Deliverable 4.3.4
Evolutionary approaches to coalition formation in dynamic nets

Due date of deliverable:	December 2007
Actual submission date:	December 2007
Dissemination level:	PU – public
Work Package 4.3:	Models and Methods for Dynamics, Evolution and Self-Organization
Participants:	UniBO CTI UCY RWTH

Authors of deliverable:

Ozalp Babaoglu (babaoglu@cs.unibo.it)
Alexander Fanghänel (alex.fanghaenel@gmail.com)
David Hales (hales@cs.unibo.it)
Spyros Kontogiannis (kontog@cti.gr)
Andrea Marcozzi (marcozzi@cs.unibo.it)
Edoardo Mollona (mollona@cs.unibo.it)
Paul Spirakis (spirakis@cti.gr)
Berthold Vöcking (voecking@cs.rwth-aachen.de)

Abstract

This report comprises the report for the D4.3.3 deliverable (in association with the relevant software[1]) for workpackage WP4.3 in Subproject SP4 of the DELIS (Dynamically Evolving Large-scale Information Systems) Integrated Project.

The essential goal of the DELIS project is to understand, predict, engineer and control large evolving information systems. In this workpackage we report on work exploring the fundamental problem of coordination in networks, particularly when agents or nodes within the network behave in a selfish way, that is, considering their own benefits rather than system level benefits.

In this report we outline two lines of work which address interaction dynamics from an evolutionary perspective and an economic game theoretical analysis of micro payments. In section 1 we give a brief outline of recent work which explores, for the case of random payoff matrices, the nature of possible evolutionary stable strategies (ESS). In section 2 we report on recent work extending the FirmNet model, inspired by P2P and novel organisational theory models of firm production. Here a traditional hierarchical (command) organisation structure is compared with a self-organised peer structure - in which nodes may move based on copying of those with higher payoffs. Interestingly, it was found that, hierarchies can out perform self-organising approaches if the number of social links is low, yet with a large number of social links self-organisation can outperform hierarchies (under certain conditions). Further it is shown that hierarchies are less efficient in terms of accumulated firm wealth. Finally in section 3 we outline recent work applying a game theoretical analysis to selfish routing in mobile devices where micro payments can incentivise cooperative routing.[2].

[1]Available from http://peersim.sourceforge.net

[2]Most papers produced within DELIS are available from the DELIS website as DELIS Technical Reports. Where this is the case references are appended with the DELIS Tech Report number in square brackets. This indicates the paper was produced within the DELIS project, not some other project.

Contents

1 On the Support Size of Stable Strategies in Random Games

In [10] we studied the support sizes of evolutionary stable strategies (ESS) in random evolutionary games. We prove that, when the elements of the payoff matrix behave either as uniform, or normally distributed independent random variables, almost all ESS have support sizes $o(n)$, where n is the number of possible types for a player. Our arguments are based exclusively on a *stability* property (proved by Kingman [9]) that the payoff submatrix indicated by the support of an ESS must satisfy. This is an exponential improvement on the frequency of ESS in random evolutionary games, compared to a previous work of our group on this matter [11].

We then combine our normal–random result with a recent result of McLennan and Berg [12], concerning the expected number of Nash Equilibria in normal–random bimatrix games, to show that the expected number of ESS is significantly smaller than the expected number of symmetric Nash equilibria of the underlying symmetric bimatrix game.

2 Peer production compared with hierarchy in dynamic networks

2.1 Introduction

In this section we overview on-going work with the FirmNet model [13] in which self-organising network structures guide a system toward co-ordinated production behaviour. We compare a set of different peer mechanisms with a traditional hierarchical organising approach. We consider the different approaches effect on firm wealth and efficiency. Here we give an overview of the main results from the work, further detail is available in the associated technical reports [13, 14].

2.2 Hierarchy and Market

For decades our common understanding of the organization of economic production has been that individuals order their productive activities either as employees in firms, following the directions of managers (hierarchy), or as individuals in markets, following price signals. Since we are in a knowledge economy, as long as the jobs become more knowledge-oriented, these classical form of organization may result ineffective.

Hierarchy uses authority to create and coordinate horizontal and vertical division of labor. Adler [2] suggests that when specialized units are told to cooperate in task that typically encouter unanticipated problems requiring novel solutions (this is typical of the knowledge economy), the hierarchical form gives higher-level managers few levels with which to ensure that the participating unit will collaborate. By their non-routine nature, such tasks cannot be preprogrammed, and the creative collaboration they require cannot be simply programmed. Hierarchy hence results week and firms have to look for other models.

Knowledge is a “public good” and hence the market/price mode forces a trade-off between production and allocation. On the one hand, production of new knowledge would be optimized by establishing strong intellectual property rights that create incentives to generate knowledge. On the other hand, not only are such rights difficult to enforce, but more fundamentally, they block social optimal allocation. Adler [2] argues that allocation of knowledge would be optimized by allowing free access because the marginal cost of supplying another consumer with the same knowledge is close to zero.

Knowledge-based jobs are also leading to increasingly incomplete employment contracts. It is not possible to state in a contract all the duties of an employer and an employee: this happens in particular when the the job is an intellectual job and the “supervisory problem” emerges. When the job is very specific, the employer cannot check if the employee is actually working well or if he really has the required skill (maybe he doesn’t really know which are the required skills for a certain job); on the other hand the employee could be less protected due to the lacks in the employment contract. Benkler [3] argues that where agents, effort or resources cannot be specified, they cannot be accurately priced or managed, he also states that human creativity is a difficult resource to specify for efficient contracting or management.

In such context "reciprocity" between employer and employee, becomes an alternative to authority: employer instead of using authority on employees, have to cooperate with them. This new form of organization is more similar to Peer-production.

Hodgsons [7] suggests that the lack of managerial control on knowledge-based jobs, especially when knowledge is tacit and cannot be codified impairs and bounds the appliance of traditional employment contracts. The nature of the contracts should evolve along with the evolution of the distribution of bargaining power.

2.3 Peer-production

By "Peer-production", we mean a form of production of goods and services entirely based on auto-organized communities of voluntary individuals with the aim to pursue a common goal. Actually Peer-production is also a mix of hierarchy and self-organization where expert members of the community give the guidelines for the production. Generally in these communities individuals work for free and they generally do this for social motivations like becoming popular or passion for a certain topic. Classical example of Peer-production are given by 'Linux" and "Wikipedia". Of course this kind of production cannot be applied on every kind of product of service. It's important that individuals working on a certain project can do this in a small time and with a limited effort of resources. This is why it is generally applied to knowledge-based services or goods for which common tools are enough. Anybody can contribute to Linux or Wikipedia spending their free time at home, they just need a PC and an internet connection.

It is also important that tasks can easily be divided into little pieces such that many people can work on them in a reasonably small time. Nowadays billions of people around the world can contribute and cooperate to realize any kind of product which requires *creativity* a *personal computer* and *an internet connection*. Now production costs, for certain kind of activities, are very low and anybody can produce or exchange several kind of goods or services (like newspapers) without the need of market-based or firm-based models. This may sound like a threat for companies but in some cases it can be an opportunity for firms ready to take advantage of such creative power constituted by millions of enthusiastic people in their business areas.

The main characteristics of the products and services realized through peer production are (after Benkler [3]):

- no hierarchy, no market: the quantity of the goods realized with peer production cannot be decided neither by an authority (as in the hierarchy), nor by price (as in the market). It is decided by the spontaneous encounter between tasks and skills;
- individuals involved in a peer community must hold the production means (the same happens in the market);
- the final product is a "common good" with a public open license.

These characteristics are perfectly respected by the Open Source communities. In recent year we have seen how the main benefit of the peer production have been given to the Open source community. Von Krogh and von Hippel [16] have shown how these communities have benefited greatly form the advent of the Internet, which has enabled members to interact and share resources extensively. The most famous example is the Linux operating system which has gained lot of popularity through the years and millions of buyers and users worldwide.

However for such kind of projects the main ingredients are the creativity and the passion of the thousand of people daily working on them. Many are the reasons why the reasons why they decide to contribute for free, as make experience on important projects and establish contacts which other developers. On the other hand there are firms like IBM and Intel having employee working on open source projects, meaning that not everybody is really involved for free.

Three are the main conditions that must be respected to allow peer production (after Tapscott and Williams [15]):

- the object to be produced must be information-oriented;
- tasks must be divisible into small independent parts such that individuals can participate contributing with minimal time effort;
- costs relating the integrations of the produced parts and for their reviews must be minimal.

Beside of this, an other important aspect of the peer production is that communities must be endowed of a system for *peer reviews* for controlling the quality of the product and *leaders* able to govern and handle the interactions between members and merge heterogenous contents produced by them.

Peer production hence, provides a framework, within which individuals who have the best information available about their own fit for a task can self-identify for the task [3]. This provides an information gain over firms and markets, but only if the system develops some mechanism to filter out mistaken judgments agents make about themselves. This is why practically all successful peer production systems have a robust mechanism for peer review or statistical weeding out of contributions from agents who misjudge themselves.

We believe that the mechanisms which are behind the peer production can be derived from the mechanisms behind P2P networks. Both firms and P2P networks are complex adaptive systems (CAS) and we believe that common methods for organizing both exists.

2.4 The FirmNet Model

Modeling and simulation constitute a fundamental element of the research design. We used an agent-based model to simulate interaction among professionals holding specific expertise and project managers, which receive from clients tasks to be performed. In our model firms are represented indirectly as a network of agents that receive tasks from clients and offer to professional a reward in exchange of their collaboration. The FirmNet model [14] should be viewed as an "artificial society" type model (i.e. similar to the SugarScape model of [6]) that allows to express formally (computationally) a number of hypotheses about potential processes that may occur in organizations but in a stylised and executable manner such that experiments can be performed to deduce the consequences of those hypotheses when they are combined in complex, adaptive systems (CAS). We therefore purposefully present a simplified model in which we hope to capture the kinds of complex dynamics in which we are interested.

We implemented two different models: a *hierarchical* model (HI) and a *marlek-based model* (we called it self-organizing, SO). The aim is to understand which one of the two decision making processes performs better in some specific environment. In both models we have two kind of agents: the *node-skill agents* (*NS*) and the *node-task agents* (*NT*). These agents are nodes in a Peer-to-Peer Network; they all hold a certain skill (S) that they use to perform some task. Agents NT, in addition, play the role of project managers, have direct contacts with clients and receive a certain Task to be completed. To complete a task, it is necessary to complete three jobs, each requiring a different skill and NT agents need to form a team attracting NS agents, which hold the required skill. Thus, the difference between NT and NS agents is that NT agents arbitrage, on behalf of the firm, the relationship between skills and clients. Hence, the model simulates an organizational network in which teams arise having certain skills. In both models the designed organization network is a Peer-to-Peer Network in which each node has a maximum number of links (network degree). Each link is bidirectional; a connection of a node a to another node b implies a connection of node b to node a. Links are undirected so the entire network can be considered as an undirected graph where each vertex is a node and each edge is a link. In both models we can distinguish two phases: an *Interaction* phase and a *Network Evolution* phase. The first is the one in which NT try to complete the task, also asking its neighbors; the second is the one in which the network evolves according to a self-organizing or a hierarchical policy, forming teams of specialists. However the main differences between the HI and the SO model are the following: in HI all the NS agents periodically receive a fixed wage even though their *skill* is not used for any job and once they are asked for working on a certain job, they cannot refuse unless they are busy; in SO instead, a NS will accept a call for a job only if not busy and if the margin (commission) it will receive from NT is greater than its own satisfaction threshold; moreover, NS will not receive a fixed salary at a fixed period, but only the commission payed by the NT after the entire task is completed.

So with these two models we try to develop a number of hypotheses about potential process that may occur in an organization. We do this computationally using computer simulation. As we said in our model firms are represented as Peer-to-Peer networks and we performed simulation using PeerSim[3], an agent-based platform for testing P2P protocols developed at the Department of Computer Science of the University of Bologna.

The simulation time is divided into cycles. The network is composed of 1000 nodes (agents) and at cycle 0 the 25% of the them receive a task to be completed; so we have 25% of NT and 75% of NS. The tasks are produced selecting at random three values from a set of five elements ($J \in \{1, 2, 3, 4, 5\}$); the receiving nodes will then act as a *project manager* and will start looking for employees among their immediate neighbors (this is the interaction phase). After this, with a certain probability the Network Evolution phase takes place. Basically the nodes invoking this phase select a random node in the network and then some rewiring action takes place (changing of the links). In the HI model only NT perform this phase and the selection is made on the entire network. In the SO model this phase can be performed by anybody and the selection is made only on a portion of the network. In the current version of both our models, when a node i selects a node j for the evolution phase, if i is a NT it will keep j into account only if it has a skill suitable for the task, else it will be discarded. As we said in the HI model only NT invoke this phase searching for NS with the right skill; in SO this phase can be invoked by anybody: if it is invoked by a NT there will be a preferential attachment for NSs, if it is invoked by a NS the preference will be NTs. If this preference cannot be realized, a preference will be given to those nodes having higher degree.

2.5 Hierarchy versus Self-Organisation

Essentially the hierarchy approach (HI) assumes that manager agents (NT) have the intelligence and power to recruit nodes with required skills from the entire population. But to achieve this all nodes receive a payoff if they work on tasks or not. The self-organising approach (SO) relies on all nodes moving in the network based on their own local decision processes - based on their own self-interest.

Here we give a brief overview of the results of a comparison between the hierarchy (HI) and self-organisation approaches (SO). Figure 1 shows the pseudo-code of the interaction phase for both HI and SO approaches.

We performed a large number of experiments with our models trying to work out which one performs better in some situation investigating both in a dynamic environment (in which NT periodically

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

```
At each cycle if N_i == NT :
    if (N_i != busy) && (N_i.task.contains(N_i.skill))
    then N_i = busy; N_i.task.remove(N_i.skill);

    if (N_i.task != empty) then for all N_i neighbors j:
    if (N_i.task.contains(N_j.skill) && (N_j != busy)
            && (N_j.benefit > N_j.beta)) then
            N_j = busy;
            N_j.task.remove(N_j.skill);
            endif
     endif
```

```
At each cycle if N_i == NT :
if (N_i != busy) && (N_i.task.contains(N_i.skill))
then N_i = busy; N_i.task.remove(N_i.skill);

if (N_i.task != empty) then for all N_i neighbors j:
if (N_i.task.contains(N_j.skill) && (N_j != busy)
        N_j = busy;
        N_j.task.remove(N_j.skill);
        endif
endif
```

Figure 1: Interaction phase pseudo-code: the code on the left is executed in the Self-Organizing model; the one on the right in the Hierarchical model.

substitutes its task with a new different one) and a static environment (in which the task is always the same). We adopted two measures for evaluating them: the percentage of completed tasks (Pct) calculated as the ratio between the number of tasks completed and the number of tasks injected and the firm wealth (wealth) calculated as the total wealth of NTs minus the total wealth of NS.

We performed experiments with different network degree values (maximum number of links of each node: d3, d5, d10, d20). We found that hierarchy outperforms self-organizing in terms of percentage of task completed. As the size of intra-organizational networks increases (this latter measured as the "degree"), SO algorithm improves. The wealth created under hierarchical mechanism is higher than SO only for low network degree. When degree of intraorganizational network increases, SO algorithm produces significantly more profits. The clear argument that emerges from the simulation experiments is that the selection of a hierarchal or a market-based mechanism of control entails the trade off between reliability and costs. At a cheaper price SO gives the same good performaces as HI but only when the organisational environment keeps a high frequence of interactions together with a large number of links at the same time. Hence, it is clear that the efficiency of the market-based model depends on the structure of the network that can be formed, for example, when the network degree is high.

2.5.1 Static Environment

In this section we give an illustration of the results obtained with the static environment, where manager nodes (NT) periodically receive always the same task.

In figures 2 and 3 results are shown for the hiearchy approach and two variants of the self-organising approach (CSLAC and CSLAC2). We can note how in general the hierarchy gives better results than the two self-organizing mechanism in terms of percentage of completed tasks (PCT). Interestingly we found that CSLAC2 gives better results than CSLAC and that it performs even better than the hierarchy when the network degree is 20. While in CSLAC nodes make new links with nodes having high utility in CSLAC2, new links are made in a more accurate way: for example when a NT selects a NS, it will try to link to him only is it has the right skill. We believe that these more accurate rules are the reason why CSLAC2 performs better than CSLAC. In terms of firm wealth, we can note how hierarchy is costly because all nodes are paid even if they are not productive. Of course also here CSLAC2 performs better than CSLAC for the same reasons mentioned before.

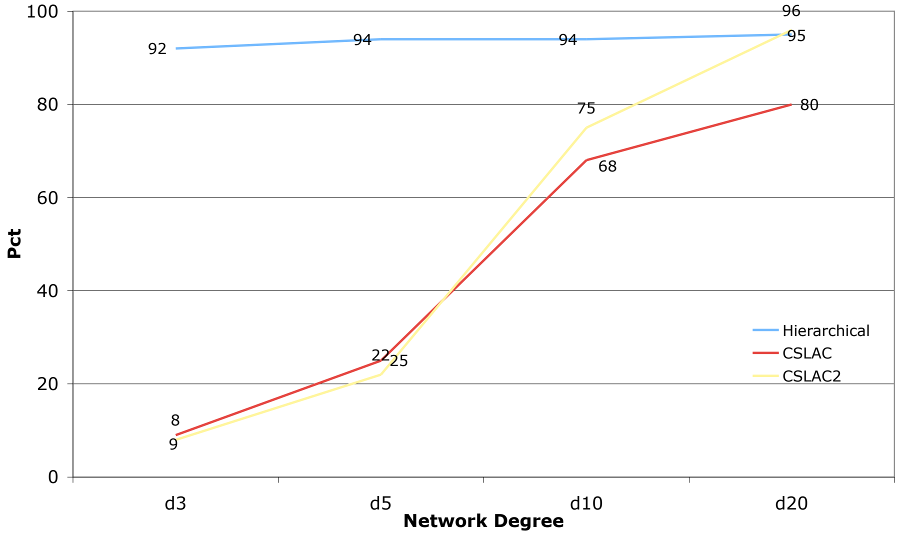

Figure 2: Percentage of completed task (PCT) for networks with different node degrees (undirected links and uniform degree over all nodes) in a static task environment. Here all NT (manager nodes) receive the same tasks over time. CSLAC and CSLAC2 are two different self-organsing peer protocols. Note that the Hierarchical approach outperforms the self-organsing approaches over most of the chart. This is because in this scheme the NT nodes learn to recruit the right nodes with the right skills to complete the given tasks.

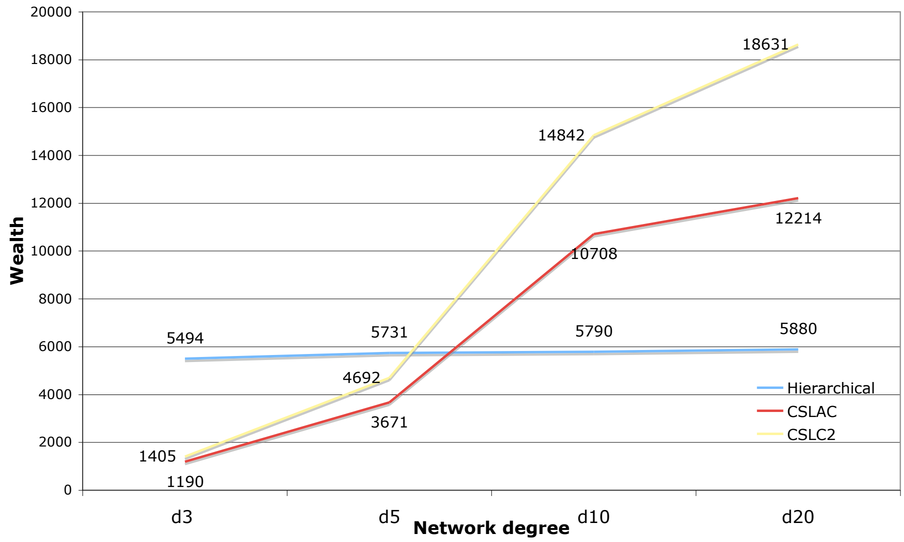

Figure 3: Total population wealth for the same experiments as shown in figure 2. Note that the self-organised approaches - CSLAC and CSLAC2 outperform the Hierarchical organisation when the degree increases above about five. This is because the hierarchical organisation requires that all nodes be given some payoff even if they are not productive. In the self-organising approach unproductive nodes do not get payoff. Hence wealth in-equality is higher (not shown here).

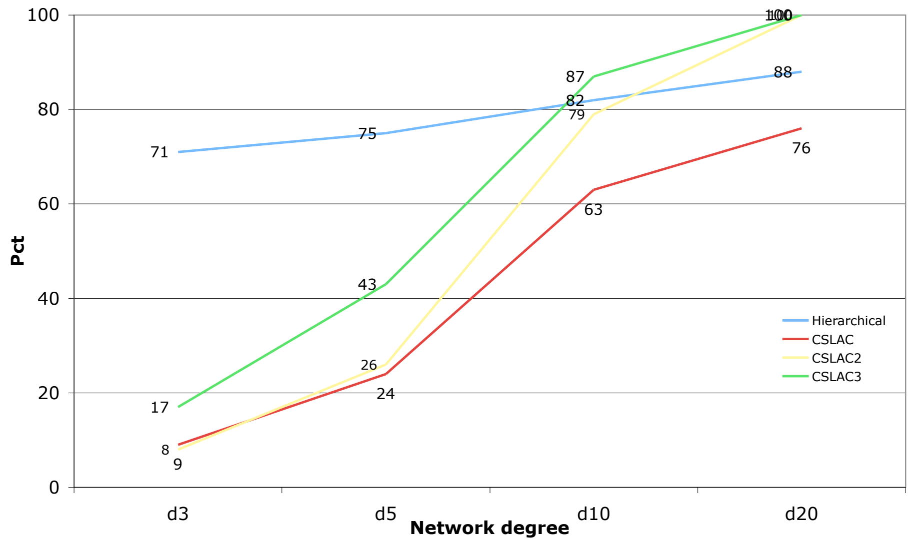

Figure 4: Percentage of completed tasks (PCT) for different node degrees in a dynamic task environment. Here manager nodes receive randomly generated tasks rather than the same tasks (as previously). The hierarchical organisation does less well than previously. The self-organising peer protocols CSLAC and CSLAC3 do better when the degree is more than about ten. Essentially, since tasks are random over time manager nodes (NT) in the network are less able to benefit from hierarchy because they can not predict which other nodes to recruit.

2.5.2 Dynamic environment

This section contains the results obtained with the dynamic environment. Here nodes receive tasks every 30 cycles. They are randomly generated and different from the previous one.

In figures 4 and 5 we can note how in terms of PCT, here the hierarchy gives worse results than in the static environment: this means that authority is less effective when the task rapidly changes. On the other hand CSLAC performs better here, while CSLAC2 gives more or less the same results. We think that in an environment in which tasks rapidly change (typical of a knowledge economy) a self-organising methods can give good results even though also here we can note how these good performances are obtained as the size of intra-organizational networks increases (this latter measured as the network degree). In terms of firm wealth, also here we can note that hierarchy costs much more than SO.

In this setting we introduced the CSLAC3 algorithm characterized by the fact that no rewiring action takes place. Surprisingly we found that it performs better than CSLAC and CSLAC2. We think that this may be due to the fact that with CSLAC and CSLAC2 when nodes move often lose links and sometimes they may produce isolated nodes that do not improve the performances of the firm. On the other hand with CSLAC3 nodes are never isolated and NTs, since always well linked, can, over time, complete a large number of task. We believe that this must be investigated making a structural analysis of the network.

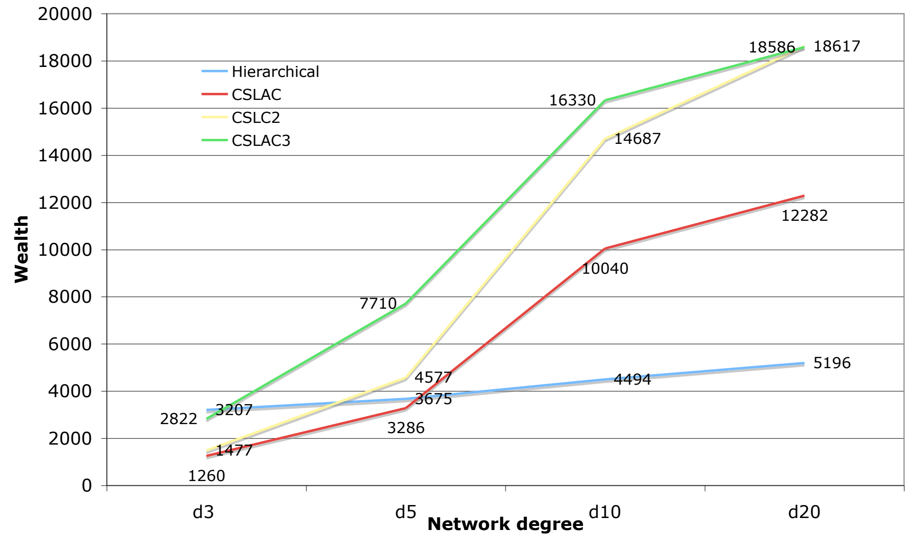

Figure 5: Total population wealth for the dynamic task environment. Hiarachy again performs poorly due to the requirement to give payoff to all nodes even when they are not productive. The failure of the central manager nodes (NT) to recruit the right skills leads to more unproductive nodes than in the static task environment when managers could predict the skills they need.

2.6 Summary

The work presented here contributes to the area of studies that is concerned with emergent organizational problems in a knowledge economy. In addition, we present an attempt to bridge studies in computer science, dealing with the nature and the mechanisms of P2P networks evolution and organization theory.

In our work, we wanted to explore under what circumstances networks of specialist may self-organize. In this respect, we believe that the P2P network is a useful concept to address novel forms of organization in a knowledge economy. On these lines, we investigated how agents, with individual incentives and decision-making rules, interact locally and give raise to global organizational structures. We studied how different individual decision-making rules lead to different emergent network of skills, with different performances.

We performed experiments with both the Hierarchical and the Self-Organizing model in order to highlight the differences between the two models and discover in which circumstances they perform better.

3 Payments in Wireless Networks

In hybrid communication networks mobile devices connect in an ad hoc fashion to a base station, i.e., an access point to the wired part of the network. In these scenarios it is of particular importance to reduce the energy consumption for up-link connections from the mobile devices to the base stations as mobile devices have rather limited battery power. Since energy requirements increase super-linear in the distance between two devices, the usage of intermediate nodes forwarding packets can significantly reduce the transmission power in comparison to directly transmitting to the base station. Using mobile devices as relay stations, on the one hand, might also increase the Quality of Service (QoS) due to a reduction of interference. On the other hand, however, the QoS suffers from an increase in latency if packages need to be forwarded several times until they reach the wired part of the network. For this reason only a relatively small number of hops seems to be acceptable. Although the benefits of using multihop connections are convincing from a global point of view, one might ask why participants in a commercially operated network should forward packets of other participants, as this only drains the battery of the forwarding node, thus, bringing a negative utility to that participant. The usual response to this objection is that the forwarding nodes should receive a payment for forwarding packets.

Previous work in this area considers games, in which selfish players want to connect their terminals at the lowest possible cost [4]. Thereby they are allowed to use links for free that other players already established. Others discuss micro-payments [8] or incentive compatible payment schemes [5].

In [1] we present a game theoretic study of hybrid communication networks in which mobile devices can connect in an ad hoc fashion to a base station, possibly via a few hops using other mobile devices as intermediate nodes. The maximal number of allowed hops might be bounded with the motivation to guarantee small latency. Our model assumes that intermediate nodes on an uplink path are reimbursed for transmitting the packets of other devices. The reimbursements can be paid either by a benevolent network operator or by the senders of the packets using micropayments via a clearing agency that possibly collects a small percentage as commission. These different ways to implement the payments lead to different variants of the hybrid connectivity game.

Our equilibrium analysis gives a game theoretical motivation for the implementation of micropayment schemes in which senders pay for forwarding their packets. We further support this evidence by giving an upper bound on the Price of Anarchy for this kind of hybrid connectivity games that is independent of the number of nodes, but only depends on the number of hops and the power gradient.

4 Conclusion

The aim of this work package, in general, has been to bring together game theoretic proof and evolutionary dynamics. The work outlined in section 1 has shown how formal focus on the ESS within game interactions can lead to proofs in general over randomised payoff matrices. In section 2 we have seen how out-of-equilibirum networks perform with both hierarchical and self-organised protocols. Developing and applying a formal approach to such networks will be a highly interesting area of work going forward. It would appear that within sufficiently stable task environments organisational networks could converge close to an efficient equilibrium point. If this could be known a priori by the “manager nodes” then it could be possible for them to act efficiently bringing the network to optimal equilibria. But this needs further study.

References

[1] Ackermann, H. and Briest, P. and Fanghänel, A. and Vöcking, B. (to appear) Who Should Pay for Forwarding packets?

[2] Adler, P. (2001). “Market, hierarchy, and trust: The knowledge economy and the future of capitalism”. Organization Science: 215-234.

[3] Benkler, Y. (2002) “Coases Penguin, or, Linux and The Nature of the Firm”. Available at http://www.benkler.org/

[4] Eidenbenz, S. and Kumar, V. S. A. and Zust, S. (2003) Equilibria in topology control games for ad hoc networks. Proc. of the 3rd joint workshop on Foundations of mobile computing (DIALM-POMC), pp. 2–11.

[5] Eidenbenz, S. and Anderegg, L. and Wattenhofer, R. (2007) Incentive-Compatible, Energy-Optimal, and Efficient Ad Hoc Networking in a Selfish Milieu. Proc. of the 40th Ann. Hawaii International Conference on System Sciences (HICSS), pp. 293.

[6] Epstein, J. M. and R. Axtell (1996), "Growing Artificial Societies: Social Science from the Bottom Up". MIT Press and Brookings Institution Press: Cambridge, Massachusetts, and Brookings Press: Washington, D.C.

[7] Hodgson, G. M. (1999) "Economics & Utopia"; London, Routledge.

[8] Jakobsson, M. and Hubaux, J. P. and Buttyán, L. (2003) A Micro-Payment Scheme Encouraging Collaboration in Multi-hop Cellular Networks. Financial Cryptography, pp. 15–33.

[9] Kingman, J. F. C. (1988) Typical polymorphisms maintained by selection at a single locus. J. of Appl. Prob., 25:113125, (special issue).

[10] Kontogiannis, S. and Spirakis, P. (2007) On the Support Size of Stable Strategies in Random Games. In 4th Symposium on Stochastic Algorithms, Foundations, and Applications (SAGA 2007), Lecture Notes In Computer Science 4665, Springer Publications. [DELIS-TR-0505]

[11] Kontogiannis S., Spirakis P. (2005) Counting Stable Strategies in Random Evolutionary Games, In 16th Annual International Symposium on Algorithms and Computation (ISAAC'05), Springer-Verlag [DELIS-TR-0176]

[12] McLennan, A. and Berg. J. (2005) The asymptotic expected number of nash equilibria of two player normal form games. Games & Econ. Behavior, 51:264–295, 2005.

[13] Mollona, E. and Marcozzi, A. (2007) "Self-Organisation of Knowledge-Based Dispersed Economic Activity and the Scope of Firms: An Agent-Based Approach". Full paper. In proc. of EURAM 2007. Paris, France. [DELIS-TR-0375]

[14] Mollona, E.; Marcozzi, A. (accepted) FirmNet: The Scope of Firms and the Allocation of Task in a Knowledge-Based Economy. *Journal of Computational and Mathematical Organization Theory (CMOT)* [DELIS-TR-0535]

[15] Tapscott, D. and Williams, A. D. (2007) Wikinomics (La collaborazione di massa che sta cambiando il mondo). Rizzoli Editore.

[16] Von Hippel, E. and von Krogh, G. (2003) "Open-source software and the 'private-collective' innovation model: Issues for organization science". *Organization Science, 14/2: 209–223.*