Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

USING EVOLUTIONARY MODEL DISCOVERY TO DEVELOP ROBUST POLICIES

Alex Isherwood

The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102, USA

Matthew Koehler

The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102, USA

David Slater

The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102, USA

ABSTRACT

Agent-based models can be a powerful tool for evaluating the impact of policy decisions on a population. However, analyses are traditionally beholden to one set of rules hypothesized at the conception of the model. Modelers must make assumptions of agent behavior that are not necessarily governed by data and the actual behavior of the true population can thusly vary. Evolutionary model discovery (EMD) seeks to provide a solution to this problem by leveraging genetic algorithms and genetic programming to explore the plausible set of rules that can explain agent behavior. Here we describe an initial use of the EMD system to develop robust policies in a resource constrained environment. In this instance, we extend the NetLogo implementation of the Epstein Rebellion model of civil violence as a sample problem. We use the EMD framework to generate 23 plausible populations and then develop policy responses for the government that are robust across the plausible populations.

1 INTRODUCTION

This work focuses on addressing a common problem with simulation-based policy development studies and analyses (while a general issue, this is often associated with the use of agent-based models). Namely, that the analyst(s) create one simulation, perhaps perform a parameter sweep of "important" parameters (a design of experiments), analyzes the data, draw one or more conclusions and moves on. Unfortunately, there may be many simulations, or agent behavior rules, available for a given scenario or situation. There is a growing body of literature under the label of "Inverse Generative Social Science," iGSS, that is attempting to address this issue (Epstein 2023). Here the typical agent-based modeling process, e.g., Generative Social Science (Epstein 2006), is turned on its head and the agents become the output of interest rather than the data the agents generate. Essentially, how many different agent models produce the dynamics of interest? Here one goes from data to agents, rather than the usual direction from agents to data. There are a number of emerging techniques to do this from more inferential DeDeo et al. (2010) and Rand (2019) to more heuristic (Ren et al. 2018), (Gunaratne 2019) and (Gunaratne and Garibay 2020). More discussion of these other methods can be found in Isherwood and Yelnick (2023)). Here we focus on Evolutionary Model Discovery (EMD) introduced by Gunaratne (2019) and expanded in Gunaratne and Garibay (2020). While the original EMD codebase was made available, for this effort we reimplemented it in Java making use of the Evolutionary Computation in Java (ECJ) framework (ECJ) (Luke 2010). This was done to simplify its use and expand the EMD frameworks functionality to include both NetLogo (Wilensky 1999) and MASON (Luke et al. 2005) (The MITRE EMD codebase may be found here: strategyMining.) In this paper we attempt to leverage the ability to create many different simulations each with acceptable performance to develop a robust policy intervention. Here we define robustness as creating a large deviation in performance from the original simulation across all EMD-generated plausible populations.

2 METHODS

For this experiment we extended the "rebellion" model included with the standard NetLogo installation (Wilensky 1999). This model was articulated by (Epstein 2002). The model proposes a simple mechanism for generating the bursty dynamics of human uprisings and is premised upon the interactions among the ratio of cops and rebels within the vision of a given agent, coupled with their risk tolerance and their perceived legitimacy of the government. The code used for the following experiment can be found as an example within the Strategy Mining github repository referenced above.

While we stared with the rebellion model included with NetLogo, we extended it to better exercise the EMD framework. First, a small-world social network among the citizens was created with NetLogo's networks extension, nw:generate-small-world, with a clustering exponent of 2.0). The network was created to add an additional factor for EMD. Specifically, by including the social network agents could now also use information gathered from the social contacts in their decision to rebel or not in addition to information gathered by "looking around" their immediate vicinity (as defined by the vision parameter in the original model).

Secondly, the original "determine-behavior" function was also modified to make it compatible with EMD generated rules instead of the default (hand coded) rules. This is necessary because EMD uses a genetic program (GP) to reconfigure agent decision rules and a genetic algorithm (GA) to optimize the GPs. This being the case, rules must be strongly typed so the GP knows valid and invalid combinations. This is an important point, a typical NetLogo model will not be compatible with EMD without modification. More detail on how to modify a NetLogo model can be found at MITRE GitLab site (see above), igssWorkshop, and in (Isherwood and Yelnick 2023). It is also worth stressing that EMD only creates new combinations of atomic behaviors defined a priori by the modeler. For example, if you are simulating vehicle and specify how to turn left, how to turn right, and how to move forward; EMD will create novel combinations of those three rules but will not create a new rule such as "move backward." To continue this example, EMD could create an ability to back up with two consecutive turns in the same direction and then move forward, but would not create a "move in reverse" capability for the agents. A detailed discussion of how this works within the EMD system can be found within this video by Ms. Yelnick presented as part of the second iGSS Workshop (link above).

In addition to restructuring the code and decision logic of the agents, EMD requires a fitness metric so one may define simulation performance. This fitness metric is used by the GA for optimization. In attempting to match the original behavior of the model, the original model was run 300 times for 500 ticks each with the number of active (rebelling) agents recorded at each time step. Four lists were then recorded to describe the rebellion dynamics in each model run, with a rebellion defined as a continuous period in which more than 10 agents were rebelling (active):

1. The maximum number of active agents in each rebellion
2. The sum of active agents in each rebellion
3. The duration in ticks of each rebellion
4. The interarrival time between each rebellion

The fitness metric was then designed to compute the Kolmogorov-Smirnov (KS) statistic between the summary lists of a new model run and each of the merged summary lists from the 300 ground truth runs. The final metric is the sum of the four KS statistics. To define a "good" EMD-generated rule, i.e., one that performs similarly enough to the original to be considered a plausible population, a novel approach was defined. A threshold was determined by running the base rule (original agent behavior rules) through the fitness metric 300 times to determine its natural variance in fitness, shown in Figure 1.

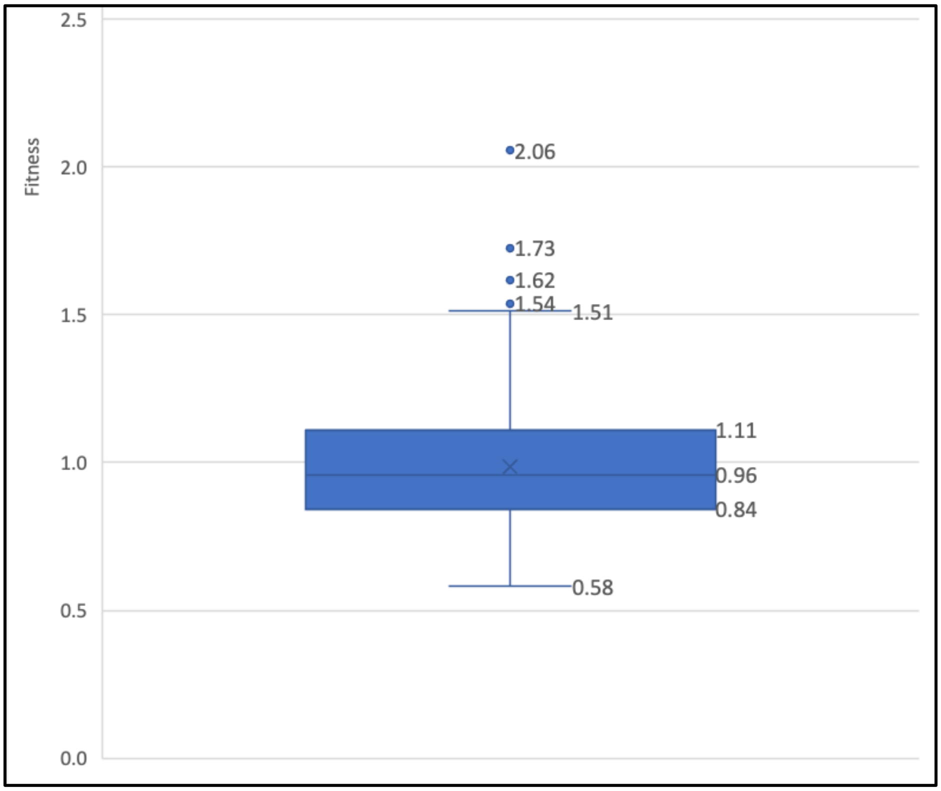


Figure 1: The variation in score of the base rule over 300 trials.

In order to define a threshold we chose the 75th percentile (1.11), in effect a defined upper bound of how much the same rule could plausibly vary against itself in a sufficiently large sample. This provided us with a principled way to define the amount of deviation a simulated run could have against our reference model and still be considered well performing. This approach can be used when multiple series of data are available or can be generated. A fitness metric can be developed to simply determine the distance between series, or in this case, a more customized metric to focus on specific patterns within the series.

Phase one of this experiment was to run the EMD framework to create a set of agent models all of which had similar dynamics to the original model (where we define similar as the statistic defined above is less than or equal to 1.11). As EMD uses a heuristic search, some collections of rules come up more than once and are, therefore, sampled more than others. A sample of rules created by the EMD framework that were used at least 10 times with a mean fitness of 1.11 or less are shown in Table One. Due to the specific combining factors used, many of the rules tried by the genetic algorithm would simplify to the same rule. Therefore, a python script was developed to simplify each rule before evaluating it to ensure only unique rules were examined.

All of the rules shown in Table 1 created rebellion dynamics within the 1.11 KS statistic threshold and, thus, they were used in phase two of the experiment. These rules can be considered the plausible rule space, or the plausible populations, for matching data generated by the base rule. Using the above EMD-generated rules with similar behavior to the original model, phase two of this effort was comprised of a parameter variation experiment designed for policy analysis and development exercise.

Rule #	Rule Content
0	"(greaterThan (subtract (reportGrievance) (multiply (reportArrestProb) (multiply (reportArrestProb) (reportAversion)))) (reportThreshold))",
1	"(greaterThan (subtract (reportGrievance) (multiply (reportArrestProb) (multiply (reportArrestProb) (multiply (reportArrestProb) (reportAversion))))) (reportThreshold))",
2	"(greaterThan (subtract (reportGrievance) (multiply (reportArrestProb) (reportAversion))) (reportThreshold))",
3	"(greaterThan (subtract (subtract (reportGrievance) (multiply (reportArrestProb) (reportAversion))) (multiply (multiply (reportArrestProb) (reportAversion)) (reportAversion))) (reportThreshold))",
4	"(greaterThan (subtract (subtract (reportGrievance) (multiply (reportArrestProb) (reportAversion))) (multiply (reportArrestProb) (reportAversion))) (reportThreshold))",
5	"(greaterThan (subtract (subtract (reportGrievance) (multiply (reportArrestProb) (reportAversion))) (multiply (reportArrestProb) (multiply (reportArrestProb) (reportAversion)))) (reportThreshold))",
6	"(greaterThan (subtract (reportGrievance) (multiply (reportArrestProb) (reportGrievance))) (reportThreshold))",
7	"(greaterThan (subtract (subtract (reportGrievance) (reportArrestProb)) (multiply (reportAversion) (multiply (reportAversion) (reportArrestProb)))) (reportThreshold))",
8	"(greaterThan (multiply (subtract (subtract (reportAversion) (reportArrestProb)) (multiply (reportAversion) (reportAversion))) (reportAversion)) (reportThreshold))",
9	"(greaterThan (subtract (reportGrievance) (multiply (reportArrestProb) (reportArrestProb))) (reportThreshold))",
10	"(greaterThan (subtract (subtract (reportGrievance) (reportArrestProb)) (multiply (reportArrestProb) (reportAversion))) (reportThreshold))",
11	"(greaterThan (subtract (subtract (multiply (multiply (reportAversion) (reportAversion)) (multiply (reportAversion) (reportAversion))) (reportArrestProb)) (reportArrestProb)) (reportThreshold))",
12	"(greaterThan (subtract (reportGrievance) (reportArrestProb)) (reportThreshold))",
13	"(greaterThan (subtract (subtract (reportGrievance) (multiply (reportArrestProb) (add (propNeighborsActive) (reportGrievance)))) (multiply (reportArrestProb) (add (propNeighborsActive) (reportGrievance)))) (reportThreshold))",
14	"(greaterThan (subtract (multiply (multiply (reportAversion) (reportAversion)) (multiply (reportAversion) (reportAversion))) (reportArrestProb)) (reportThreshold))",
15	"(greaterThan (subtract (subtract (subtract (subtract (reportGrievance) (reportArrestProb)) (reportArrestProb)) (reportArrestProb)) (reportArrestProb)) (reportThreshold))",
16	"(greaterThan (subtract (subtract (reportGrievance) (reportArrestProb)) (reportArrestProb)) (reportThreshold))",
17	"(greaterThan (subtract (subtract (subtract (reportGrievance) (reportArrestProb)) (reportArrestProb)) (reportArrestProb)) (reportThreshold))",
18	"(greaterThan (subtract (reportGrievance) (multiply (reportArrestProb) (add (propNeighborsActive) (reportGrievance)))) (reportThreshold))",
19	"(greaterThan (subtract (subtract (subtract (reportGrievance) (reportArrestProb)) (reportArrestProb)) (multiply (reportAversion) (reportArrestProb))) (reportThreshold))",
20	"(greaterThan (subtract (reportGrievance) (multiply (multiply (reportArrestProb) (reportArrestProb)) (reportArrestProb))) (reportThreshold))",
21	"(greaterThan (subtract (reportGrievance) (multiply (multiply (reportArrestProb) (reportAversion)) (reportAversion))) (reportThreshold))",
22	"(greaterThan (subtract (multiply (multiply (reportAversion) (reportAversion)) (reportAversion)) (add (propLinksActive) (reportArrestProb))) (reportThreshold))",
23	"(greaterThan (subtract (multiply (multiply (reportAversion) (reportAversion)) (reportAversion)) (add (add (propLinksActive) (reportArrestProb)) (reportArrestProb))) (reportThreshold))"

Table 1: The twenty-four best performing rules.

More specifically, given the twenty-three plausible populations, what policy changes might the government enact to lessen the severity of rebellion dynamics? Since the government does not know which of the twenty-three plausible populations they are dealing with, the policy most likely to succeed will be the one that is a high performer across all plausible populations. In our hypothetical policy development scenario the government has two policy “levers” and a budget constraint. The policy levers include varying the number of police officers (or "cops" as they are referred to in the Rebellion model) used to contain a rebellious outburst and spending money on a message campaign to increase the population’s perception of the legitimacy of the government. The government can pull one or both of these levers subject to a limit on the total expenditure. The parameter experiment varied the following parameters in the rebellion model:

1. “Cop density” (0-.10 in increments of 0.002)
2. “Government legitimacy” (0-1 in increments of 0.02)

For each permutation of parameter combinations, the mean number of active agents across all ticks was recorded.

3 VALIDATION

As this was a study to better understand how EMD could be used within a policy analysis/development context, we did not engage in a formal validation process for the model. The model was simply face validated after the extension from the rebellion model that is included with NetLogo. Strictly speaking we feel the model as used achieved Level 1 Empirical Relevance (Axtell 2005), qualitative macroscopic correspondence to its referent. Here the referent was the original rebellion model as described by Epstein (2002) and the bursty dynamics seen in rebellion sizes over time. Here, our extension also produces bursty dynamics in rebellion size over time; therefore, we conclude that our extension did not result in undue violence to the underlying generating mechanisms of the original rebellion model and was fit for its intended purpose of exercising the EMD framework.

4 RESULTS

Figure 2 summarize our experimental results. Each of the twenty-four (the twenty-three behaviors shown in Table 1 and the original agent behavior) plausible populations are shown as a heatmap based upon the mean rebellion numbers for thirty runs at each parameter point. Darker regions indicate lower mean rebellion numbers, while lighter regions indicate higher rebellion numbers. Figure 2 highlights the utility of using EMD to produce plausible populations. To reiterate, all these populations generate rebellion dynamics that are consistent with the original model (our referent in this case). However, as shown by the differing shapes of the heatmaps some of these populations do not pay great attention governmental legitimacy, e.g., population 21, while others do not pay attention to cop density, e.g., populations 8 and 14. Had we relied on a single population type for this analysis we might have created a very fragile policy recommendation. If we had used population 21 for our analysis when the “real” population was of type 8, for example, we would have reached conclusions that would have had no impact on the situation facing the government.

The results of the parameter variation experiment were plotted logarithmically, as the number of active agents could vary between 0-1000, model runs with a mean number of active agents between 0-10 were still of interest. The Figure 3a shows the logarithm of the mean number of active agents across all time steps for each permutation of parameters for ten model runs. Figure 3b shows the aggregated results for the EMD generated rules (the disaggregated results are shown in Figure 2). EMD generated three groups of rules that appeared visually. The first (Rules 0-5, 21) approximately matched the distributions of active agents across all parameter variations from the base rule. The second group (Rules 6-7, 9-10, 12-13, 15-20) was similar to the first when CopDensity was less than 4, but above that threshold had agents that were unwilling to become active. The final group (Rules 8, 11, 14, 22-23) ignored the GovernmentLegitimacy parameter and only considered CopDensity in the determination to rebel.

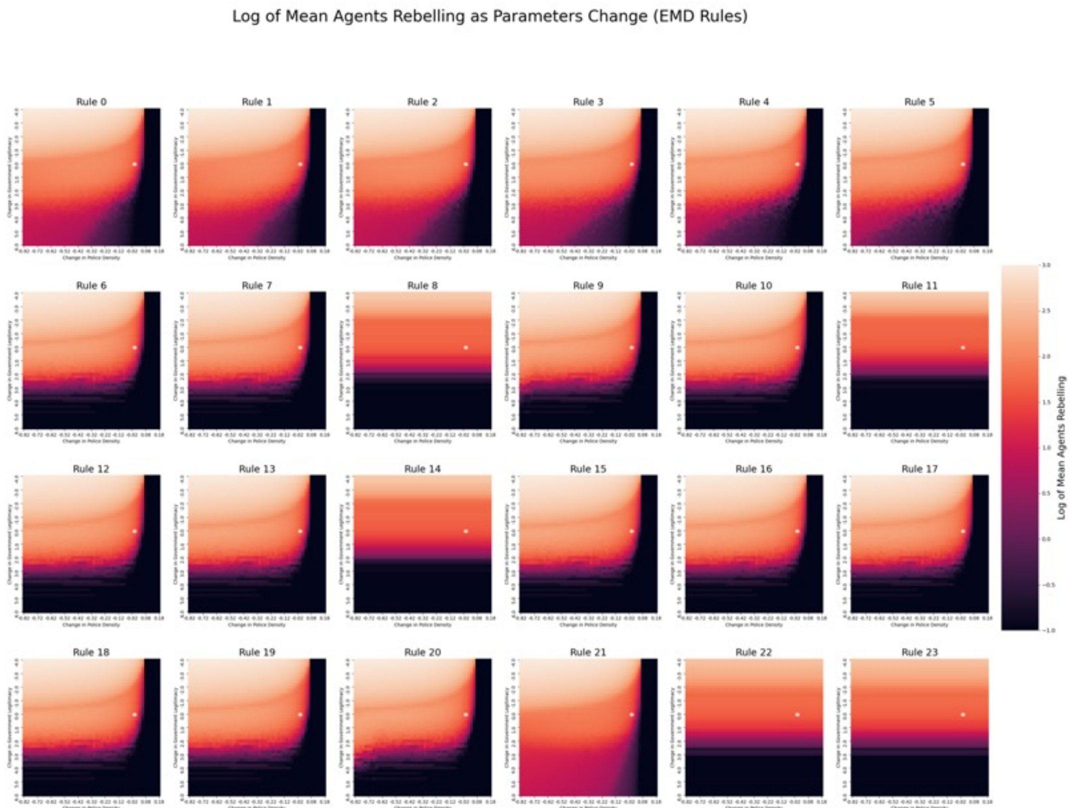


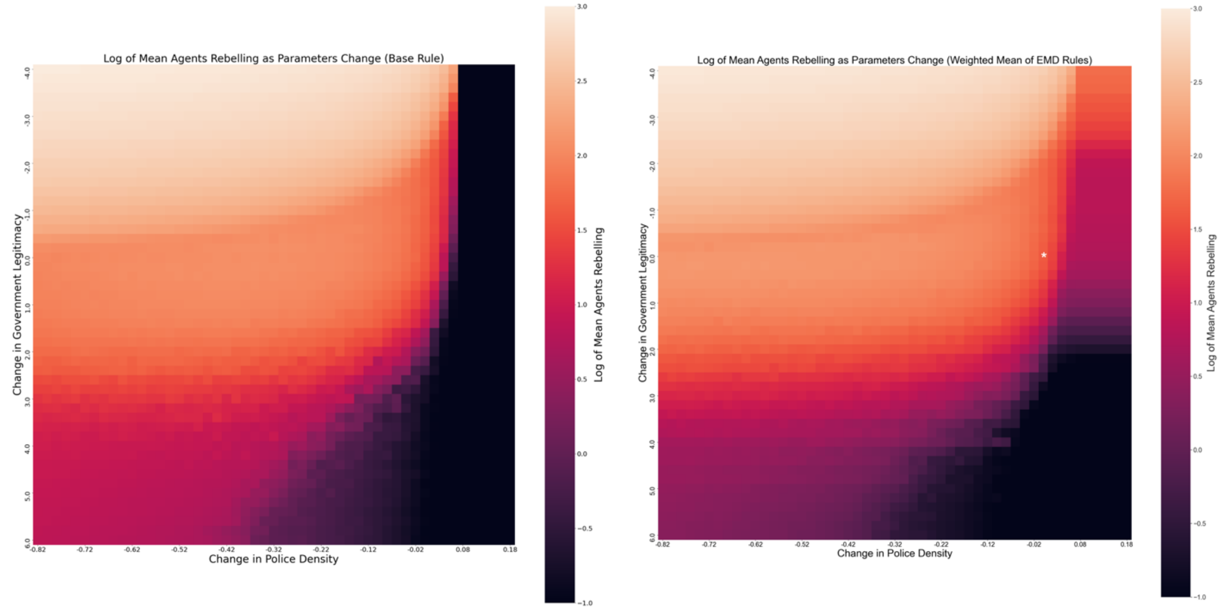
Figure 2: Overall results of the experiment for each plausible population, the starred point is the original parameter settings used in the port of the Epstein model.

In taking the aggregate results of the 24 EMD-generated rules, we took the weighted mean of them across each parameter combination, using the mean fitness scores as weights. The result create a plot that cautions against following the extreme cases of any single run. We see that for GovernmentLegitimacy greater than 0.85, the base rule has no active agents for any CopDensity. This is shown in Figure 3. However, across the aggregate of cases in which the base behavior is seen, cases do appear where agents will still rebel in these instances. Policy interventions can be interpreted as movement across these parameter plots with the white star as the starting point from the default model parameters.

As can be seen in Figure 3, there is a striking difference between Figure 3a and Figure 3b. This is significant as it means that while the EMD generated rules generate largely the same rebellion dynamics they respond differently to policy changes. That means that if policies are developed based upon only a single behavior rule, and the actual population uses a different rule, the impact of the policy could be very different than what was expected (the location of the base case rule is shown with the small star in Figure 3b). Figure 3 also highlights the additional policy space potentially available to be leveraged that may not appear useful when explored with a single behavior rule.

5 CONCLUSION

This example serves to demonstrate the utility of EMD for agent-based modeling problems, as well as techniques for construction and tuning of models. For problems where stochasticity and repeatability are



a. The overall dynamics seen in the DOE with the base rules.

b. The overall dynamics across all EMD generated rules within the DOE.

Figure 3: The rebellion dynamics seen in the DOE from the base (a) and from the EMD generated rules.

concerns, EMD offers a solution for exploring the rule space for other combinations of rules that can match a behavior. Policy problems are one such area where decisions should not be beholden to variance of any one rule. This approach allows for data driven consideration of all hypothesized model factors, rather than just one combination of them.

ACKNOWLEDGMENTS

This work was funded by the MITRE Independent Research and Development Program. Approved for Public Release; Distribution Unlimited. Public Release Case Number 23-1398.

REFERENCES

DeDeo, S., D. C. Krakauer, and J. C. Flack. 2010. “Inductive Game Theory and the Dynamics of Animal Conflict”. *PLoS Computational Biology* 6(5):e1000782.

Epstein, J. M. 2002. “Modeling Civil Violence: An Agent-Based Computational Approach”. *Proceedings of the National Academy of Sciences* 99(suppl_3):7243–7250.

Epstein, J. M. 2006. “Remarks on the Foundations of Agent-based Generative Social Science”. *Handbook of Computational Economics* 2:1585–1604.

Epstein, J. M. 2023. “Inverse Generative Social Science: Backward to the Future”. *Journal of Artificial Societies and Social Simulation* 26(2):1–9.

Gunaratne, C. 2019. *Evolutionary Model Discovery: Automating Causal Inference for Generative Models of Human Social Behavior*. University of Central Florida. Electronic Theses and Dissertations, 6871.

Gunaratne, C., and I. Garibay. 2020. “Evolutionary Model Discovery of Causal Factors Behind the Socio-Agricultural Behavior of the Ancestral Pueblo”. *Plos one* 15(12):e0239922.

Isherwood, A., M. J. M. K. D. S. W. T., and M. Yelnick. 2023. “Modeling Macaque Fighting Dynamics with the Evolutionary Model Discovery Framework to Understand Its Application and Utility”. In *Proceedings of the 2022 Conference of the Computational Social Science Society of the Americas*. Computational Social Science Society of the Americas.

Luke, S. 2010. “The ECJ owner’s manual”. *San Francisco, California, A User Manual for the ECJ Evolutionary Computation Library*:1–206.

Luke, S., C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. 2005. “MASON: A Multiagent Simulation Environment”. *Simulation* 81(7):517–527.

Rand, W. 2019. “Theory-Interpretable, Data-Driven Agent-Based Modeling”. *Social-Behavioral Modeling for Complex Systems*:337–357.

Ren, Y., V. Cedeno-Mieles, Z. Hu, X. Deng, A. Adiga, C. Barrett, S. Ekanayake, B. J. Goode, G. Korkmaz, C. J. Kuhlman et al. 2018. “Generative Modeling of Human Behavior and Social Interactions Using Abductive Analysis”. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 413–420. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Wilensky, U. 1999. “NetLogo”. *http://ccl. northwestern. edu/netlogo/*.

AUTHOR BIOGRAPHIES

ALEX ISHERWOOD is a Data Scientist at the MITRE Corporation. He received his BS in Computer Science from the College of William and Mary. His email address is aisherwood@mitre.org.

MATTHEW KOEHLER is the Generative Social Science and Computational Finance Group Lead at The MITRE Corporation. He received his AB from Kenyon College, his MPA from Indiana University, his JD from George Washington University Law School, and his Ph.D. in Computational Social Science from the George Mason University Krasnow Institute for Advanced Study. His email address is mkoehler@mitre.org.

DAVID SLATER is a Principle Operations Researcher within the MITRE Corporation’s Operations Research Department. He has spent the last twelve years at MITRE applying Applied Mathematics to some of the United States’ most critical and challenging Systems Engineering problems. His research interests include complex systems analysis, network science, and complexity science. He received his B.S from Michigan Technological University in 2005 and a M.S. and Ph.D. in Applied Mathematics from Cornell University in 2009 and 2011, respectively. His email address is dslater@mitre.org.