

Chapter 6

Large Simulations and Small Societies: High Performance Computing for Archaeological Simulations

Xavier Rubio-Campillo

6.1 Introduction

Computational simulation of societies has found in archaeologists one of the most promising fields of applications, as can be observed in the number of related publications (Doran et al. 1994; Doran 1999; Lake 2000; Diamond 2002; Kohler and van der Leeuw 2007; Costopoulos and Lake 2010; Lake 2013). The reason of this interest can be found in the link between the basic objectives of archaeology and simulation. Archaeology attempts to understand human behavior through the detection and analysis of spatio-temporal patterns related to the location and type of found structures and objects. Uncertainty is inherent to this process, as any research based on data from the archaeological record should cope with it, both in time and space (Crema et al. 2010). Within this context, computer simulation is the perfect virtual lab because it is capable of dealing with mathematically intractable problems such as the interaction of complex human behavior (Galán et al. 2009). They are suited to explore different hypotheses capable of explaining detected patterns, as well as to validate them, at least within the context of the simulation model. Agent-Based Models is one of the most widely spread simulation techniques; its basic concept of entities with individual decision-making processes interacting within a common environment is well suited to explore the type of questions faced by archaeologists.

From a broader perspective a common trend of Agent-Based Models simulating human behavior is a gradual increase in their complexity (Bonabeau 2002). These research projects often have at their disposal a vast amount of data that can be used to define the value of several parameters, as well as the behavior of the different agents.

X. Rubio-Campillo (✉)
Barcelona Supercomputing Center, Barcelona, Spain
e-mail: xavier.rubio@bsc.es

The results are large scale simulations executed in High-Performance Computing (HPC) infrastructures (i.e. supercomputers), created to explore realistic scenarios, develop policy analysis and manage emergency situations, to cite some applications (Leitao et al. 2013; Macal et al. 2008).

The emergence of these large scale simulations brings also a theoretical issue: the intractability of Agent-Based Models understood as computational problems. From this perspective the cost of executing an ABM in a computer follows an exponential curve. This is crucial if we want to understand the emergent behavior of an ABM given any set of initial conditions. Within this perspective the computational cost for the solution will increase exponentially to the number of parameters of the model. The consequence of this property is that, at a given point, we will not be able to solve an increasingly complex ABM in a reasonable amount of time, regardless the computer we use. We can have better computers, but the requirements will be orders of magnitude larger than our capabilities.

There is an ongoing debate in the community of archaeologists developing ABMs in relation to this increasing complexity. Some works try to follow the trend, developing detailed models designed to test hypotheses on realistic scenarios. Simultaneously a different group of modelers are focused on theory building, creating abstract simulations with simpler scenarios (for this division see Lake 2001, 2013). The discussion is centered on the best way to use the efforts of the community, taking into account the fact that realistic models cannot be fully defined on the basis of the fragmented data collected by the discipline. In this sense, the development of large scale complex models with weak assumptions could strongly bias the interpretation of the archaeological record, thus decreasing the scientific quality of research (Premo 2010). Paradoxically the development of abstract models without realistic backgrounds can keep away most potential audience from simulations, thus preventing most archaeologists from using this interesting tool.

Regarding HPC one could think that this technology cannot really offer anything to archaeologists, given that it seems focused on boosting large scale ABMs with huge scenarios and millions of agents. The models used in archaeological research never have these requirements, so where is the need for additional computational power? This chapter will show that the benefits of HPC are not constrained to execute large simulations. In particular we will explore:

- How can we speed up the execution of a computer simulation of a small-scale society?
- Is it possible to improve Agent-Based Models with an increase on computational power?
- What pitfalls and issues do the introduction of HPC create?

Not only is HPC important for accelerating the performance of the simulations, but we also need a boost on computing capabilities in order to answer some of the methodological questions arised during recent years. Some deficiencies of archaeological ABMs are related to the lack of these capabilities, specially regarding realistic hypothesis testing models. An immediate improvement caused by the use

of additional computers is a better exploration of parameter spaces, but there are others that can be even more interesting for the discipline. The introduction of advanced decision-making processes and the development of multiscale simulations could also improve the scientific quality of computer simulations developed in archaeology.

Next section defines background concepts needed for the chapter. This is followed by a discussion about the properties of Agent-Based Models and the potential approaches to executing them using HPC resources. This section also provides an overview of existing software and applications usable for achieving this task. Fourth section deals with the methodological issues solved (and generated) by the introduction of HPC. Finally, different concluding remarks are provided to summarize the ideas of the chapter, as well as future lines of research.

6.2 Background

Some technical concepts need to be defined before continuing the discussion. A general overview of an HPC infrastructure can be seen in Fig. 6.1:

- The *Central Processing Unit (CPU)* is the atom of any computing system. It is responsible for calculating arithmetical and logistical operations. Most existing

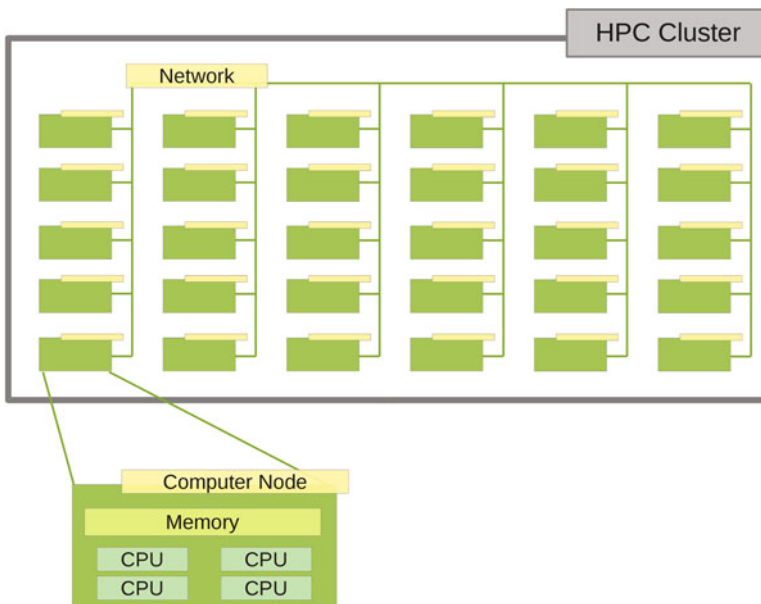


Fig. 6.1 General architecture of an HPC infrastructure

systems combine the power of different CPUs, and are known as multi-core processors.

- A *computer node* is a set of processors that share memory.
- An *HPC infrastructure* is a large set of computer nodes interconnected with some type of network. This can be as simple as some computers communicating through internet (i.e. grid computing), or a supercomputer where nodes are interconnected with high-speed networks.

Different hardware components can be used to execute a simulation. If we want to distribute the process in more than one CPU we have two choices:

- Multiple processes in one node. The different CPUs of a computer node share the access to RAM (Random Access Memory). As a consequence, all of them can read and write in the same memory space. If a computer simulation wants to use more than one CPU it needs a mechanism to avoid conflicts generated by simultaneous writing of a particular memory section. This can be achieved with different techniques, being threads the most widely used. Another solution, OpenMP, is interesting for scientific applications given its capabilities for easily adding multicore execution to any software (Dagum and Menon 1998; Kuhn et al. 2000).
- Multiple nodes. The computer nodes of a distributed system (i.e. clusters, supercomputers, etc.) do not share memory. Any process being executed in more than one node must have some way to communicate among each section of the execution. This is achieved using the MPI (Message Passing Interface, MPI Forum 2009) protocol, that specifies how a distributed (or parallel) application must send and receive data using the network.

By definition, these solutions are complementary. State-of-the-art HPC applications use an hybrid approach to parallelism: a simulation is split into different nodes, while each section exploits all the cores of a given node.

6.3 Developing HPC-Based Simulations

From a technical point of view the distribution of an agent-based simulation in different computer nodes is a complex task. There is a strong, inherent coupling of the various components of the system. They are constantly interacting with each other, so communication is always intensive.

Each agent needs to gather knowledge from their environment, as well as from other agents in order to execute their decision-making processes. Once this phase is completed, there is a possibility that the agents will modify the environment (and so will the other agents). These mechanics translate, in terms of parallelization, to the need of sharing several layers of environmental data and agents between computer nodes, at every single time step. Furthermore, the execution of the agents' actions cannot usually be distributed within a single computer node (i.e. OpenMP), because

there can be several conflicts related to agents accessing and modifying the same data at the same time.

Based on these constraints the best way to distribute the execution of an ABM over different computer nodes is to adopt a case-based approach. There is no optimal way to distribute an ABM, but a set of different solutions that can be applied to similar problems. The user should choose a particular method to split the execution amongst the nodes, depending on the nature of the model and the properties of the system. This idea is the basis of different initiatives like GridABM (Gulyás et al. 2011), a framework of template solutions for distributing these type of simulations.

Taking this perspective the majority of archaeological simulations share the following properties:

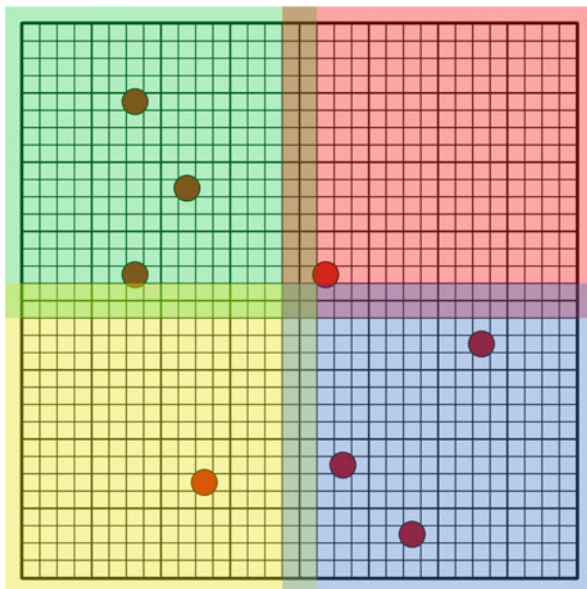
- Importance of environment. Hypothesis testing models developed in Archaeology usually focus on the strong relation between agents and the surrounding landscape. For this reason, these ABMs will need a substantial amount of spatially structured information, traditionally stored in a Geographical Information System (Conolly and Lake 2006). These platforms are spatial databases where data is stored as vectors (database records with spatial coordinates) and raster maps (bi-dimensional matrices containing regularly distributed values). The agents' environment will be modelled with a combination of these concepts, generating diverse information such as digital elevation models, site distribution, vegetation indexes, etc.
- Low number of agents. Past societies were not as dense as the ones we live in today. We will not need to reproduce situations with millions of agents, as simulations will usually model small-scale societies.
- Intensive local communication. A world without telecommunications means that, within a reasonable scale, an agent will only interact with the environment and agents that are spatially close to it. These interactions can be intense, but in any way an agent will need to know the entire situation of the world, just the section of it within a given interaction range.

6.3.1 Distribution of Archaeological ABM in Different Computer Nodes

These properties suggest that the best choice for splitting an archaeological ABM is to use spatial partitioning: each computer node owns a section of the entire simulated scenario, containing the different landscape data as well as the agents.¹ This is the solution used in Pandora (Rubio-Campillo 2013) and Repast-HPC (Collier and North 2011), the only HPC frameworks used in published archaeological ABMs.

¹A different approach is seen in Long et al. (2011), where agents are grouped based on the network of interactions.

Fig. 6.2 Spatial partitioning of an Agent-Based Model. Each *color* represents the section of the world owned by a different computer node, including raster maps with environmental information and agents. The zone where different *colors* overlap is the border zone



Pandora defines the environments where agents live as a set of layers containing raster map structures, following GIS standards. The world of the simulation is evenly divided among computer nodes, and each one of them will own a section of the environment, as well as the agents located in this part of the world. This layout is depicted in Fig. 6.2. Information in the border between adjacent nodes (raster maps and agents) will be communicated to neighbours every time step execution, in order to keep up-to-date data in the entire scenario. The size of this buffer border will be defined as the maximum interaction range of any agent, being the absolute horizon of actions of any component of the simulation. This approach is also adopted by Repast-HPC, and solves the issue of distribution for models of past societies without global interaction. The solution is highly scalable, given the fact that every computer node will need to communicate, at most, with eight neighbouring nodes (if nodes own rectangular regions), independently of the total size of the simulation.

Unfortunately, the issue of collision between agent's actions is still present, because two agents *living* in different computer nodes could be modifying the same bordering data at the same time. There are different techniques to avoid this conflict, but most of them can be computationally intensive (i.e. rollbacks). This overhead is affordable if the ABM is CPU intensive and the possibility of conflict is not high, but this is seldom the case.

Pandora, for example, takes a simpler approach. It is based in the segmentation of the spatial section owned by every computer node in four equal parts numbering 0–3, as it can be seen in Fig. 6.3. The agents contained in all 0 sections will be executed simultaneously without the possibility of conflicts, given the fact that they are not adjacent. Once all of them are finished, modified border data is sent to the

Fig. 6.3 The execution of any computer node is divided in four different sections, that are executed sequentially

0	1	0	1
2	3	2	3
0	1	0	1
2	3	2	3

neighbors, and a new section will begin its execution (1, 2 and finally 3). Once all of them are executed, the entire state of the simulation is serialized and a new time step can be evaluated.

The pitfall of this solution is that agents in section 0 will always be executed before agents in sections 1–3. Depending on the model the consequences of this effect can be nonexistent, or introduce artifacts in the outcome. As usual, a careful choice between the different strategies is needed, based on the existing scenario.

6.3.2 *Simultaneous Execution of Agents*

Parallelization of a simulation on different computer nodes is needed but is not sufficient. Every node should be able to use its complete set of CPU cores to simultaneously execute its agents. Again, the problem of conflicts between agents' actions is a barrier that must be broken if we want to avoid artifacts on simulation results.² To fix this we need to take a closer look at the way ABMs execute the set of agents (the scheduling system), in order to find properties useful to solve the problem.

Most of the time needed to execute an ABM is spent in the same task: the moment when the agents gather information, choose a particular set of behaviors, and execute them. These processes are always mixed in a single method, executed by every agent every time step. This is the approach taken by the three most popular ABM platforms: *tick* in Netlogo (Wilensky 1999), *step* in MASON (Luke 2011) and RePast (North et al. 2007).

It is worth to note that the first two processes do not modify anything, as the agent is only evaluating potential course of actions depending on existing data. For this

²For example two agents modifying at the same time the same cell of a given raster map.

reason we could simultaneously execute the decision-making process of different agents. This solution is safe of conflicts if the agents only choose a set of actions (without applying them).

Pandora uses this approach to split the step of an agent in three different methods. In the first one, *updateKnowledge*, an agent cannot modify the environment or other agents; it only gathers information. In the second one, *selectAction*, the agent executes the decision-making process and choose an action (it still cannot modify anything). Once every agent has chosen what it wants to do, Pandora executes the actions of the agents sequentially. Finally, the third method that a user can specify is *updateState*, where any agent can modify its internal state evaluating the results of its actions. This cycle *Explore–Decide–Apply* allows Pandora to distribute the execution amongst different CPU cores of a node, as the first two steps (the most computationally expensive) can be easily parallelized. The third one is executed sequentially, thus avoiding conflicts between the actions of the agents.

This structure could seem more complicated than just defining one method but, from a theoretical point of view, the division of an agent's execution in these three steps is more consistent than the traditional ABM approach. The single method implementation mixes the different stages of an agent's cycle, that should be correctly specified while building the model (see Fig. 6.4). Dividing the execution as shown here avoids this problem, forcing coherence during the transition from theory to code.

In addition this solution provides a cleaner interface to implement agents with advanced artificial intelligence, when this requirement is needed. The AI algorithm will be executed in one phase of the cycle (*decide*), that encapsulates the entire decision-making process and keeps it separated from the rest of the model.

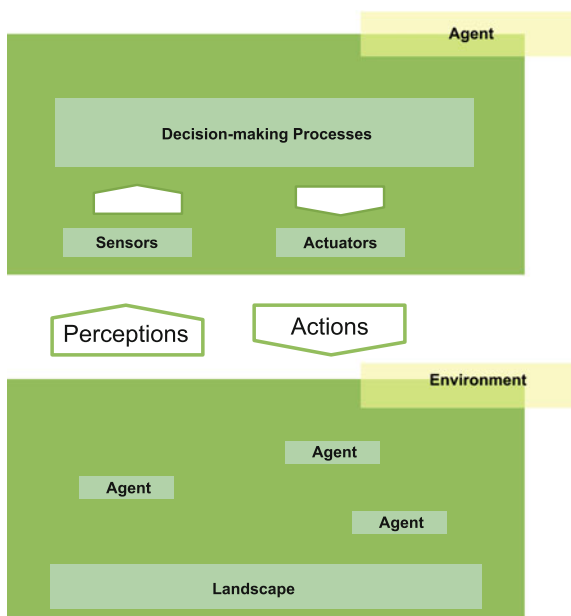


Fig. 6.4 The execution cycle of an agent

6.3.3 *Serializing Distributed Simulations*

If every CPU of every computer node is simultaneously executing agents there is an additional issue that needs to be dealt: bottlenecks during the serialization of the results. The scientific analysis of an ABM simulation will usually need to keep track of the evolution of the entire system from the beginning until the end of an execution. This means that the simulation needs to provide access to the state of the simulation at any given time step and at multiple levels of scale (from summary statistics to the state of individual agents). Besides, most HPC systems use a job cue where programs are submitted in order to be executed. This batch system is not interactive, so the researcher will usually access the outcome after the simulation finishes. The entire simulation state should be stored in a single file while it is running, in order to analyze it after the execution. The different nodes should wait to store their information while the rest are writing. This heavily affects the performance of the simulation, wasting resources and time due to the high volume of data to serialize.

One solution to the serialization bottleneck is that every computer node writes into a different file. The problem of this approach is that it requires postprocessing of different files in order to merge results, and this is also a costly operation. The optimal strategy, chosen by both frameworks (Repast-HPC and Pandora) is the distribution of the serialization through HDF5 (Folk et al. 1999). It is an open-source library specifically designed to store the outcome of scientific visualizations. It allows for simultaneously serializing data from several computer nodes in a structured, binary format. In addition, raster maps stored using HDF5 can be loaded by GIS applications (e.g. Quantum GIS, Sutton et al. 2009) in order to minimize the postprocessing of the data.

6.3.4 *Theoretical Performance of Distributed ABMs*

We can compute the improvements of distributing ABMs taking a look at theoretical wall time (real-world time needed to finish an execution). We can define the time needed for a model with local interaction as:

$$WallTime_L = N \cdot timeSteps \cdot (E + I \cdot L) / numCpus$$

being N the number of agents in the simulation, E the time an agent spends interacting with the environment in a given time step, I the time spent interacting with another agent and L the number of agents within range of interaction.³ Similarly, the wall time for a model with global interaction is defined as:

$$WallTime_G = N \cdot timeSteps \cdot (E + I \cdot N) / numCpus$$

³This would be the cost for the majority of archaeological ABMs.

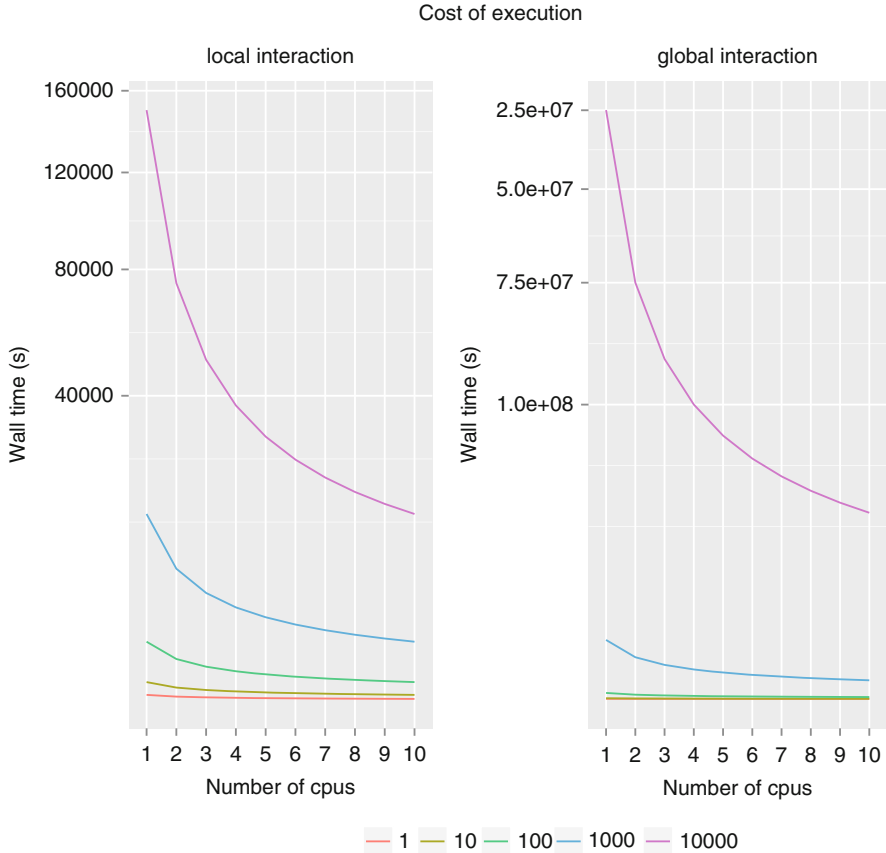


Fig. 6.5 Theoretical comparison of temporal costs of execution for Agent-Based Models with parameters $timeSteps = 1,000$, $E = 0.005$ s, $I = 0.001$ s, and $L = 10$ for (*left*) simulations where only local interaction exist and (*right*) simulations where agents interact with any other existing agent. Y axis shows (with a root squared scaling) the time needed to execute simulations with different number of agents (defined by *color*) for given cpus (X axis)

Figure 6.5 compares the theoretical cost of executing an ABM for both local and global interaction scenarios.

This decrease in wall time is the maximum boost in performance that a distributed simulation can achieve. Overheads generated by communication, serialization and other processes will lower performance, to the point that the increase in the number of computer nodes could slow the execution of very simple models. In this context the modeller is responsible for choosing the optimal infrastructure capable of accelerating the simulation execution (Witek and Rubio-Campillo 2012b).

6.4 Computational Solutions of Methodological Problems

We will explore three methodological discussions that can be enriched with the introduction of increasing computing power: parameter sweeps, exploration of size dependent emergence, and the problem of self-fulfilling prophecies.

6.4.1 *Parameter Sweeps*

The first benefit from having a large set of CPUs is that you can execute a higher number of runs for a given experiment. Archaeological problems deal with a high degree of uncertainty, so a large percentage of models embody stochastic processes. The consequence is that we cannot analyze a simulation based on one run, as different executions provide different results; we need to execute several runs, and their number will increase depending on the degree of stochasticity of the model.

An HPC infrastructure minimizes the time needed to achieve this task but it is important to note that this problem has not been thoroughly explored, because the proper number of simulations is difficult to know; how many runs of a given scenario must be executed before being sure that the results are fully understood? A common technique is to compare the average and standard deviation of particular outcomes for different sets of the scenario (e.g. 10, 100 and 1,000 runs), but it still remains a tricky issue for most models. In any case this is a field that should be addressed in the near future.

If we want to grasp how the model is affected by different initial conditions we also need to execute parameter sweeps. The technique consists on exploring the combination of possible initial values of the simulation. This analysis allows the researcher to fully understand the importance of the parameters as well as the relation between them. If the number of parameters is large this could be an impossible task, as a correct analysis should explore the whole range of combinations; as a consequence the scientific quality of the model will decrease, because the effect of the different parameters in the outcome will not be fully understood.

This problem is even more important in archaeological models, as it is difficult to give realistic values for most parameters (given the mentioned uncertainty). Some models avoid this problem fitting real data to simulated data. The problem is that this is not always possible, because the solution requires high-quality archaeological data that usually is unavailable. As a result archaeology-related ABMs often suffer from the problem of having too many parameters that are not explored neither calibrated.

One of the possible solutions is the adoption of HPC. This would allow, at a technical level, to execute the needed number of simulations in order to perform sensitivity analysis using different techniques (i.e. genetic algorithms, see Stonedahl and Wilensky 2010). Nevertheless the difficulties to understand the outcome remain, as the analysis of the results will be extraordinarily complex, and every new parameter will increase the problem at an exponential rate. Besides, HPCs are

not designed to execute a large number of simple simulations. The cost of the interconnectivity between computer nodes is extremely high, and is not really used if every run is executed in a different node. In the end, a simple cluster or cloud-based services would have a better trade-off between cost and capabilities, being HPC best suited to face the following two issues.

6.4.2 *Size Dependent Emergence*

A quick glance at ABMs published in archaeological journals suggest that they tend to be small (for a survey of recent simulations see Lake 2013). The number of agents will seldom exceed the order of thousands, so the idea of having a large-scale simulation with zillions of agents do not seem related to the usual case studies. If a large process needs to be modeled it is easier to jump from one scale to a higher rank, thus adding a level of abstraction to the behavior of the agents. In the end the best practice is the choice of a resolution as lowest as possible; it will avoid the computational requirements and complications of huge simulations.

The pitfall of this approach is that some times the optimal spatial, temporal and behavioral resolutions do not fit well. Imagine the simulation of migration processes at a continental scale such as hominin dispersion from Africa. Models designed for this task have extreme low resolutions (hundreds of year per time step, thousands of kilometers per discrete spatial position). Given the temporal and spatial span it seems the best choice, and it has been adopted by several research works during the last decades (Mithen and Reed 2002; Nikitas and Nikita 2005; Hughes et al. 2007)

A closer look to the processes being modeled makes clear that the different resolutions do not fit: there is a gap between the choice of spatiotemporal scale and the scale of the behavior defined in the agents. On the one hand, space and time resolutions are chosen to model a process that is being developed at a huge scale. On the other hand, the behavior seems to be modeling a small-scale event (in the case of dispersion, the movement and reproduction of human groups). The truth is that behavior is not suited to the scale, because in reality during a time step any human would be able to move wherever he/she wants around the simulated world.

For example, in the cellular automata defined in the model *Stepping Out* (Mithen and Reed 2002, p. 436) a time step is defined as 250 years. During this time a cell can colonize its neighbors, being all of them triangles with sides of 322 km. Why a further cell cannot be accessed during this huge time interval? A cell at thousands of kilometers from the point of origin could be colonized; if this was not the case was for other reasons, not because it was physically impossible: this constraint is introduced in the model to deal with the divergence of the scales. In the end, a paradox emerges from the model: to understand the reasons why humans colonized entire continents we need to explore behavior at a small scale. If we want to correctly address the relation between behavior, spatial scale and time scale, we need to let the migration process emerge, instead of applying the behavior suitable for one scale to worlds defined at other scales while forcing artificial constraints.

A second issue related to the size of simulations arise from the concept of emergence, typical of ABMs: how can we model mechanisms that can only be understood for a large number of agents? For example, warfare tactics are strongly related to the size of an army. The tactics used in a given period are linked to the number of available soldiers that were deployed in a battlefield. Studies on battlefield archaeology of the eighteenth century (Rubio-Campillo et al. 2012) shows how linear warfare of this era cannot be correctly modeled while trying to simulate a few individuals. The reason is that the studied cultural traits (firing systems) were designed to be used with thousands of soldiers, given the lack of accuracy of existing fire weapons. Some of the hypotheses regarding command and control, and even individual behavior follow the same reasoning, as do other types of traits. In this context, we cannot simplify the simulation using fewer agents, as the behavior we want to explore is linked to the number of them that are interacting at a given time step.

Both issues (incorrect relation between scales and large-scale behaviors) can be included inside the general concept of size-dependent emergence. True enough, lots of emergent behaviors are detected in small worlds, but certain phenomena can only be observed in large-scale scenarios (Murphy 2011). HPC is essential in these cases, as we will need to create larger simulations, more costly in terms of computer power and more difficult to analyze if an HPC is not available.

6.4.3 *Solving the Dilemma of Self-fulfilling Prophecies*

The two previous topics proof that some models cannot be simplified or split in simpler simulations. As a consequence, the results can be more complex to understand and justify. Parallel to these issues there is another one that, in our opinion, seems to be the most important issue of state of the art archaeological simulations: the definition of behaviors.

The vast majority of published ABMs are based on the classic SugarScape model by Epstein and Axtell (1996). The model consists of a discrete world, defined as a finite bi-dimensional matrix, where a set of agents interact between them as well as with the environment. This behavior is defined as a list of rules; if one condition applies, a simple behavior is executed (e.g. if there is no food the agent will move to the adjacent place with more resources). In other words, the decision-making process of the agent is wired, as its choices are completely predefined. As a consequence it is incapable of finding solutions and reacting to conditions not devised by the modeler. This is a weakness of the methodology if we think that the interest on ABMs is precisely the emergence of large scale behaviors from the interaction of these simple rules. In theory these behaviors should not be explicitly defined in the original model, but as we said behavior is predefined, so there exist an important thread of circular explanations (Macal and North 2010).

Moreover, if everything in an ABM is related to the agent's behavior, why is it not analyzed like a parameter? Behavior is arbitrarily defined based on archaeological

assumptions, so it should also be explored to ensure scientific quality. Most models do not face this issue, so we are not really sure if different behaviors (even slight variations) affect the observed outcome as no sensitivity analysis is performed to study it.

The consequence of this problem is that critiques against ABM focus on the fact that they are self-fulfilling prophecies: emergent traits are not related to the problem, but to the way it was programmed, and if we can take a look at the code, we will learn the implicit or explicit assumptions the modeler introduced in order to achieve the final outcome.

The solution of this issue is critical for the future of ABM. For simple models it can be argued that the assumptions are less important, because such models can be replicated and understood without problems. Theory building models guarantee the scientific quality, as the number of parameters is small and behavior is so simple that the emergence of non-expected behavior can be understood with proper analysis.

The same cannot be said about hypothesis testing simulations. They are usually more complex because their goal is to understand realistic scenarios. For this reason these models create agents with several traits and a large list of conditions and rules, as simpler agents would not be able to take decisions based on the amount of data being used. In the end it will become impossible to understand, even by the creator of the model, which system properties emerged from the simulations, and which ones from the way the agents were programmed. Even though the problem is often not explicit, its consequences are so important that we have to wonder if hypothesis testing models are really useful to understand social change, except for some approaches that use excellent datasets and simple mechanisms, like evolutionary archaeology (Premo 2010; Lake 2013).

Any solution to this issue needs to avoid the design of the traditional rule-based agents. Luckily enough Artificial Intelligence has developed, in the past decades, several alternatives to the modeling of decision-making processes. The change would provide access to a plethora of well-known methods that could be integrated in the agents' design such as goal oriented agents based on atomic actions.

Imagine that we want to explore the foraging strategies of Hunter-Gatherers in a realistic landscape. Instead of defining every way in which an agent can interact with the environment, we could define its basic goal as survival (i.e. getting each time step a given level of food) and a set of simple actions that it can use to achieve the goal. The only possible actions would be moving to another location or forage the place where the agent currently is. Every action would have potential associated rewards and costs, and the decision-making process would consist on the choice of a single action each time step. The whole process can be seen as a fully observable stochastic state model, or Markov Decision Process (MDP). In this context, the agent needs to explore possible options and the impact on its future, thus choosing the best one given its knowledge of the environment. Figure 6.6 shows this process, where the agent explores its future applying different sets of actions.

The benefit of this approach is that MDPs are well known models of decision-making in the field of Artificial Intelligence. There are different ways to solve it, like the family of A* algorithms or the UCT approach (Bonet and Geffner 2012).

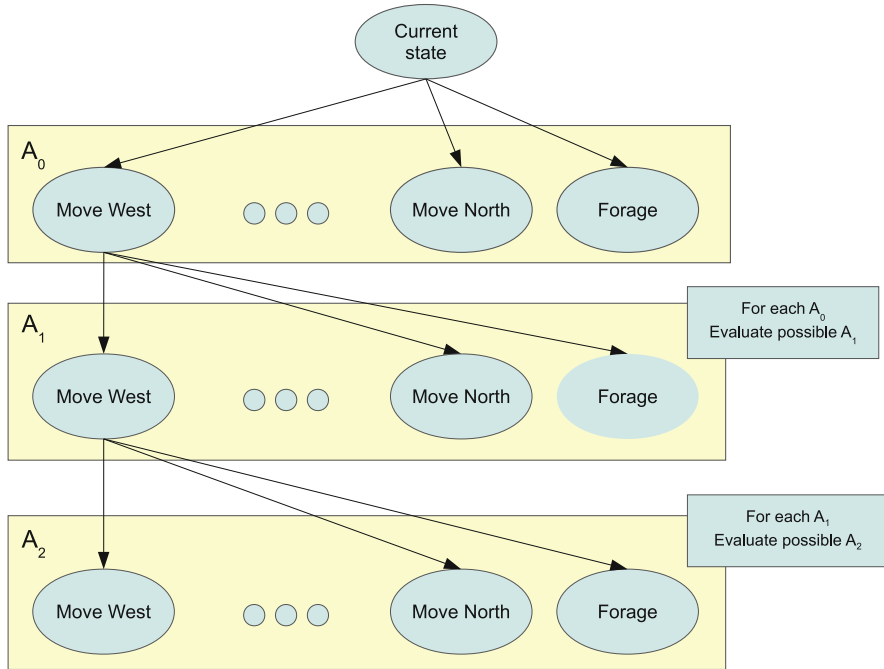


Fig. 6.6 Exploration of actions using a Markov Decision Process

This is a CPU consuming algorithm that acts as a planner for the agent: given a set of goals, the agent will simulate on its own the accumulated effects of executing different available actions. Thus, at every time step potential rewards and costs will be explored for a large set of possible actions, given a defined policy and a search depth. When the process is finished the agent will choose to execute the best action for the next state (when the algorithm will be executed again).

This change in the architecture of the agents solves the problem of implementing realistic scenarios at several levels:

1. Definition of the model. It is simpler to think on a small set of possible actions than a large set of rules. Moreover, if archaeological data is scarce these rules can not be defined at all (but we can still define the small actions).
2. Complex behavior without complex rules. Instead of defining rules the modeler defines goals and atomic actions. They are more understandable and for this reason the model can be best understood by archaeologists that did not implement the model.
3. Verification. The researcher has different available algorithms to solve the MDP. They are already understood and published by AI experts, so the modeler can use third-party implementations instead of developing new code.

4. Extension. The division of behavior between potential actions and decision-making process allows for the exploration of different approaches (e.g. what if we introduce a cognitive model, social interaction or partial information?).

To summarize this approach, in order to model realistic scenarios we need realistic decision-making processes. They have been researched by Artificial Intelligence during the last decades, so we should use their knowledge to create new types of agents, capable of acting in a realistic way while maintaining scientific quality. It is clear that this approach requires additional computing power, and HPCs are the key that will allow its introduction into archaeological ABMs.

6.5 Concluding Remarks

The use of simulation in science always require further computational power. Only high-performance computing (HPC) resources are capable of dealing with large simulation scenarios containing agents applying artificial intelligence algorithms with high computational costs. Even though the problems solved by archaeological ABMs are somewhat different than other fields, the capabilities of these systems can also be exploited. We have explored different options to accelerate the execution of the simulations, but also to improve the methodological framework and scientific quality of these models.

As we already know hardware infrastructures evolve at a fast pace; any software solution must be independent of the characteristics of the underlying system. On one side of the spectrum we find the supercomputers, where hundreds or thousands of nodes are located inside a facility with high-speed interconnection, forming a homogeneous and reliable environment. The increasing trend of cloud computing is present on the other side, where a theoretical unit of computing can, in reality, shape multiple and varied hardware infrastructures linked with medium or low-speed interconnections (Armbrust et al. 2009). The debate about which system is better is far from the aim of this chapter, but it is important to note that, from our perspective, they serve different purposes. Cloud computers are definitely needed when no supercomputing infrastructure is available. Even if a supercomputer is an option, cloud computers are more cheaper when the agents inside our model do not interact constantly. If the model is communication-intensive, on the other hand, a supercomputer is the best available choice, as its performance is optimal for this kind of executions (Wittek and Rubio-Campillo 2012a). The election of the platform used to execute an ABM will be increasingly difficult given the varied options, and the modellers should be able to choose based on the properties of their own simulations.

Additionally the constant improvement of available hardware components provides new ways for scientific simulation to exploit new computing capabilities. For example, Graphics Processor Units (GPUs) are increasingly being used to boost the performance of particular sections of a simulation (D'Souza et al. 2007). Even though it is unclear how small-scale ABMs could benefit from this approach, some examples have been suggested (Wittek and Rubio-Campillo 2013).

To conclude, this chapter provides an overview of the impact of HPC in archaeological ABMs. It is a decisive advance on the quality of the models focused on small-scale societies with a high demand for complex behaviors. The boost in computing capabilities provides solutions to different methodological issues derived from the use of this technique. In any case, we should be aware that, even if they are extremely powerful, they are just a tool; it is the responsibility of the researchers to develop models capable of exploiting these resources and solve interesting problems.

Acknowledgements Special thanks to Jose María Cela, Miguel Ramírez and two anonymous reviewers for their suggestions and comments on the topic and the preliminary versions of the text. The author is part of the SimulPast Project (CSD2010-00034) funded by the CONSOLIDER-INGENIO2010 program of the Spanish Ministry of Science and Innovation.

References

- Armbrust M, Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I (2009) Above the Clouds: A Berkeley View of Cloud Computing. Tech. Rep. UCB/EECS-2009-28, University of California, Berkeley
- Bonabeau E (2002) Agent-based modeling: methods and techniques for simulating human systems. *Proc Natl Acad Sci USA* 99(3):7280–7287
- Bonet B, Geffner H (2012) Action Selection for MDPs: Anytime AO* vs. UCT. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, pp 1749–1755
- Collier N, North M (2011) Repast HPC: A Platform for Large-Scale Agent-Based Modeling. In: Dubitzky W, Kurowski K, Schott B (eds) Large-Scale Computing Techniques for Complex System Simulations. Wiley, Hoboken, pp 81–110
- Conolly J, Lake M (2006) Geographical Information Systems in Archaeology. Cambridge University Press, Cambridge
- Costopoulos A, Lake MW (eds) (2010) Simulating Change: Archaeology into the Twenty-First Century. The University of Utah Press, Salt Lake City
- Crema E, Bevan A, Lake M (2010) A probabilistic framework for assessing spatio-temporal point patterns in the archaeological record. *J Archaeol Sci* 37(5):1118–1130
- Dagum L, Menon R (1998) OpenMP: an industry standard API for shared-memory programming. *Comput Sci Eng* 5(1):46–55
- Diamond J (2002) Life with the artificial Anasazi. *Nature* 419(6907):567–569
- Doran J (1999) Prospects for agent-based modelling in archaeology. *Archeologia e Calcolatori* 10:33–44
- Doran JE, Palmer M, Gilbert N, Mellars P (1994) The EOS Project: Modelling Upper Palaeolithic Social Change. In: Gilbert N, Doran J (eds) *Simulating Societies*. UCL Press, London, pp 195–221
- D’Souza RM, Lysenko M, Rahman K (2007) Sugarscape on Steroids: Simulating Over a Million Agents at Interactive Rates. In: Proceedings of the Agent2007 Conference
- Epstein JM, Axtell R (1996) *Growing Artificial Societies: Social Science from the Bottom Up*. Brookings Press/MIT Press, Washington/Cambridge/London
- Folk M, Cheng A, Yates K (1999) HDF5: A File Format and i/o Library for High Performance Computing Applications. In: Proceedings of the 12th Conference on Supercomputing, Portland
- Galán JM, Izquierdo LR, Izquierdo SS, Santos J, del Olmo R, López-Paredes A, Edmonds B (2009) Errors and artefacts in agent-based modelling. *J Artif Soc Soc Simul* 12(1):1. <http://jasss.soc.surrey.ac.uk/12/1/1.html>

- Gulyás L, Szabó A, Legéndi R, Máhr T, Bócsi R, Kampis G (2011) Tools for Large Scale (Distributed) Agent-Based Computational Experiments. In: Proceedings of the Computational Social Science Society of America Annual Conference 2011
- Hughes J, Haywood A, Mithen S, Sellwood B, Valdes P (2007) Investigating early hominin dispersal patterns: developing a framework for climate data integration. *J Hum Evol* 53: 465–474
- Kohler TA, van der Leeuw SE (eds) (2007) *The Model-Based Archaeology of Socionatural Systems*. School for Advanced Research Press, Santa Fe
- Kuhn B, Petersen P, O’Toole E (2000) OpenMP versus threading in C/C++. *Concurrency Pract Exp* 12(12):1165–1176
- Lake MW (2000) MAGICAL Computer Simulation of Mesolithic Foraging. In: Kohler TA, Gumerman GJ (eds) *Dynamics in Human and Primate Societies: Agent-Based Modelling of Social and Spatial Processes*. Oxford University Press, New York, pp 107–143
- Lake M (2001) Numerical Modelling in Archaeology. In: Brothwell D, Pollard A (eds) *Handbook of Archaeological Science*. Wiley, Chichester, pp 723–733
- Lake M (2013) Trends in archaeological simulation. *J Archaeol Method Theory*. doi:10.1007/s10816-013-9188-1
- Leitao P, Inden U, Rückemann CP (2013) Parallelising Multi-Agent Systems for High Performance Computing. In: INFOCOMP 2013: The Third International Conference on Advanced Communications and Computation, Portugal, pp 1–6
- Long Q, Lin J, Sun Z (2011) Agent scheduling model for adaptive dynamic load balancing in agent-based distributed simulations. *Simul Model Pract Theory* 19(4):1021–1034
- Luke S (2011) Multiagent Simulation and the MASON Library. George Mason University. <http://cs.gmu.edu/~eclab/projects/mason/manual.pdf>
- Macal C, North M (2010) Tutorial on agent-based modelling and simulation. *J Simul* 4(3):151–162
- Macal C, North M, Pieper G, Drugan C (2008) Agent-based modeling and simulation for exascale computing. *SciDAC Rev* 8:34–41. <http://www.scidacreview.org/0802/index.html>
- Mithen S, Reed M (2002) Stepping out: a computer simulation of hominid dispersal from Africa. *J Hum Evol* 43(4):433–462
- MPI Forum (2009) Message Passing Interface (MPI) Forum Home Page. <http://www.mpi-forum.org/>
- Murphy J (2011) Computational Social Science and High Performance Computing: A Case Study of a Simple Model at Large Scales. In: Proceedings of the 2011 Annual Conference of the Computational Social Science Society of America
- Nikitas P, Nikita E (2005) A study of hominin dispersal out of Africa using computer simulations. *J Hum Evol* 49:602–617
- North M, Howe T, Collier N, Vos J (2007) A Declarative Model Assembly Infrastructure for Verification and Validation. In: Takahashi S, Sallach D, Rouchier J (eds) *Advancing Social Simulation: The First World Congress*
- Premo LS (2010) Equifinality and Explanation: The Role of Agent-Based Modeling in Postpositivist Archaeology. In: Costopoulos A, Lake M (eds) *Simulating Change: Archaeology into the Twenty-First Century*. University of Utah Press, Salt Lake City, pp 28–37
- Rubio-Campillo X (2013) Pandora: An hpc Agent-Based Modelling Framework. Software. <https://github.com/xrubio/pandora>
- Rubio-Campillo X, María Cela J, Hernández Cardona F (2012) Simulating archaeologists? Using agent-based modelling to improve battlefield excavations. *J Archaeol Sci* 39:347–356
- Stonedahl F, Wilensky U (2010) Evolutionary Robustness Checking in the Artificial Anasazi Model. In: Proceedings of the 2010 AAAI Fall Symposium on Complex Adaptive Systems
- Sutton T, Dassau O, Sutton M, Nsibandé L, Mthombeni S (2009) Quantum GIS Geographic Information System. Open Source Geospatial Foundation, Quantum GIS Development Team
- Wilensky U (1999) NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/>

- Wittek P, Rubio-Campillo X (2012a) Military Reconstructive Simulation in the Cloud to Aid Battlefield Excavations. In: Proceedings of the 4th International Conference on Cloud Computing Technology and Science, pp 869–874
- Wittek P, Rubio-Campillo X (2012b) Scalable Agent-Based Modelling with Cloud HPC Resources for Social Simulations. In: Proceedings of the 4th International Conference on Cloud Computing Technology and Science, pp 355–362
- Wittek P, Rubio-Campillo X (2013) Social Simulations Accelerated: Large-Scale Agent-Based Modeling on a GPU Cluster. In: GPU Technology Conference, San Diego. Poster DD06. http://on-demand.gputechconf.com/gtc/2013/poster/pdf/P0197_PeterWittek.pdf