Meta-ABS
Recursive Agent-Based Simulation

Jonathan Thaler

March 21, 2017

Abstract

In this paper we ask what influence recursive Agent-Based Simulation has on the dynamics of a simulation. We investigate the famous Schelling Segregation and implement our agents with the ability to anticipate their actions by recursively running simulations. Based on the outcomes of the recursions they are then able to determine whether their move increases their utility in the future or not. We investigate the dynamics of the MetaABS implementation and compare it to the movement-strategy of the original model. We hypothesize that in the case of a deterministic future this approach allows the agents to increase their utility as a group but we hypothesize that this is not the case when the future is non-deterministic as the power to predict is simply lost in this case. Also we show that alone by looking at the implementation we can raise interesting philosophical questions about agents, anticipation, information, determinism. The main contribution of this paper is the introduction of recursive agent-based simulation, a completely new method in ABS, which we termed MetaABS.

1 Introduction

The 'meaning' of MetaABS is not really clear: how can it be interpreted? It is not so much about the dynamics but more on the philosophical questions it raises. But also we wanted to check if the same happens as in the recursive simulation paper [1] : deterministic vs. non-deterministic AND one-agent recursion or all-agents recursion
we are spanning up 3 dimensions: recursion-depth, replications, and time-steps
the agent who is initiating the recursion can be seen as 'knowing' that it is running inside a simulation, but the other agents are not able to distinguish between them running on the base level of the simulation or on a recursive level
We implemented our Meta-ABS in Haskell using the functional reactive programming paradigm following the Yampa library. We believe that pure functional programming is especially suited to implement Meta-ABS due to its lack of implicit side-effects and copying of data. The code is available freely under TODO: until now the whole things is implemented in functionalReactiveABS, when project and paper is finished, copy the code-base to metaABS and insert link

2 Background

2.1 Schelling Segregation

2.1.1 Movement strategies

The original Schelling model only knew the local strategy.

Local An agent which moves selects an unoccupied place randomly relative from its current place within a rectangle of side-length 2r where its current place is at the center. The interpretation for that behaviour is that agents won't move too far as it could be costly. Also children might attend a school in this area or the

family has friends in this area, so they don't want to break that.
We introduce an additional movement-strategy

Global An agent which moves selects an unoccupied place randomly within the whole environment. The interpretation for that behaviour is that agents don't care for the moving-costs and don't really care where they end up.

2.1.2 Optimizing behaviour

The original schelling model didn't have a move-optimizing behaviour, meaning agents are just binary: if it is happy it will not move, if it is unhappy it will move but they won't care where they move.

None Agents just move depending on their movement-strategy to another place if they are not happy on the current one - they don't care how the target place is in the present or in the future, they will decide again in the next time-step. The interpretation for that behaviour is: agents want to 'just get out' at any cost, not caring what the future place will look like - it might be better or worse but they will see then.
We introduce local move-optimizing behaviours which can be interpreted as being realistic in the real-world. It is important to note that we focus on *local* instead of *global* move-optimization: the agents are limited in their reasoning-capabilities and have limited information available: they cannot check out *every* place and pick the globally best one.

Optimizing present Agents pick an unoccupied place depending on their movement-strategy and move to it if it increases their utility. The interpretation for that behaviour is: agents heard about a cool spot in town, check it out and move to it if they like it.

Optimizing future Agents pick an unoccupied place depending on their movement-strategy and move to it if it increases their utility in the future. The interpretation for that behaviour is: agents heard about a place which will be cool in the future.

Optimizing present & future Agents pick an unoccupied place depending on their movement-strategy and move to it if it increases their utility in the now and in the future. The interpretation for that behaviour is: agents heard about a cool spot in town, check it out and move to it if they like it but they also anticipate the coolness of the place in the future and if it seems that the place is going down then they won't move there.

3 Meta ABS

Informally, Meta-ABS can be understood as giving the agents the ability to project the outcome of their actions in the future. They are able to halt time and 'play through' an arbitrary number of actions, compare their outcome and then to unhalt time and continue with a specifically chosen action e.g. the best performing or in which they haven't died.

3.1 Formal description

explain the level two levels of recursion

3.2 Interpretation

TODO: how can be Meta ABS be interpreted?
how can we justify the ability of the agents to do that? real persons lack the information and the computational power to forecast such a complex problem.

4 Results

Dynamics: - global happiness over time - happiness-change between steps

Table 1: Model Configuration

Dimensions	50 x 50
World-type	Torus
Density	0.75
Similarity required	0.8
Agent-distribution	50% Red, 50% Green
Local-movement distance	5
Find-free-place retries	4

4.1 •

4.2 Future optimizing

a fraction pf agents is predicting: see what influence it has
- deterministic future vs. non-deterministic future
questions: - can it solve the difficult case which global optimizing can solve? - is it faster than global optimizing?

5 Conclusion and further research

References

[1] Gilmer, Jr., J. B., and Sullivan, F. J. Recursive Simulation to Aid Models of Decision Making. In *Proceedings of the 32Nd Conference on Winter Simulation* (San Diego, CA, USA, 2000), WSC '00, Society for Computer Simulation International, pp. 958–963.