Computational Complexity of a Constraint Model-based Proof of the Envelope of Tendencies in a MAS-based Simulation Model

Oswaldo Terán[*][†], Bruce Edmonds[*]

[*]Centre for Policy Modelling, Manchester Metropolitan University,
Aytoun Building, Aytoun Street, Manchester, M1 3GH, UK.
Tel. +44 161 247 6478 *Fax*. +44 161 247 6802
{o.teran,b.edmonds}@mmu.ac.uk

[†]Department of Operation Research and Centre for Simulation
and Modelling, Universidad de Los Andes. Venezuela
Tel. +58 274 240 2879
oteran@ula.ve

Abstract.. This paper determines the complexity of a constraint model-based proof of the envelope of a tendency in the dynamics of a Multi-Agent-based simulation model. The proof is performed via a constraint model-based exploration of simulation trajectories using forward inference, by means of which a whole fragment of the simulation model theory is investigated. Such exploration allows for all simulation trajectories defined by a range of the model's parameters and a range of the agents' choices. The paper verifies that the search is *PSPACE-complete* for an infinite number of iterations, and suggests that the search is S_i *P-complete* for a finite number of iterations.

1 Introduction

There is a *need for* studying and *proving (emergent) tendencies* in the *simulation of social systems* (including simulation of organizations). This need has been especially remarkable in those works related with elaborating or testing theories [1, 2, 4]. Such a need has not been satisfied by *traditional approaches* for exploring the dynamics of simulations models, such as Scenario Analysis and the Monte Carlo method. Neither of these approaches performs exhaustive explorations of simulation trajectories in subspaces of the simulation theory. The explored trajectories are chosen, in the first case, by a domain expert, and in the second case, randomly. Owing to these facts, those approaches cannot be used for proving tendencies in the dynamics of a simulation model - the allowed conclusions are valid either according to the expertise of a domain expert or statistically.

As an *alternative* to these traditional methods, in previous papers [6-8] a *hierarchy of computational architectures* for searching for and proving tendencies in a Multi Agent System (MAS)-based simulation model is proposed. The first architecture, that at the higher level, consists of the MAS-based model where tendencies will be searched for by the modeller. After a tendency is found, at a second architectural level, a constraint logic model[1] proof of the envelope of the simulation trajectory is proposed. In those papers, a computational technique for doing this proof efficiently is implemented and illustrate by using an example. And, at a third architectural level, a more general proof of the envelope of the tendency would be implemented by exploring a wider fragment of the simulation theory by using a syntactic driven search. As explained better in [8], *this research contributes in bringing closer the simulation and the logic programming communities*.

This paper examines the computational complexity of the procedure implemented in the second architectural level. First, in the second section, the idea of envelope is reviewed. Afterwards, in the third section, the logic-based exploration of simulation trajectories implemented for proving the envelope of tendencies in simulation models is described. Then, in the fourth section, the computational complexity of such exploration is established. And finally, in the fifth section, some conclusions are presented.

2 Enveloping Tendencies in a Simulation Model

It does not seem convenient to use always the strong concept of envelope managed in mathematics. Apart from precision on the managed concepts, the idea of making the output comprehensible for a modeller is also important. An envelope will be chosen considering the trade-off between practical usefulness for a modeller and precision (by precision we mean how close the concept is to the ideal mathematical notion of a tangent curve/surface).

Consider the case of *enveloping a single simulation output, Y*. Each trajectory will generate a sequence of real values over time, *Y*. Calling y_{ij} the output value at time instant *i* for trajectory *j*, an envelope might consist of two sequences of values over time: E_{upper} and E_{lower}, which in some sense cover all trajectories. The value of E_{upper} at time instant *i* must be greater than or equal to y_{ij} for all *j*, and E_{lower} at time instant *i* must be lower than or equal to y_{ij} for all *j*. That is, the envelope would be given by two sequences of values over time, where for each time instant all values generated by the simulation trajectories are enclosed by the two values given by these two value sets. Putting this in other words, at each time instant, *t*, the smallest interval covering all points generated by the explored trajectories is included in the interval given by the two sequences E_{upper} and E_{lower} for instant *t*.

Alternatively, first an approximating function, *f*, for the output value set *Y* that each trajectory generates might be elaborated; then, the instances of these functions (one function for each trajectory) might be enveloped.

Among the procedures of interest for enveloping tendencies in simulation studies might be the followings:

- Enveloping certain *properties of the observed tendency rather than the tendency itself*. The results might permit one to relate the simulation results to theory developments and to elaborate conclusions with respect to the theory underlying the simulation model.
- Producing a *mathematical description* of some coarse borders of the space where the tendencies have been observed. This is useful if it is difficult to describe the subspace of the tendencies directly. Then, coarse borders are chosen as a first

[1] The term ‘logical model’ means model in the logical sense, which is different to the idea of model in modeling or in simulation. In this paper, a logical model corresponds to a simulation trajectory.

approximation to the envelope and, afterwards, these enclosing borders are expressed mathematically.

- *Using extreme cases of representative or typical instances* of a tendency. It is assumed that the observed tendencies in the simulation can be grouped qualitatively as similar or close enough (in accordance with some criteria) to a finite (small) number of typical tendencies.
- *Specifying a range of parameters and choices*. This is the description used in the exploration implemented in previous papers [6-8].

3 Proving Tendencies Via a Model-based Exploration of Simulation Trajectories in a MAS-based Simulation Model

3.1 Logical Model-Constrained Exploration of Simulation Trajectories

A simulation - either an event-driven, or a finite differences, or a MAS-based - can be seen as a partial logical model. Usually, in a trajectory only a partial set of all the facts of the logical model corresponding to the trajectory are explicitly generated. This partial set consists of those facts that are relevant, either because they are required for the modeller as outputs or because they are necessary to generate the simulation transition states. The remaining facts are left as unknown.

There are different methods to specify a theory in a language. One commonly employed in logic is by using a set of formulas of the language which become the axioms of the theory. In a declarative program a simulation model is specified via a set of rules and the underlying logic of the program. *Potential trajectories are defined via non-deterministic factors* of the simulation, e.g., parameters and choices.

The idea in the referenced previous studies was to analyse the *emergence of tendencies* in a simulation by exploring a subspace of the space of trajectories. That was done via a logical model-based constraint search, where the constraints standed for the selected parameters and choices. The exploration allows a modeller to explore that fragment of the simulation theory content over a range of parameters and choices (see Figure 1). Consequently, the resulting conclusions and proofs will be valid over that fragment of the theory and, under appropriate justifications, they can be extrapolated to the whole simulation theory.

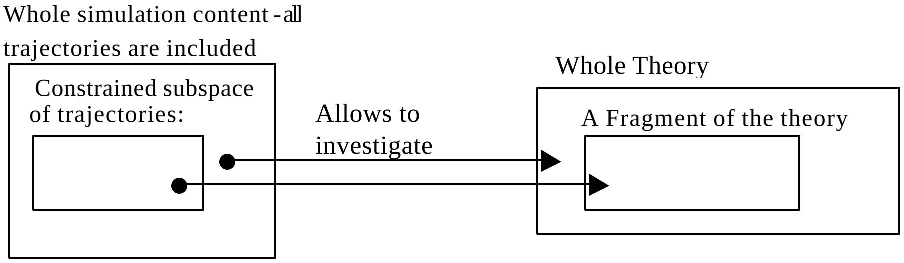
Whole simulation content - all trajectories are included
Constrained subspace of trajectories:
Allows to investigate
Whole Theory
A Fragment of the theory

Figure 1. Theory given by simulation trajectories

3.2 Logical Model Exploration for Proving the Necessity of a Tendency

The idea is to generalise about tendencies going from the observation of individual trajectories to observation of a group of trajectories generated for certain parameters and choices. In particular, it is intended to know if a certain tendency is necessary or contingent in the explored trajectories. We understand a simulation trajectory as a logical model embedded in a simulation program (a 'possible world' in semantic terms) and involving trajectories of entities (e.g., agents) inside the simulation and, hence, different from trajectories of these entities. It is a cross-product of all settings of the structure of the simulation model and all processes (e.g., agents' choices) into one path through a high-dimensional space (see Figure 2).

The character of the search is predominantly logical model, constraint, forward-chaining, and clausal ordered. A logical model is generated for each combination of parameters and choices. Each combination of parameters provides a different structure of the simulation model (see Figure 3). 'Paths' representing trajectories are generated for each structure. Then, while the simulation is going on, choices produce branch points where alternative settings for each choice turn out into a different simulation trajectory.

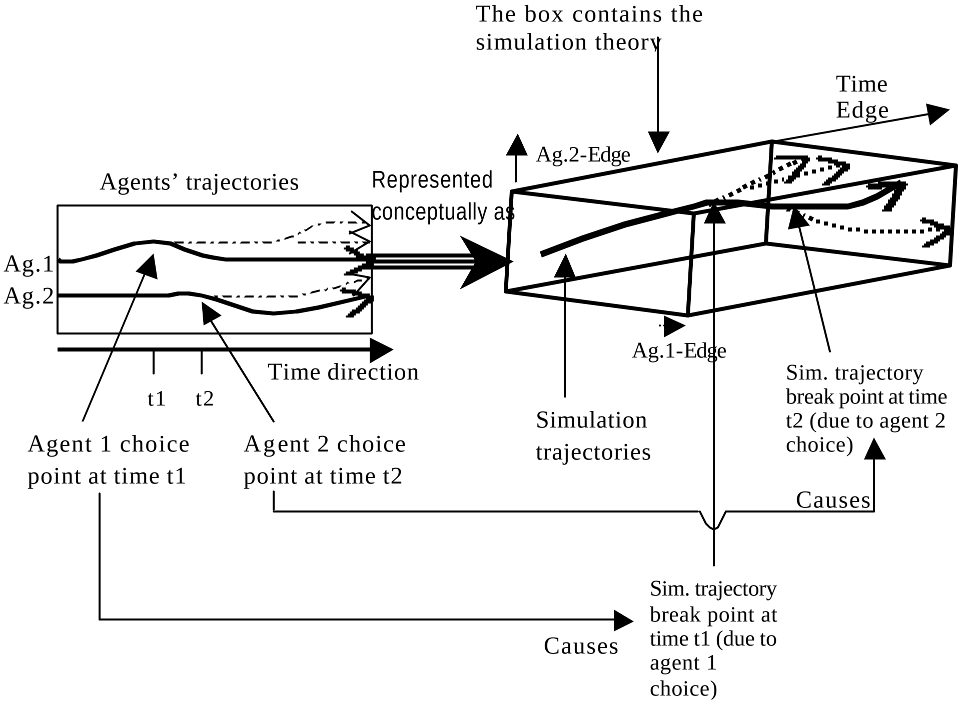
The box contains the simulation theory
Time Edge
Ag.2-Edge
Agents' trajectories
Represented conceptually as
Ag.1
Ag.2
Time direction
t1
t2
Ag.1-Edge
Sim. trajectory break point at time t2 (due to agent 2 choice)
Simulation trajectories
Agent 1 choice point at time t1
Agent 2 choice point at time t2
Causes
Causes
Sim. trajectory break point at time t1 (due to agent 1 choice)

Figure 2. Representation of a simulation theory in terms of the simulation trajectories, and of these in terms of agents' choices (for a single parameter-setting and assuming there are two agents)

This exhaustive constraint-based search over a range of possible trajectories makes it possible to establish the necessity of postulated emergent tendencies. Following a procedure similar to that used in theorem-proving [3,10], a subset of the possible simulation parameterisations and agent choices is specified, the target emergent tendencies are prearranged in the form of negative constraints, and an automatic search over the possible trajectories is performed.

Tendencies are shown to be necessary, with respect to the range of parameterisations and non-deterministic choices, by first finding a possible trajectory without the negative constraint to show the rules are consistent and then showing that all possible trajectories violate the negation of the hypothetical tendency when this is added as a further constraint. This is equivalent to showing that all possible tendencies obey the positive form of the constraint, i.e., that the positive form is true for all tendencies.

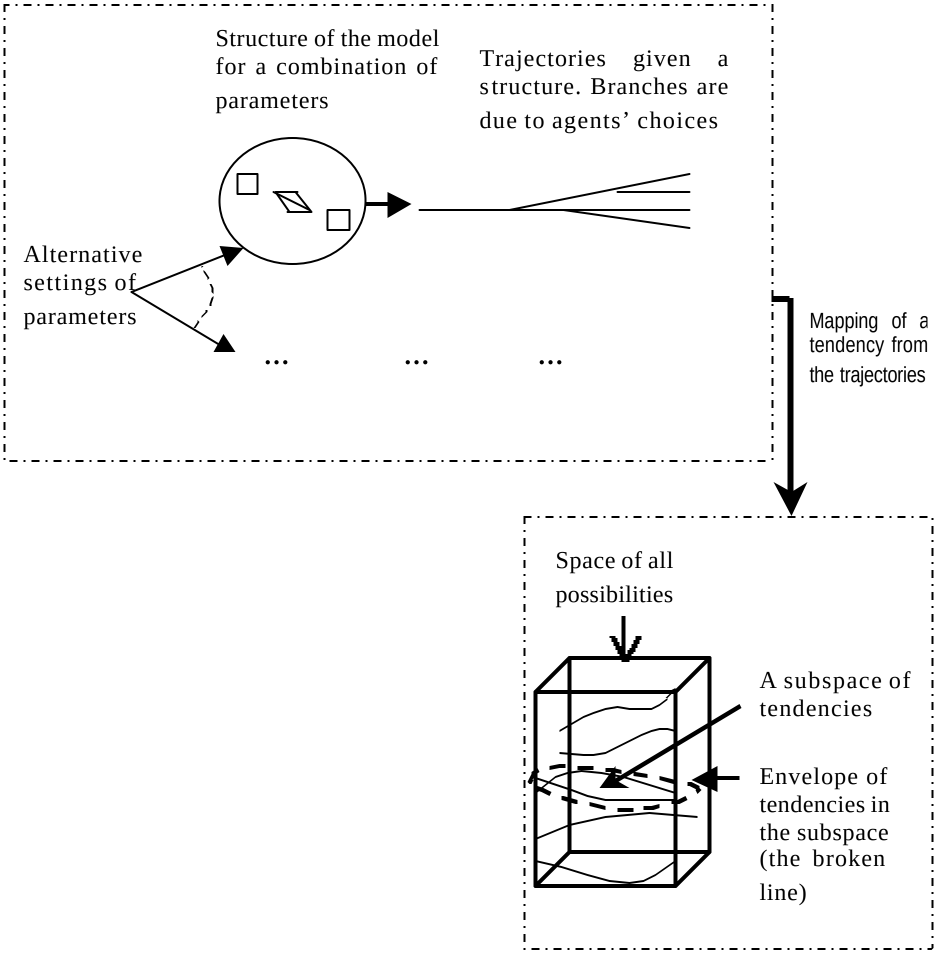
Structure of the model for a combination of parameters
Trajectories given a structure. Branches are due to agents' choices
Alternative settings of parameters
...
...
...
Mapping of a tendency from the trajectories
Space of all possibilities
A subspace of tendencies
Envelope of tendencies in the subspace (the broken line)

Figure 3. A model constraint-based exploration of the dynamics of a simulation model

4 Complexity of the Proof

The *aim* of this section is to demonstrate that the exploration of trajectories proposed in the previous section applied over an infinite (theoretically) number of iterations is *PSPACE-complete*. To make clearer the exposition, this aim is called the *target problem*. As is usual for this sort of verification, two steps are followed:

First, it will be proved that the target problem is in PSPACE by expressing it as a binary tree of depth *n*. According to Papadimitriou [5] this is sufficient (see examples in [5], pp. 455-462).

Second, it will be proved that the problem is also *PSPACE-complete* by translating another *PSPACE-complete* problem into the target problem. For this comparison, one of the problems Woolridge presents in [9] has been chosen, concretely that of agent-task-maintenance.

For the first part of the proof it must be possible to construct in polynomial space the game three, which is possible if the target problem is expressed in the form of a Boolean quantified expression (see examples 19.1 and 19.2 in [5]), as follows:

$$\$\, x_1 \;''\, x_2 \;\$\, x_3 \;''\, x_4 \;\$\, x_5 \ldots Q_n\, x_n\; (F) \qquad\qquad (1)$$

where *F* is the formula to be evaluated over the variables $x_1 \ldots x_n,$ and Q_n is the last quantifier, which will be $\$$ in case of *n impair* or $''$ in case of *n even*.

The *impair* variables correspond to the environment's action. The *deterministic* part in the state transition of the simulation will be called *environment's actions*. In the target example, it corresponds to all those changes not associated with *agents' choices*. Consequently, there is only one alternative action for the *impair* variables. The *even* variables correspond to the agents' choices (which are going to be called *agents' actions*). In the particular case of the example presented in [6], there are eight alternative agents' choices. So far, a state transition in a simulation has been divided into two parts: that *deterministic* part associated with the *existential* variables and that *non-deterministic* part associated with the *quantified* variables. A whole simulation path (or simulation trajectory) is represented by a concatenation of branches, where each branch corresponds to an assignment of values to a variable x_i.

Finally, *F* will be the question: whether the searched *tendency* has occurred in a simulation trajectory, where that trajectory is associated with an assignment of values for the variables, x_i. The whole expression *(1)* is true if for all possible assignments of values to the variables the tendency is true (remember that there is only one choice for the existential variables). As each particular assignment of values to the whole set of quantified variables corresponds to a simulation trajectory, the proof is successful if this expression is valid for all possible values the quantified variables can take! (e.g., for all possible agents' choices).

To check if the proof is successful, a boolean circuit, where and *OR* gate stands for the $\$$ quantifier and an *AND* gate stands for the $''$ quantifier, is written (see Figure 4). A leaf in this circuit is evaluated to *true* if the tendency is found in the corresponding simulation path and to *false* otherwise. The whole circuit will be *true* if and only if the tendency appears in all simulation paths. Hence, the proof is successful if and only if the circuit is true (e.g., the tendency is found in *all* paths).

These two expressions found of the problem (that is, the Boolean circuit shown in figure 4 and the expression of equation *(1)*) are sufficient to prove that the target problem is *PSPACE*. The next task is to prove that the problem is *PSPACE-complete*.

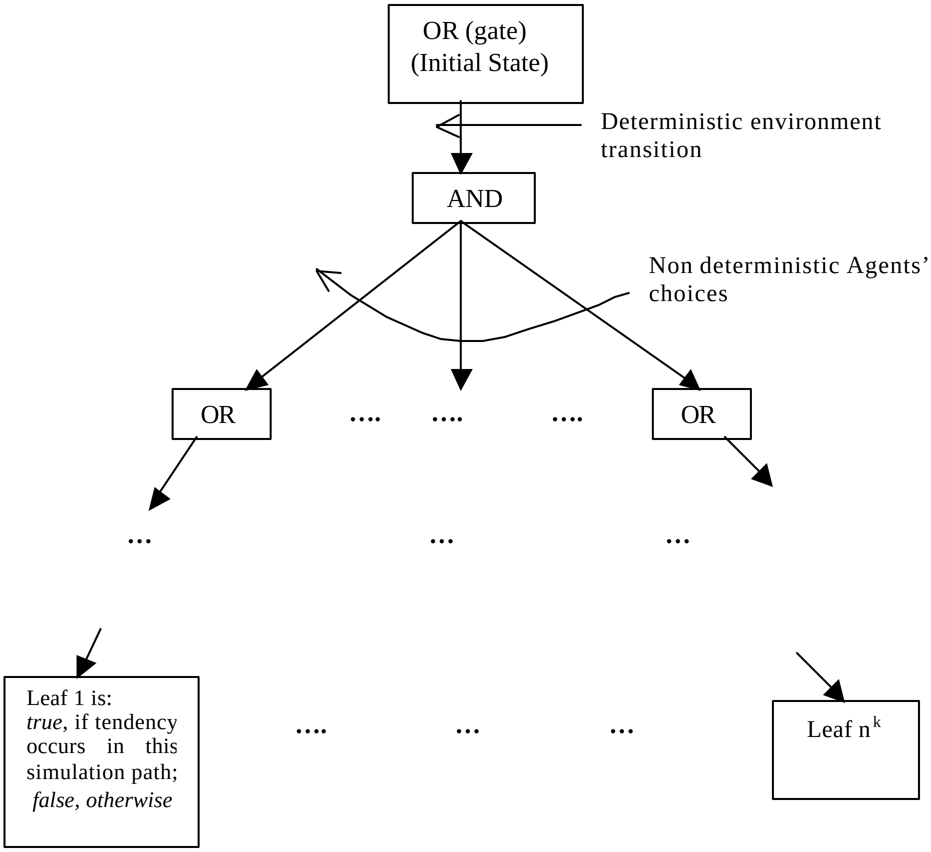
OR (gate)
(Initial State)
Deterministic environment transition
AND
Non deterministic Agents' choices
OR
OR
Leaf 1 is:
true, if tendency occurs in this simulation path;
false, otherwise
Leaf n^k

Figure 4. Boolean circuit for the target problem

Comparing with Woolridge [9], an algorithm to check the proof might be written. This will bring the example close to the one he uses when considering *maintenance tasks*. Assuming a Turing machine *M* is called recursively at each branch point (at agents' choices) and that this machine is kept in use while actions are deterministic (environment's action), the algorithm for *M* will be:

Algorithm 1:

1. If the tendency appears, then the branch is evaluated to *true (success*);
2. If there are no allowable simulation actions, the branch is evaluated to *false* (*fail);*
3. Execute the deterministic aspects of the state transition (environment action), then for each agent's choice recursively call *M;*
4. If all recursive calls in 3, are *successful* (i.e., evaluated to *true*), then *M* is *true (success).*

To prove that the target problem is *PSPACE-complete*, consider the maintenance problem in [9]. There, agents are chosen non-deterministically to act against the environment. Agent's actions are deterministic, while environment's actions are non-deterministic. The idea is to check if there is any choice of agents' actions that is successful in bringing the environment into one in a set of states whatever the environment chooses. It is like a game where agents play against the environment. Woolridge proves that the agent-maintenance problem is in NPSPACE using the following *algorithm*:

Algorithm 2:

1. if r [the run until a branch point] ends with state $\in$ *G* [the set of goals], then *M* accepts;
2. if there are no allowable actions given r, then *M* rejects;
3. non-deterministically choose an action α from *Ac* (possible agents' actions, there is one per agent) and then for each $e \in \tau$ (set of possible environment's states) recursively call M with the run $r \cdot \alpha \times e$;
4. if all of these accept, then *M* accepts, otherwise *M* rejects.

In Woolridge's problem, rather than searching for a tendency, the idea is to bring the simulation into one among a set of environment states. If the environment is brought into one of these states, it is said that the selected agents have been successful in their game against the environment. In Woolridge's example, the agents' actions are deterministic; e.g., they have only one choice, but different agents can be selected. Selection of agents corresponds to the *OR* nodes in the circuit shown in the Figure 4, each branch corresponding to the choice of a different agent. On the other hand, the environment has non-deterministic actions, and, correspondingly, their choices are associated with the *AND* gates in the circuit.

A difference between *algorithm 1* and Woolridge's algorithm (e.g., *algorithm 2*) is that in the latter, at step 3, *M* is called for each possible environment's state after an agents' choice is selected non-deterministically, while in the former *M* is called, in step 3, for each agent's choice after the (deterministic) environment action is performed. So, the deterministic action of the environment in step 3 in the former algorithm corresponds to the non-deterministic choice of agents in the latter. Consequently, though the translation of Woolridge's problem into the target problem seems straightforward, there is still a small difficulty: his case study is non-deterministic (owing to the non-deterministic choice of agents in step 3 in *algorithm 2*), while the target problem is deterministic. Woolridge's original problem is NPSPACE.

To solve the difficulty, consider the deterministic version of Woolridge's problem. Think about checking the successfulness of agents' actions in his problem once an agent has been chosen in advance at each branch point. This is a deterministic problem. It is in PSPACE but still as hard as Woolridge's original one as NPSPACE = PSPACE ([5], p. 150). Woolridge's algorithm for this deterministic version of the agent-maintenance task problem becomes:

Algorithm 3:

1. if r [the run until a branch point] ends with state $\in$ *G* [the set of goals], then *M* accepts;
2. if there are no allowable actions given r, then *M* rejects;
3. deterministically use the action α *given in advance* from *Ac* (possible agents' actions, there is one per agent) and then, for each $e \in \tau$ (set of possible sates of the environment), recursively call M with the run $r \cdot \alpha \times e$;
4. if all of these accept, then *M* accepts, otherwise *M* rejects.

The translation of the determinist version of Woolridge's problem into the target problem is straightforward from *algorithms 1* and *3*. The deterministic action of the environment at step 3 of *algorithm 1* corresponds in Woolridge's algorithm (*algorithm 3*) to the deterministic action of an agent already chosen. The recursive calls of *M* made for agents' choices in *algorithm 1* correspond in *algorithm 3* to the recursive calls of *M* for the environment’s choices. With regard to the circuit shown in Figure 4, agents’ choices in Woolridge’s problem (now deterministic) are placed at the *OR* gates and environment (non-deterministic) choices in Wooldridge’s problem are placed at the *AND* nodes. Therefore, the deterministic version of Woolridge’s maintenance problem has been translated into the target problem, and, consequently, the target problem is also *PSPACE-complete*.

It has been demonstrated that the target problem is *PSPACE-complete* for an infinite number of iterations, *i*. Using the experience accumulated so far in this proof for *i* infinite (particularly useful is the expression of the problem in the circuit given above), and theorems 17.8 (especially its corollary 2) and 17.10 in [5], it should be possible to prove that the problem is S_i*P-complete* if the number of iterations, *i*, is finite.

5 Conclusion

This paper has verified that the complexity of a constraint model based exploration of simulation trajectories for proving the envelope of tendencies in the dynamics of a MAS-based simulation model is *PSPACE-complete* for an infinite number of iterations and has suggested that it is S_i*P-complete* for a finite number of iterations, *i*.

Proving the envelope of tendencies in simulation outputs is an alternative to traditional methods used for examining simulation outputs. The former allows elaborating more general conclusions than the latter.

As explained better in [8], constraint exploration of simulation trajectories brings closer the simulation and the logic programming communities. This paper contributes in making clearer a property of high interest to both of these communities, namely the computational complexity of a constraint exploration of simulation trajectories

Acknowledgements. The research reported here was funded by CONICIT (the Venezuelan Governmental Organisation for promoting Science), by the University of Los Andes, and by the Faculty of Management and Business, Manchester Metropolitan University.

References

1. Axtell, R., R. Axelrod, J. M. Epstein, and M. D. Cohen, “Aligning Simulation Models: A Case Study and Results”, *Computational Mathematical Organization Theory*, 1(2), pp. 123-141, 1996.
2. Carley K., M. Prietula, and Z. Lin, “Design Versus Cognition: The Interaction of Agent Cognition and Organizational Design on Organizational Performance”, *Journal of Artificial Societies and Social Simuation* 1(3), 1998 (accessible at: http://www.soc.surrey.ac.uk/JASSS/1/3/4.html).
3. Loveland, D. W., *Automated Theorem-proving: A Logical Basis*, North-Holland Pub., Amsterdam, 1978.
4. Moss, S., “Social Simulation Models and Reality: Three Approaches”, *MAB’s 98: Multi-agent Systems and Agent-Based Simulation*, Paris, 1998 (accessible at http://www.cpm.mmu.ac.uk/cpmrep35.html).

5. Papadimitriou, Christos, *Computational Complexity*, Addison-Wesley Publishing Company, California, USA, 1994.
6. Terán Oswaldo, Bruce Edmonds and Steve Wallis, “Mapping the Envelope of Social Simulation Trajectories”, MABS2000 @ ICMAS-2000: The Second Workshop on Multi Agent Based Simulation, Boston, July 9, 2000. Published in: Moss, Scott and Paul Davidsson (Editors), *Multi Agent Based Simulation (MABS-2000), Lecture Notes in Artificial Intelligence, Vol. 1979*, Springer Verlag, Berlin
7. Terán Oswaldo, Bruce Edmonds and Steve Wallis, **“**Determining the Envelope of Emergent Agent Behaviour via Architectural Transformation”, ATAL-2000: The Seventh International Workshop on Agent Theories, Architectures, and Languages, Boston, July 7-9, 2000. Published in: Castelfranchi, C. and Y. Lesperance (Editors), *Intelligent Agents VII. Agent Theories, Architectures, and Languages. Lecture Notes in Artificial Intelligence, Vol. 1986*, Springer-Verlag, Berlin
8. Terán Oswaldo, Bruce Edmonds and Steve Wallis, “Constraint Exploration and Envelope of Simulation Trajectories”, First Workshop on Rule-Based Constraint Reasoning and Programming at the First International Conference on Computational Logic (CL2000), July 24-28, 2000, Imperial College, London, UK (this paper is accessible at: http://www.pst.informatik.uni-muenchen.de/personen/fruehwir/cl2000r.html; and at: http://arXiv.org/abs/cs/0007001)
9. Woolridge, Mike “The Computational Complexity of Agent Design Problems”, in *Proceedings Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, Boston, MA, USA, July 10-12, 2000, pp. 341-348.
10. Wos, L., *Automated Reasoning*: *Introduction and Applications*, Prentice Hall, London, 1984.
11. Zeigler, Bernard, *Theory of Modelling and Simulation*, Robert E. Krieger Publishing Company, Malabar, Fl, USA, 1976.