Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

COMBINATORIAL TESTING FOR PARAMETER EXPLORATION

Chiara Maalouf
Megan Olsen
M S Raunak

Department of Computer Science
Loyola University Maryland
4501 N. Charles Street
Baltimore, MD 21210 USA

ABSTRACT

Agent-based models often have large parameter spaces that cannot be fully explored. How can we determine the roles parameters play in defining the resulting simulation behavior? We propose using covering arrays, often used in test case generation for software testing, to strategically decrease the search area in simulation parameter space exploration. We test our approach on the Wolf-Sheep Predation model, and find that the number of parameter combinations need to be explored may still be high for some models. We propose next steps for the use of covering arrays in exploring an agent-based model's parameter space.

1 INTRODUCTION

Computer Simulation allows researchers to study complex systems without the expense or time of running real world experiments. Many studies utilize agent-based models (ABM), where agents represent individual actors, and parameters affect the behavior of agents and the environment where they reside. Simulation modelers are generally interested in identifying the impact of the parameter values and their combinations on the outcome of simulation experiments. However, the parameter and outcome relationship cannot usually be predicted without running the simulation. One challenge of learning the relationship between parameter value combinations and outcomes is parameter sensitivity, the potentially drastic change in outcome with the slightest change in parameter values (Miller 1974). Parameters have varying degrees of sensitivity, and therefore it can be hard to pin point which parameter value combinations result in which outcome. Knowing that a specific set of parameters leads to a specific outcome does not necessarily inform the result of a slightly different parameter set. A brute-force effort to understand the modeled system under different scenarios requires trying every possible combination of parameter values. However, except for the most trivial models, this approach creates a massive number of combinations to test and is therefore generally intractable. For example, to test all combinations in a system with ten parameters each with a range of values [1,10], we would generate 10 billion tests. If every test took 1 second, it would take almost 32 years of computational time to run all tests. Hence, one has to be strategic in parameter space exploration.

We propose utilizing "covering arrays" (Kuhn et al. 2013) for this purpose. A covering array table is a compact mathematical object that ensures t-way interactions between parameter values within its rows. Each row of the table works as a parameter set that captures multiple t-way (e.g. 2-way, 3-way) interactions among the parameters. If we are only interested in studying pairwise (2-way) interactions of all model parameters, a much smaller covering array can be generated than all possible combinations, reducing the parameter exploration space significantly, yet strategically. This notion of coverage comes from the idea of how well a set of chosen configurations (parameter-set) "covers" the set of all configurations. Although covering arrays are successfully used to test many software types, they have not yet been utilized for

examining simulation parameter spaces. In this paper we present the first study on the effectiveness of using covering arrays to identify parameter set combinations to predict simulation results.

2 METHODS AND RESULTS

We use the NetLogo Wolf-Sheep Predation Model (Wilensky 1997), a 2D grid predator prey model with wolves, sheep, and grass, to study the benefit of using a covering array. This model has 7 parameters: Initial-number-wolves (0-250), Initial-number-sheep (0-250), Grass-regrowth-time (0-50), Wolf-reproduce (0-20), Sheep-reproduce (1-20), Wolf-gain-from-food (0-100), and Sheep-gain-from-food (0-50). Wolves need to eat sheep to gain energy and sheep need to eat grass to gain energy. An agent's energy decreases with every step. If a wolf eats a sheep, the sheep dies. A patch of grass eaten by a sheep cannot be eaten again until it has regrown. This model has over 6.9 trillion parameter combinations. Even if only half were feasible combinations, they cannot all be used in experiments. Our goal is to determine what model parameters result in a stable system, such that all agent types will continue to coexist indefinitely as studied in Olsen, Laspesa, and Taylor-D'Ambrosio (2018). We used the NIST ACTS program to generate covering arrays from the parameter space (Borazjany, Lei, Kacker, and Kuhn 2012). A 2-way covering array of all possible parameter values has 63,000 combinations. To determine if enough data can be generated for learning, 20,000 of the 2-way parameter combinations were randomly chosen and run four times for 100,000 time steps. Then the agent numbers for the last 1000 steps were analyzed to determine model stability. The four runs are to ensure that results are due to model parameters, not random number seeds.

Of the combinations run, less than 1% of the results were a stable system. Although prior work appeared to imply that stable systems are common in this model (Olsen, Laspesa, and Taylor-D'Ambrosio 2018), that does not appear to be the case. As the 20,000 were a random sample of the 2-way parameter sets, we determined that the 2-way combinations could not provide enough examples of stable systems to predict stability based on our parameter sets. We next examined the 3-way parameter sets, of which there are 92,151. We randomized this set and ran 20,000 initial parameter sets. Again, less than 1% of the results were stable, implying that a 3-way combination of parameters is still insufficient for finding enough stable systems to draw conclusions about how parameters affect outcomes.

3 CONCLUSIONS

Covering arrays appear to be a useful tool for decreasing the parameter space for parameter exploration of ABMs. However, the number of parameter combinations that must be run to generate the data to determine stable situations for the Wolf Sheep model remains computationally expensive. Further investigation is necessary to determine if a model with a different type of dynamic may be better suited to the approach, and if so, what level of coverage is necessary to generate the required data for parameter optimization.

ACKNOWLEDGMENTS

This work is partially supported by NSF grant #1626262 and NIST grant 70NANB18H278.

REFERENCES

Borazjany, M., Y. Lei, R. Kacker, and R. Kuhn. 2012. "Combinatorial Testing of ACTS: A Case Study". In *First International Workshop on Combinatorial Testing (CT 2012), IEEE Fifth International Conference on Software Testing, Verification and Validation (ICST 2012*, 591–600.

Kuhn, D. R., R. Kacker, and Y. Lei. 2013. *Introduction to Combinatorial Testing*. Chapman Hall / CRC.

Miller, D. 1974. "Sensitivity analysis and validation of simulation models". *Journal of Theoretical Biology* 48(2):345–360.

Olsen, M., J. Laspesa, and T. Taylor-D'Ambrosio. 2018, July. "On Genetic Algorithm Effectiveness for Finding Behaviors in Agent-based Predator Prey Models". In *Proceedings of the Summer Simulation Multi-Conference (SummerSim'18)*. Bordeaux, France: Society for Computer Simulation International.

Wilensky, U. 1997. "NetLogo Wolf Sheep Predation Model". Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.