Evolutionary Computation Applied to Agent-Based Simulation Modeling of Climate and Social Dynamics

Jeffrey K. Bassett and Claudio Cioffi-Revilla

Mason-Smithsonian Joint Project on Climate Change and Society
Center for Social Complexity, Krasnow Institute for Advanced Study
George Mason University, Fairfax, Virginia 22314, USA
{jbasset1,ccioffi}@gmu.edu
https://cs.gmu.edu/~eclab/projects/cdi/

Abstract. Detailed population distribution data are often unavailable for building spatial agent-based models of climate change effects on humans, especially for past, historical data. A key challenge is to generate approximations of historical (or even future) populations, to initialize our models, based on certain spatial qualities of the landscape. We use evolutionary algorithms (EAs) to tune a relatively simple "placement algorithm" or "settlement algorithm." The EAs generate modern settlements that match LandScan data. When generating historical populations, we include as much information as is available, and then let our algorithm generate missing parts of the spatio-temporal distribution. We illustrate this procedure based on evolutionary computation with the NorthLands model of climate change and social dynamics, using MASON and ECJ. Results are positive and encouraging, highlighting additional research directions.

A critical task in developing viable, empirically-based agent-based models (ABM) of complex systems with coupled human, artificial, and natural (CHAN) component subsystems is the specification of initial population distributions. Theory, observation, and experience can provide significant help, but in most cases finding proper population distribution values is a hard, non-trivial task in ABM research. This is especially so for ABMs on climate change and societal dynamics, where spatial and temporal scales are relatively large and theory and observations on population distributions are incomplete.

This paper presents a novel methodological procedure for obtaining population distributions in ABMs of CHAN systems using genetic algorithms (GA) from evolutionary computation (EC). The GA-based procedure is illustrated with the recent MASON NorthLands ABM, created to analyze climate change scenarios and societal consequences (Cioffi et al. 2015). Results show the advantage of our GA-based procedure over other approaches, such as manual tuning.

The next section provides and introduction with motivation and background on earlier related research, followed by sections describing, experimenting with,

and discussing our proposed GA-based procedure. The final section provides a summary.

1 Introduction

1.1 Motivation: Research Questions

Consider the goal of developing a viable, empirically-calibrated agent-based model (ABM) of a complex system with coupled human, artificial, and natural (CHAN) component subsystems, such as one or more geographic regions of the Earth system. How are initial population distributions determined when extant theory, data, or other sources are insufficient or unavailable? Which procedures can provide valid answers to such questions? How can the procedures be tested, demonstrated, and improved? Research questions such as these are critical for the task of determining proper population distributions in ABMs of CHAN systems used for analyzing and understanding impacts of climate change on society as well as the natural and built environments.

As pointed out elsewhere, "computational simulation modeling provides a viable scientific methodology, specifically through geospatial agent-based models (Cioffi, 2014: ch.10; Heppenstall et al., 2012; Railsback and Grimm, 2012). Such models combine a set of features, such as: (1) ability to selectively represent all empirical entities of interest (social, artificial, and natural) as computational objects endowed with (i.e., encapsulating) attribute-variables necessary to determine the state of each entity (overcoming the challenge of high dimensionality); (2) ability to model all necessary spatio-temporal features, such as weather, landscapes, and human activity that co-evolve over time (overcoming fragmentation in traditional disciplinary models); (3) ability to implement relevant systems and processes directly informed by social and biophysical theories (leveraging all necessary disciplinary knowledge within a unified framework); and (4) ability to manipulate variables and change scenarios, including at run-time, for conducting virtual experiments that yield empirically valid results (enabling experimental science *in silico*)"(Cioffi et al., 2015: 2).

MASON (Multi-Agent Simulator Of Neighborhoods) (Luke et al. 2005) is a Java toolkit for building ABMs, while ECJ (Luke 2010) is a highly configurable Java toolkit that can thus be used to create a variety of different Evolutionary Algorithms. Further motivation is provided by the opportunity to use MASON and ECJ in combination, for which both systems are optimized (Cioffi, De Jong & Bassett 2012).

1.2 Relevant Literature

Recent comprehensive reviews of the literature on spatial agent-based social simulation models that focus on socio-natural and socio-engineered systems (i.e., the most relevant class of models) include An et al. (2014) and Cioffi (2015), in addition to Batty (2013), Heppenstall et al. (2012), Kohler and van der Leeuw

(2007), Liu et al. (2007), and Railsback and Grimm (2012). The most common strategy for populating a spatial ABM at initialization is to assign agents to random locations and let the burn-in phase at run time determine where they will be subsequently distributed, based on the model's dynamics. Another strategy is to use a known empirical distribution, such as provided by census data, but for simulations extending into the past this is not a viable solution. Neither of these strategies works well for models that have large spatial and temporal scales, because they either take too long to run (former case) or information for initialization is unavailable (latter case).

2 Proposed Procedure

This section describes our proposed general procedure, prior to presenting illustrative results in the next section. The main goal is to obtain viable estimates of population distributions for use in a geospatial agent-based model. By "viable" we mean population values that have internal and external validity (i.e., accuracy and empirical correspondence, respectively) as well as reliability (measurement consistency over space and time). First, we describe the algorithm for initial settlement nucleation by agent placement, followed by the evolutionary algorithm (EA).

2.1 Agent Placement/Settlement Nucleation Algorithm

During initialization, agents are placed on the ABM's "map" a few at a time, based on certain spatial features that exist. As agents are placed, subsequent agents decisions are affected by decisions made by earlier agents. In other words, agents will tend to be attracted to locations that have already been settled, thus growing towns and cities.

The MASON modeling library allows for a number of map layers to be defined. These layers contain geospatial information in the form of grids that can be displayed independently or together. The data from these layers can be accessed by agent, or other parts of the program. Several map layers define the features that affect settlement. The specific layers we use are: *elevation* above sea level, proximity to *fresh water* (i.e., large rivers and lakes), proximity to potential *ports*, and *temperature* averaged over a 50 year period. These and other features described below are ethnographically appropriate (i.e., informed by cultural anthropology), given the target system and research questions addressed by the model; different features apply to other cases. In this case they represent features that actors deem significant in their lives, and relate to growing seasons, transportation, and trade, among other human activities.

The locations of previously settled agents also affects current and future settlement decisions. These are modeled as two layers. The first defines the locations of the already settled population, and the second is derived from the first, and adds a diffusion effect, making it attractive to settle in locations that neighbor locations that are populated. The diffusion grid is generated by iterating through

each cells in the population grid, retrieving the population value for the given cell, and then adding this value to all the cells in the diffusion grid that make up the Moore neighborhood of the corresponding cell.

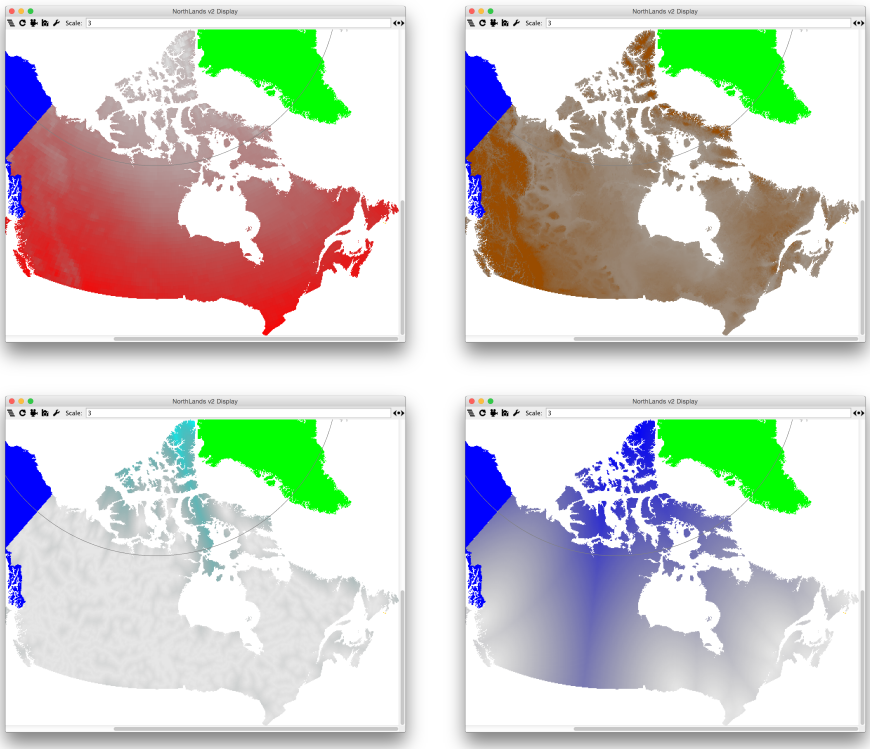

Fig. 1. Spatial distribution of desirability by average temperature (upper left), elevation (upper right), fresh water (lower left), and ports (lower right).

Accordingly, six spatial layers affect settlement decisions. Four represent *natural qualities* or physical features of a given location: temperature ($\boldsymbol{t}$), elevation ($\boldsymbol{e}$), proximity to fresh water ($\boldsymbol{w}$), and proximity to a port ($\boldsymbol{p}$), as shown in Figure 1. Two more spatial layers represent *social qualities*: settled population ($\boldsymbol{s}$) and population diffusion ($\boldsymbol{f}$). Each of these six variables represents a vector containing the appropriate values for all the cells in Canada. These are combined to form another layer called desirability, using the following formulas:

$$\boldsymbol{n} = c_t z(\boldsymbol{t}) + c_e z(\boldsymbol{e}) + c_w z(\boldsymbol{w}) + c_p z(\boldsymbol{p}) \tag{1}$$

$$\boldsymbol{n'} = normalize(\boldsymbol{n}) \tag{2}$$

$$\boldsymbol{a} = c_s \boldsymbol{s} + c_f c_s \boldsymbol{f} \tag{3}$$

$$\boldsymbol{d} = \begin{bmatrix} (n'_1)^{c_x} + a_1 \\ (n'_2)^{c_x} + a_2 \\ \vdots \\ (n'_m)^{c_x} + a_m \end{bmatrix} \tag{4}$$

Here, $\boldsymbol{n}$ is a vector denoting desirability based on natural factors alone, m is the length of $\boldsymbol{n}$ (and all the other vectors as well), $\boldsymbol{a}$ is a vector representing social factors alone, and $\boldsymbol{d}$ is a vector representing total desirability at each grid cell on the map (in this case Canada). This equation is performed on every grid cell, thus defining a new, aggregate layer for desirability. The constant c_x is an exponent with the effect of adjusting the emphasis on high desirability areas, allowing the demand for the most desirable areas to be increased appropriately. The function $z()$ transforms a vector into a set of standardized z-score values, where $z(\mathbf{x}) = (x_i - \mu)/\sigma$, and the $normalize()$ function is a unity-based normalization function, rescaling values into the range [0, 1].

When an agent selects the area it will settle, it essentially considers every grid cell in Canada, and then chooses one of those locations randomly, with a likelihood that is biased by the desirability. This is implemented using a technique that is common in evolutionary algorithms called a roulette wheel.

A roulette wheel is essentially implemented as follows. Pick a random number y between 0 and $\sum d_i$. Then traverse the vector $\boldsymbol{d}$, summing the values as you go. If the sum exceeds y, then the previous vector value in $\boldsymbol{d}$ is chosen, and the grid cell associated with it is selected for settlement. The speed of this process can be greatly improved by keeping a vector of partial sums of $\boldsymbol{d}$, and then using a binary search to find the appropriate index.

Because our desirability calculation is dependent on agents that have already been placed, we must update the roulette wheel from time to time. In theory we should update it after every agent is placed, but this would be too time consuming. Instead, we decided to place agents in groups, and then perform the update. For our experiments, we place 100000 agents before performing another update. The consequence of this is that the placement decisions made by agents will not be affected by any other agents that were placed previously that were also in the same group.

When agents are placed, they are placed 10 at a time. In other words, a location is chosen, and then 10 agents are placed in that one location. Then another location is chosen, 10 more agents are place, and so on. According to the LandScan data, the total population in Canada in 2005 is just over 30 million, and this is how many agents we ultimately place.

2.2 Evolutionary Algorithm

Our EA was implemented using the ECJ toolkit, given that the NorthLands model is implemented in MASON (Luke et al. 2005) and the two toolkits are built to operate together in a highly efficient way.

Representation Individual agents in our EA are represented as a list of real valued numbers. These comprise key constants to the agent placement algorithm (see 2.1). Specifically, the following constants are defined as genes: c_e, c_w, c_p, c_t, c_s, c_f, c_x. All genes are constrained to the range [-1, 1], except for c_s, c_f, and c_x, which are constrained to ranges [0, 30], [0, 10], and [0, 40], respectively.

Selection and Reproduction The parent and offspring populations in our EA are non-overlapping. This simply means that each new generation consists solely of newly produced offspring. In other words, no individuals from proceeding generations are allowed to survive. We felt that this would allow for greater variation, and avoid becoming trapped in local optima, thus improving the search.

During reproduction, parents are selected using tournament selection with a tournament size of 2. This provides a reasonably strong and consistent selection pressure, without being overwhelming.

All selected parents are varied using two reproduction operators: two-point crossover and Gaussian mutation. Two-point crossover was chosen over one-point crossover because it is known to have fewer issues with gene linkages (De Jong, 2006). The Gaussian mutation operator is applied to every gene in each individual, with a fixed and relatively low amount of variation ($\sigma = 0.05$), which we determined experimentally using sensitivity studies. A low value for σ allows the EA better convergence on solutions later in the run. The crossover counterbalances this, as it is known to produce large amounts of variation, but only early in the run, when it is most useful.

All experiments were performed with populations containing 50 individuals. In general, crossover requires populations of at least this size to be effective. Similarly, all experiments were run for 50 generations, which preliminary experiments determined was sufficient to converge on a solution.

Fitness Function The fitness of an individual is calculated by using the gene values as parameters to the placement algorithm and generating a population. This population is then compared to the LandScan data to see how similar the two are. LandScan data is a GIS product created at Oak Ridge National Laboratory that estimates global population distributions at a 1km scale.

There are two distinct aspects of the populations that are compared: a spatial distribution, and the rank distribution of the cells. A good fit for the spatial distribution indicates that agents are in roughly the correct locations and concentrations. Rank distribution, on the other hand, gives an indication of whether agents tend to create communities with similar densities and in similar proportion to actual human communities.

Spatial Distribution A simple approach to measuring a spatial distribution is to calculate the Kullback-Leibler divergence of the two distributions (Kullback and Leibler, 1951). This produces a result in the range [0, 1], thus creating a type of similarity metric, with a 1 indicating a perfect match.

However, we were concerned that human populations tend to be highly concentrated into population centers within relatively small boundaries. Even under the best of circumstances, it is unlikely that our algorithm would be able to generate cities in exactly the same locations as the ones that humans chose to nucleate. Our concern was that our placement algorithm might generate a city very close to the location of an actual city, but still far enough away so that there is little impact on the similarity metric. In other words, the algorithm would get almost no credit, even though the generated and real cities were actually quite close.

In order to create a more forgiving similarity metric, we first applied a Gaussian smoothing algorithm to both populations before comparing them. Figure 2 provides an example of the effect that this has on the spatial layer, essentially spreading the population out into larger masses. For the Gaussian smoothing, we performed 2 passes using a 9×9 kernel with $\sigma = 3.0$.

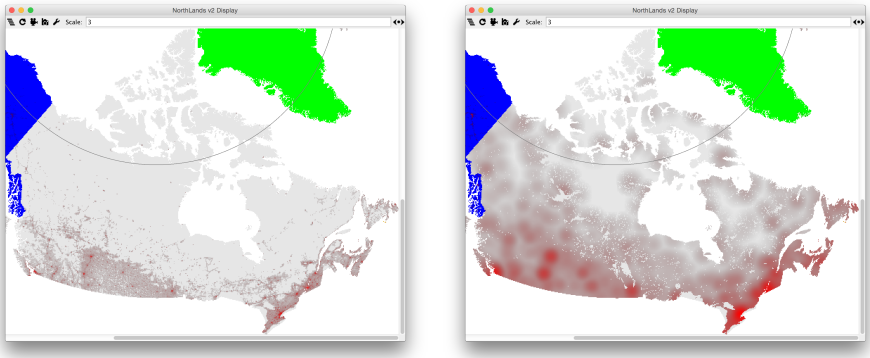

Fig. 2. The left image is population distribution derived from 2005 LandScan data. The color map has been altered to be non-linear in order to emphasize the low population grid cells. The right image is the same population data with a Gaussian smoothing algorithm applied for use with the EA fitness function.

Rank Distribution The rank distributions of settlement sizes (also called Zipf's or Zipfian distributions) were compared using the Kolmogorov-Smirnov statistic, which has value in the range [0, 1], with 0 being an identical match.

Combining Measures We combined the spatial and rank measures as follows: $f(x) = s(x)(1 - r(x))$, where $s(x)$ is the spatial measure and $r(x)$ is the rank measure, both described above. This creates the fitness values used for an individual in the EA.

3 Further Illustrative Results

In this section we provide additional illustrative results using the same example for populating the Canadian boreal and Arctic regions of the MASON NorthLands model. 30 runs of the EA were performed, and the solutions evolved all tended to be roughly similar. These coefficient values are fairly representative: $c_t = 0.90$, $c_p = -0.085$, $c_w = -0.2$, $c_e = -0.17$, $c_s = 4.0$, $c_f = 2.0$, and $c_x = 35.0$. Figure 3 shows the spatial distribution of aggregate desirability (left) generated by these values, which were used to generate the corresponding population distribution (right).

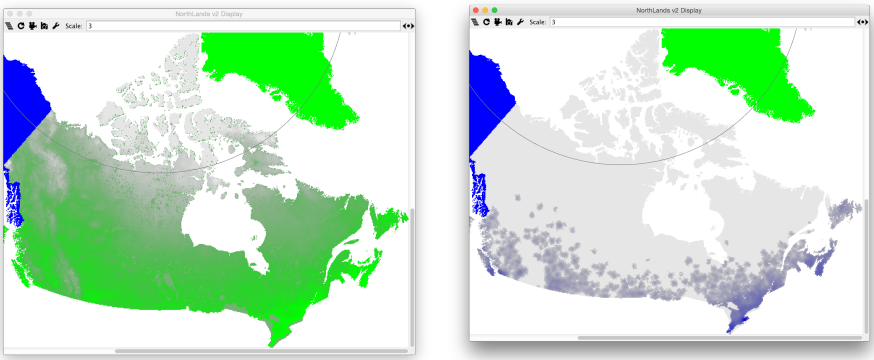

Fig. 3. Aggregate desirability (left) is composed of the desirability components in figure 1, combined using the coefficients found by the EA. The generated population (right) used the aggregate desirability to place agents in the appropriate locations.

By contrast, Figure 4 shows a side-by-side comparison of real data (left) and a population generated from the best evolved result (right; fig. 3, right).

Finally, a different perspective is gained by comparing and contrasting the rank-size distributions for real and simulated data. As shown in Figure 5, the two distributions are quite similar and minor differences can be explained. The largest cities are somewhat larger in the empirical (LandScan data) distribution, because the NorthLands model does not priviledge settlement formation over other factors that operate in the real world. At the other extreme, small towns or villages are not measured with great accuracy by the LandScan data, whereas data collection is exact (zero measurement error) for the simulated data.

4 Discussion

4.1 On the proposed procedure

The procedure we have presented and demonstrated requires fairly standard tools from evolutionary computation and is generally applicable to a broad vari-

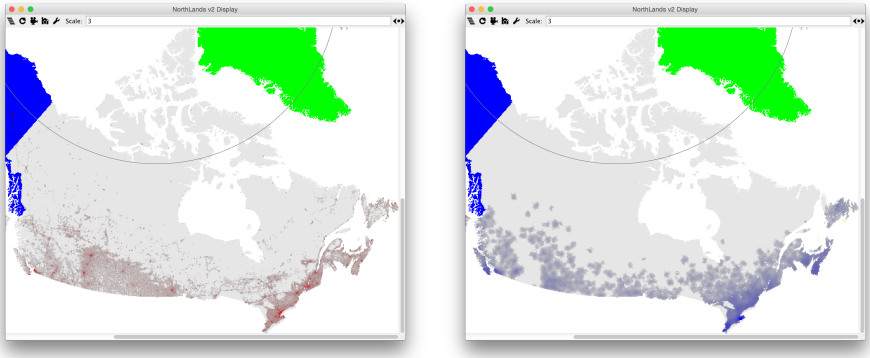

Fig. 4. A side-by-side comparison of the 2005 LandScan data (left) and a population generated from the best evolved result (right).

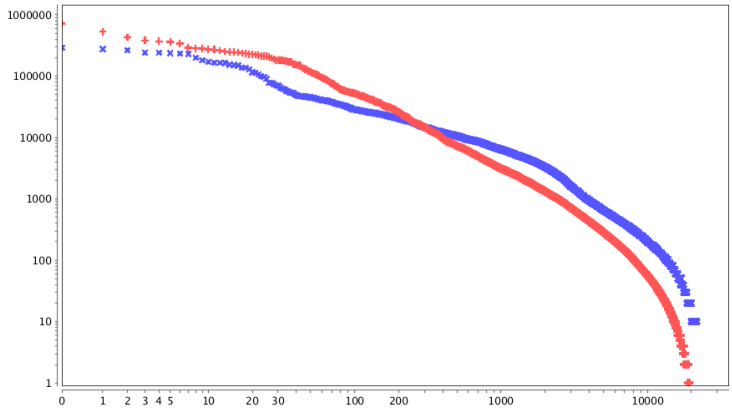

Fig. 5. A comparison of population rank-size distributions on a log-log scale. Rank and population are represented by the x- and y-axis, respectively The generated population (blue 'x') is very similar to that in the LandScan data (red '+'), although the population seems to be somewhat skewed away from areas of high density and toward areas of lower density.

ety of situations characterised in the Introduction. The situation addressed is not uncommon in computational social science using spatial agent-based models, so the same procedure could be used to populate other entities that can be placed according to a set of identified features. For examples, certain types of buildings or infrastructure, not just human agents or households (as in NorthLands).

Another advantage of the proposed procedure is that it is also susceptible to improvements as evolutionary computation (and advanced toolkits, such as ECJ) also improve. Finally, this EA procedure can be easily tested in other MASON-based social simulation models, because ECJ is ideally designed to operate with models created with MASON.

4.2 On the results

The context of these results is provided by the broader Mason-Smithsonian Joint Project on Climate and Society, of which NorthLands is a core part. These results represent the latest of several iterations that were developed using a spiral development model, and in that time NorthLands has evolved from a single model architecture to one based on a "federated" framework.

The results demonstrated here have intrinsic value for the proposed procedure, as well as extrinsic value for marking progress in the development and analysis of NorthLands. These results on population distributions provide more robust foundations for the simulation of population migratory movements when climate change stress the population through a variety of causal mechanisms, as in a network of influences that generate rural and urban migratory movements.

4.3 Future research

The following are envisioned developments:

1. We are developing an approach to incorporating more detailed census information into our placement algorithm. Population information often exists at the province, county and municipal levels, and we are planning to constrain the placement of certain agents to match this.
2. We are developing a mechanism for agent movement that uses an approach that is very similar to our placement algorithm, but includes additional factors in decision-making, such as available infrastructure, wealth, agent satisfaction and agent preferences.
3. We plan to perform sensitivity studies, particularly in different regions and time frames, in order to determine the robustness of these results and their sensitivity to cultural factors.

Acknowledgements

Paper prepared for the Sixth Annual Conference of the Computational Social Science Society of the Americas (CSSSA), Santa Fe, New Mexico, October 29

– November 1, 2015. Funding for this study has been provided by the US National Science Foundation, CDI Program, grant no. IIS-1125171, which supports the Mason-Smithsonian Joint Project on Climate Change and Societal Dynamics, and by the Center for Social Complexity at George Mason University, in collaboration with the Smithsonian National Museum of Natural History.

This product was made utilizing the LandScan (2005)^{TM} High Resolution global Population Data Set copyrighted by UT-Battelle, LLC, operator of Oak Ridge National Laboratory under Contract No. DE-AC05-00OR22725 with the United States Department of Energy. The United States Government has certain rights in this Data Set. Neither UT-Batelle, LLC nor the US Department of Energy, nor any of their employees, makes and warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of the data set.

References

An, L, A Zvoleff, J Liu & W Axinn: Agent-Based Modeling in Coupled Human and Natural Systems (CHANS): Lessons from a Comparative Analysis, Ann Assoc Amer Geog 104 (4): 723–745 (2014)

Batty M, The New Science of Cities, MIT Press (2013)

Cioffi-Revilla, C: Introduction to Computational Social Science: Principles and Applications. London and Heidelberg. Springer (2014)

Cioffi-Revilla, C: A Unified Framework for Convergence of Social, Engineering, and Natural Sciences. In W S Bainbridge & M Rocco. Handbook of Science and Technology Convergence. London and Heidelberg. Springer (2015)

Cioffi-Revilla C, De Jong K, Bassett J: Evolutionary Computation and Agent-based Modeling: Biologically-inspired Approaches for Understanding Complex Social Systems. Comput Math Org Th 18(3):356–373 (2012)

Cioffi-Revilla, C, Honeychurch W, Rogers JD: MASON Hierarchies: A Long-Range Agent Model of Power, Conflict, and Environment in Inner Asia. In J Bemmann & M Schmauder (eds.), Complexity of Interaction Along the Eurasian Steppe Zone in the First Millennium CE. Bonn, Germany: Rheinische Friedrich-Wilhelms-Universität Bonn Press, 89–113 (2015)

Cioffi-Revilla C, Rogers JD, Schopf P, Luke S, Bassett J, Hailegiorgis A, Kennedy W, Froncek P, Mulkerin M, Shaffer M, Wei E: MASON NorthLands: A Geospatial Agent-Based Model of Coupled Human-Artificial-Natural Systems in Boreal and Arctic Regions. Proc Soc Sim Conf SSC2015, 11th Conf Eur Soc Sim Assoc, Groningen, The Netherlands, Sept 14–118 (2015)

De Jong, K. A. Evolutionary Computation: A Unified Approach. MIT Press, Cambridge, Mass. (2006)

Heppenstall AJ, Crooks AT, See LM, Batty M (eds): Agent-based Models of Geographical Systems. Springer, New York (2012)

Kohler TA, van der Leeuw SE, eds. The model-based archaeology of socionatural systems. School for Advanced Research on the, 2007.

Kullback S and Leibler RA: On information and sufficiency. Ann. Math. Stat., 22:7986, (1951)

Liu J, Dietz T, Carpenter SR, Alberti M, Folke C, Moran E, . . . Taylor W: Complexity of Coupled Human and Natural Systems. Science 317:1513–1516 (2007)

Luke S, Cioffi-Revilla C, Panait L, Sullivan K: MASON: A Java Multi-Agent Simulation Environment. Simulation: Trans Soc Modeling Simulation Int 81(7):517–527 (2005)

Luke S: The ECJ owners manual. Technical report, George Mason University (2010)

Railsback SF, Grimm V: Agent-Based and Individual-Based Modeling: A Practical Introduction. Princeton University Press, Princeton NJ (2012)