Efficient Parameter Exploration of Simulation Studies

Megan Olsen
Department of Computer Science
Loyola University Maryland
Baltimore, MD, USA
mmolsen@loyola.edu

M S Raunak
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD, USA
raunak@nist.gov

***Index Terms*—simulation, parameter exploration, covering arrays, combinatorial approach**

I. Extended Abstract

Simulation is a useful and effective way to analyze and study complex, real-world systems. It allows researchers, practitioners, and decision makers to make sense of the inner working of a system that involves many factors often resulting in some sort of emerging behavior. Scenarios such as the spread of a pandemic, the operations of an autonomous vehicle on busy streets, or the flow of patients in an emergency room can be studied with simulation models. Agent based modeling or ABM is a common modeling technique used in simulating and studying such complex systems. In these models, agents are individual autonomous entities that make decisions about their actions and interactions within the environment. The factors that influence the agent's decision making process and thus drive the simulation outcome are commonly known as parameters. A typical agent-based simulation model will include many parameters, each with a potentially large set of values. The number of scenarios with different parameter value combinations grows exponentially and quickly becomes infeasible to test them all or even to explore a suitable subset of them. How does one then efficiently identify the parameter value combinations that matter for a particular simulation study? In addition, is it possible to train a machine learning model to predict the outcome of an agent-based model without running the agent-based model for all parameter value combinations?

We experiment with an ABM known as HeatBugs [3]. The agents in HeatBugs move in a 2D grid, in which only one agent can exist in a given location. Each agent gives off heat to its environment, affecting other agents. Agents have an ideal temperature, and will move if needed to acquire their ideal temperature. The primary output in this model is the agents' level of unhappiness, based on if they are able to maintain their ideal temperature. The behavior of the agents and their environment are defined by eight parameters: the number of agents, speed of evaporation of heat in the environment, how much heat diffuses to neighboring locations, the min/max temperatures for agents, min/max heat agents output, and how likely they are to move. If we hold the number of agents steady, there are approximately 40 trillion parameter value combinations for this model. It is infeasible to test more than a very small fraction of these combinations.

We propose utilizing covering arrays [1] to create t-way (t = 2, 3, 4, etc.) combinations of parameter values to significantly reduce the parameter value exploration space. A t-way covering array is a matrix of values that ensures the existence of all t-way combinations of parameter values. For example, if we have 10 binary parameters, the total number of possible value combinations is $2^{10} = 1024$. If one is interested in only covering all 3-way value combinations of these 10 binary variables, that can be achieved with a covering array matrix of only 13 rows, each row containing 10 values. Those 13 rows include every possible 3-way combination of parameter values. From the perspective of a simulation model study, the covering array replaces the need for running 1024 simulations with only 13 simulation runs.

In our prior work we showed that covering arrays were useful for decreasing the parameter space in another agent-based model [2]. We now build on that work by applying it to the HeatBugs model [3] and then training a machine learning model by using the covering arrays to select our training and test data. For HeatBugs we create a 2-way covering array consisting of 33,051 combinations as our training data, and a 3-way covering array consisting of slightly over 3.97 million combinations as our testing data. We then run the simulation model four times with varying random number seeds for each of the combinations in both the 2-way and 3-way arrays, to account for the impact of stochasticity in the simulation results.

We investigate if the 2-way covering array provides enough information about the model that we can use it to train a random forest machine learning model to predict if the unhappiness level will be steady at the end of the simulation for other unseen parameter value combinations, tested on the 3-way covering array data. We define steady unhappiness level as a low standard deviation of unhappiness across the final 500 time steps of the simulation run. We create a binary category to predict the happiness outcome based on whether all runs of that parameter set are steady, or not. To the best of our knowledge, predicting agent-based model results via machine learning has not previously been attempted, and it is generally assumed that you cannot predict the outcome without running the simulation model.

Our results show that a 2-way covering array [1] provides sufficient training data to train our random forest to predict if the unhappiness level will be steady by the end of the simulation. We train and test on three definitions of "steady": standard deviation $<= 1$, $<= 0.5$, and $<= 0.2$. As the cutoff for "steady" increases, the classes become less balanced. When testing using the 3-way covering data for each definition of steady, we have a class balanced accuracy of 95.2% for standard deviation $<= 1$, 96% for standard deviation $<= 0.5$, and 96.7% for standard deviation $<= 0.2$. It is expected that as class imbalance increases, accuracy would decrease, however the numbers are still quite high even in our most imbalanced data. These results imply that a random forest machine learning model can be predictive in general on ABM results with only a small fraction of the full parameter space used to generate the training data. This is an exciting possibility for exploring parameter values in simulation studies.

In this presentation, we discuss the problem of handling large parameter spaces, the new use of covering arrays to decrease the space for this type of problem, and the application of machine learning to predict the result of ABMs using the covering array data to choose a representative part of the parameter value space.

REFERENCES

[1] R. Kuhn, "An Introduction to Covering Arrays", https://math.nist.gov/coveringarrays/coveringarray.html, 2006.
[2] C. Maalouf, M. Olsen, M Raunak, "Combinatorial Testing for Parameter Evaluation". Proceedings of the 2019 Winter Simulation Conference, Dec. 2019.
[3] "NetLogo Heatbugs model." http://ccl.northwestern.edu/netlogo/models/Heatbugs. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.