Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Selfish versus unselfish optimization of network creation

Johannes J Schneider[1] **and Scott Kirkpatrick**[2]

[1] Institute of Physics, Johannes Gutenberg University of Mainz, Staudinger Weg 7, 55099 Mainz, Germany
[2] School of Engineering and Computer Science, The Hebrew University of Jerusalem, Givat Ram, Jerusalem 91904, Israel
E-mail: schneidj@uni-mainz.de and kirk@cs.huji.ac.il

Received 13 July 2005
Accepted 29 July 2005
Published 11 August 2005

Online at stacks.iop.org/JSTAT/2005/P08007
doi:10.1088/1742-5468/2005/08/P08007

Abstract. We investigate several variants of a network creation model: a group of agents builds up a network between them while trying to keep the costs of this network small. The cost function consists of two addends, namely (i) a constant amount for each edge an agent buys and (ii) the minimum number of hops it takes sending messages to other agents. Despite the simplicity of this model, various complex network structures emerge depending on the weight between the two addends of the cost function and on the selfish or unselfish behaviour of the agents.

Keywords: random graphs, networks

ArXiv ePrint: cs.NI/0508029

J. Stat. Mech. (2005) P08007

 1742-5468/05/P08007+13$30.00

Contents

1. Introduction

The Internet, which has changed computing by making access to information and to computing resources available to the world, is a complex system worthy of study in its own right. It came into being through the federation of many different networks originally designed to serve different purposes for different communities, and does not have strong central management. Attempts to understand its behaviour at the lowest level, that of connectivity and topography, have shown that the simplest model, a graph in which sites are connected at random, must be discarded in favour of a graph in which the distribution of the number of links is very heavy tailed. Most sites have very few connections, but there is a power law tail containing very few, very highly connected sites [1]. The earliest maps of the Internet [2] tended to show a fairly tree-like structure, but these were gathered by searching out from a single point along shortest paths, producing essentially a minimum spanning tree to the selected destinations. More extensive searches [3] show that there are many cross-links as well. The most recent, detailed studies of the links between subnetworks, building blocks of the Internet, show an average connectivity of more than six links per site in addition to power law tails in the degree distribution [4, 5].

Other networks, like the World Wide Web, networks of actors with joint movies as edges between them and various kinds of networks between humans, exhibit similar properties [1, 6]. Besides these investigations of the properties of the Internet and of related networks, one is interested in the basic mechanisms leading to this type of network. The growth and shaping of a network is a stochastic process of considerable interest, but the lack of central control makes the use of statistical mechanical models for such complex systems suspect. In fact, such a network appears to be a 'game', in which many independent agents manage its components to suit their own needs. Recent attempts to analyse the implications of this thought [7] have focused on the 'price of anarchy', a catchy term given to the ratio of the cost of the worst-case selfish but stable solution in such a game to the 'social optimum' in which all players cooperate to produce the best solution for the system as a whole. This ratio may be a constant, or may increase with N, the number of players in the game.

The equilibrium in which each agent has chosen a specific configuration for the assets that it manages and, given the configurations chosen by the other agents, has no incentive to change, is called a pure Nash equilibrium [8]. Even when the social cost is carefully defined to be the sum of the individual costs which individual players optimize, the social optimum may not be a Nash equilibrium, if individual incentives destabilize it. Nonetheless, the search for a pure Nash equilibrium or more generally when the agents' information or choices are more restricted, a selfish optimum, is a stochastic process of great interest both as a variation on statistical mechanics and as a model of how large complex systems behave in the real world.

A recent model that we shall consider is 'network creation', introduced in [9] and extended to models which represent actual peer-to-peer data sharing networks formed as overlays in the actual Internet, in [10]. This is the first of what may be many models in which the differences between selfishly driven optimization and global optimization can shed light on real-world complex systems. Determining the full range of possible selfishly optimal behaviour in such models is at least as difficult as combinatorial optimization, and has in fact spawned new complexity classes in computer science [11].

2. The network creation model

In the network creation model, one agent is assigned to each node i of the graph. Each edge between a pair (i, j) of nodes is owned either by the agent on node i or the agent on node j, but it can be used by any agent for transferring messages. The cost of sending messages will depend on our model of message traffic. For simplicity and generality, we shall assume that each agent must send an equal number of messages (one, without loss of generality) to all other agents. This uniform model will never generate the power law tails seen in the real Internet, but it proves to exhibit surprisingly rich behaviour. It requires that the graph be connected. Thus each agent must purchase sufficient edges in order to be connected to all other agents, but as this occurs in an asynchronous parallel process, agents can take advantage of edges purchased by others. Buying an edge costs an amount α. In this model, α is simply a constant and does not depend e.g. on the distance between the nodes i and j or on the required bandwidth for the connection. The cost of sending each message is given by the number of hops (the number of links that it passes over) in the shortest path between the sender and the receiver. The cost of one hop is an arbitrary parameter, which it is conveniently set to 1.

If the network is already connected, each agent still has the problem of whether it wants to buy additional edges in order to reduce the costs induced by the number of hops or to take a larger amount of hops into account in order to reduce the costs incurred by the edges. The decision to which it gets at the end depends strongly on the value of α: if α is smaller than 1, then of course a complete graph with edges between every pair of nodes is preferable. However, if α is very large, then the network is surely only a tree, such that the condition that the graph must be connected is fulfilled and no edge more than needed is added. The most interesting question for us is what structures the networks created by the agents have for intermediate values of α. These structures will depend not only on the value of α but also on the decision of the agents as to when to buy an additional edge. This decision depends also on the behaviour of the agents, whether the agents are selfish, i.e., if they only consider whether their own costs decrease, or not, i.e., if they consider whether the sum of the costs of all agents decreases.

Related to this behaviour, one can write down cost functions for the single agents: the cost function of a selfish agent on node i can be written as

$$\mathcal{H}_{\text{selfish}}(i) = \sum_{\substack{j \\ \text{agents}}} (\alpha \times \eta_{ij} + d_{ij}) \tag{1}$$

with $\eta_{ij} = 1$ if the agent on node i has bought an edge to node j and 0 otherwise and with d_{ij} being the distance between the nodes i and j measured in hops as described above. Analogously, the cost function of an unselfish agent on node i is given as

$$\mathcal{H}_{\text{unselfish}}(i) = \sum_{\substack{j \\ \text{agents}}} \alpha \times \eta_{ij} + \sum_{\substack{k \\ \text{agents}}} \sum_{\substack{j \\ \text{agents}}} d_{kj}. \tag{2}$$

Thus, in the unselfish scenario, each agent considers not only the costs of their own links but also all the costs induced by the distances.

One can also define an overall cost function for the system which is given as

$$\mathcal{H}_{\text{total}} = \sum_{\substack{i \\ \text{agents}}} \sum_{\substack{j \\ \text{agents}}} (\alpha \times \eta_{ij} + d_{ij})\,. \tag{3}$$

This cost function is independent of the behaviour of the agents: it is identical to the sum over all selfish cost functions $\mathcal{H}_{\text{selfish}}(i)$ of the single agents. But it is also basically equal to the cost function of an unselfish agent, as each agent can only make decisions due to the connections they bought on their own and not on the connections bought by other agents.

In these distances d_{ij}, one can also consider the constraint that the graph has to be connected: the agent on node i adds up the number of hops messages take to any other node j to which it is directly or indirectly via other nodes connected and stores this number in the d_{ij}. For all other nodes, which it cannot reach, it sets $d_{ij} = L$ with L being a large number. L has to be larger than α and the maximum possible number of hops in the system to ensure that the graph will be connected at the end of the simulation.

3. Simulation details

Simulating this model, one mostly starts with an empty graph without edges between the nodes, because this is the natural starting point. For comparison, additionally to this 'from scratch' scenario, we investigate a 'from complete' scenario in which one starts with a complete graph in which all pairs of nodes are directly connected via an edge which is randomly owned by one of the nodes. Whereas the 'from scratch' scenario can be justified by the historic development, as there were no connections at the beginning, the 'from complete' scenario can be pictured as if there were a planning meeting of all agents in which they started with a complete graph and with edges which were already assigned to them and they determined in a random order which edges should be removed.

Besides the creation of a starting configuration, a simulation must specify rules for moves, i.e., a prescription for how to change the configuration. An obvious choice for a move from the point of view of an agent is of course a buy/sell-move. One chooses a pair (i, j) of nodes at random with $i \neq j$. If there is no edge between them, then the agent on

J. Stat. Mech. (2005) P08007

node i determines whether it is preferable for it to buy an edge to j or not. According to the behaviour of the agent, it will make its decision. If the situation is improved in its view by buying an edge then the agent on node i buys an edge to j. However, if there is already an edge between i and j, then the buy/sell-move asks the owner of the edge whether it is preferable for it to remove it. This buy/sell-move can also be accepted if the costs for the agent stay the same, i.e., if the move leads to an equally good configuration, as it is anyway nowadays the habit of managers to frequently restructure their companies. If such a restructuring does not lead to any deterioration, it is to be accepted.

Besides this buy/sell-move, which is surely a natural move, one can also implement a switch-move: here a triple (i, j, k) of nodes is randomly chosen in such a way that the agent on node i owns an edge between i and j and that there is no edge between i and k. Now the agent on node i is asked whether it is preferable to delete the edge to j and to add an edge to k at the same time. This additional move might lead to even better configurations. The simulation might otherwise get stuck at configurations which could easily be improved by this switch-move, while the successive application of corresponding sell- and buy-moves would not be accepted because one of them would lead to a deterioration.

After having performed many such moves, the simulation run usually reaches a configuration which is not changed any longer. In the case of unselfish behaviour of the agents, in which the simulation is performed basically according to the total cost function of the system, the simulations often end at a local or even the global minimum in the energy landscape. However, such a local minimum is not always reached due to the ownership of a link: an agent can only delete an edge if it is owned by it. Thus, if it is rather well but not optimally connected because another agent bought an edge to it and if it is not advantageous to buy a further edge as the costs for it would be larger than the savings in the reduction of the distances, it will have to stay with this locally non-optimal situation. The configurations in which simulation runs end up are some kinds of local minima. In optimization, one differentiates between local and global minima: global minima have cost function values which are optimal for the problem, there is no configuration at all with a better cost function value; in contrast, local minima only have a cost function value which is better than the cost function values of all configurations which can be reached by the application of one move from this local minimum. In the world of multi-agent systems, a Nash equilibrium corresponds to the global optimum. We end up here at local minima in which the simulations get stuck and which cannot be improved by the application of any move available to an agent. We test for reaching such a local minimum explicitly. Before stopping the simulation we explore all possible moves that are available to a single agent. It could also be that a simulation run never gets stuck in a stable configuration because the system can jump between equally good configurations.

Thus, we cannot let a simulation run through a possibly infinite loop. In our implementation, 10 000 steps are performed. In each step, the first 100 moves are tried in a random way, i.e., the nodes which buy, sell and switch edges and their neighbours are selected at random. If all of these 100 moves are rejected, then all possibilities for moves are checked deterministically but in a random order. This will ensure that in each step at least one move is accepted. If during this complete search for a move to be accepted no acceptable move is found, then the simulation run is already stuck in a local minimum or

J. Stat. Mech. (2005) P08007

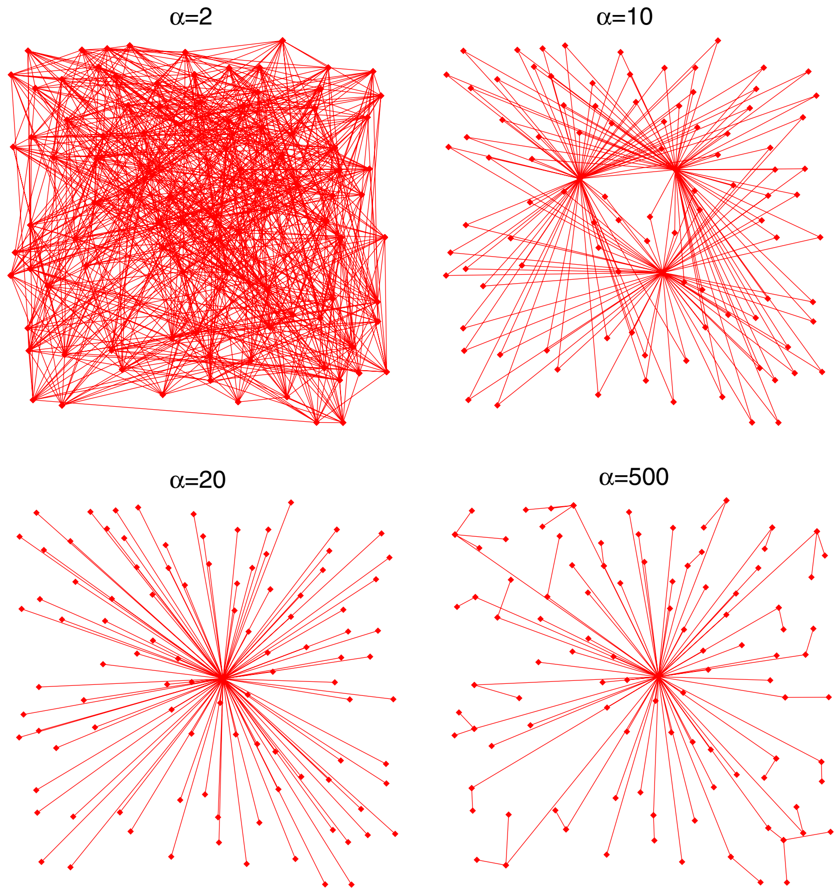

Figure 1. Examples of resulting configurations: depending on the value of α, simulations lead to different interesting final configurations.

Nash equilibrium, such that the simulation can be stopped ahead of time. Otherwise, the simulation proceeds with the next step.

In our simulations, we either use only buy/sell-moves (b/s) or use both buy/sell-moves and switch-moves in a random order (b/s + sw) with equal probability. Furthermore, we either start 'from scratch' (fs) or 'from complete' (fc) and the agents exhibit either a selfish or an unselfish behaviour. Thus, we have all in all eight different scenarios. For each of these scenarios, we consider several values for α, namely 0.5, 0.7, 1, 1.3, 1.5, 1.7, 2, 2.3, 2.5, 2.7, 3, 3.3, 3.5, 3.7, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 300, 400 and 500. For each scenario and for each value of α, we performed 100 simulation runs over which the results are averaged.

4. Computational results

The final configurations of our simulations strongly depend on the size of α: obviously, the simulations end with fully connected graphs for $\alpha < 1$, as buying an edge is cheaper than letting a message perform an additional hop. Figure 1 shows typical final configurations reached at larger values of α. Although these examples were found using

J. Stat. Mech. (2005) P08007

selfish optimization, from scratch, similar results are found starting from a complete graph, or using global optimization. For $\alpha = 2$, the example graph found is dense, but no longer complete. For different configurations found at $\alpha = 2$, the number of edges in the configurations varies between 99 and 917 with an average of 613. We often get a star with one centre node to which all other nodes are connected for $\alpha = 20$. For even larger values of α, the simulations mostly produce trees which, at first, are close to stars in structure. But also some rather unexpected structures occur: for $\alpha = 10$, we often get a structure with three centre nodes. All other nodes are connected to two of these three centre nodes. There are two edges between these three nodes, such that messages need only up to two hops to get from the sender to the receiver. In graph theory terms, the solution for intermediate values of alpha maintains the diameter of the network, the longest minimum path between any two points, at two hops.

Now we want to study the behaviour of our simulations and the final configurations statistically.

Figure 2 shows the minimum, mean and maximum numbers of steps that the simulations took. Please note that there is a maximum of 10 000 steps: if the simulation has not reached a stable Nash equilibrium or a local minimum, it breaks after this number of steps. It might be that such a broken simulation might have ended soon after, but it also might have gone on forever. Furthermore note that the minimum number of steps is always 1: thus, even if the simulation starts in a Nash equilibrium or a local minimum, such that no move is accepted, one step is counted. There are indeed simulations which were started from such minima: as already mentioned, a complete graph with edges between all pairs of nodes is optimal for $\alpha < 1$. Thus, if we start from our 'from complete' scenario for some $\alpha < 1$, there is no move leading to any improvement such that the simulation can already stop at this point. For selfish agents working on $\alpha = 1$, we find that the runs are only ended by the rule that a simulation cannot exceed 10 000 steps. Here many neighbouring configurations, which can be transferred via a move to each other as the move does not lead to a deterioration, are degenerate.

Then in the range of intermediate values of α, we see several structures, of which the most interesting is the one for unselfish agents using buy/sell-moves only: here the number of steps the simulation needs to come to an end strongly depends on whether α, when it takes an integer value, is even or odd. The calculation time is much larger for even integer values, as can be seen at the minimum, at the mean and at the maximum numbers of steps in the range $2 \leq \alpha \leq 18$. The time is often only limited by the maximum number of allowed steps, whereas simulations with odd integer values always come to an end in much less than 10 000 steps. The reason for this at first sight strange behaviour is that every distance d_{ij} is counted twice in the Hamiltonian for the unselfish agents, namely once as distance d_{ij} from i to j and once as distance d_{ji} from j to i. Now if an unselfish node decides to add a further edge for a cost value of α which reduces the distances of e.g. $\alpha/2$ pairs of nodes by an amount of 1, this move leads to an equally good configuration. Thus, both this move and its inverse move are accepted, leading to degenerate configurations between which the simulation can jump endlessly. The final configurations of these simulation runs contain several centre nodes; all other nodes are connected to two of these centre nodes, which are partially connected to each other.

Such degeneracies do not occur for odd values of α, as here adding or deleting an edge leads to either a better or a worse configuration. If we had used a factor of 1/2 with which

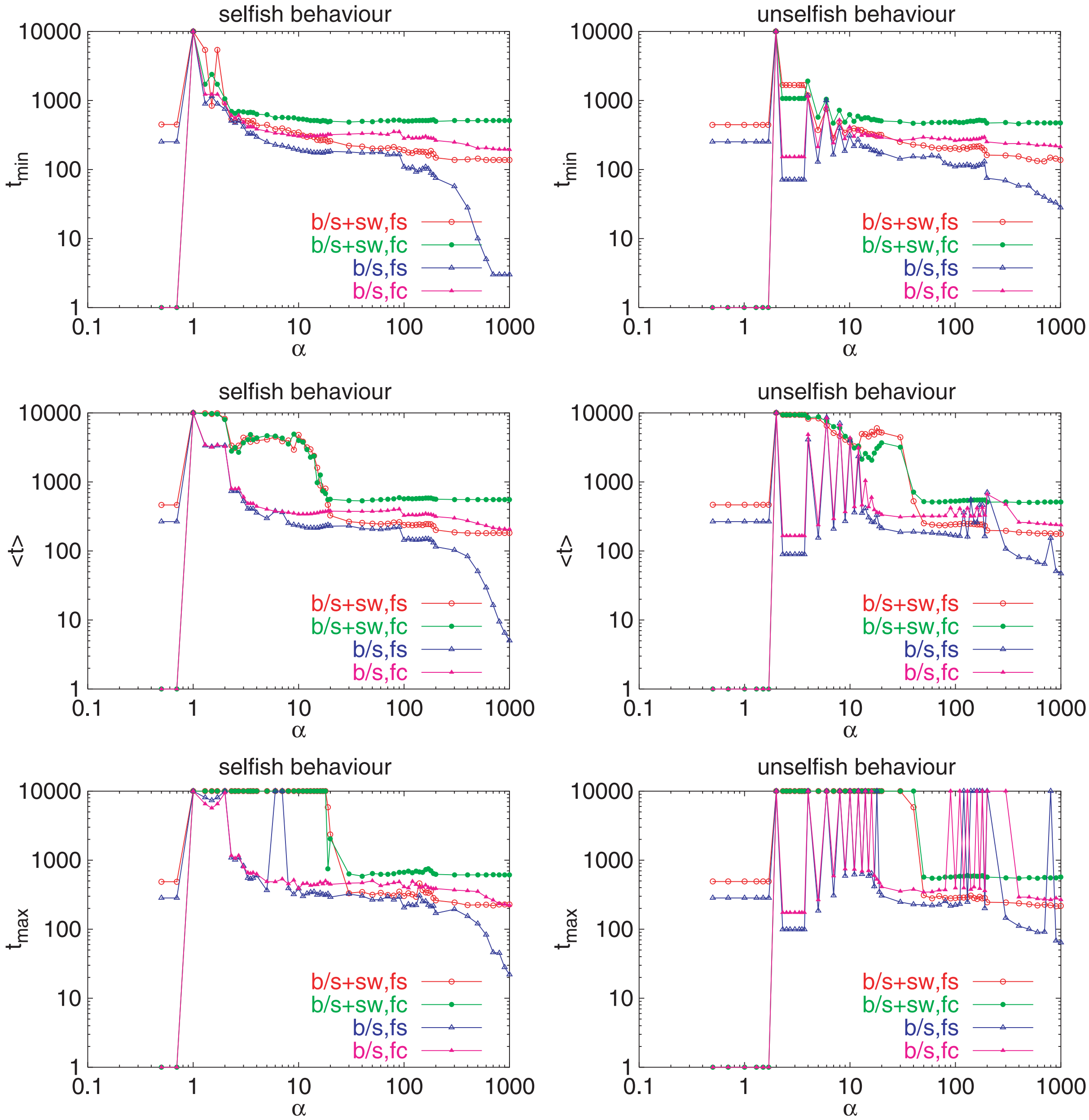

Figure 2. Minimum (top), mean (middle) and maximum (bottom) numbers of steps until the system reaches either a Nash equilibrium or a local minimum: the results for the selfish behaviour are shown on the left, for unselfish behaviour on the right. In each graphics, empty symbols mark results for the 'from scratch' scenario (fs), filled symbols results for the 'from complete' scenario (fc). Results for using both moves (b/s + sw) are shown as circles, results for using buy/sell-moves only (b/s) as triangles.

we had multiplied the sum over the distances in the cost function of unselfish agents, we would see degeneracies at all integer values of α and much quicker convergence on the non-integral numbers in between.

J. Stat. Mech. (2005) P08007

Sometimes we also get an increase in the calculation time at $\alpha \approx N$ in the unselfish case using only the buy/sell-moves. Here some of our simulations also run into configurations which are degenerate with their neighbours.

Please note that these times have so far been given in steps. However, there is a large difference in computing time between simulations with selfish agents and simulations with unselfish agents. Working with selfish agents, only the distances from this agent to all other agents have to be determined. Here the calculation time for a move goes with $\mathcal{O}(N)$. For unselfish agents, the new distances between all pairs of agents have to be determined, so there the calculation time goes with $\mathcal{O}(N^2)$. Thus, we had to spend weeks of calculation time on dozens of work stations for our simulations of unselfish agents, whereas we could perform the simulations of selfish agents within a comparatively short time.

Next we have a look at the minimum, mean and maximum numbers of links in the final configurations, which are shown in figure 3. We get similar pictures here for the various scenarios: for $\alpha < 1$, we generally get completely connected graphs with $N \times (N-1)/2$ edges. As we always use systems with $N = 100$ nodes, this number is 4950. For large values of α, the number of edges is always given by $N - 1$, so we always have trees here. In the intermediate regime, there is a gradual decrease of the number of edges in the system, which is nearly monotonic. For selfish agents using both the buy/sell-moves and the switch-moves, we get a large plateau for the maximum number of links at intermediate values of α. This maximum number is given by 196, i.e., by $2 \times (N-3)+2$; the corresponding structure is the graph with three central nodes which was shown in figure 1.

Furthermore, we are interested in the average distance between two different nodes in the network, which is measured as the number of hops a message needs to get from one node to the other. Looking at figure 4, we find three different regimes depending on the value of α: for small α, we always get completely connected graph, such that the average distance between two nodes is 1. Increasing α, the average distance increases to a value of ≈ 2. We get instead a plateau at this value for intermediate α. For large α, the average distance increases again. In this regime, we find a significant difference between the simulations working both with buy/sell-moves and switch-moves or using only buy/sell-moves: without switch-moves, the average distance explodes with increasing α, whereas the increase is comparatively small for those simulations in which switch-moves were implemented. Using both moves, selfish and unselfish behaviours of the agents lead to roughly the same average distances. But unselfish behaviour pays off when using only the buy/sell-moves.

This result can be interpreted from two different points of view. From the point of view of a moralist, one can say that unselfish behaviour is superior to selfish behaviour and thus leads to overall better results for all agents. Therefore, one should never allow one's own ego to dictate the decisions that one makes. On the other hand, from the point of view of optimization, one has to state that in the case of the unselfish agents, basically the cost function of the overall system was considered when performing a move. Thus, every move tried to minimize this overall cost function, whereas the selfish agents worked only on their local part of the cost function. Therefore, from the point of view of an optimizer, it is quite clear that unselfish optimization has to lead to solutions better than or at least as good as selfish optimization.

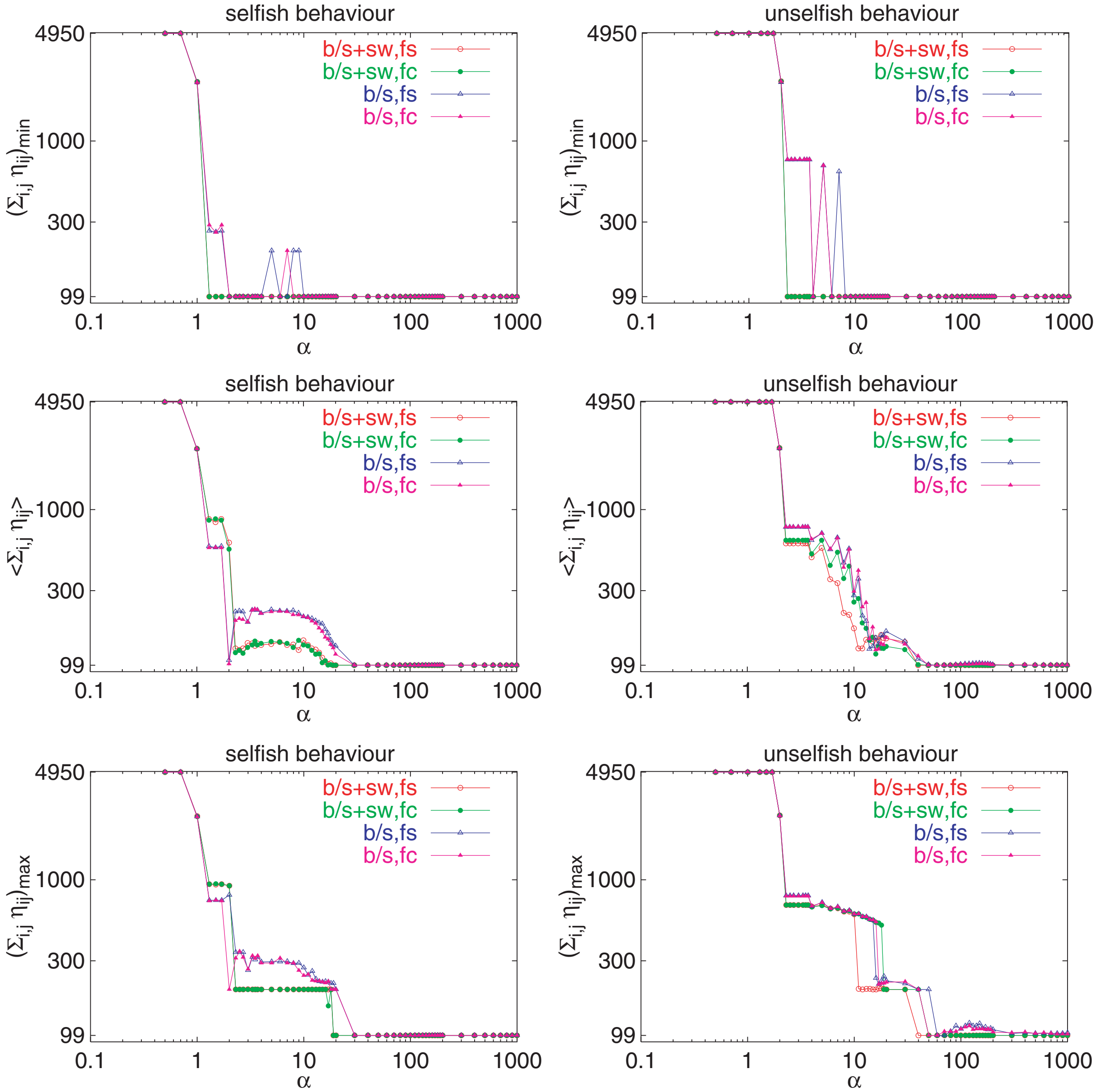

Figure 3. Minimum (top), mean (middle) and maximum (bottom) numbers of links in the final configurations.

Finally, we have a look at the probabilities of the resulting configuration being a tree or a star, which are shown in figures 5 and 6. For selfish agents, the probability that the resulting configuration is a tree is nearly 1 at $\alpha = 2$ if only buy/sell-moves are used. In this case, the probability drops to zero on increasing α and then increases again for $\alpha \geq 10$. For $\alpha \geq 30$, it is again equal to 1. If also using the switch-moves, the probability increases from zero to ≈0.75 at $\alpha \approx 2.5$, then decreases and stays at roughly a plateau of ≈0.6 for $3 \leq \alpha \leq 10$ and then increases to 1, reaching this value at $\alpha \approx 20$. The corresponding picture for the probability that the resulting configuration is a star is rather similar, except that the probability breaks down from 1 to 0 at $\alpha \approx N$.

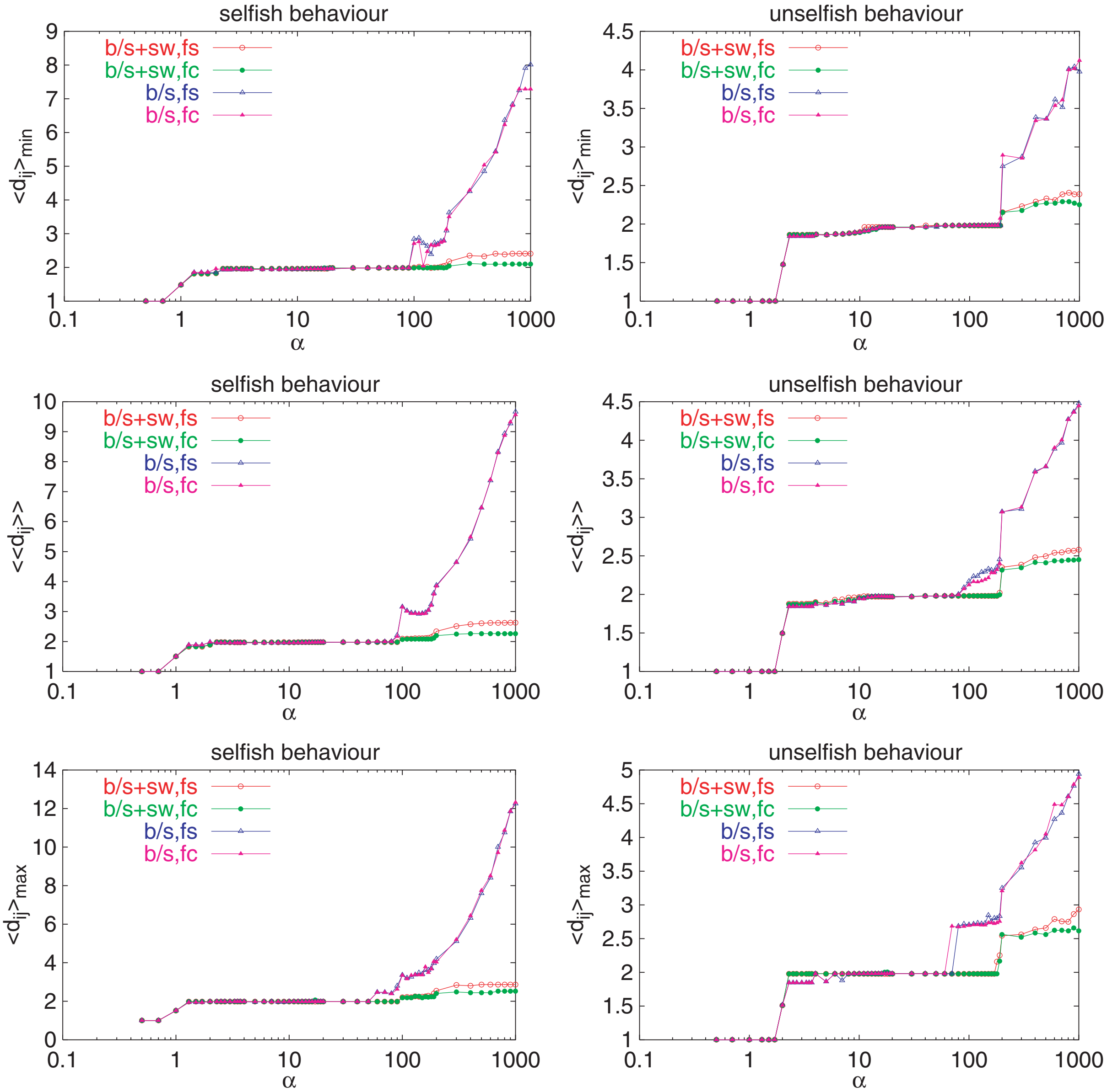

Figure 4. Minimum (top), mean (middle) and maximum (bottom) average distance in the final configurations.

In the case of unselfish behaviour, we get a different picture for these probabilities: the probability increases from 0 to 1 in the range $2 \leq \alpha \leq 50$ with a short breakdown at $\alpha = 20$. For large α, we get different results: if using only the buy/sell-moves, the probability that the resulting configuration is a tree breaks down when α approaches N and finally fluctuates around 0.8 for large α. If also working with switch-moves, the probability stays at 1. Again the picture is the same for the probability that the final configuration is a star, except that there the probability breaks down for $\alpha > N$. This decrease is performed gradually if only buy/sell-moves are used and abruptly if both move types are used.

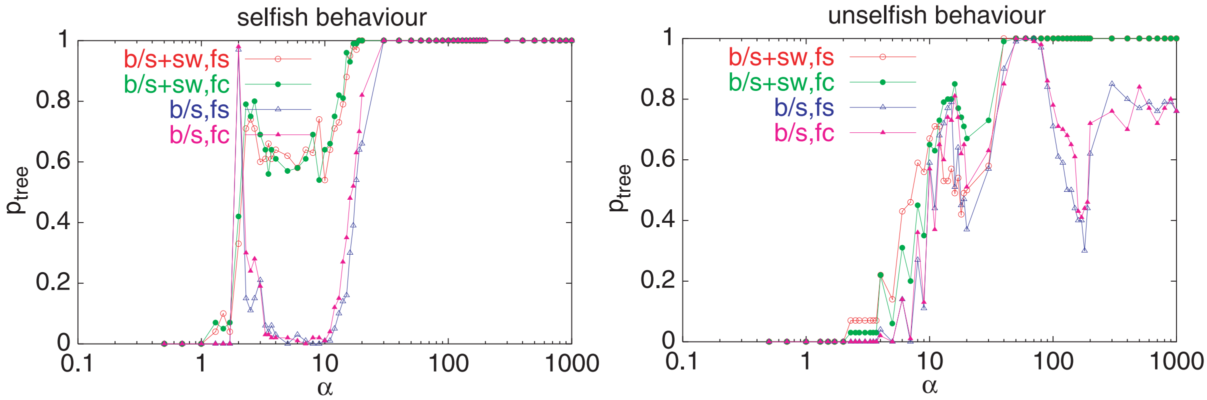

Figure 5. Probability that the resulting configuration is a tree.

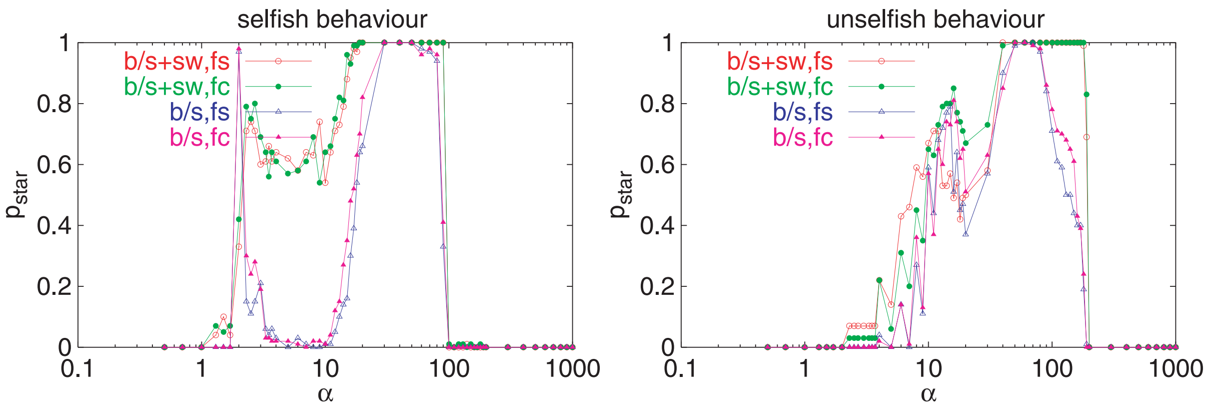

Figure 6. Probability that the resulting configuration is a star.

Summarizing, if a simulation run ends in a tree for $\alpha < N$, then it is always a star, whereas the probability for a star vanishes for $\alpha > N$.

5. Conclusion and outlook

In this work, we have explored the structures generated when multiple agents construct a computer network, within a highly simplified model of the decisions that are made to achieve this. We find that some persistent and rather complex structures emerge in the intermediate regimes of the parameters that characterize the network building process. In addition to the complete graph and the star, the two idealized optimum solutions (the first too expensive to be a realistic solution and the second too fragile to be a reliable solution for a large scale network), we find trees and multi-centred stars resulting from the process of network formation which we simulate. The trees are more easily formed than stars and, while loss of a single node separates the tree into a few parts that can communicate only within each part, there is no central site which controls all communication, as in the star. The multi-centred star and other approximate solutions that result from the simulations offer the germ of a more reliable approach to network formation—no pair of sites is very far apart and each pair can communicate over multiple distinct routes.

Of course, this network creation model could be criticized in various ways for not being close to reality. For example, the agents although trying to keep their costs small have an infinite amount of money. Secondly, the cost for buying a link is simply a constant and does not depend on the distance between two nodes or on the bandwidth. Here it is also unrealistic that a link in only owned by one of the two adjacent nodes and that every node is free to send messages via this link. Third, the bandwidth problem within real networks is not considered at all in this model. Instead, the model emphasizes only the number of hops a message needs to get from the sender to the receiver, which is often only a side-issue in real networks. Finally, the configurations which emerge from this network are quite unrealistic. In particular, the power law distribution in the connection numbers of the nodes in real networks is not reproduced in the model.

Our most interesting result for this model is the comparison between the outcomes of selfish or unselfish behaviours of the agents. From the point of view of optimization, one would prefer to work with unselfish agents as these consider the overall cost function of the problem, which is to be minimized. In contrast, selfish agents only consider some local part of the cost function. However, working with selfish agents saves a lot of computing time: as only the distances from one agent to all other agents have to be evaluated for selfish agents, it takes a computing time $\mathcal{O}(N)$, whereas all distances have to be evaluated for unselfish agents, taking time $\mathcal{O}(N^2)$. Moreover, even comparing the computing time in number of steps or of moves, the simulations with selfish agents often outperform those with unselfish agents. If we define the quality of a network by the average distance in hops between two arbitrary nodes, which this network creation model attempts to optimize, we find that we get equally good configurations for small and intermediate values of α for both kinds of behaviours. Only for large α do simulations with unselfish agents lead to better configurations. Thus, we can summarize by saying that simulations with selfish agents are mostly superior to simulations with unselfish agents in that they lead to equally good results in much shorter computing times.

The model should serve as a starting point for future, more detailed investigations, with more plausible models for the message traffic required, or a more concrete description of the history and hierarchy with which the actual Internet has been formed.

References

[1] Pastor-Satorras R and Vespignani A, 2004 *Evolution and Structure of the Internet* (Cambridge: Cambridge University Press)

[2] Burch H and Cheswick W, *Mapping the Internet*, 1999 *IEEE Comput.* **32** 97

[3] Broido A and Claffy K, *Internet topology: connectivity of IP graphs*, 2001 *SPIE Int. Symp. on Convergence of IT and Communication (Denver, CO, USA, Aug. 2001)*

[4] DIMES@home http://www.netdimes.org

[5] Shavitt Y and Shir E, 2005 *Preprint* cs.NI/0506099

[6] Barabasi A-L and Albert R, *Emergence of scaling in random networks*, 1999 *Science* **206** 509

[7] Koutsoupias E and Papadimitriou C H, *Worst-case equilibria*, 1999 *Lect. Notes Comput. Sci.* **1563** 404

[8] Nash J F, *Equilibrium points in n-person games*, 1950 *Proc. Nat. Acad. Sci.* **36** 48

[9] Fabrikant A, Luthra A, Maneva E, Papadimitriou C H and Shenker S, *On a network creation game*, 2003 *PODC Proc.* p 347

[10] Chun B G, Fonseca R, Stoica I and Kubiatowicz J, *Characterizing selfishly constructed overlay routing networks*, 2004 *Proc. INFOCOM* at press, http://www.ieee-infocom.org/2004/Papers/28_4.PDF

[11] Fabrikant A, Papadimitriou C H and Talwar K, *The complexity of pure Nash equilibria*, 2004 *Proc. STOC* p 604

J. Stat. Mech. (2005) P08007