Evolutionary Robustness Checking in the Artificial Anasazi Model

Forrest Stonedahl and **Uri Wilensky**
Center for Connected Learning and Computer-Based Modeling
Northwestern University, Evanston, IL, USA
forrest@northwestern.edu and uri@northwestern.edu

Abstract

Using the well-known Artificial Anasazi simulation for a case study, we investigate the use of genetic algorithms (GAs) for performing two common tasks related to robustness checking of agent-based models: parameter calibration and sensitivity analysis. In the calibration task, we demonstrate that a GA approach is able to find parameters that are equally good or better at minimizing error versus historical data, compared to a previous factorial grid-based approach. The GA approach also allows us to explore a wider range of parameters and parameter settings. Previous univariate sensitivity analysis on the Artificial Anasazi model did not consider potentially complex/nonlinear interactions between parameters. With the GA-based approach, we perform multivariate sensitivity analysis to discover how greatly the model can diverge from historical data, while the parameters are constrained within a close range of previously calibrated values. We show that by varying multiple parameters within a 10% range, the model can produce dramatically and qualitatively different results, and further demonstrate the utility of sensitivity analysis for model testing, by the discovery of a small coding error. Through this case study, we discuss some of the issues that can arise with calibration and sensitivity analysis of agent-based models.

Motivation

Agent-based modeling[1] is a technique that is becoming increasingly popular for many scientific endeavors, due to the power it has to simulate complex adaptive systems in a variety of natural and social environments (Bankes 2002; Bryson, Ando, and Lehmann 2007; Goldstone and Janssen 2005; Wilensky and Rand in press). In an agent-based model (ABM), there are many agents operating according to simple rules, but the resulting interactions between agents lead to the emergence of complex aggregate-level behavior. The resulting aggregate behavior of an ABM (especially one that aims at high fidelity to real-world systems), is often dependent on a large number of controlling parameters. However, because of the complex nature of the emergent patterns, and the nonlinear interactions between these parameters, the outputs of ABMs can rarely be characterized by simple mathematical functions, and formal analytic methods usually prove insufficient (Edmonds and Bryson 2004). Furthermore, the computational time required to run an ABM, together with the large number of parameters often makes it infeasible to exhaustively compare all combinations of parameter settings. Additionally, ABMs are predominantly stochastic in nature, leading to variability of results, even when run multiple times with identical simulation parameters. As a result, the rigorous analysis of agent-based models remains a challenging task, and proper methodology for efficient analysis is still at a formative stage. In this work, we offer a case study about the use of one particular approach, genetic algorithms (GAs), to accomplish two common model analysis tasks: parameter calibration, and sensitivity analysis. For this case study, we chose to examine the *Artificial Anasazi* model (Dean et al. 2000), which is a well-known agent-based simulation from the field of archeology.

[1] Sometimes also referred to as *multi-agent modeling*, *multi-agent based simulation*, or *individual-based modeling*

Background and Related Work

Artificial Anasazi model background

The *Artificial Anasazi* model (Dean et al. 2000; Axtell et al. 2002; Gumerman et al. 2003) simulates the rise and fall of the prehistoric Kayenta Anasazi population living in Long House Valley, in northeastern Arizona from the years 800-1350 AD. This agent-based model simulated the residential and agricultural practices of an artificial society at the unit of individual households. It used geographic, rainfall, and various forms of archaeological survey data to achieve a high degree of verisimilitude with respect to historical reality. Moreover, after calibrating their model, the researchers found a reasonably good correspondence between the model and the real history, for both qualitative spatial settlement patterns, and population over time (Axtell et al. 2002).

A particular inspiration for the *Artificial Anasazi* model is to help understand the "fall." Archaeological records demonstrate that the Kayenta Anasazi abandoned the region around 1300 AD. However, the reason for this departure has been debated. One of the primary findings from the *Artificial Anasazi* model is that environmental factors alone were not sufficient reason for a complete exodus; the valley could have continued to support a modest population (Axtell et

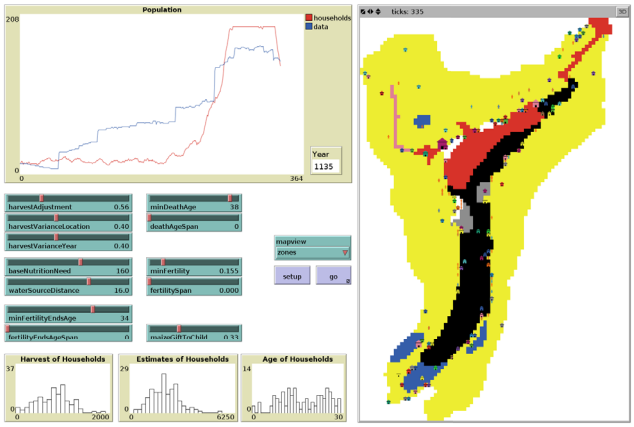


Figure 1: Graphical interface of Janssen's Artificial Anasazi model (implemented in NetLogo).

al. 2002). However, for a full discussion of the *Artificial Anasazi* model, we refer the reader to the original sources.

In this work, we analyze the replication of this model[2] by Janssen (2009), which was implemented in the NetLogo modeling environment (Wilensky 1999). The model parameters were hard-coded in Janssen's replication, so we converted these variables into "explicit" model parameters that are controllable via the model's graphical interface (see Figure 1), as well as making a few minor compatibility changes so the model would run in NetLogo 4.1.[3]

Several reasons motivate our choice of the *Artificial Anasazi* model for this case study. First, whereas many agent-based models are "abstract" models, that demonstrate qualitative trends or emergent phenomena, *Artificial Anasazi* is an example of a "facsimile" model in the typology of agent-based models, which attempts to closely match historical data (Gilbert 2008). Second, it is a particularly well-known ABM that has received considerable attention, both in the press (e.g. (Kohler, Gumerman, and Reynolds 2005)), and from the agent-based modeling community in general. Third, there have been several previous calibration efforts using this model (Dean et al. 2000; Janssen 2009), as well as published (univariate) sensitivity analysis (Janssen 2009). We will compare with these prior analyses as we discuss results in the sections below.

Kohler et al. (2000) developed a similar ABM of the nearby Mesa Verde region during this time period, and further elaboration of this model used the *Cultural Algorithm* framework to embed (and evolve) social intelligence within the system (Reynolds et al. 2005; Kobti, Reynolds, and Kohler 2006). Despite the commonality of using evolutionary algorithms, our work differs in that we are performing model analysis tasks externally to the model, rather than incorporating evolution as a mechanism within the ABM.

[2]Download available: `http://www.openabm.org/site/model-archive/ArtificialAnasazi`

[3]The exact model file we used is available at: `http://ccl.northwestern.edu/ALIAS/LHV_robustness.nlogo`.

Related methodological research

There are many ways of analyzing and checking for robustness in ABMs, including a variety of approaches for calibrating model parameters and performing sensitivity analysis. (Gilbert 2008; Wilensky and Rand in press; Chattoe, Saam, and Möhring 1997). However, our present work focuses on the use of genetic algorithms for these tasks.

This is not the first time that genetic algorithms have been proposed for parameter calibration and sensitivity analysis of computer simulations. In particular, Miller's (1998) seminal work on *active nonlinear testing (ANT)* proposed the use of metaheuristic search algorithms for a range of tasks for computer simulations. Specifically, Miller demonstrated how both calibration and a form of multivariate sensitivity analysis could be achieved on the well-known World3 system dynamics (SD) model (Meadows et al. 1974), using either genetic algorithms or a hill-climbing approach. SD models share several features with ABMs, such as nonlinear interactions between parameters. However, SD models tend to model change of aggregate (macro-level) quantities, whereas in ABMs macro-level dynamics emerge from interactions between agents at the micro-level. Additionally, SD models are often deterministic, whereas ABMs are almost always stochastic in nature, requiring the examination of a number of "trials" to evaluate the model's behavior. As we will discuss later, the stochasticity of model run results brings up several important questions about what it means to perform robustness checking on an ABM. Moreover, we believe that the concepts of *active nonlinear testing* deserve further investigation within the context of agent-based models of complex adaptive systems.

Little work has been done in this area, with a few notable exceptions. Calvez and Hutzler (2005) proposed the use of genetic algorithms for tuning the ABM parameters, and demonstrated several parameter tuning tasks on a model of ant foraging (Wilensky 1997). One of these tasks was an mock calibration task, which sought parameters that would yield model output closest to data which had already been generated by the model. In contrast, the *Artificial Anasazi* model represents a *real* calibration task, attempting to match real historical data. Other cases of using genetic algorithms to search the parameter-space of ABMs include: finding optimal seeding strategies for viral marketing in an social network ABM (Stonedahl, Rand, and Wilensky 2010), and discovering various forms of emergent collective behavior in flocking ABMs (Stonedahl and Wilensky 2010b). We are not aware of previous instances of using genetic algorithms to perform sensitivity analysis on an agent-based model.

Calibration Task

Task description & prior work

Broadly construed, the *calibration* of an ABM may refer to any process by which changes are made to the model or its model parameters, such that the behavior of the resulting model is closer to a desired behavior. In this paper, we will more narrowly define *calibration* to be the common case of searching for model parameter settings that yield output

that is closest to specified *reference pattern*. (We will assume that only the model's parameters may be varied, and the model's code is a fixed entity.) In the case of the *Artificial Anasazi* model, we are following two previous calibration efforts (Dean et al. 2000; Janssen 2009), though we will primarily compare with Janssen (2009) because differences could exist between Janssen's replication and the original model, and also because the original authors' calibration process was not well documented.

Both previous calibration efforts chose the target reference pattern to be the time-series of historical population data (number of households), and sought to minimize an error measure, which defined the "distance" between the simulated population history and the real population history. Following (Dean et al. 2000), we will denote the historical population data with a vector of length 550, X_t^h), where t is the number of years since 800 AD, and similarly denote simulated data with vector X_t^s.

Previous calibration efforts used multiple error measures of the difference between X_t^s and X_t^h, specifically the three L^p norms (L^1, L^2, and L^∞). However, prior work found little difference between the choices of error function, and Janssen (2009) specifically found that both the L^1 and L^2 measures yielded the exact same optimal calibrated settings. For simplicity our work focuses on the L^2 measure, which is also called the Euclidean distance between the vectors X_t^s and X_t^h. Furthermore, minimizing the L^2 measure yields the same result as minimizing the *mean squared error* when comparing two sequences (the absolute magnitude of the error measures will differ, but finding parameters that minimize $f(x)$ is equivalent to finding parameters that minimize $\sqrt{f(x)}$, for $f(x)$ positive).

Janssen (2009) used a factorial experiment (grid-based sweep) for performing the calibration. Due to computational constraints, Janssen varied only 5 parameters, with 7 to 9 choices for each parameter. In contrast, using a genetic algorithm (or other search-based) approach to calibration makes it feasible to explore a much larger parameter space. Our calibration effort explores a 12-dimensional parameter space, with a wider range of parameter values, and with higher resolution. For a comparison of the parameter calibration ranges we used with the prior calibration effort by Janssen, see Table 1. Of course, there is no magic bullet; the model can only be run so many times within a finite time limit. Given a the same amount of computational time, the GA approach can only run the model with the same number of different parameter-settings that the grid-based approach can. However, the GA is a heuristic method that can adaptively explore more advantageous portions of a larger parameter space. The intuition is that by harnessing the biologically-inspired mechanisms of mutation, recombination, and natural selection, the GA will be able to evolve parameter settings that minimize the error measure, and thus calibrate the model. Pragmatically, it is often infeasible to perform calibration with fine resolution on a medium-to-large number of parameters with a grid-based approach. For instance, an exhaustive grid-based search on the parameter space defined for the GA in Table 1 would involve 6.5×10^{16} combinations of parameters, and would require a million processors each running for over a million years to complete.

Parameter	**Janssen Range** low-high (inc)	**GA Range** low-high (inc)
HarvestAdjustment	0.54-0.7 (0.02)	0.5-1.5 (0.01)
HarvestVarianceLocation	0-0.7 (0.1)*	0-0.5 (0.01)
HarvestVarianceYear	0-0.7 (0.1)*	0-0.5 (0.01)
BaseNutritionNeed	160	100-200 (5)
MinDeathAge	26-40 (2)	26-40 (1)
DeathAgeSpan	0 (const)	0-10 (1)
MinFertilityEndsAge	26-40 (2)	26-40 (1)
FertilityEndsAgeSpan	0 (const)	0-10 (1)
MinFertility	.095-.185 (.015)	0.0-0.2 (0.01)
FertilitySpan	0 (const)	0-0.1 (0.01)
MaizeGiftToChild	0.33 (const)	0-0.5 (0.01)
WaterSourceDistance	16 (const)	6-24 (0.5)

**varied in lock-step, as a single variance parameter*

Table 1: Parameter ranges (low, high, and increment) for the GA calibration task, compared with ranges explored in a previous grid-based calibration by Janssen (2009).

Search Method

The GA we employed was a standard generational genetic algorithm (Holland 1975), with a population size of 30, a crossover rate of 0.7, and a mutation rate of 0.05, using tournament selection with tournament size 3.

The value to be assigned to each model parameter was individually encoded in binary using a Gray code.[4] The concatenation of binary sequences for all model parameters forms the genome for an individual in the GA.

Full generational replacement is used, meaning that from each generation of 30 individuals, 30 children are created to replace the parent generation. Each child is created by first using tournament selection to preferentially choose one or two parents with better fitness values, and then performing either sexual or asexual reproduction with the parent(s), followed by per-bit mutation.

To evaluate fitness, the individual is decoded into the component parameter values, the model is run 15 times with those parameters, and fitness function is calculated as the average L^2 error value from these replicate runs. During tournament selection, individuals with lower fitness function values (lower average error) are preferred. The choice to minimize the average 15 replicate runs follows from the previous calibration efforts (Axtell et al. 2002; Janssen 2009), although we also examine the alternative of searching for the single best run in a second follow-up calibration experiment.

To monitor/verify the progress of the GA, for each new "best-so-far" model parameter values that the GA found, an additional 30 independent replicate runs were performed and logged, providing an unbiased (and more confident) estimate of the average L^2 error for those parameter settings. We will refer to this process as *best-checking*, and the verified

[4]Gray codes create a smoother mapping between numeric values and binary strings than traditional "high-order" bit encodings.

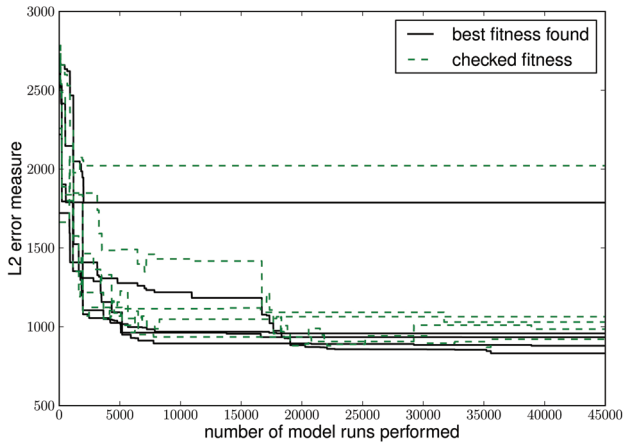


Figure 2: GA performance for the *calibration-15* task.

value as the *checked fitness*. (The GA does not make use of *checked fitness* information; rather, this monitoring is extrinsic to the search process.)

Our GA implementation employed BehaviorSearch[5], which is a tool we have developed that interfaces with NetLogo to automate the exploration of ABM parameter-spaces using genetic algorithms or other meta-heuristic search techniques (Stonedahl and Wilensky 2010a; 2010b).

Calibration15 experiment

Using the setup described above, we performed 5 GA searches for parameter settings that yield the best *average of* 15 *model runs*. We will refer to this as the *calibration-15* experiment. Each search went for 100 GA generations, corresponding to running the simulation a total of 45,000 times, with a small number of additional runs used for the extrinsic *best-checking* process. A single GA search required approximately 16.5% of the 272,160 runs required by the factorial-sweep approach employed by Janssen (2009), so the five searches together still required less computation than the grid-base approach. To provide an idea of computational running time, in total these searches required approximately 2500 CPU-hours ($\approx$ 104 CPU-days). Search time is dominated by the time required to run the model and the time spent on genetic operations is inconsequential. Thus, in this paper we will report computational effort in terms of the number of simulation runs performed.

An examination of search performance of the five *calibration-15* searches shows that one of the five prematurely converged to a suboptimal solution, whereas four of the five reached reasonably good levels of calibration (see Figure 2). The best parameter settings found from *calibration-15* experiment (as well as results from later experiments) are given in Table 2. These parameter settings yielded a mean L^2 error value of 891.4 ($\sigma = 65.8$) from running the model 30 times, which was lower than the mean L^2 error of 945.3 ($\sigma = 80.0$) for the Janssen calibrated settings. Both distributions of error appeared normally distributed (Shapiro-Wilkes test, $p < 0.01$), and the finding that the GA's mean error was less than for the Janssen settings appeared statistically significant (Student's t-test, $p < 0.01$). However, we happened to decide to run the simulation 100 times with each of these settings, and the picture suddenly changed.[6] With 100 replicate runs, the mean L^2 error for the GA parameters was 943.1 ($\sigma = 324.5$), and the mean L^2 error for the Janssen settings was 930.6 ($\sigma = 194.4$); the GA now appeared to have found worse (less calibrated) parameters.

This led us to examine the distribution of error among the 100-replicates for each case (see Figure 3), which turned out to be non-normal. In general, the GA-provided settings usually offer a (slightly) better match with historical data, but there are a few high-error outliers (that raise the mean error value), and these outliers appear more likely with the GA's settings than with Janssen's. These outliers are apparent in the visual comparison of the 100 GA and Janssen simulated histories against the historical data (Figure 4). The median L^2 error for the GA was 860.4, compared to 893.8 for the Janssen settings, and a randomly chosen run with the GA settings is almost twice as likely to have better performance than one chosen from the Janssen settings (65.9% vs. 34.1%).

The trade-off present here may be described in terms of confidence versus accuracy. Given three hypothetical choices, which of the following represents the *best-calibrated* parameter settings for an ABM?

1. simulated results are always somewhat close to historical
2. simulations are often quite close, but occasionally far off
3. simulated results occasionally match historical data perfectly, but are usually far off

Answering this question is difficult, particularly in facsimile-type models of historical events, since there is only one recorded version of history to compare against (and even for that, the data may be uncertain). We believe that this question warrants explicit consideration whenever a model calibration is performed, and that the choice of distributional comparison may require estimates of the likelihood of history having unfolded in the way that it did, and consideration of plausible alternative histories. For the most part, these estimates and theories will be subjective in nature, which is why it is especially important that they are explicitly addressed during the calibration process. The choice of distributional comparison for calibration will also depend partially on the goals for building the model.

In some cases, one distribution of error may *dominate* another, in the sense that every error value in one distribution is lower than some corresponding error value in the other distribution. In this situation, choosing the "better calibrated" settings is simple, and comparing the mean values is sufficient. However, we would like to emphasize that because

[5]Download available: `www.behaviorsearch.org`

[6]We include this vignette partially as a reminder that statistics must be interpreted with care, and that the distributions of variables resulting from multi-agent-based simulations may be irregular.

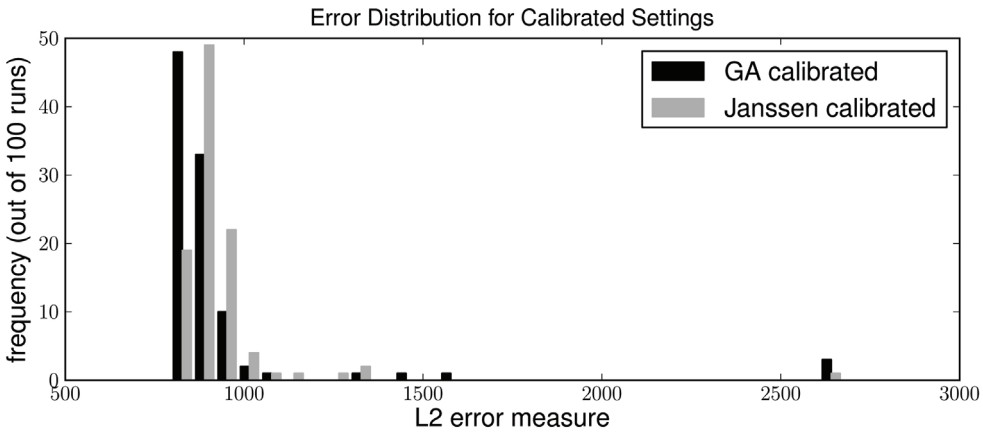
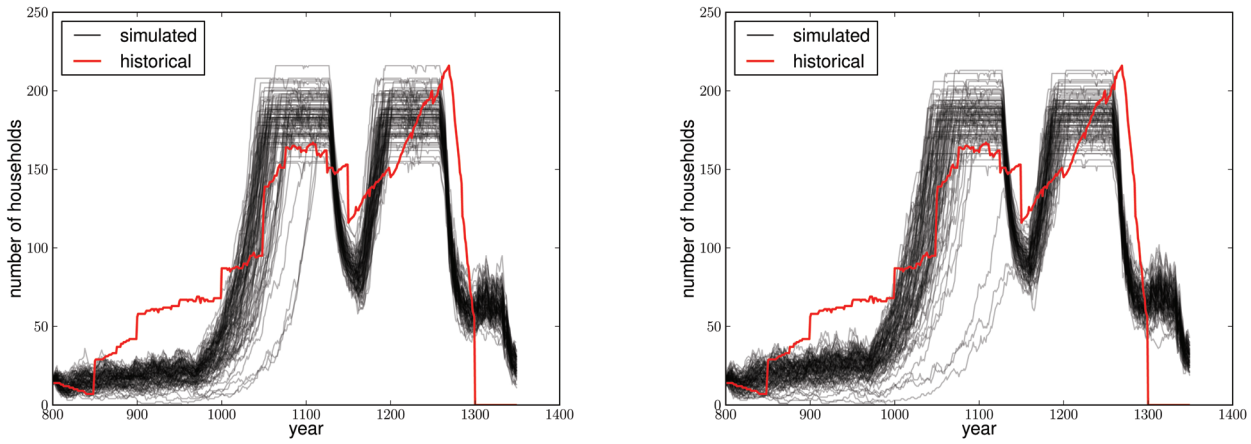


Figure 3: A histogram displaying the distribution of error values across multiple runs, comparing the GA calibrated settings with the calibrated settings previously found by Janssen (2009).



(a) *Janssen calibration*

(b) *GA: calibration-15*

Figure 4: Simulated population histories from 100 model runs, showing both Janssen's calibrated settings (a) and the GA's calibrated settings from the *calibration-15* experiment (b), plotted in comparison to the historical data. The flat tops of the simulated trajectories are artifacts of populations reaching simulated carrying capacity, as discussed further in (Janssen 2009).

of model stochasticity, calibrating ABMs requires comparing one distribution with another, rather than a single result. The issues we have preliminarily touched on here are part of a potentially much deeper discussion, which is outside the scope of this case study; in future work we plan to formulate a more rigorous and general framework for addressing both calibration and sensitivity analysis in ABM.

In the case of the Artificial Anasazi model, the GA's distribution of error seems slightly superior to us than Janssen's, given that it usually provides a closer match, and it seems reasonable that in some alternate histories an unlikely adverse chains of events (e.g., poor harvests for many years in succession) could have caused the population's trajectory to be significantly lower (as seen in Figure 4). However, the differences in error values are generally slight, and one could easily argue that both the GA's as Janssen's settings are equally calibrated; both recreate some features of the historical trajectory while failing to produce others. The fact that GA was searching a significantly wider range of parameters than Janssen's grid-based approach, yet was not able to find substantially better calibration, suggests that previous calibration efforts on this model were not missing important fruitful areas of the parameter space. However, as the 5 GA searches only covered a small region of the extremely vast search space, this conclusion is somewhat speculative.

Calibration-1 experiment

The results of the previous experiment led us to wonder how different the results of model calibration would be if we were instead seeking parameters that yielded the single best run, rather than the smallest average error. Investigating this is interesting for several reasons. First, it might discover settings that occasionally match the historical data, even if average error is poor. Second, running the model once is much quicker than running the model 15 times, and although it gives a noisier signal about calibration error, the GA might be able to use this faster noisier fitness function to lead to parameters that provide good average performance as well. Because the *calibration-1* experiment requires fewer model runs than the *calibration-15* experiment to evaluate fitness, we were able to increase our genetic algorithm settings to use a population of 90, running for 200 generations, for a total of 18000 simulation runs. We also increased the mutation rate to 3%, as a larger population can generally support a larger mutation rate. Similar to before, we used a *best-checking* routine, this time recording the minimum error from 30 independent replicate runs, each time the GA discovered a new "best." Again we ran 5 searches with these settings, to reduce the risk of reporting anomalous results.

We took the parameter settings corresponding to the lowest *checked fitness* L^2 error (see Table 2), and ran the simulation 100 times with those settings. The lowest L^2 error obtained from this was 733.6, which is substantially lower than the 823.5 error that was the best from the 100 runs with Janssen-calibrated settings. These single best runs are compared in Figure 5. However, the *average error* for these parameter settings was 962.4, which is somewhat larger than the mean error for Janssen or *calibration-15*. Essentially, the best *calibration-1* parameters cause more variation in model run results (compare Figure 6 with Figure 4), which can sometimes lead to a better historical fit, but provides a worse fit if averaged.

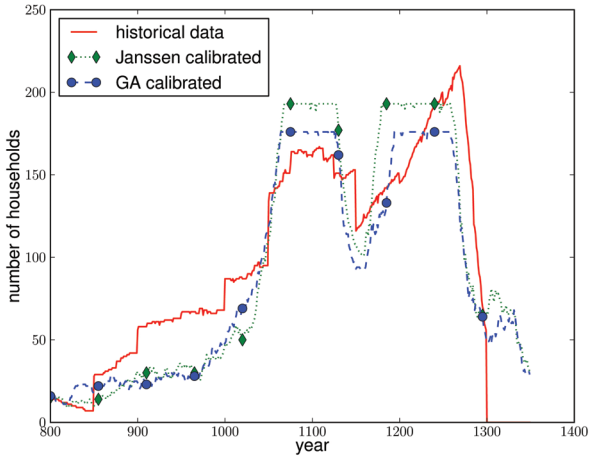


Figure 5: The single best runs found from 100 replicate runs with the settings from Janssen (L^2 error $= 823.5$) and the *calibration-1* experiment (L^2 error $= 733.6$), compared with historical data.

This contrast highlights a potential problem with calibrating to get the lowest average error. In order to obtain the absolute lowest average error, every model run would have to be identically equal to the historical data. In general, such a result would indicate a very unrealistic model, where only one path through history is possible. Over the past century, our increased recognition of chaos theory and the effects of path dependence in the social science domain (e.g., (Brown et al. 2005; Batty 2007)) strongly suggests that small changes in the initial conditions, or chance events early in the process, should significantly influence the historical trajectory. In other words, while a well-calibrated model should be able to produce something resembling the historical data, at least some variation in outcomes is a desirable trait for model credibility. Accordingly, one could argue that the *calibrate-1* experiment provides the best calibrated settings.

Sensitivity Analysis Task

Sensitivity analysis is a particularly important task, since the robustness (or lack of robustness) of a model with respect to changes in model parameters provides considerable information about the complex system being modeled. However, despite its importance, it is also a practice that is too often neglected by ABM practitioners; if it is performed at all, it often covers only a few parameters, or neglects potentially nonlinear interactions between parameters. Some form of sensitivity analysis is a necessary part of ABM verification and validation (Gilbert 2008), as well as replication (Wilensky and Rand 2007). However, the term "sensitivity analysis", does not refer to a single precise technique or

Parameter	Janssen calibration	GA calibration-15	GA calibration-1	GA sensitivity-15	GA sensitivity-corr
HarvestAdjustment	0.56	0.67	0.64	0.6104	0.5264
HarvestVarianceLocation	0.4	0.47	0.44	0.436	0.436
HarvestVarianceYear	0.4	0.23	0.5	0.424	0.408
BaseNutritionNeed	160	200	185	144	164.8
MinDeathAge	38	37	40	40	41
DeathAgeSpan	0	3	10	1	1
MinFertilityEndsAge	34	36	29	37	31
FertilityEndsAgeSpan	0	9	5	3	0
MinFertility	0.155	0.13	0.17	0.16585	0.14105
FertilitySpan	0	0.09	0.03	0.0155	0.0031
MaizeGiftToChild	0.33	0.31	0.47	0.3102	0.35310
WaterSourceDistance	16	10	11.5	17.44	16

Table 2: Parameter ranges (low, high, and increment) for the GA calibration task, compared with ranges explored in a previous grid-based calibration by Janssen (2009).

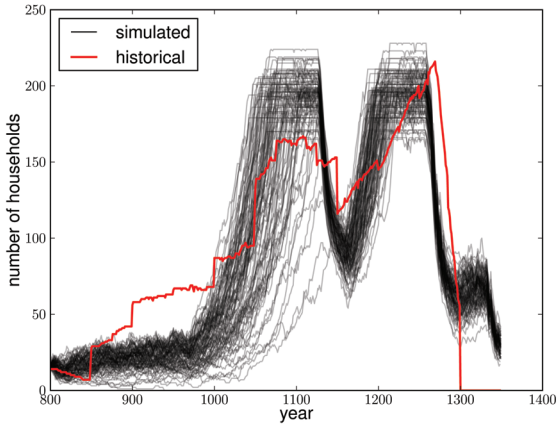


Figure 6: Simulated population histories from 100 model runs with the best *calibration-1* parameters, plotted against historical data.

methodology; rather, the term is broadly applied to class of related techniques that share the goal of determining what factors cause model results to change, and with what magnitude (Chattoe, Saam, and Möhring 1997). In this paper, we focus only on the specific approach of varying model parameters in the vicinity of some "default" parameter settings. In the case of the Artificial Anasazi model, a partial univariate sensitivity analysis has already been performed. Specifically, Janssen (2009) examined the effect of singly varying each of the five variable parameters from their calibration (HarvestAdjustment, HarvestVariance, MinDeathAge, MinFertilityEndsAge, MinFertility) while holding all other parameters constant (fixed at the previously calibrated values). While this approach does provide insight into the model dynamics near the calibrated point, we are interested in the related question of how robust the model is to changes in multiple parameters simultaneously. Specifically, if model parameters are each constrained to be within a relatively small range of the calibrated values, how far "off" can the model's output be? Exploring this question is one form of multivariate sensitivity analysis, as discussed in Miller's (1998) work on Active Nonlinear Testing. Similar to Janssen's calibration approach, a grid-based factorial parameter-sweep could be employed for small numbers of parameters being swept at low-resolution. However, again we propose an alternative approach of using a genetic algorithm to evolve parameter settings that yield results that are significantly different from the model's desired outcome (i.e. the historical data).

Sensitivity-15 experiment

Our first sensitivity analysis experiment was to search for parameter settings, within a small margin of the calibrated settings from Janssen (2009), that would yield the highest average L^2 error measure across 15 runs. Following Miller (1998), we chose to allow each parameter to range within $\pm 10\%$ of its calibrated value. Notice that we only have to change two small things in order to switch from performing model calibration to sensitivity analysis: we restrict the search space to a narrower range for each parameter, and we attempt to maximize (rather than minimize) the same error function (L^2 distance) used for calibration.

Mirroring the *calibrate-15* experiment, we used the same GA settings, and performed 5 searches, each of which ran the model a total of 45000 times[7]. All five of these searches found parameter settings yielding L^2 error values that were more than 4 times greater than the calibrated Janssen settings error (930.6). For the best settings found (again, listed in Table 2), the average L^2 error was 3918.6 ($\sigma = 249.7$); Figure 7(a) visually displays 100 simulated histories with these settings. While our experiment differs in flavor from that of Janssen (2009), it is still instructive to compare our results with that of the univariate sensitivity analysis previously performed. Specifically, we note that when varying each of 5 parameters singly, the highest relative L^2 error gain was 50% (within the $\pm 10\%$ parameter range), and even the sum of the highest errors for each parameter is only around 150%, which is still small compared with the

[7]However, running time in hours was over 80% longer, as these runs tended to create a much greater number of agents

$> 300\%$ increase in error discovered through the GA's multivariate search. This disparity is due in part to the GA manipulating more parameters to which the model is sensitive (such as BaseNutritionNeed), and also to the nonlinear interactions between parameters.

Figure 8 displays the distribution of best parameter values found by the GA in each of the 5 searches that cause such a dramatic discrepancy from historical data. The different GA searches sometimes found different settings from one another, but there are still some clear trends. In particular, they consistently discovered high values for HarvestAdjustment, HarvestVarianceLocation, MinFertility, and MinFertilityEndsAge, while they unanimously selected the lowest possible BaseNutritionNeed value in the range. In other words, the model is particularly sensitive to these parameters. For the most part, these parameter settings match our intuitions. In order to achieve the an extremely large population, there should be more bountiful harvests, a higher reproduction rate for creating households, and low nutritional requirements per household. The other parameters' values are relatively scattered throughout the range, and it is apparent that it is not necessary for them to be assigned a specific value in order to achieve large error.

There was, however, a curious trend regarding the two HarvestVarianceX parameters, which raised two questions:

1. Why does more variation in the crop yield from different fields (HarvestVarianceLocation) result in larger populations?
2. Why is yield variation over time (HarvestVarianceYear) not similarly correlated?

Addressing question 1, we first confirmed this was not a fluke by running the model 100 times with the best *sensitivity-15* settings, except using the lowest HarvestVarianceLocation value in the $\pm 10\%$ range (0.36), and we found a more than 10% decline in L^2 error (t-test, $p < 0.01$). Next, we examined the model code, and discovered that the HarvestVarianceLocation was affecting agricultural quality as the variance of a normal distribution centered around 1.0, but that agricultural value was not allowed to be negative, so was thus truncated at 0. As a result, increasing the variance also increases the distribution's mean value. The relevant excerpt from the NetLogo model code is as follows:

```
ask patches [
  ;...
  set quality ((random-normal 0 1)
      * harvestVarianceLocation) + 1.0
  if (quality < 0) [set quality 0]
]
```

This explains question 1 from above, and it stems from a reasonable modeling choice, although the outcome shows that one must take care in the interpretation of model parameters. To answer question 2, we looked for where (HarvestVarianceYear) was used in the code, only to find that it wasn't. Instead, HarvestVarianceLocation was *also* affecting variation over time; whereas HarvestVarianceYear was initialized and then never referred to again. This was clearly a bug in the Artificial Anasazi model,[8] which we had uncovered as a result of performing this sensitivity analysis. Admittedly, a careful code audit, or other forms of analysis, could also have helped find this bug. Nonetheless, our GA-based multivariate sensitivity analysis provided the information that led to the discovery of the bug in this published model, which lends further support for the utility of this approach.

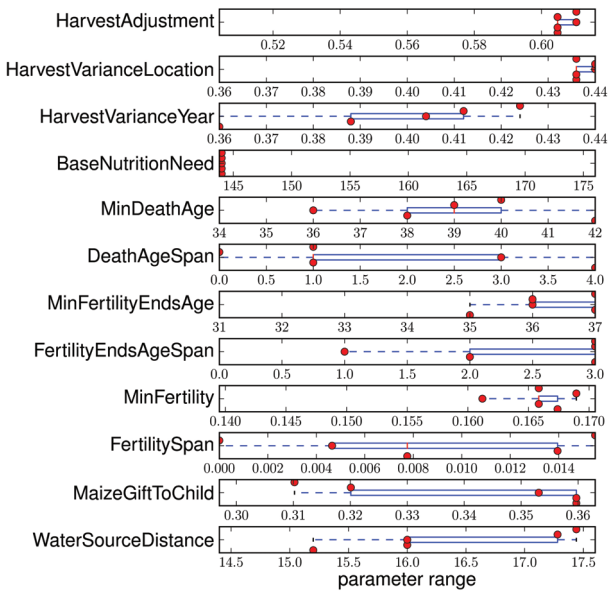
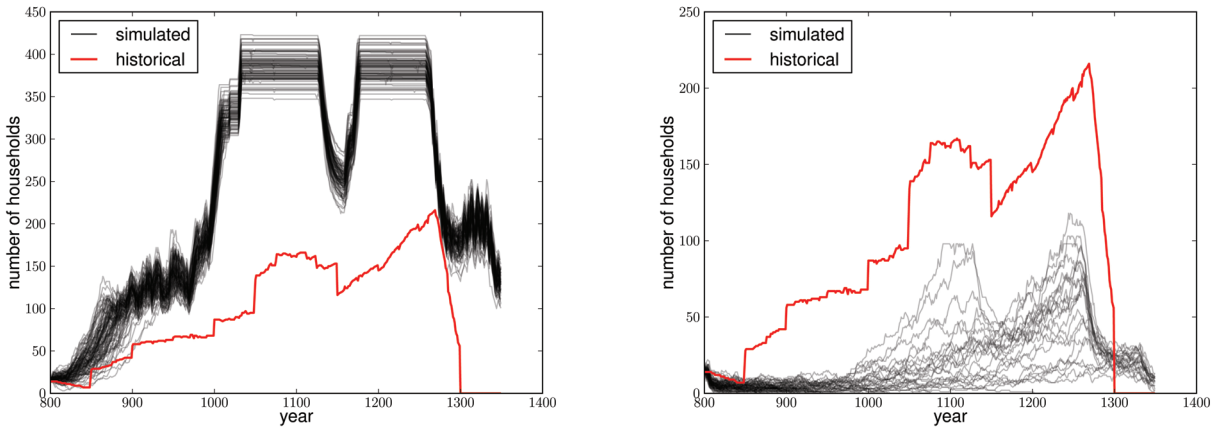


Figure 8: Distribution of "best" parameter settings found in each of the 5 GA searches of the *sensitivity-15* experiment. Actual parameter values are displayed as solid circles, while the boxes and whiskers display the middle 3 runs, and full extent of the data, respectively. The center x-value in each plot corresponds to the Janssen calibrated settings.

From the results, it seems possible that it would be sufficient to only test the extreme settings ($+10\%$, and -10%), rather than checking all values in between. With 12 parameters, this would only require $2^{12} = 4196$ combinations of parameter settings, which is a feasible number to enumerate. This may often be the case, but in general one cannot be sure that nonlinear interactions between parameters would not cause the optimal/extreme results to fall elsewhere in the viable range. For models with very large numbers of parameters, and small viable ranges for each parameter, allowing only 2 or 3 choices for each parameter may be prudent, together with a genetic algorithm approach.

We also performed a *sensitivity-1* experiment, using similar settings as the *calibration-1* experiment, searching for parameters that would cause the largest L^2 error for a single model run. However, the results were very similar to the *sensitivity-15* experiment, and are thus omitted for the sake of brevity.

[8] We reported this issue in personal correspondence with the model author. We also note that this minor error did not affect any of the results previously obtained in (Janssen 2009).



(a) *sensitivity-15* (b) *sensitivity-corr*

Figure 7: Simulated histories from 100 runs with the best sensitivity experiment settings, compared with historical data.

Sensitivity-corr experiment

Although the *sensitivity-15* experiment produced results of a different quantitative magnitude than results from calibrated values, they were still qualitatively similar (see Figure 7(a)). We were interested in whether we could use a different error measure for a sensitivity analysis to find simulated histories with a different general shape. As a measure for qualitative difference, we chose the Pearson product-moment correlation coefficient (r) between the simulated (X_t^s) and historical (X_t^h) population sequences. As an example, the single run with the largest L^2 error value (4524.3) from the *sensitivity-15* experiment still had a quite high positive correlation ($r = 0.83$) compared with the historical data.

Using a genetic algorithm with the same settings as the *sensitivity-1* experiment (population 90, 200 generations, 3% mutation), we ran 5 searches for parameters (within the $\pm 10\%$ range) that would yield the smallest correlation coefficient (r) value. The best (lowest correlation) parameter settings are listed in Table 2, yielding an average correlation of $r = -0.18$. Whereas the largest L^2 error measure was achieved by an unrealistically large Anasazi population, the smallest correlation was achieved by population decline and extinction, which are also consistently achievable within the $\pm 10\%$ range of calibrated values. Of 100 runs (shown in Figure 7(b)) using the best parameters for non-correlation, the lowest correlation for a single run was -0.6, which had a relatively long lingering decline with the population reaching 0 in the year 994 AD. Interestingly, because of our chosen measure, slow population declines cause greater negative correlation with the data than when the population dies out almost immediately. This led the GA to find runs that were on the brink of extinction, and thus out of the 100 runs, there are a few runs that are still highly correlated with the historical data (the closest matches in 7(b)). Though the Pearson correlation-coefficient was reasonably effective in this case for finding qualitatively different runs, it is worth emphasizing that it may not always be appropriate. Developing a variety of error measures for search-based sensitivity analysis that correspond well with human intuitions about what constitutes qualitatively different behavior of a system is a ripe area for future work.

Conclusions

To summarize, we have presented a series of 5 experiments using genetic algorithms to perform tasks relating to ABM calibration and sensitivity. In the calibration tasks, we demonstrated that the genetic algorithm could find calibrated parameters that were better (in some respects) than parameters previously discovered in a grid-based sweep. This process brought up important aspects of calibration (judging distributions of error, rather than simply mean error), which researchers should attend to during model analysis. In the sensitivity tasks, we demonstrated that the genetic algorithm approach can consistently find parameter settings that yield both dramatically and qualitatively different results. Additionally, the multivariate sensitivity analysis highlighted several instances of anomalous model behavior, leading us to discover a bug in the Artificial Anasazi model's code. This emphasizes the utility of sensitivity analysis as a technique for model testing and verification. Several of the general issues about search-based robustness-checking that arose as a result of this case study deserve further consideration beyond the preliminary discussion we presented here; in future work we plan to develop a methodological framework which will discuss trade-offs between different error measures, distributional comparisons, spatial and temporal data, and approaches for analysis. Future work should also include a comparison of genetic algorithms with other metaheuristic search algorithms (e.g. hill climbing, particle swarm optimization), for ABM robustness checking tasks. However, the results of the present study are both thought-provoking and promising, and it is our hope that ABM practitioners will adopt methods like these to improve the rigor of model analysis.

Acknowledgments We gratefully acknowledge support from the National Science Foundation (grant IIS-0713619), and we thank Northwestern's *Quest* high performance computing cluster for providing computational resources.

References

Axtell, R.; Epstein, J.; Dean, J.; Gumerman, G.; Swedlund, A.; Harburger, J.; Chakravarty, S.; Hammond, R.; Parker, J.; and Parker, M. 2002. Population growth and collapse in a multiagent model of the Kayenta Anasazi in Long House Valley. *Proceedings of the National Academy of Sciences* 99(Suppl 3):7275.

Bankes, S. 2002. Agent-Based Modeling: A Revolution? *PNAS* 99(10):7199–7200.

Batty, M. 2007. *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals*. The MIT Press.

Brown, D.; Page, S.; Riolo, R.; Zellner, M.; and Rand, W. 2005. Path dependence and the validation of agent-based spatial models of land use. *International Journal of Geographical Information Science* 19(2):153–174.

Bryson, J. J.; Ando, Y.; and Lehmann, H. 2007. Agent-based modelling as scientific method: a case study analysing primate social behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences* 362(1485):1685–1698.

Calvez, B., and Hutzler, G. 2005. Automatic Tuning of Agent-Based Models Using Genetic Algorithms. In *MABS 2005: Proceedings of the 6th International Workshop on Multi-Agent-Based Simulation*.

Chattoe, E.; Saam, N.; and Möhring, M. 1997. Sensitivity analysis in the social sciences: problems and prospects. In Troitzsch, K.; Gilbert, N.; and Suleiman, R., eds., *Tools and Techniques for Social Science Simulation*. New York: Physica-Verlag. 273.

Dean, J. S.; Gumerman, G. J.; Epstein, J. M.; Axtell, R. L.; Swedlund, A. C.; Parker, M. T.; and McCarroll, S. 2000. Understanding anasazi culture change through agent-based modeling. In Kohler, T., and Gumerman, G., eds., *Dynamics in human and primate societies: agent-based modeling of social and spatial processes*. Oxford, UK: Oxford University Press. 179–205.

Edmonds, B., and Bryson, J. J. 2004. The insufficiency of formal design methods " the necessity of an experimental approach - for the understanding and control of complex mas. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 938–945. Washington, DC, USA: IEEE Computer Society.

Gilbert, G. 2008. *Agent-based models*. Sage Publications, Inc.

Goldstone, R., and Janssen, M. 2005. Computational models of collective behavior. *Trends in Cognitive Sciences* 9(9):424–430.

Gumerman, G.; Swedlund, A.; Dean, J.; and Epstein, J. 2003. The evolution of social behavior in the prehistoric American Southwest. *Artificial life* 9(4):435–444.

Holland, J. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

Janssen, M. A. 2009. Understanding artificial anasazi. *Journal of Artificial Societies and Social Simulation* 12(4):13.

Kobti, Z.; Reynolds, R.; and Kohler, T. 2006. The emergence of social network hierarchy using cultural algorithms. *International Journal of Artificial Intelligence Tools* 15(6):963–978.

Kohler, T.; Kresl, J.; van West, C.; Carr, E.; and Wilshusen, R. 2000. Be there then: a modeling approach to settlement determinants and spatial efficiency among late ancestral pueblo populations of the Mesa Verde region, US southwest. In Kohler, T., and Gumerman, G., eds., *Dynamics in human and primate societies*. 145–178.

Kohler, T.; Gumerman, G.; and Reynolds, R. 2005. Simulating ancient societies. *Scientific American* 293(1):67–73.

Meadows, D.; Behrens, W.; Meadows, D.; Naill, R.; Randers, J.; and Zahn, E. 1974. *Dynamics of growth in a finite world*. Wright-Allen Press Cambridge, MA.

Miller, J. H. 1998. Active nonlinear tests (ANTs) of complex simulation models. *Management Science* 44(6):820–830.

Reynolds, R.; Kobti, Z.; Kohler, T.; and Yap, L. 2005. Unraveling ancient mysteries: reimagining the past using evolutionary computation in a complex gaming environment. *IEEE transactions on evolutionary computation* 9(6):707.

Stonedahl, F., and Wilensky, U. 2010a. BehaviorSearch [computer software]. Center for Connected Learning and Computer Based Modeling, Northwestern University, Evanston, IL. Available online: `http://www.behaviorsearch.org/`.

Stonedahl, F., and Wilensky, U. 2010b. Finding forms of flocking: Evolutionary search in abm parameter-spaces. In *Proceedings of the MABS workshop at the ninth international conference on Autonomous Agents and Multi-Agent Systems*.

Stonedahl, F.; Rand, W.; and Wilensky, U. 2010. Evolving viral marketing strategies. In *GECCO '10: Proceedings of the 12th annual conference on genetic and evolutionary computation*. New York, NY, USA: ACM.

Wilensky, U., and Rand, W. 2007. Making models match: Replicating an agent-based model. *Journal of Artificial Societies and Social Simulation* 10(4):2.

Wilensky, U., and Rand, W. in press. *An introduction to agent-based modeling: Modeling natural, social and engineered complex systems with NetLogo*. Cambridge, MA: MIT Press.

Wilensky, U. 1997. NetLogo Ants model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U. 1999. *NetLogo*. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-based Modeling.