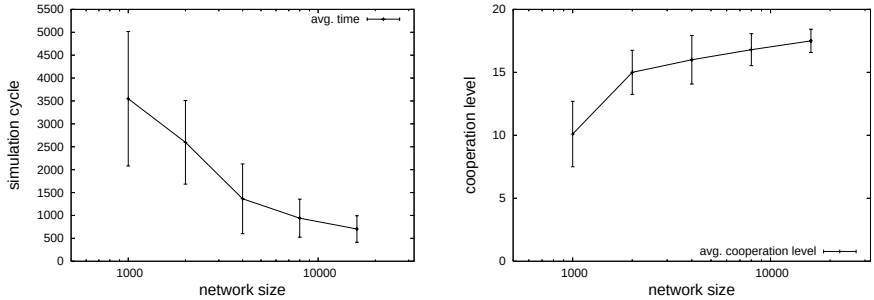
In order to get a general picture we therefore ran a number of experiments and considered their average values and variances. In addition we ran experiments to test the scalability of the process by using networks of different sizes.

5.2 Different Network Sizes

To determine scalability of the SLAC approach in the WL game we performed 10 simulation experiments each for networks of size 1000, 2000, 4000, 8000, 16000. Each experiment was started with a unique random generator seed and executed for 5000 simulation cycles.

What we are interested in is the average time needed to pass from a stable state to a higher stable state (a more socially optimal equilibrium) and the time needed to get a high (possibly the highest) social utility.

Results are shown in figure 4. The average time between phase transitions is shown in figure 4(a), while figure 4(b) shows the average strategy level reached at the end of simulation (5000th cycle).



(a) Time for cooperation increase (b) Cooperation level at 5000 cycles

Fig. 4. Average time needed to increase cooperation level in different network sizes and cooperation level reached at 5000 cycles. 90% confidence intervals are shown. As can be seen larger networks reach higher cooperation levels more quickly. Ten runs were performed for each network size. The x-axis is on a log scale.

It is interesting to notice that from the cooperation level point of view the system reverse-scales, in fact it seems to have better performance in bigger networks. In larger networks it takes less time between phase transitions. Moreover the runs for large networks produce lower variance in results since the confidence interval gets smaller and smaller with increasing network size.

The reverse scaling property appears to be a direct consequence of the stochastic nature by which transitions to higher equilibria occur. As discussed above, it takes the coincidental mutation of neighboring nodes to a higher strategy to produce a system-wide transition. A larger network increases the chance of such a fortuitous coincidence. [4]

We hypothesize that the network acts as a kind of "social ratchet" always selecting the socially optimal strategies to spread yet resisting the spread of less optimal strategies. We tested our hypothesis by performing further experi-

[4] In addition this is related to the mutation rate used. We deductively derived an approximate expression which captures this relationship for a similar "tag" model in [Hales(2001)]

ments in which we turned off mutation in SLAC. We present those results in the following section.

We also tested that the ratchet effect was due to the differential strategy and link copying based on utility, rather than just random changes, by running simulations in which nodes only mutated their strategy and links when they found another node with higher utility. In this case the results showed no dynamic behavior but rather average strategy value oscillated around the mean (of 10) with low variance.

5.3 Seed Injection

In order to test our "social ratchet" hypothesis we tuned off mutation completely in SLAC[5] and artificially created a pair of neighbor nodes with a higher strategy and "injected it" into a running network. We found that this was sufficient to cause a rapid transition of the entire network to this higher strategy. Hence for any network, injecting two nodes linked together and playing the highest possible strategy will suddenly bring the network to the highest strategy. Conversely when nodes playing low strategies are injected into the population they are not copied, rather they quickly adopt the existing higher strategy from other nodes.

This is consistent without our "social ratchet" hypothesis, as discussed above. The network functions to select only those more socially optimal examples and resists less socially optimal examples. This process operates when new strategy variants are produced via random mutation or directly introduced.

It is interesting to notice how the network reacts when a very large number of nodes playing low strategies are introduced. We found that even if 50% of nodes are suddenly set to play a low strategy then the network does not fall to the system-wide adoption of a lower equilibrium but rather the network quickly recovered. This is illustrated in figure 5 where starting from a network with lowest possible strategy different strategies are periodically injected.

[5] In addition we changed the utility comparison stage of SLAC such that nodes only copied others if utility was stricktly greater than their own. This increases stability of the network.

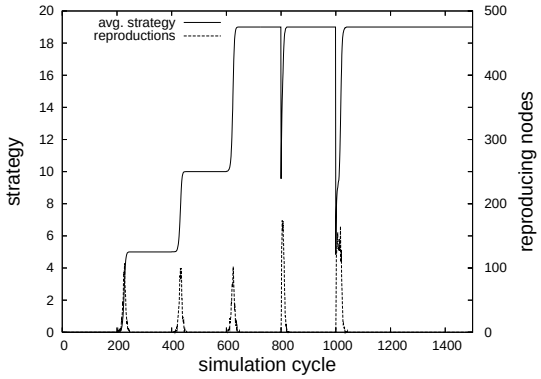


Fig. 5. Experiment with no mutation and node injection (4000 node network). Initially all nodes are set to strategy 0 (minimum). Two nodes playing strategy 5 are injected at cycle 200. Two nodes playing strategy 10 are injected at cycle 400. Two nodes playing strategy 19 (maximum) are injected at cycle 600. To test robustness, 1000 nodes playing strategy 0 (minimum) at injected at cycle 800. 2000 nodes playing strategy 0 are injected at 3000. The lower dashed line shows the number of node movements (copying between nodes) made in each cycle.

6 Discussion and Conclusion

We applied the previously proposed SLAC network evolution algorithm to the Weakest Link (WL) game. The WL game requires nodes to interact with several neighbors simultaneously to obtain utility and has several equilibria which produce increasingly levels of social optimality. The SLAC algorithm specifies that each node myopically and greedily attempts to maximize its own utility by copying other random nodes (strategy and neighbor links) if they have higher utility. Nodes have no knowledge of the underlying game or of the other nodes behavior.

We found that SLAC acts as a "social ratchet" always selecting a higher equilibrium, when it appears in the population, while resisting invasion by nodes following less socially desirable strategies. When strategy change is endogenous, with nodes mutating strategies randomly, the network needs to wait for a chance occurrence of a set of nodes following strategies with higher social benefit in order to increase social benefit. When nodes do not apply mutation, and change is exogenous, then the "injection" of a pair of nodes following strategies producing higher social benefit is sufficient to flip the entire network to a more socially

optimal strategy. However, it requires over 50% of nodes to be set to a less socially optimal strategy to push the network down to a lower equilibria.

The "social ratchet" effect is emergent rather than programmed into the nodes. The nodes myopically act for own benefit. Yet the evolution of the network selects behaviors that appear highly social. It is almost as if some "hidden hand" turns the "social ratchet" towards ever more socially optimal behavior.

Previous applications of SLAC were to two player games in which only pairs of nodes needed to coordinate to achieve the social optimum and the social optimum was a non-equilbirium outcome (e.g. the singe round, two player Prisoner's Dilemma). The results present here from the WL game generalize further the possible application of SLAC as a mechanism for robustly selecting socially optimal behaviors within evolving networks and have also given insights in to the way SLAC works. We do not have a proof capturing the conditions under which SLAC will select the social optimum - currently we only have simulation results for specific games and payoffs. Our ideal future aim would be a proof such that for a given game we could know a priori if SLAC would select the social optimum or not. This would allow for a principled selection of SLAC to solve a given specific application scenario (say in a peer-to-peer system).

Our primary motivation for this work is to developed self-organising and robust yet light-weight P2P protocols which could be deployed in a variety of specific scenarios. We are therefore interested in the general aggregate properties of essentially trivial behavioral algorithms (at the node level). What is surprising is that such simple node behavior can produce high levels of social cooperation and coordination at the network level. We have recently proposed a variant of SLAC for automatic programming "by example" in always-on P2P [Hales and Babaoglu(2006)]. We have also applied SLAC to a P2P file-sharing scenario [Hales and Edmonds(2005)]. These results give us some optimism that more trickily coordination problems could be tackled by SLAC[6].

From the perspective of social science we might speculate that human social systems could embody similar processes particularly where large populations exist in a state of constant "social flux" and where individuals observe and copy those achieving higher social benefits. In these situations, behavior that appears highly beneficent and socially enlightened may in fact be no more than the result

[6] Though as we have discussed elsewhere SLAC is still open to certain kinds of malicious node behavior [Arteconi and Hales(2005)].

of a long history of greedy myopic individual behavior. If this sounds depressing, take heart. An alternative interpretation is that although individuals may behave selfishly, have little foresight and care not for others, the society as a whole can guide them toward socially optimal behavior without the need for centralized control, draconian punishments or recourse to the gods.

Acknowledgements

Thanks go to Mark Jelasity, Ozalp Babaoglu, Edoardo Mollona and Andrea Marcozzi, all from the Department of Computer Science at the University of Bologna, for many discussions relating to aspects that influenced the paper.

References

[Arteconi and Hales(2005)] S. Arteconi and D. Hales. Greedy cheating liars and the fools who believe them. Technical Report UBLCS-2005-21, University of Bologna, Dept. of Computer Science, Bologna, Italy, Dec. 2005. Also available at: http://www.cs.unibo.it/pub/TR/UBLCS/2005/2005-21.pdf.

[Aumann and Sorin(1989)] R. Aumann and S. Sorin. Cooperation and bounded recall, March 1989.

[Bowles(2003)] S. Bowles. *Microeconomic Theory: Behavior, Institutions and Evolution*. Princeton University Press, 2003.

[Camerer and Marc(1997)] C. Camerer and K. Marc. Coordination in organizations: a game-theoretic perspective, 1997.

[Colell et al.(1995)Colell, Whinston, and Green] A. M. Colell, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[Hales(2001)] D. Hales. Cooperation without memory or space: Tags, groups and the prisoner's dilemma. In *MABS '00: Proceedings of the Second International Workshop on Multi-Agent-Based Simulation-Revised and Additional Papers*, pages 157–166, London, UK, 2001. Springer-Verlag. ISBN 3-540-41522-X.

[Hales(2005)] D. Hales. Self-organising, open and cooperative p2p societies – from tags to networks. In *Proceedings of the 2nd Workshop on Engineering Self-Organsing Applications*, volume 3464 of *LNCS*. Springer, 2005.

[Hales(2006)] D. Hales. Choose your tribe! - evolution at the next level in a peer-to-peer network. In *Engineering Self-Organising Systems. Proc. of the 3rd Workshop on Engineering Self-Organising Applications*, volume 3910 of *LNCS*. Springer, 2006.

[Hales and Arteconi(2006)] D. Hales and S. Arteconi. Slacer: A self-organizing protocol for coordination in peer-to-peer networks. *IEEE Intelligent Systems*, 21(2):29–35, Mar/Apr 2006.

[Hales and Babaoglu(2006)] D. Hales and O. Babaoglu. Towards automatic social bootstrapping of peer-to-peer protocols. Technical Report UBLCS-2006-19, University of Bologna, Dept. of Computer Science, 2006.

[Hales and Edmonds(2005)] D. Hales and B. Edmonds. 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, 2005.

[Holland(1993)] J. Holland. The effect of labels (tags) on social interactions, 1993.

[Monderer and Shapley(1996)] D. Monderer and L. Shapley. Potential games, 1996.

[Myerson(1997)] R. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1997.

[Riolo et al.(2001)Riolo, Cohen, and Axelrod] R. Riolo, M. D. Cohen, and R. Axelrod. Cooperation without reciprocity. *Nature*, 414:441–443, 2001.

[Van Huyck et al.(1990)Van Huyck, Battalio, and Beil] J. B. Van Huyck, R. C. Battalio, and R. O. Beil. Tacit coordination games, strategic uncertainty, and coordination failure. *American Economic Review*, 80(1):234–48, March 1990. available at http://ideas.repec.org/a/aea/aecrev/v80y1990i1p234-48.html.

[Van Huyck et al.(1991)Van Huyck, Battalio, and Beil] J. B. Van Huyck, R. C. Battalio, and R. O. Beil. Strategic uncertainty, equilibrium selection, and coordination failure in average opinion games. *The Quarterly Journal of Economics*, 106(3):885–910, August 1991. available at http://ideas.repec.org/a/tpr/qjecon/v106y1991i3p885-910.html.

[Van Huyck et al.(1992)Van Huyck, Gillette, and Battalio] J. B. Van Huyck, A. B. Gillette, and R. C. Battalio. Credible assignments in coordination games. *Games and Economic Behavior*, 4(4):606–626, October 1992. available at http://ideas.repec.org/a/eee/gamebe/v4y1992i4p606-626.html.

A Proof: Potential Function Existence in Weakest Link Game

Fix $i \in N$ and $s_{-i} \in \{0, 1, \ldots, K\}^{n-1}$. Also let $s_i, s_i' \in \{0, 1, \ldots, K\}, s_i > s_i'$. Given our specification (1) above, if $u_i(s) - u_i(s_{-i}, s_i') > 0$, then it must be $s_i' < \min_{j \in N \setminus i} s_j = \min s_{-i}$, in which case $u_j(s) - u_j(s_{-i}, s_i') > 0$ for all $j \in N \setminus i$.

In turn, this implies

$$P(s) - P(s_{-i}, s_i') = u_i(s) - u_i(s_{-i}, s_i') + \sum_{j \in N \setminus i} (u_j(s) - u_j(s_{-i}, s_i')) > 0,$$

as desired. Yet, such P is not a potential: consider that if $u_i(s_{-i}, s_i') - u_i(s) > 0$, then $s_i > \min_{j \in N \setminus i} s_j = \min s_{-i}$, in which case, as long as $s_i' < \min_{j \in N \setminus i} s_j$, it may well be $u_j(s_{-i}, s_i') - u_j(s) < 0$ for several j.

In our simulations, at each cycle each node either *reproduces* (with probability P_{rep}) or *plays* (with probability $1 - P_{rep}$). In the former case, either mutates, with, probability $\epsilon << P_{rep}$, choosing a random location in the network and a random strategy, or else makes a random comparison with another player and copies links and strategy whenever this latter is doing better. Hence, all those who play at cycle t initiate theirn own WL game involving all their neighbors, whether old or brand new. Yet, these games affect only the initators' payoffs.

In order to reason about social optima, this whole situation may be approximated as follows: every node, at each cycle, plays a number of WL games which equals the number of its neighbors plus 1. In particular, if N denotes the whole population (or vertex set) and (N, E_t) denotes the network at cycle t, with $E_t \subset N^{(2)} = \{A \subseteq N : 2 = |A|\}$, then the set of i's neighbors at cycle t is $E_t(i) = \{j \in N \setminus i : \{i, j\} \in E_t\}$. Furthermore, at each cycle t a total number n of weakest link games is played, each with player set $E_t(i), 1 \leq i \leq n$. Let Γ_i^t denote the weakest link game played by i and all its neighbors at cycle t. Then, at each cycle t we have a whole game Γ^t involving all players and consisting of the sum or composition of the n games Γ_i^t, i.e., with some abuse of notation, $\Gamma^t = \sum_{i \in N} \Gamma_i^t$. This composite game does not seem to admit a potential. Still, it is definitely a pure common interest game where the unique social optimum attains when $s_i = 19$ for every $i \in N$, in which case $u_i(s) = 20 \cdot (|E_t(i)| + 1)$ for every $i \in N$ and every cycle t.

Affiliation of authors

- Giovanni Rossi: The University of Bologna, Dept. of Computer Science, Mura Anteo Zamboni 7, 40127 Bologna, Italy. giorossi@cs.unibo.it
- Stefano Arteconi: The University of Bologna, Dept. of Computer Science, Mura Anteo Zamboni 7, 40127 Bologna, Italy. arteconi@cs.unibo.it
- David Hales: The University of Bologna, Dept. of Computer Science, Mura Anteo Zamboni 7, 40127 Bologna, Italy. dave@davidhales.com