Dynamic design of experiment with MAS to approximate the behavior of complex systems

François Klein, Christine Bourjot, Vincent Chevrier

Dynamical design of experiment with MAS to approximate the behavior of complex systems

Francois Klein and Christine Bourjot and Vincent Chevrier

LORIA
BP 256
54506 Vandoeuvre les Nancy Cedex, France
email: surname.name@loria.fr

Abstract. This paper is about the understanding and control of complex systems modeled by a multi agent system. We propose an approach to experimentally approximate the behavior of the system as a function of its parameters. The originality of our proposition is the ability to dynamically generate data to train the model and therefore reducing the number of executions needed to build the approximation while maximizing its accuracy. We detail our proposition and provide an example of how to put it into practice. Some experiments are presented and the obtained results suggest its viability.

1 Introduction

The context of this work is the study and the analysis of complex system (CS), i.e. systems being able to produce a collective response from interactions between simple individuals. What is interesting is the ability of such systems to have a collective response such as their complexity is out of the individual simplicity. Therefore the complexity is not in the individuals but in the interactions between them: the properties of the CS can be qualified of emergent.

These systems are characterized mainly by non linear dynamics (small fluctuations when near some critical point can provide significant modification of the system), by sensibility to initial conditions and to parameters (small changes on a parameter produce different patterns). Thus the overall properties cannot be understood simply by examining separately the components.

For example, the following (and well known) equation models the dynamic of a population (which varies from 0 to 1) according to time: $x_{n+1} = Rx_n(1 - x_n)$ where R is the reproductive rate and is a parameter. In this example, a bifurcation occurs at several values of R. The population size to which the system converges is an attractor. For a given value of R, several attractors can be exhibited, leading to an oscillation between them.

As in many situations, the understanding and control of a complex phenomenon cannot easily be envisaged on itself but is undertaken on a model. Multi-agent models [1] provide concepts to describe such systems as a set of agents that act in a shared environment and which interact directly or via their environment. Therefore, they are good candidates [3] to model complex systems because of their explanatory abilities and because of the opportunity to observe the consequence of local changes on the global phenomenon. Once the multi-agent model is built (and assessed as valid), it can be used to explain the mechanisms that underlie the dynamics of the complex system (examples can be found in the modeling of self-organized societies in biology [2] or [4]), and to predict the behavior of the system when initial conditions are changed or when some perturbation occurs.

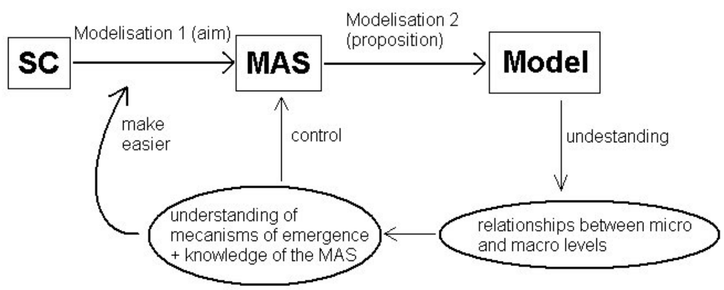

Fig. 1. Steps in CS modeling in order to understand and control them

This paper focuses on the aim of understand and analyze MAS's behavior, to make easier the modeling of a CS by a MAS, and thus to help the understanding and the control of a CS. This is the first step of a long-term study, so the paper does not introduce a modeling method for any CS (cf fig. 1). We propose to experimentally study a MAS to model some of its global behaviors from its parameters. In the next section, we define the problem. Section 3 describes our proposition and section 4 provides an example of use. In section 5 we discuss advantages and drawbacks of such an approach and propose some future works.

2 Position of the problem

From now, we put the stress on the MAS to take advantage of the opportunity of experimentation they offer : the possibility to try experiments at will, the easiness to set any value for a parameter, the capacity of automation, etc. The MAS behavior is defined as the set of parameters related to the agents' behavior and to the environment dynamics. Provided some initial conditions, the system will produce a collective property (or pattern). Our issue is then to establish a relationship between these parameters (and initial conditions) and the characteristics of the global behavior. This issue can be broken down into three parts:

1. understanding: knowing the global behavior of the system on a sub space of the parameters space
2. controlling: determining a set of values for the parameters in order to obtain a desired global behavior
3. optimizing: finding a set of values for the parameters that provides the optimal desired global behavior according to a criterion of that behavior.

According to the properties of complex systems, answering such questions ***a priori*** is a hard task. Two main trends can be used to deal with this issue[5, 6]:

1. to use a formal framework in which the MAS can be expressed and to study the behavior in that framework, or
2. to study experimentally the MAS by doing several executions of it with various parameter settings and to approximate the behavior from these data (one execution of the system with fixed parameters is called a *replication*).

Several authors[7–11] claim that the study of MAS needs experimental approaches and make different proposals. For example, [12] explicitly refers to the experimental process: "'*the development of any agent system, however trivial, is essentially a process of experimentation*"'. Indeed, we need an experimental approach to cover the inputs space, where the MAS's behavior will be very irregular and hard to model with a formal approach. Unfortunately, the same observation triggers a huge number of necessary replications, that will soar with the number of parameters or with the accuracy of the exploration of the inputs space. The question that comes up is : "Is it possible and how to model the behavior of a MAS in following an experimental process, efficient enough to potentially be put into effect ?" Of course, the aim of the model is to help us to draw up the relationships between the parameters and the global behavior of the MAS. The next section describes such an experimental process.

3 Proposition

3.1 Generalities

In the remaining of the paper, we consider only the understanding and control of the multi-agent system by using an experimental approach. Different works deal with the same issue by characterizing the global function of the system [13] or [7]; or by providing some key steps to follow while studying the system [14]. The key characteristic of our proposition is the reduction of replications by the definition of dynamical designs of experiments.

From our perspective, the behavior of a system is defined by:

1. a set of parameters (or inputs) that corresponds to the values the designer can set in the system before executing it.
2. a set of outputs that corresponds to values that can be observed on the system. As the systems we are studying are dynamic, such values can be observed over the time, each output being a sequence of values.

Executions of the model are expected to *reproduce*[1] the main features of the multi-agent system behavior. The behavior can be viewed as a function that links inputs and outputs. That means, for a given set of input values, the execution of the model enables the computing of the output values.

In this context, we can define more precisely what is understanding and controlling a system.

Understanding means to answer the following: provided a set of input values, what will be the system response (the values of output)?

Controlling is the complementary question: we fix a desired output and we wish to know a value (or a set of values) for the input such as to obtain the output.

In both cases, the problem is to build an approximation of the behavior of the system. In the remaining of the article we hypothesize that such a behavior can be approximated as a function linking inputs to outputs.

The difficulty is now to make a balance between accuracy of the model (the quality by which it answers the previous questions) and the complexity of the model itself in size or time to build it.

We propose to do that in a four-steps approach:

1. restrict the search spaces to the "‘interesting areas"’ by considering a subspace of the possible input and output according to the purpose of the study and to the a priori knowledge we can acquire on the system;
2. select a relevant model according to the knowledge we got on the system and to the purpose of the study;
3. train the model from experimental data;
4. validate it.

3.2 Simplification: restriction of domains

Even if it corresponds to a complete knowledge about the system, building a model on the whole space; that is considering the entire input and output; suffers of several drawbacks:

1. practically, the size of the space can be considerable. It can increase significantly if we consider the time and/or the system stochasticity for the outputs. The time to run the replications of the system makes this approach unrealistic;
2. in some areas of the inputs space, the behavior of the system is without any interest (because it will never correspond to an actual case, or because the configuration of parameters is obviously a non-sense, or because the system is already known as being chaotic; etc)

[1] As it is a model, a peculiar point of view is chosen and the term "‘reproduce"’ has to be taken from that perspective.

3. among all the inputs and outputs, only some of them can be of interest according to the aim of the study: some inputs can have very few influence on the behavior; among outputs, some are not relevant per se or have to be pre-processed to become of interest (for example, in the population dynamics system, the positions of agents are of few interest to understand it and it would be preferable to compute their dispersion).
4. the behavior of the system can qualitatively vary for different range of parameters values and it can be worth considering separately these behaviors by several approximations.

That's why we can remove some elements among all the inputs and outputs of the studied MAS, we can restrict some other and have to replace some output values by some new ones corresponding to a processing (aggregation) of the previous ones in order to synthesize information. We call *indicators* the considered output values.

This first step aims at replacing all the input parameters by some pertinent ones (control ones, with respect to the aim of the study); and all the outputs values by relevant indicators. This step will limit the size of the inputs and outputs spaces. We expect it to make easier the approximation of the system.

3.3 Model selection

Now we are interested in establishing a relationship between some parameters governing local behavior and some indicators of the global phenomenon.

As this relationship can be chosen between several different models, we have to select one by putting into balance its accuracy of representation, its simplicity, its easiness of use, its compacity, the time it needs to be built, its expressiveness, etc.

An example of representation can be parametric expressions. The behavior is viewed as a function $F(p) = i$ (where p and i are respectively the parameters and the indicators) equivalent to an equation governed by two parameters (A, B) such as $F(p) = i = A.p + B$. It can also be a statistic estimator such as a neural network [15](with maybe some parameters to set up for efficiency purpose).

Choosing a representation means determining the family of the model (linear equation, perceptron with n-layers, ...) according to preliminary studies of the system: it is not relevant to suppose the system to behave linearly if it can be shown with few replications that it is not.

The choice of the model is also depending on its goal: if we are aiming at controlling the system, the model has to be inversible (or at least on some part of its range).

3.4 Training of the model: dynamical design of experiments

Once a representation is chosen, it has to be instantiated with the system to study. As mentioned above, we have to balance the accuracy of the model with the time needed to build/train it. In other words, we wish to minimize the number of replications needed to build the model while maximizing the accuracy of its forecast. This problem is known as design of experiments [16]. A design of experiments (DOE) is a list of experiments to be undertaken according to their specific value of parameters.

However, most of the DOE are applied to create simple models, typically linear ones, and we are considering complex systems, thus we need a specific kind of DOE.

We propose to build dynamically the DOE; that is, determining the value of parameters of the next replications according to the results already obtained. The interest of dynamical DOE is to make homogeneous the accuracy of the model by choosing "pertinent" parameters setting, and as a result, either to reduce the number of replications needed to train the model, or to raise the accuracy for a given number of replications. The execution of the DOE ends when the desired accuracy is reached, or when a limit (on the number of replications for example) is reached.

Thus in a given area of the parameters space E, either the model is built and we know that its accuracy is reached, either the model has no answer and we know this is because some limit has been stated.

The way the model's accuracy is made homogeneous on E hardly depends on the model chosen. To be applicable, this supposes that the model is not sensible to the order of the data used to train it; and that the accuracy can be assessed during runtime. The assessment can be exact (e.g. the mean error when doing a linear regression) or can be estimated through heuristics.

Therefore, the principle is to generate dynamically DOE by iteratively assessing the accuracy and if not sufficient, generating new settings for replication.

3.5 Validation

Once the model trained, it has to be validated, that is to be assessed as an efficient approximation of the system. It is done by comparing the prediction of the model with the response of the system on random values of inputs.

4 Instantiation of the proposition

As our proposition is based on general principles, we illustrate in this section its use on an example. It has to be mentioned that we don't aim to obtain a model that fit well a MAS but we rather wish to emphasis how we put into practice our principles. However, the model used and the algorithm to dynamically design experiments are generic enough to be reused for other systems.

4.1 Toy example: population dynamics

The problem we chose is that of the population dynamics. It consists in a population of agents that live in a shared environment. Environment is a toroid grid where each cell contains a given amount of food which increases linearly. Agents move randomly in the environment in which they can get food to increase their energy level. If its energy level is too low, the agent dies; if it exceeds a given threshold, agent gives birth to a new agent.

We implemented this problem by a cyclic simulator in C++.

We are interested by the behavior of the population: in which conditions will the population extinct, will it stabilize after a number of cycles, will it evolve between some limits, etc.

This problem is well know and has been implemented in many simulation tools (such as netlogo or starlogo[2]) and it can be described as the following equation: $x_{n+1} = Rx_n(1 - x_n)$ which informs of the behavior of the system.

Value of R	Behavior of the system
$0 < r < 1$	Extinction
$1 < r <= 3$	Growth to a constant size
$3 < r < 3.4$	Oscillation between two values
$3.4 < r < 3.57$	Oscillation between four values
$r > 3.57 = r_0$	Oscillation between more values and chaos

Even if such an equation informs on the global behavior, it cannot answer questions related to the individuals parameters: what happens if birth threshold is above α? What is the minimum value for the growth of the food to guarantee a stable population?, etc.

4.2 Steps of the approach

Domain and range restrictions The multi-agent system is described by many parameters: the initial number of agents, their initial level of energy, *the food growth rate* t_x, *the maximum amount of food to be eaten by cycle* Q_t, *the energy loss by cycle* D_p, *the birth threshold* S_e, the size of the environment, the maximum amount of food on a cell.

The values that can be measured during execution of the system are, at local level, every features of the agents and of the cells, and at global level, any value that can be computed on a set of elements from one replication (such as the population size, the minimum, maximum, or average value of a feature); or some statistics computed from those previous values after several replications.

The choice of inputs (parameters) and outputs (indicators) to study is tied to what we wish to know about the population behavior.

The equation informs that the initial population (and experimentally stated not too close to zero) has no influence on the global behavior. We decided to fix as constant the size of the environment and the maximum amount of food on a cell. Some other values (agent's initial orientation for example) have no influence on the behavior and are randomly set up for each replication.

Finally, the set of parameters is made of

- *the food growth rate* $t_x \in [0.01, 1]$,

[2] http://education.mit.edu/starlogo/, http://ccl.northwestern.edu/netlogo/

– *the maximum amount of food to be eaten by cycle* $Q_t \in [2, 10]$,
– *the energy loss by cycle* $D_p \in [1, 9]$,
– *the birth threshold* $S_e \in [10, 100]$.

As indicators, we choose to run several replications, to stop them after a fixed number of cycles and to consider the mean of the population, its standard deviation and the number of replications that converge to zero.

Model choice As we wish to assess if a dynamical design of experiments can effectively reduce the number of replications, we decide to instantiate a general purpose model for each indicator, linking it to the chosen local parameters.

Let us call N the number of parameters taken into consideration (in our case, $N = 4$), N is also the dimension of the input space. We call *mesh* a subspace of the input space included between 2 fixed values for each parameter, so a mesh is marked off by 2^N points.

Our model supposes that the input space is composed of a finite number of strictly distinct meshes, and that the MAS behaves such as to be linearly interpolated in each mesh.

As the domains of both inputs and outputs are numerical, we decide of an arbitrary threshold δ associated to the system, that represents the relative error on the linearity accepted in a mesh. Considering a mesh, the average slope avg for a parameter can be computed with 2^{N-1} pairs of points, and δ is defined such as, for each parameter, the average difference between the slope defined by 2 points and avg is under δ, relatively to the average output value in the mesh (see figure 2 for $N = 2$).

If we succeed in building a minimal set of points, defining the meshes of the model, such as the above property holds, the model combines the qualities of simplicity, compacity, and accuracy seeing that only the areas where the behavior of the indicator is hard to model are filled with small meshes.

This model has been chosen for two other reasons : firstly, it allows the control of the MAS, thanks to its inversibility. Indeed, it is easy to find, if it exists, the value of a parameter that triggers a searched value of the indicator.

Secondly, the model is in accordance with the studied MAS, because we use only 4 parameters, thus 2^4 points per mesh, that is sufficient to compute a relevant average of the slope, and not too many to compute it quickly. Another sign that shows the adaptation of the model to the MAS is that, according to observation, the monotonous variation of a parameter triggers a monotonous variation of the indicators, so there is no oscillation, and a set of linear approximations fits such a system.

Dynamical design of experiments We present here the algorithm to train the model, we consider only one output indicator for clarity purpose, but the principle is the same for several.

Initially, S corresponds to the the summits of E and their values for the indicator. S is made from one sole mesh that is not considered as explored. The principle of the algorithm is to verify if the property holds on every non explored mesh of E and if not to divide the mesh into smaller ones and recursively consider them. As we cannot ensure the size of mesh on which the hypothesis can hold, we fix a limit to the size of a mesh. Therefore when considering the mesh, if it is under the limit size, it is marked as unexploitable.

The algorithm stops and constructs a model whose complexity in size is upper bounded. At the end, a mesh is either explored, either unexploitable.

The algorithm is as follow:

- For each non evaluated mesh $m \in S$
  - if m under limit size, mark it as unexploitable
  - otherwise,
  - For each dimension of E,
    - compute the average avg of the slope of F
    - compute the average difference σ between avg and the slope of F along an edge of the mesh.
    - if $\sigma < \delta$, the mesh is considered as explored
    - otherwise divide the mesh in smaller ones and add them into E as non explored

To divide a mesh in smaller ones, we consider the middle of each segment between two summits in mesh.

In our case we decide to consider at most 16 segments on one dimension. As we have $N = 4$ dimensions, the number of points to consider in order to build S is at most 17^4, This is the number of points we should consider, if we used a non-dynamical DOE, to obtain a similar model accuracy.

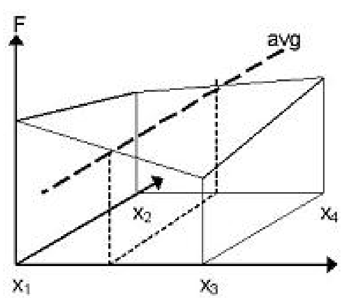

Fig. 2. Approximation of F over a mesh: over the $\overrightarrow{x_1 x_2}$ dimension, we compute $F(x_1) - F(x_2)$ and $F(x_3) - F(x_4)$ and compare the average with δ.

Validation Validation of the model is only relevant for explored mesh and can be done by statistically comparing the approximate values furnished by the model and the ones furnished by the system execution.

4.3 Results and comments

We undertook different experiments with three outputs and different values for δ. For each response of the system we did ten replications in order to assess the stochasticity.

We consider the results according two points of view:

1. one about the approach: can we reduce the number of replications needed to approximate the system ?
2. one about the model itself: are the predictions of the model reliable ?

According to the indicator used, we can observe a limitation of the number of data from 30% up to 75%.

About the quality of the model, since the training stops whether the size of a mesh is minimal (and results are not exploitable) or the mesh is explored (and can be used to predict the system response), we have to consider only the response of the system in the explored part. Preliminary tests suggest an average errors of 10%.

But we found out that some results were redundant on the three output indicators. For instance, as we compute the proportion of replications that converge to the death of the population, in each point of the inputs space, if all of the replications lead to zero, we can deduce that the mean and the standard deviation of the population converge to zero too. This calls into question our preliminary study to choose the outputs, and we wish to emphasize the weight of that part of the method exposed in the paper for the further applications.

As the main objective with those experiments was to assess the feasibility of dynamical DOE, the results we obtained encourage us to continue: we can significantly reduce the number of replications to understand a system.

However, we made raw choices: the mesh division is systematic and not contextual, we can imagine to built mesh by selecting points using a criteria that both optimize the relevance of data used to train the model and the size of the mesh and therefore increasing the number of explored ones; the threshold used to define the model is arbitrary and can vary over E, it would be of interest to select it according to the data we obtained.

That's why we have to confirm our preliminary results.

5 Conclusion and further works

We consider in this paper the understanding and control of a complex system modeled by a multi-agent one. We propose an approach to experimentally approximate the behavior of the system as a function of its parameters. Of course, as a Mas can be considered as a complex system itself, our proposition can be used for the design of such systems.

The originality of our proposition is the ability to dynamically generate data to train the model and therefore reducing the number of replications needed while maximizing the accuracy of the approximation; and its anytime property:

the system can be stopped at any step and is able to produce a solution of which quality is dependent of the number of steps. The feasibility of the approach has been assessed through some experiments on a population dynamics phenomenon.

We highlighted the necessity of carefully study the system that must be modelled, to choose relevant indicators of the global behavior, and an adapted family of models that fits this behavior. Although our implementation is using a particular design of experiments, since only some points of the inputs space can be computed, it is generic enough to be reused with other phenomenon.

We succeeded in answering the question of the possibility to model a MAS behavior by an experimental process, in showing a method that reduces the number of replications with an accuracy refinable at will. Thus, we are able to understand and control, in a first time, complex systems modeled by a MAS, and at long term, any CS in modelling it by a known MAS.

Future works will deal with time depending indicators, to allow us to study and control the perturbations of a CS (and of a MAS), instead of the simple consequences of initial conditions. They will also continue the experiments to confront this approach with other phenomena, and with different models and dynamic DOE to compare their strengths and drawbacks.

References

1. G. Weiss, *Multiagent systems a modern approach to distributed artificial intelligence*, Mit Press, 1999
2. S. Camazine , JL. Deneubourg, N. Franks, J. Sneyd, G.Theraulaz and E. Bonabeau *Self-Organization in Biological Systems.* Princeton University Press, 2001.
3. J. Fromm. *The emergence of complexity* Kassel University Press, 2004.
4. V. Thomas, C. Bourjot, V. Chevrier and D. Desor. *Hamelin : A model for collective adaptation based on internal stimuli.* In Stefan Schaal, Auke Ijspeert, Aude Billard, Sethu Vijayakumar, John Hallam, and Jean-Arcady Meyer, editors, From animal to animats 8 - Eighth International Conference on the Simulation of Adaptive Behaviour 2004 - SAB04, Los Angeles, USA, pages 425-434, Jul 2004.
5. S. Kazadi, M. Chung, B. Lee and R. Cho *On the Dynamics of Puck Clustering Systems*, Robotics and Autonomous Systems, 46(1), pp. 1-27, 2004.
6. H D.Van Parunak, R. Savit andRL. Riolo *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users Guide.* Proceedings of Multi-agent systems and Agent-based Simulation (MABS'98), 10-25, Springer, LNAI 1534, 1998.
7. T. De Wolf, T. Holvoet and . Samaey *Engineering Self-Organising Emergent Systems with Simulation-based Scientific Analysis,* , Proceedings of the Fourth International Workshop on Engineering Self-Organising Applications (Brueckner, S. and Di Marzo Serugendo, G. and Hales, D. and Zambonelli, F., eds.), pp. 1-15, 2005
8. C. Bourjot and V. Chevrier. *A Platform for the analysis of artificial self-organized systems.* In 2004 IEEE International Conference on Advances in Intelligent Systems - Theory and Applications - AISTA'2004. (Luxembourg). 2004.
9. F. Zambonelli, M.P. Gleizes, M. Mamei and R. Tolksdorf, *Spray Computers: Frontiers of Self-Organization*, in proc of 1st International Conference on Autonomic Computing, 268-269, New York (NY), May 2004.

10. B. Edmonds *Using the Experimental Method to Produce Reliable Self-Organised Systems*, The 2nd International Workshop on Engineering Self-Organising Applications (ESOA 2004) at 3rd AAMAS, New York, July 2004.
11. B. Edmonds and J. Bryson *The Insufficiency of Formal Design Methods -The Necessity of an Experimental Approach for the Understanding and Control of Complex MAS*, (936-943) AAMAS 04.
12. M. Wooldridge and N. R. Jennings, *Pitfalls of agent-oriented development.* In Proceedings of the 2nd International Conference on Autonomous Agents, pages 385-391, 1998.
13. O. Teran, B. Edmonds and S. Wallis *Mapping the Envelope of Social Simulation Trajectories.* Multi Agent Based Simulation 2000 (MABS2000), Boston, MA, 8th-9thJuly, 2000. Lecture Notes in Artificial Intelligence, 1979:229-243 (2001).
14. M. Fehler, F. Kluegl and F. Puppe *Techniques for Analysis and Calibration of Multi-Agent Simulations*, in proc of ESAW 04, 2004.
15. A. Aussem and D. Hill, *Wedding connectionnist and algorithmic modelling towards forecasting Caulerpa Taxifolia development in the north-western Mediterranean sea*, Ecological Modelling, pp.225-236, 1999.
16. JPC. Kleijnen *Statistical tools for simulation practionners*, Marcel Dekker, 1986.