Stability*of Multi-Agent Systems†

Maria Chli, Philippe De Wilde ‡,
Jan Goossenaerts, Vladimir Abramov, Nick Szirbik §,
Luis Correia, Pedro Mariano, Rita Ribeiro ¶

Abstract – *This work attempts to shed light to the fundamental concepts behind the stability of Multi-Agent Systems. We view the system as a discrete time Markov chain with a potentially unknown transitional probability distribution. The system will be considered to be stable when its state has converged to an equilibrium distribution. Faced with the non-trivial task of establishing the convergence to such a distribution, we propose a hypothesis testing approach according to which we test whether the convergence of a particular system metric has occurred. We describe some artificial multi-agent ecosystems that were developed and we present results based on these systems which confirm that this approach qualitatively agrees with our intuition.*

Keywords: Distributed systems, stability, games, decision making, equilibrium distribution.

1 Introduction

Multi-agent systems is a growing field mainly because of the recent development of the Internet as a means of circulating information, goods and services. Many researchers have contributed valuable work in the area in the recent years. However what is still missing is a clear notion of the stability of multi-agent systems.

The agents of a multi-agent system are computer programs in a distributed environment that execute tasks on behalf of their human owners. These tasks often involve decision-making. Stability, understood intuitively as the property of a system that exhibits bounded behaviour is perhaps the most desired feature in the systems we design. It is important for us to be able to predict the response of a multi-agent system to various environmental conditions prior to its actual deployment. This is why we believe that a clear mathematical definition of the concept will allow us to develop tools and methods for its analysis.

2 Background

Computer scientists often talk [5, 9] about stable or unstable systems without having a verifiable definition of stability. On the other hand, control engineers have a very well established definition [6], which however, is not suitable for multi-agent systems. This is mainly because the agents have to make decisions which is different than changing the values of variables, performing arithmetical operations or differentiation and integration. Similarly the widely accepted notion of stability, which comes from the field of dynamical systems [8, 4], is not appropriate for multi-agent systems either. This is because this definition is restricted for closed systems. Agents, as they act on behalf of humans, are not isolated from the real world. They constantly have to deal with new concepts and changing input. Similarly the definition of stability in the context of population dynamics is not adequate [10]. In this field the research concentrates on grouping agents into types and studying how populations of different types evolve by looking at the size of the population. However, the agents may not always fall into classes, as they are independent individuals.

We propose that considering agents as utility maximising players who take decisions in a game, instead of computer programs, would be a more appropriate approach that concentrates on the actions of the agents rather than implementation issues. In addition we want to account for systems that are as close to reality as possible. In order to do this we have to cater for systems with a varying number of agents. This is the reason we work with *ecosystems of agents*. In an ecosystem, new agents can appear and existing agents can vanish.

In this work we propose a definition of stability for multi-agent systems based on the stationary distribution of a stochastic system. We provide simple example systems to illustrate this.

---

*Funded by IST-1999-10304

‡Chli and De Wilde are with the Electrical and Electronic Engineering Department, Imperial College, London, UK. Email: {m.chli,p.dewilde}@imperial.ac.uk

§Goossenaerts, Abramov and Szirbik are with the Information and Technology Department, Technical University of Eindhoven, Netherlands. Email: {j.b.m.goossenaerts,v.a.abramov,n.b.szirbik}@tm.tue.nl

¶Correia, Mariano and Ribeiro are with the Department of Informatics, New University of Lisbon, Portugal. Email:{luis.correia, pedro.mariano, rita.ribeiro}@di.fct.unl.pt

3 Markov Chains Primer

This section presents a brief and by no means exhaustive introduction to the theory of Markov Chains which underlies the approach taken in this work to defining stability. More comprehensive introductions to Markov Chain theory and stochastic processes in general are available in [7] and [1].

Definition: We say that $(X_n)_{n\geq 0}$ is a *Markov chain* with *initial distribution* $\lambda = (\lambda_i : i \in I)$ and *transition matrix* $P = (p_{ij} : i, j \in I)$ if:

1. $Pr(X_0 = i_0) = \lambda_{i_0}$
2. $Pr(X_{n+1} = i_{n+1} \mid X_0 = i_0 \ldots X_n = i_n) = p_{i_n i_{n+1}}$

The crucial point is that a Markov process is memoryless, which means that the current state of the system is the only state that is required to describe its subsequent behaviour. We say that a Markov process $X_1, X_2, \cdots, X_m$ has a stationary distribution if the probability distribution of X_m becomes independent of time (m).

We view a multi-agent system as a countable set of states I with implicitly defined transitions between them. At time n the state of the system is the random variable X_n.

3.1 Definition of stability

A system considered under the terms described above is said to be stable when the distribution of the state of the system converges into an equilibrium distribution. In other words when $Pr(X_n = j) \to \pi_j$ as $n \to \infty$ $\forall j \in I$. Our definition can also be stated by: *A stochastic process $x_1, x_2, x_3, x_4, \ldots$ is **stable** if, the probability distribution of x_m becomes independent of the time index m for large m.*

3.2 Dealing with a varying number of agents

We want the definition of stability we propose to apply to systems with a varying number of agents (ecosystems). The system state will be represented by an infinite vector $\mathbf{X}$ that has one or more elements for each agent and a number of elements to describe general properties of the system state that are not particular to any single agent. We model an agent being 'dead' by setting the vector elements for that agent to some predefined value (e.g. -1).

4 Example games

A couple of example-'toy' games we dealt with are given below. We will explain how our definition of stability applies to them and then investigate how we deal with more complicated ones.

4.1 The 3-player +/- game

The three players of this game, can each be in one of two states '+' or '-'. If two players are in the same state the third goes into that state. If all three players are in the same state, there is probability 0.9 that they all remain in that state and probability 0.1 that they all change state. This game has eight states: (1) —, (2) –+, (3)-+-, (4) +–, (5) +-+, (6) -++, (7) ++-, (8) +++.

From the rules of the game we devise the state transition probabilities. For example the probability of going to state 8 (+++) from state 1 (—) is 0.1, the probability of remaining in state 8 is 0.9, the probability of going from state 8 to state 7 (++-) is 1, or the one of going to state 4 (+–) from state 6 (-++) is 0 and so on. Thus we construct an 8x8 state transition matrix. It is shown in Table 1.

State Id	1	2	3	4	5	6	7	8
1	0.9	0	0	0	0	0	0	0.1
2	1	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	1
8	0.1	0	0	0	0	0	0	0.9

Table 1: The transition matrix for a 3-player +/- game

We often refer to the transition matrix as $\mathcal{P}$. Once we have this we can check whether an equilibrium distribution exists or not by solving the following simultaneous equations for π:

$$\begin{aligned} \pi\mathcal{P} &= \pi \\ \sum \pi_i &= 1 \end{aligned}$$

If *one or more solutions* exist then the system is *stable*. In the case where only one solution exists, this solution will be the equilibrium distribution into which the system will converge. In the case where there exists more than one solution, this means that the system can converge into one of many stable solutions/distributions of states. It is uncertain in which one the system will fall, but as soon as it does it then remains into that one forever and this why we consider it stable.

In this example we saw how we can devise state transition probabilities from the rules of a game, which we formulated from a multi-agent system scenario. These transition probabilities allowed us to predict whether the system will eventually reach an equilibrium distribution or not, in other words whether the system is stable or not.

4.2 The simple random walk

An example of an unstable system is the simple random walk, where we start in state 0 and at each timestep, if we are in state K, we go either to state $K+1$ or to state $K-1$ with equal probabilities. It can be shown that the distribution of the state X_m, where m is the time, is centred on zero but its variance is proportional to $\sqrt{m}$ which means that the p.d.f. is not independent of m.

4.3 A coin game

In another slightly more complicated game, N players toss a coin in pairs. The loser gives to the winner a unit of its wealth. Players are destroyed if their wealth is below 1 and can generate another player if their wealth is more than 4. The sum of all players' wealth should be 6.

If we try out this game with three players we realise that its state space consists of 19 states. Solving the system of simultaneous equations shown above we conclude that it has an equilibrium distribution. It is remarkable however; the fact that such a simple game, with so few players has so many states. The number of states rises tremendously as we increase the number of players and adjust the wealth in the system accordingly. For example, if we try the same game with four players instead of three we will be presented with a state space of around 70 states.

At least theoretically one could evaluate the transition probabilities of the multi-agent system and through a simple eigenvalue analysis find its equilibrium distribution, if it exists. In practice due to the state space explosion problem in real multi-agent systems, this approach is not viable. We therefore propose statistical analysis as a means of testing for the probabilistic behaviour of more complex multi-agent systems.

Multi-agent systems are often used to analyse problems such as trading in a stock exchange or transportation problems. This type of situations can easily be formulated as games and their stability can be analysed in the way we propose. In the next section we show how we formulated a trading scenario and a loads transportation scenario as games and comprehensively analysed their stability. We also present the results we obtained from extensive simulations and statistical analysis.

5 Experiments

We worked on various different models we developed in Java, in order to test and experiment with our definition of stability.

As these models are somewhat more complicated than the simple games we saw in the previous section we do not go about finding out the state space of each one of them. Their state spaces are probably quite large. What we do instead is to check that several indicative metrics do reach a stationary distribution after the system has been left to run for a while.

5.1 Trading Simulation Model

5.1.1 Scenario

Initial setting There is an array of N different resources in this model. There are M traders each with its capital, its discounting percentage and its available resources. All traders are endowed with the same amount of wealth; the amount of each resource given to each trader is randomly calculated. A discounting percentage is used when the trader recalculates its prices. Finally, the scenario involves tasks that are generated by agents. A task requires some resources and produces some others.

At each clock tick every trader with its turn issues a task and advertises it amongst the other traders. Each task carries a random combination of required and produced resources. Every trader gives an offer for the task (provided that they possess the required resources). The cheapest offer is selected. If the issuer cannot pay for any offer then the task is not executed. Otherwise, it selects an offer and the task is executed. The required resources are subtracted from the task executor's set of resources, the produced resources are added to the issuer's set of resources and the issuer pays to the executor an amount of money equal to the price for executing the task. Finally each trader recalculates its prices according to its discounting percentage and whether its offer was accepted or not.

Generation and Destruction of agents When a trader is sufficiently rich, i.e. its wealth exceeds a certain threshold; it generates a new trader to which it gives half its wealth. Also, the parent trader endows the child trader with half of its resources. The new trader inherits its generator's discounting factor. When a trader's wealth goes below zero then it is destroyed.

5.1.2 Stability Conditions

In order to consider the system stable, its state must reach a stationary distribution when it has been left to run for a while. Strong indications of stability would be that metrics such as the proportion of traders that execute tasks, the number of traders, the prices of he resources, the wealth per trader would all reach stationary distributions when the system has been left to run for a while.

5.1.3 Experiments

In order to explore the entire parameter space we performed an extensive and exhaustive set of experiments, for as many combinations of initial parameter values and perturbation sizes as possible. The parameter values we varied for this model were the discounting factor's, prices' and resources' order of magnitude and the initial, generation and destruction wealth for the traders.

An unstable system We performed an experiment with 50 traders, 10 different types of resources. The simulation was left running for 10^4 time ticks. Each trader is endowed with 10^6 monetary units and a random amount of each of the 10 resources. The amount from each resource it gets is of the order of 10^3 (calculated randomly). The resources' prices are initially of the order of 100 monetary units (calculated randomly). The discounting factor's initial order of magnitude is 10^{-3}. A trader can generate a new trader if its wealth exceeds 1.5×10^6 and it dies if its wealth goes below 0.

In Fig.1 we show how the ratio Tasks per Trader varies throughout time. However, the graph is not enough to demonstrate stability of the system; this is why we used statistical analysis.

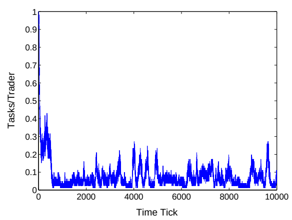

Figure 1: TRADE_UNSTABLE: Tasks per Trader

Our aim was to show that after the system has been left to run for some time, the metrics mentioned above reach a stationary distribution. We utilised statistical hypothesis testing which indicates whether two samples come from the same distribution to establish if this was the case. More specifically we used t-test for the equality of the means and F-test for the equality of the variances. For in-depth information on these tests one can refer to any statistics textbook such as [3, 2].

The tests showed that when the model is run with these initial conditions results in an unstable system.

A stable system If we re-run the above experiment with the same initial parameters, only changing the discounting factor's initial order of magnitude from 10^{-3} to 10^{-13} and the prices' initial order of magnitude from 10^3 to 10^4 the system's behaviour is significantly different.

We can see from Fig. 2 that the system looks much more stable than it did before. We performed the hypothesis tests for each metric as described above. We saw that for all of them there was significant evidence in the 5% significance level the metrics reached a stationary distribution. Therefore the system is *stable* when run with those specific initial conditions.

Perturbations It would be interesting to find out how the system responds when we disturb its normal running by introducing a perturbation.

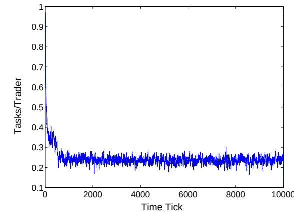

Figure 2: TRADE_STABLE: Tasks per Trader

We started an experiment with the same initial conditions as the first one we described (which was unstable). However, this time we injected a shock into the system. At time tick 4000, the prices of all the resources of each trader are increased arbitrarily by 500%. We then allowed the simulation to run until time tick 10^6 and observed the results. Fig. 3 shows how the ratio of tasks per trader varies for this experiment.

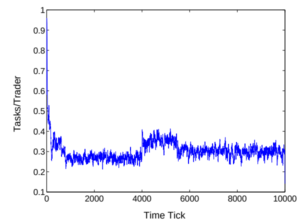

Figure 3: TRADE_PERT_1: Tasks per Trader (before and after the shock)

The tasks per trader ratio seems to follow a much more stable trend after recovering from the shock at time tick 4000. When we perform statistical testing we notice that at the 5% significance level there is significant evidence that the values for the metric tasks executed per trader after the shock come from a common distribution. In general, the system after the shock is stable and the effect of the shock is noticeable.

In another experiment the system was started with the same initial parameters as the stable system shown above.

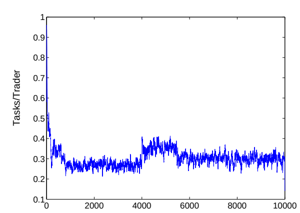

Figure 4: TRADE_PERT_2: Tasks per Trader (before and after the shock)

At time tick 4000 the following shock was injected into the system. The wealth of each trader was increased by 60%. Fig. 4 shows that the effect of the perturbation was quite significant. We showed, once more using hypothesis testing at the 5% significance level, that there is evidence that after the recovery from the perturbation the values of this metric belong to a common distribution, i.e. the metric is stable after it recovers from the shock. In general the system after the substantial effect of the perturbation has more or less elapsed goes into a stable phase.

5.2 Loads' Transportation Model

5.2.1 Scenario

Initial setting We define a number N of cities and a map of interconnections between these cities. Some cities are directly connected to each other while some others are not. The distance between two cities that are directly connected takes one day to travel. For example, if the cities A and B are connected via city C (i.e.: A $\rightarrow$ C $\rightarrow$ B), it takes two days to travel from A to B. The simulation operates with a granularity of one day. A number of M lorries travel around the map each of them being at a city at any particular day. 0 up to M lorries can be at city at a time. Loads are generated at a certain rate at each city and have a specific city as a destination e.g. a load can be generated at city A and be destined for city B. Each lorry can carry up to K loads, from which it earns rewards. Rewards are added to the lorry's wealth.

Load Allocation and Distribution The loads located at a city are allocated to the lorries present at that city randomly. A lorry has an algorithm which decides the route it will follow based on information such as the city it currently is in and the destinations of the loads it carries. Every D number of days the lorries are required to reduce their wealth by a certain amount. This is a form of taxation.

Generation and Destruction of Agents When a lorry's wealth exceeds a certain threshold, it generates a new lorry at the city it currently is and endows it with half its wealth. When a lorry's wealth is below a certain threshold then it is destroyed.

5.2.2 Stability Conditions

According to our definition of stability the system will be considered stable if its metrics show evidence that they reach a distribution after some time the system has been running and remain in this distribution forever. The proportion of lorries that carry loads, the proportion of loads that have been carried, the loads carried per lorry, the wealth per lorry, the number of lorries are the metrics we observe to decide on whether the system is stable or not.

5.2.3 Experiments

An unstable system We performed an experiment with 100 lorries and a network of 500 cities connected to each other with a connectivity percentage of 40% between the cities of the network. The simulation was left running for 20 000 time ticks. Each lorry is endowed with 10 monetary units. The maximum capacity of a lorry is 3 loads. A lorry can generate a new agent if its wealth exceeds 50 and it is destroyed if its wealth goes below 0. Tax equal to 2 monetary units is deducted every 5 time ticks. A new load is generated in each city every 3 time ticks. The maximum number of loads that can be waiting in a city at any time is 5.

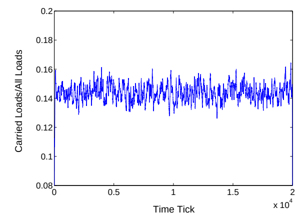

Figure 5: LOAD_UNSTABLE: Carried Loads Ratio

The graph in Fig. 5 shows how the carried loads ratio varies with time. Hypothesis testing at the 5% significance level showed that there is significant evidence that the system is unstable when run under those initial conditions.

A stable system In an attempt to make the system more stable we run it again, performing another experiment. We keep the initial conditions mostly the same as they were for the previous experiment. However, this time we change the intercity connectivity percentage from 40% to 90%. Fig. 6 shows the behaviour of the carried loads ratio for this experiment.

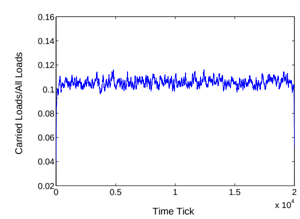

Figure 6: LOAD_STABLE: Carried Loads Ratio

The graph depicts a fairly stable system. This is also the evidence we get when we perform statistical testing on the metrics ratio of working lorries, carried loads, loads per lorry, wealth per lorry.

Perturbations In an attempt to challenge the behaviour of the system when something unexpected happens we introduce perturbations. It is interesting to see what their effect is, if any, and whether it is justifiable.

During this experiment we inject a shock into the system. The initial conditions are the same as those for the previous experiment where the system was stable. This time, at time tick 4000, we increase the minimum wealth a lorry has to have in order to be able to generate a new lorry from 50 to 5×10^7 monetary units, keeping all other parameters the same.

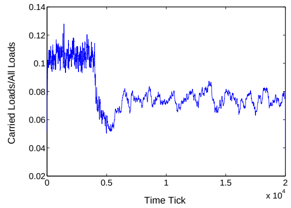

Figure 7: LOAD_PERT_1: Carried Loads Ratio (before and after the shock)

This perturbation will naturally cause the frequency of new lorry generation to decrease. Due to the decrease in the number of lorries, there are more jobs for the remaining lorries to carry out; therefore the working lorries ratio increases dramatically, as is the wealth per lorry and the loads per lorry ratio. However, the remaining lorries do not seem to be enough to cope with the number of loads that are generated in the system therefore the carried loads ratio drops after the shock occurs.

We know from the previous experiment that the system is stable under those specific initial conditions. It is evident from the graph in Fig. 7 that the system is driven to instability after the shock is injected. This is a case where a perturbation causes a stable system to become unstable. In another experiment we performed we saw that it is possible the opposite can happen, i.e. an unstable system can become stable after it has undergone a perturbation.

This simulation was started with the same initial parameters as the experiment for the unstable system we showed above. We injected a shock at time tick 4000, which was to decrease the maximum number of loads in each city by 80%.

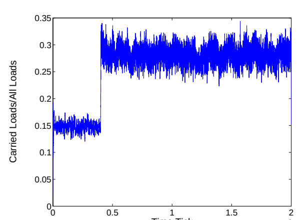

Figure 8: LOAD_PERT_2: Carried Loads Ratio (before and after the shock)

As there are now less loads in the network of cities, there are less jobs for the lorries, hence the population decreases. Consequently, the remaining lorries have greater workload. The working lorries ratio increases. The system seems to work much more efficiently after the injection of the shock, as the carried loads ratio is boosted from 15% to 30%. In addition to all these, hypothesis testing confirms that after the system has recovered from the shock the metrics we examine fall in stationary distributions. Therefore, we can deduce that there is significant evidence that after a small transient phase the system is stable.

6 Conclusion

This has been a study of the concept of stability of ecosystems. We began by evaluating definitions of stability that already existed in well-established fields of mathematics. We then explained the reasons these definitions are not suitable for application in the context of multi-agent systems and ecosystems. Subsequently, we proposed a definition of stability which is the only one which takes into account the game nature of multi-agent systems, is relevant to systems with a varying number of agents and is supported by the mathematical framework of stochastic systems.

The study included the design and development of experimental multi-agent platforms which we analysed, with the intention of illustrating the validity of our definition. Moreover, we introduced statistical hypothesis testing as a means of applying our definition analytically to quantify the stability of complex multi-agent systems.

References

[1] D. R. Cox and H. D. Miller. *The Theory of Stochastic Processes.* Chapman and Hall, 1972.

[2] J. Crawshaw and J. Chambers. *A Concise Course in Advanced Level Statistics.* Stanley Thornes Pub Ltd, 2001.

[3] R. A Fisher. *Statistical methods for research workers, 14th ed.* Oliver and Boyd, 1970.

[4] N. Glance, Hogg T., and B. A. Huberman. Computational ecosystems in a changing environment. *Intl. J. of Modern Physics*, C(2):735–753, 1991.

[5] Balakrishnan Hari and Srinivasan Seshan. Analyzing stability in wide-area network performance. *ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, 1997.

[6] Hassan K. Khalil. *Nonlinear Systems.* Prentice Hall; 2 edition, 1995.

[7] J. R. Norris. *Markov Chains.* Cambridge University Press, 1997.

[8] Lawrence Perko. *Differential Equations and Dynamical Systems.* Springer Verlag, 2001.

[9] J. D. Thomas and K. Sycara. Heterogeneity, stability, and efficiency in distributed systems. *Proceedings of the 1998 International Conference on Multi Agent Systems*, 1998.

[10] J. W. Weibull. *Evolutionary Game Theory.* MIT Press, 1995.