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

CHOOSING THE RIGHT ENTITY SIZE TO MINIMIZE DISCRETIZATION ERROR IN DISCRETE EVENT SIMULATION MODELS

Leonardo Chwif

Industrial Engineering
Mauá Institute of Technology
Praça Mauá, 1
São Caetano do Sul, SP 09580-900, BRAZIL

Wilson Pereira

Associate Consultant
Simulate Simulation Technology
Av. Prof. Lineu Prestes, 2242
São Paulo, SP 05508-000, BRAZIL

José Arnaldo Barra Montevechi

Production Engineering and Management Institute
Federal University of Itajubá
Av. BPS, 1303
Itajubá, MG, 37500-903, BRAZIL

ABSTRACT

In discrete-event simulation models, the way we establish the relationship between a real-world object and the model entity (a single indivisible object flowing through the model) is crucial to some classes of problems due to possible computational unfeasibility. In addition, the entity size also relates to results accuracy and simulation running time - a subject barely explored in the literature. In this paper, these questions were investigated through case studies which supported our initial hypothesis about the general relationships involved. Then, a simple algorithm was developed for correctly choosing the best entity size to provide the desired accuracy, measured as a discretization error, with promising results. The limitations of the algorithm are addressed and some directions for future research are pointed.

1 INTRODUCTION

Entity size in simulation models, as will be soon defined, is a property linked to the model complexity. There is a consensus amongst the simulation community that a simple model is preferable to a complex one (Salt 1993; Chwif et al. 2000; Robinson 2011). According to Robinson (2011), the reasons are:

- Simple models can be developed faster;
- Simple models are more flexible;
- Simple models require less data;
- Simple models run faster;
- The results are easier to interpret since the structure of the model is better understood.

Two components of complexity are the scope and level of model details (Robinson 2011). For example, when simulating manufacturing systems, the level of detail can range from the entire facility to a single work-center. The scope is smaller in the latter case when compared to the former. On the other hand, the work-center could be modeled with associated processing times, breakdown times, shift patterns, and material handling equipment, among others. In this case, the level of detail is much higher, although the scope remains the same - the work-center (Chwif et al. 2000). However, the scope and level of detail could be considered as "structural complexity" because it is the property of the conceptual model. This and other concepts on model simplification are presented in a survey by van der Zee (2019).

When a conceptual model is converted into a computer model, there is another dynamic complexity element: the number of entities within the model. The more entities a computational model has to deal with, the more memory and processing time will be required since the number of current and future events depends on the number of entities. In this sense, fewer entities within a model should be better. Franciscan William of Occam (c. 1287-1347) postulated the Occam's Razor Principle (or Law of Parsimony): "*Non sunt multiplicanda entia sine necessitate*" or "Entities must not be multiplied beyond necessity".

This paper addresses the entity-sizing problem by showing the advantages and disadvantages of lowering the number of entities in a simulation model. It is organized into the following: Section 2 provides some preliminary concepts since there are no past literature foundations for this issue. Section 3 presents two case studies when the issue of entity size is analyzed in practice. Discussions are provided in Section 4, and finally, in Section 4, conclusions and further work are presented.

2 PRELIMINARY CONCEPTS AND LITERATURE REVIEW

Simulation is a widely used technique in various fields to analyze and understand complex phenomena. There are different types of simulation, which vary according to the nature of the problem to be analyzed and the tools available to model it. The main types of simulation are discrete event simulation, systems dynamics or continuous simulation, and agent-based simulation (Amaral et al. 2022).

Discrete event simulation is used to model systems in which changes occur at specific times and not continuously over time. On the other hand, continuous simulation is used for systems that change continuously over time and are described by differential equations. Agent-based simulation involves creating models that simulate the behavior of entities, known as agents. These agents can represent individuals, groups, processes, or machines, and interact with each other as well as the environment in which they operate (Santos et al. 2021). Hybrid simulation, in turn, combines elements of, at least, two of the aforementioned techniques, allowing the modeling of complex systems that present discrete events and continuous dynamics (Brailsford et al. 2019).

Hybrid simulation has become an increasingly important tool for analyzing complex systems in various fields, such as engineering, economics, and biology. Mosavi et al. (2019) present a hybrid simulation approach for fluid modeling, while Zhang et al. (2019) use a hybrid simulation approach to model complex biological systems. One advantage of hybrid simulation is its ability to handle complex and nonlinear systems, where continuous and discrete-event simulation alone may not be sufficient. In addition, hybrid simulation allows the inclusion of models from different time scales. For instance, Kumar and Kumar (2018) presented a hybrid model (system dynamics and discrete-event simulation) to simulate the manufacturing of gold and silver products. The refining banks were modeled using system dynamics principles to determine the relationship between the deposition rate and the quantity of residual gold in the bank. For the melting & anode casting process and, for the melting & bullion casting process, discrete-event simulation was used.

However, hybrid simulation also presents disadvantages, such as the need for care in choosing models for each component of the system, the complexity of implementation, and the difficulty in analyzing and interpreting the results obtained (Scheidegger et al. 2018).

In some cases, such as in the modeling of fluids, grains, and other materials, hybrid simulation can be used in conjunction with discrete event simulation. The discretization of entities can be used to model the materials in a discrete manner, while continuous simulation is used to model the macroscopic behavior of the materials. However, the choice of the most appropriate simulation technique depends on the specific characteristics of the problem to be analyzed and the tools available for its modeling.

Harrel et al. (2011) define each component of a system as Entities, Activities, Resources, and Controls. Banks et al. (2009) define the entity as an object of interest within a system, such as parts or clients. Since this work focuses on entity sizing, we will define an entity as any item processed throughout the system, such as products, customers, and documents.

Regarding entity sizing, the focus of this work is on situations where an entity represents multiple real-world objects. Thus, we will define **entity size** as the number of real-world objects represented by a single entity in a simulation model. For instance, in a coaching work with which one of the authors was involved, he had to build a model for a tomato can process unit to check whether the facility, given the mix of products, could handle the demand. In this case, it was defined that one entity would be equivalent to 120 tomato cans, i.e., entity size = 1/120 (entities per can). We can also define **entity multiplicity** as the inverse of entity size, whose numeric value may be more "palatable" (e.g.: 120 cans per entity).

Kogler and Rauch (2018) surveyed several simulation models for the wood supply chain. In this case, this issue arises since the entity size could be one log, one full log truck, one ton, etc. depending on the simulation model and objectives. Another example can be found in Chen et al. (2002). They studied logistics activities in a chemical plant; since most chemical production involves continuous flow; entity discretization was an important issue. The plant produces three different types of dry chemicals in one production line, one at a time. The chemical product is then transferred to intermediate silos, which is again transferred to one of two specific silos: for packing the product in 2 tons package or 5 tons package. Since it is a discrete event model (although, according to the authors, the model could be built with the continuous simulation paradigm), they needed to choose a convenient entity size. They initially established that one entity was equivalent to 2 tons (because of the small package capacity). However, since there were 2 tons and 5 tons packages, and the latter is not multiple to the former, they finally decided to assume that one entity equals 1 ton. Literature shows that this kind of rationale is common, and the same ratio will be used in one of the case studies described in Section 3. Nevertheless, a question arises: is it the best value?

Although there are recognized frameworks and formalisms for both hybrid discrete and continuous systems (Giambiasi et al. 2001; Lau et al. 2014), as well as several works on discrete event simulation for continuous systems (Fioroni et al. 2007; Kabadurmus et al. 2010; Béchard and Côté 2013), there are practically no references in the literature referring to the specific issue of the optimal entity size into a discrete event simulation model (for the multiple entity cases). Most discussions on the discretization of continuous systems tackle the discretization of the time (Sterman 2000; Giambiasi and Carmona 2006; Nutaro 2007; Murata et al. 2010). In the next section, we will expand the understanding of this issue by analyzing one prototype and two practical cases.

It is possible to infer that if the entity multiplicity (1/entity size) is low, the simulation model should have more entities and take more time to run. Nevertheless, if the multiplicity is too high, the computational time should be optimized, but the precision of results can be the primary concern (due to discretization errors). Therefore, we suggest that there should be an optimum size that balances precision with running time. Figure 1 addresses this point.

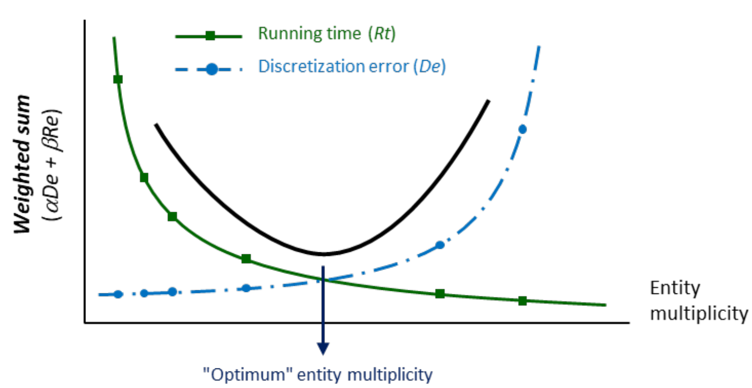

Figure 1: Balance between discretization error versus running time.

3 CASE STUDIES

3.1 Prototype Case

To evaluate the influence of entity size and its precision, we first build a simple M/M/1 queuing model with λ equal to 6 entities per hour (mean interarrival time of 10 minutes) and μ equal to 7.5 entities per hour (mean service time). In this baseline scenario, we assumed that one entity is equal to 1 ton. Then, we created alternative scenarios varying entity size. Scenario 1 establishes that one entity = 10 tons; scenario 2: one entity = 100 tons; scenario 3: one entity = 1,000 tons. To create 3 equivalent models, the mean interarrival times and the mean service times were multiplied by the entity size in each scenario. The KPIs evaluated were total production, in tons, at the end of the simulation period and server utilization. Other KPIs such as queuing size or time were not evaluated since they are dependent on entity size and will not allow us to compare results. Table 1 summarizes the experiment: all errors were calculated regarding the baseline scenario. Total simulation time and warm-up period for all scenarios were dictated by scenario 3: 10,000,000 minutes of warm-up and 100,000,000 minutes of results collection period. Replications were dimensioned to provide less than 1% precision around the mean of KPIs (in all scenarios, 5 replications were conducted). The simulation model was developed in SIMUL8 simulation software in an Intel Core i7 9700F @ 3.00GHz computer. Simulation running time was relatively fast, varying from around 30 seconds (best case - scenario 3) to 1 minute (worst case - baseline scenario).

Table 1: Results from the prototype model.

	Baseline	Scenario 1	Scenario 2	Scenario 3
Production (tons)	9,999,065.8	9,999,074.0	10,012,540.0	10,066,800.0
Server utilization	80.0%	80.0%	80.0%	80.5%
Abs. relative error (production)	N/A	0.0001%	0.1348%	0.6774%
Abs. relative error (utilization)	N/A	0.0247%	0.0505%	0.6019%

As can be seen from Table 1, the maximum error occurs in scenario 3 (entity size is 1,000 tons/entity), but its value is below 1%. Therefore, no huge lack of precision will occur if we increase entity size, at least in this simple model.

3.2 PP-Production Facility

The objective of this study (a consultancy project) was to evaluate bottlenecks and the productive capacity with the addition of a 3rd extrusion line in a polypropylene (PP) production facility. The model included the receiving process, extrusion, and expedition of granulated plastic material (PP). There were 7 kinds of raw materials which were received either from trucks carrying big bags or bulk trucks. The arrival policy of trucks was triggered when each silo reached reorder point level. Raw material was directed to one of the seven storage silos, depending on the raw material type. It was considered a production plan with 6 families of products, each family composed of a mixture of raw materials. When the production of one given line started, the material was transferred to intermediate silos that would feed the extrusion line. After the extrusion process, the product would go to the final silos that would feed the homogenization process and packaging, which could be either big bags or 25 kg packs. Production lines set-up and breakdowns were also considered in this model. The schematic layout is depicted in Figure 2.

The focus here is to define the entity size for the product (Polypropylene granules - see Figure 3). Their characteristic is between 3 to 5 mm plastic granules with an approximate density of 0.9 g/cm^3. Therefore, the weight of each granule is around 0.06 g. It is clear that it is not worth assigning one entity equal to 1 granule. By doing this and given that the extrusion average velocity is around 1,650 kg/h for each line, then in one hour, not less than 80 million entities will be “flowing” through the model, which will not be computationally feasible.

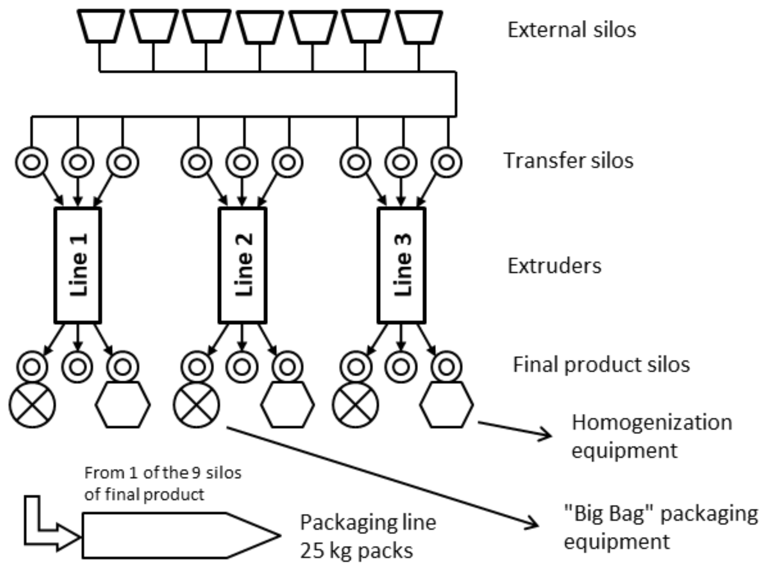

Figure 2: Schematic layout of polypropylene facility.

Figure 3: Polypropylene granules.

So, in the project, it was decided that 1 entity = 1 ton, based on personal experience because, besides the easy interpretation (1 entity = 1 ton), 1 ton is one good resolution for external silo content (that has the maximum capacity of 240 tons). Nevertheless, at this point in the project, we did not conduct a detailed study regarding the entity size. The original model was developed in SIMUL8 simulation software and it ran relatively fast for a 1-month production plan simulation. Since the model was initialized with PP stocks from the production plan, no warm-up period was required. Replications were dimensioned to provide less than 1% precision around the mean of KPIs.

Now, this model was retaken and configured by a parameter called entity size (=1/multiplicity), so variations can be made. This was done from the external silos to the final process (packaging). For the receiving process, we maintained 1 entity = 1 load of the truck which was converted to entity numbers depending on the parameter entity size for external silos feeding.

The key performance indicators (KPIs) were:

- Queue sizes and queuing times for big-bags and bulk trucks;
- Minimum, average, and maximum levels of material for each external silo;
- Production line and post-extrusion equipment utilization;
- Overall production level, with breakdown by line and type of package.

Average results of 100 runs for each KPI were stored, ranging the entity size from 0.03125 entities per ton (multiplicity = 32, since 32 tons equals 1 entity) to 32 entities per ton (multiplicity = 0.03125), along with the running time for each trial, equivalent to 100 simulation runs. Against our expectation, results were quite insensitive to entity size, especially the most important one: the overall production (see Table 2 - other KPIs will be omitted due to space constraints).

Table 2: Simulation results for the PP-production facility.

Entity size	Overall production (ton)	Trial running time (sec)
0.03125	-	-
0.0625	3,133.4	17
0.125	3087.3	18
0.25	2988.7	19
0.5	2986.5	22
1	2957.2	30
2	2943.8	42
4	2944.3	64
8	2944.6	109
16	2944.6	202
32	2944.7	384

With entity size equal to 0.03125 entities per ton, the model provided an error because it was configured to do so when there is stock out of the external silo. It occurred because the precision is very low with this entity size (32 tons = 1 entity).

Since the overall production did not seem to change above 32 entities per ton, this value became our reference. Then, we calculated the relative error of this KPI for every other value of entities per ton. Figure 4 shows the relative error of overall production by varying entity size.

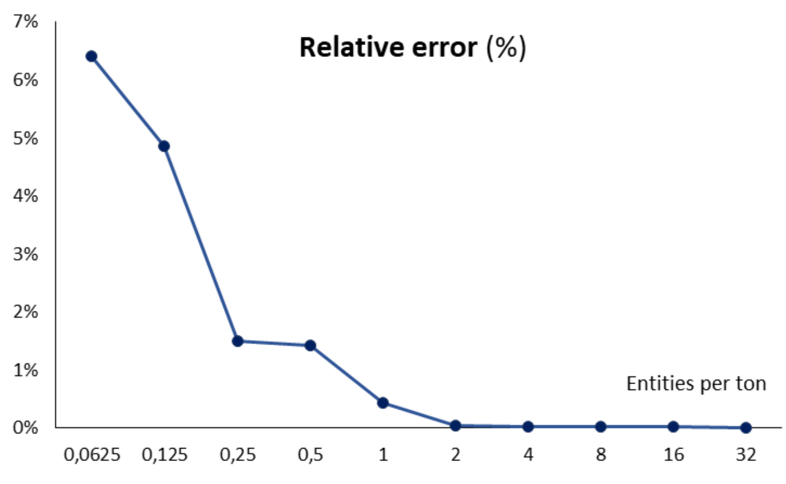

Figure 4: Relative error by varying entity size.

It is clear to see that, for entity size higher than 2 tons per entity, the error was near 0. For an entity size of 0.0625, the error was above 6%, which is still relatively low when evaluating the variation of entity size (from 32 to 0.00625).

On the other hand, simulation running time was very sensitive to entity size. Figure 5 shows simulation running time versus entity size.

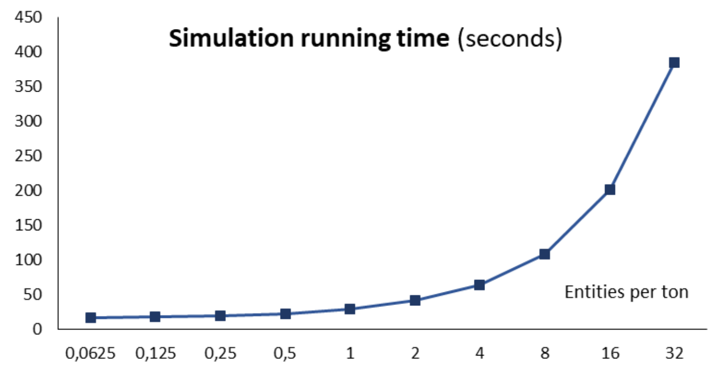

Figure 5: Running time versus entity size (entities per ton).

As hypothesized, running time follows practically an exponential pattern as entity size rises, which can lead to unfeasible running times. The case study model is of medium size complexity but if we deal with complex models, it is clear that entity size choice could be critical.

3.3 Ice Cream Bar Production Facility

In this project, a simulation model was used to analyze a new ice cream production line. Although all the production stations for the ice cream process were modeled, the focus of this work was to identify improvements in the raw materials supply for this line.

The production process included the following steps: after pumping the mix from the flavor vat, the mixture passed through freezers (2) at a constant rate and was fed to the extruder. There were two extruders where some stock-keeping units (SKUs) were also variegated, being fed at a constant rate. Sticks were automatically added at this point. After the extrusion process, each bar was put on a plate conveyor that passed through the hardener. Chocolate (and inclusions) would be automatically pumped into the chocolate dip tank as well as the dazzle at dazzle tank (for SKUs that require dazzle). Bars would be automatically transferred from the hardening end conveyor by the slat conveyor to be dipped in the tanks. The dipping process batched in 12 bar lanes across. Then, the bars would be automatically transferred into the wrapper station by a pick-and-place robot (6 lanes per movement) and after wrapping, the 6 bars would be split into 3 + 3 that would feed the two-carton equipment. Each carton would pack 3 bars and after cartooning, it would pass through a metal detector and weight check (all equipment alongside out conveyor) and then goes to the case packer machine that would pack 12 cartons per case.

For simulation purposes, each part of the process (Flavor vats, Freezer, Extruder, Hardener, Chocolate tank, Wrapper, Cartoner, Metal detector, weight check, and case packer) was treated as a "black box" running at freezer velocity (300 bars/min). Figure 6 shows the overall process sequence.

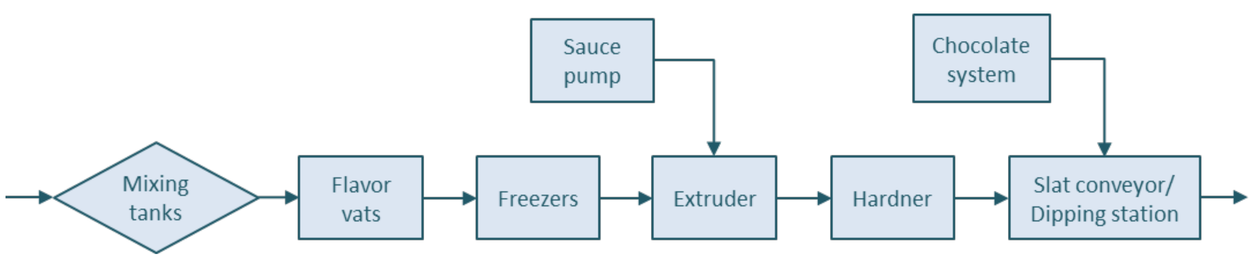

Figure 6: Steps for ice cream bars production.

For feeding the line with raw materials there were some operators: freezer/flavor vat operator, that fed the initial ice cream mix; sauce pump operator, that fed sauce pump; chocolate operator that fed chocolate bars into the chocolate tank; wrapper operator that fed wrap rolls; packer operator that fed the cartoner equipment; hoypack operator that fed the case packer machine with corrugate boxes. There was also one QA person to check if process parameters were within desirable limits. All replenishment logic was guided by two parameters for each raw material: replenishment quantity and replenishment point.

We simulated the line considering several production plans, breakdowns, and cleaning between the production of different SKUs. Since it was a very fast line, running at 300 bars/minute, we considered that one simulation entity was equivalent to 360 bars, which was equivalent to 10 cases (each case contains 36 bars). This would result in 1.2 minutes/entity. In the original study, we performed several scenarios varying replenishment parameters, the number of operators, machines, and production plan, but for studying the effect of entity size we fixed the initial scenario with a fixed production plan and configured the simulation model to allow to parametrize entity size. The running length was set to 1-month of production with no warm-up since the objective of the simulation is to evaluate the system under a 1-month program plan. Replications were set to provide less than 1% precision around the mean of KPIs.

We ran the model considering that one entity equals 36, 144, 180, 216, 360, 540, 648, 900, 1,800, and 10,800 ice cream bars. We also considered 18 bars per entity and 1 bar per entity, despite that the case packer used 36 bars in a single case. For confidentiality, all KPIs were renamed to KPI 1-KPI 62. These KPIs covers machine and operator utilization, stock-outs (binary KPI), average time to replenishment, and other indicators.

Table 3 shows simulation results for each entity size (10 replications per trial), omitting most KPIs due to space constraints. Two KPIs are very important: KPI 1 (total production of bars in one month) and KPI 2 (the asset intensity). KPI 3 is the total number of entities produced, which presents huge differences across scenarios since in each scenario we varied entity size.

Table 3: Simulation results for ice cream bar production facility.

Entity size	KPI 1	KPI2	KPI3	Trial running time (sec)
1	-			-
18	10,818,971.30	80.40	601.054.0	1,082
36	10,774,684.80	80.46	299,296.8	493
144	10,741,392.00	80.21	74,593.0	128
180	10,692,000.00	79.84	59,400.0	103
216	10,739,001.60	80.19	49,717.6	87
360	10,773,576.00	80.45	29,926.6	52
540	10,448,190.00	78.02	19,348.5	42
648	10,736,776.80	80.18	16,569.1	38
900	10,771,110.00	80.43	11,967.9	30
1,800	9,984,060.00	74.55	5,546.7	21
10,800	10,841,448.35	80.95	1,003.8	12

Once again, the most important results were quite insensitive to entity size, showing higher differences for entity size over 1,800 bars per entity. If we consider the results for 18 bars per entity as reference (we did not run the scenario of 1 bar per entity due to simulation time constraints) and compare the result for 900 bars per entity (removing KPI 2 because it clearly differs) we can plot the relative difference errors across all 62 KPIs (Figure 7). For the majority of KPIs, relative errors were less than 1% with an average of -0.5%. For KPI 5, it provided -8,7%. KPI 5 is the percentage of planned maintenance which is very low (around 2%), so this error has little impact.

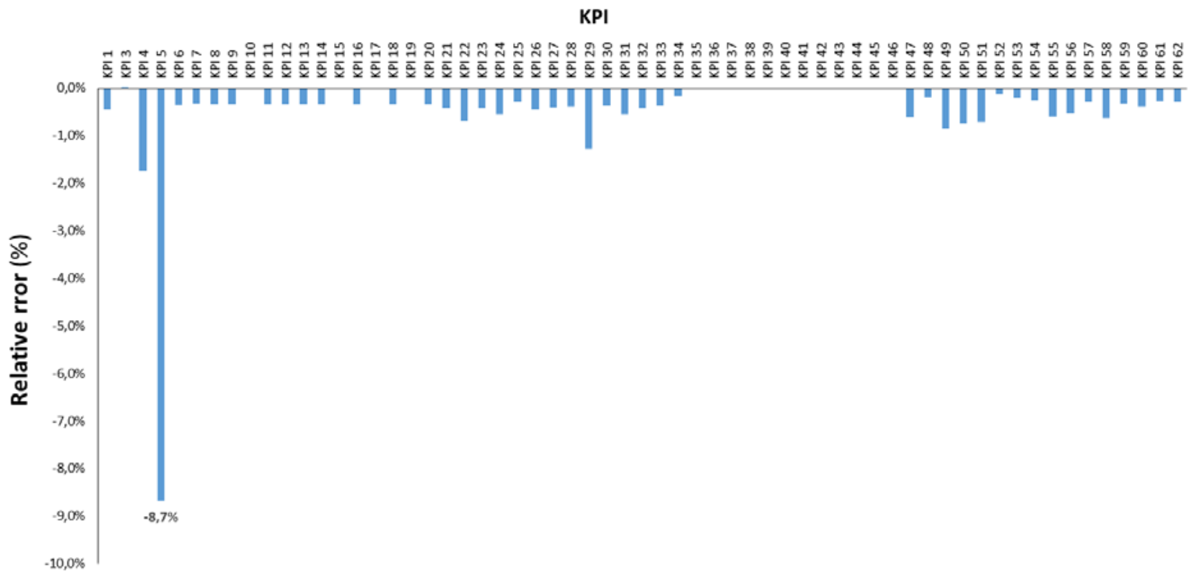

Figure 7: Relative error across the KPIs for 900 bars per entity x 18 bars per entity.

For simulation running time analysis, Figure 8 shows a plot of the running time in seconds versus entity size. It is important to note that, because of entity size is equal to 1 bar per entity, the running time was estimated initially by SIMUL8 simulation software as 6 hours, so the run was not performed due to time constraints.

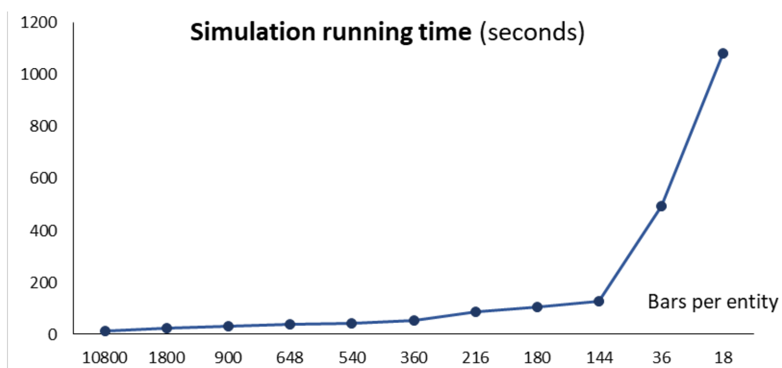

Figure 8: Running time versus multiplicity (bars per entity).

4 DISCUSSIONS

Based on these cases studies, we can reach the following conclusions:

- More entities in a model provide better results accuracy: true, however, the accuracy in general is quite insensitive to entity size only providing significant errors when entity size is very low.
- More entities in the model provide longer running times: true. Usually, it resembles an exponential curve.

The central question is which is the "optimum" entity size? Of course, if one applies Occam's Law, there is a tendency to lower the number of entities inside the model, leading to an accuracy loss or even errors, as the second case study showed. If we adopt the hypothesis that currently, computer and simulation software technology provides relatively short runs, then we are looking to a parallel question: "What is the minimum entity size that can provide the desired accuracy?".

From the results obtained, precision decreases exponentially when increasing entity size and simulation time decreases exponentially when increasing entity size. We perform some adjustment tests and discovered the best-fit mathematical function using Least Squares Method for several examples. Figure 9 shows how one KPI varied according to entity size and the best-fit mathematical function.

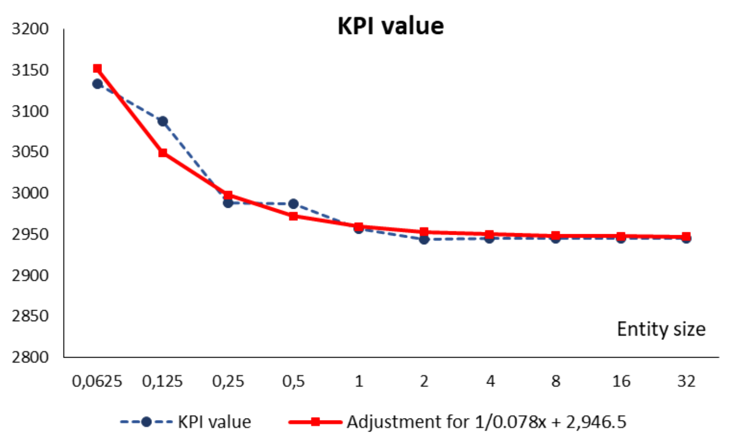

Figure 9: KPI value versus entity size - empirical and theoretical.

Therefore, we can address an algorithm to derive the optimum entity size to reach desired KPIs precision:

1. Parametrize the simulation model to account for different entity sizes. Although this can generate some work, it is normally feasible to do this.
2. Choose two different entity sizes X1 and X2. Choose one KPI and simulate entity sizes 1 and 2. The number of replications to provide the desired confidence interval must be already determined. Record KPI for X1 (Y1) and KPI for X2 (Y2).
3. Using the equation provided below, create a two-variable system of equations, and determine parameters *a* and *b*, by substituting the points (X1, Y1) and (X2, Y2):

$$f(x) = \frac{1}{ax} + b$$

4. Recall that if *x* tends to infinite, the KPI tends to *b*, therefore, it is the KPI with the best precision;
5. Choose an entity size X3, compute *f*(X3), and compare it to ideal precision *b*. If the difference is acceptable, record X3. If not, increase X3 until the difference is admissible.
6. Go back to Step 1 if there are more KPIs to compute (new iteration);
7. The "optimum" entity size will be the maximum entity size found in all iterations (all KPIs).

This algorithm is based on the premise that there is an asymptotic convergence of a given KPI when increasing entity size, as observed in the case studies. Furthermore, to keep the algorithm simple and straightforward, only 2 points (minimum number necessary) are used to infer parameters *a* and *b*. Therefore, deviation errors may occur. If one would like to minimize these errors, we advise using more points and use the least square method as shown in Figure 9.

Notice that the algorithm above only is concerned with precision. If the running time for any entity size is computationally prohibitive, then we must "relax" the desired precision to allow less computational effort or change the model to consider a simpler one that runs faster.

5 CONCLUSIONS AND FUTURE WORK

This paper addresses the issue of verifying the relationships between entity size and both results accuracy and computational performance for discrete event models in which entities can be multiple of a given unit. Looking at the literature, we found that research is scarce in this area; therefore, this work can be considered one of the first to explore this issue.

Given the case studies, it was possible to show that the risen hypotheses were true (more entities in the model provide better results accuracy and provides longer running times). This work “proved” Occam’s razor principle, indicating that we can use an entity size that does not take too much time to run, and still have the results lying in a good precision range. We found that is also possible to derive an algorithm to determine the “best entity size” to obtain the desired precision (minimize discretization error), which is the major advance of it, since when a minimum desired precision is reached, the computational running effort may be saved. If several scenarios would have to be run, since it was shown that computational running effort varies exponentially with entity size, the total time of a simulation study can be shortened. Future research can be done to evaluate the time savings of this study.

As another future research, we intend to deep study the errors involved in the proposed algorithm to enhance its robustness. Since this algorithm was developed from a generalization of case studies, it is also advised to further test it in other cases.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Stewart Robinson for his invaluable insights for this article improvement.

REFERENCES

Amaral, J. V. S., J. A. B. Montevechi, R. C. Miranda, and W. T. S. Junior. 2022. “Metamodel-Based Simulation Optimization: A Systematic Literature Review”. *Simulation Modelling Practice and Theory* 114:102403.

Banks, J., J. S. Carson II, B. L. Nelson, and D. M. Nicol. 2009. *Discrete-event Simulation*. 5th ed. New Jersey: Prentice-Hall.

Béchard, V., and N. Côté. 2013. “Simulation of Mixed Discrete and Continuous Systems: An Iron Ore Terminal Example”. In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 1167-1178. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Brailsford, S. C., T. Eldabi, M. Kunc, N. Mustafee, and A. F. Osorio. 2019. “Hybrid Simulation Modelling in Operational Research: A State-of-the-Art Review”. *European Journal of Operational Research* 278(3):721–37.

Chen, J. C., Y. M. Lee, and P. L. Selikson. 2002. “A Simulation Study of Logistics Activities in a Chemical Plant”. *Simulation Modelling Practice and Theory* 10(3-4):235–245.

Chwif, L., M. R. P. Barretto, and R. J. Paul. 2000. “On Simulation Model Complexity”. In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fish-wick, 449-455. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fioroni, M. M., L. A. G. Franzese, C. E. Zanin, J. Furia, L. de Toledo Perfetti, D. Leonardo, and N. L. da Silva. 2007. “Simulation of Continuous Behavior Using Discrete Tools: Ore Conveyor Transport”. In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1655-1662. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Giambiasi, N., B. Escude, and S. Ghosh. 2001. “GDEVS: A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems”. In *Proceedings of the 5th International Symposium on Autonomous Decentralized Systems,* March 26-28, Dallas, Texas, 464-469.

Giambiasi, N., and J. C. Carmona. 2006. “Generalized Discrete Event Abstraction of Continuous Systems: GDEVS Formalism”. *Simulation Modelling Practice and Theory* 14(1): 47-70.

Harrel, C. R., B. K. Ghosh, and R. Bowden. 2011. *Simulation Using ProModel*. 3. ed. New York: McGraw-Hill Education.

Kabadurmus, O., O. Pathak, J. S. Smith, A. E. Smith, and H. Yapicioglu. 2010. “A Simulation Methodology for Online Process Control of Hot Mix Asphalt (HMA) Production”. In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 1522-1533. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kogler, C., and P. Rauch. 2018. “Discrete Event Simulation of Multimodal and Unimodal Transportation in the Wood Supply Chain: a Literature Review”. *Silva Fennica* 52(4).

Kumar, A., and S. Kumar. 2018. "Higher Production Plan Realization Through Dynamic Simulation". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 4097-4098. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lau, S. C., M. Lu, and C. S. Poon. 2014. "Formalized Approach to Discretize a Continuous Plant in Construction Simulations". *Journal of Construction Engineering and Management* 140(8):04014032.

Mosavi, A., S. Shamshirband, E. Salwana, K. Chau, and J. H. M. Tah. 2019. "Prediction of Multi-Inputs Bubble Column Reactor Using a Novel Hybrid Model of Computational Fluid Dynamics and Machine Learning". *Engineering Applications of Computational Fluid Mechanics* 13(1):482–92.

Murata, M., J. Satsuma, A. Ramani, and B. Grammaticos. 2010. "How to Discretize Differential Systems in a Systematic Way". *Journal of Physics A: Mathematical and Theoretical* 43(31):315203.

Nutaro, J. J. 2007. "Discrete-Event Simulation of Continuous Systems". In *Handbook of Dynamic System Modeling*, edited by Fishwick, P.A. New York: Chapman and Hall/CRC.

Robinson, S. 2011. "Choosing the Right Model: Conceptual Modeling for Simulation". In *Proceedings of the 1993 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 1428-1440. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Salt, J. D. 1993. "Keynote Address: Simulation Should be Easy and Fun!". In *Proceedings of the 1993 Winter Simulation Conference*, edited by G. W. Evans, M. Mollaghhasemi, E. C. Russel, and W. E. Biles, 1-5. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Santos, C. H., J. A. B. Montevechi, J. A. Queiroz, R. C. Miranda, and F. Leal. 2021. "Decision Support in Productive Processes through DES and ABS in the Digital Twin Era: A Systematic Literature Review". *International Journal of Production Research* 60(8):2662–2681.

Scheidegger, A. P. G., T. F. Pereira, M. L. M. Oliveira, A. Banerjee, and J. A. B. Montevechi. 2018. "An Introductory Guide for Hybrid Simulation Modelers on the Primary Simulation Methods in Industrial Engineering Identified through a Systematic Review of the Literature". *Computers & Industrial Engineering* 124:474–92.

Sterman, J. D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York: McGraw-Hill Higher Education.

van der Zee, D. J. 2019. "Model Simplification in Manufacturing Simulation - Review and Framework". *Computers & Industrial Engineering* 127: 1056–1067.

Zhang, D., E. A. Del Rio-Chanona, P. Petsagkourakis, and J. Wagner. 2019. "Hybrid Physics-Based and Data-Driven Modeling for Bioprocess Online Simulation and Optimization". *Biotechnology and Bioengineering* 116(11):2919–30.

AUTHOR BIOGRAPHIES

LEONARDO CHWIF graduated in Mechanical Engineering (Mechatronics Specialization) in 1992 at the University of São Paulo and received his M.Sc. degree in 1994 and his Ph.D. in Simulation in 1999 from the same University. Upon graduation, Dr. Chwif joined the Brazilian branch of Mercedes-Benz (truck division). Then he joined the Brazilian branch of Whirlpool Corporation. He spent one year at Brunel University as a research visitor at the Centre for Applied Simulation Modeling. Currently, he is the CEO of Simulate Simulation Technology. Dr. Chwif also teaches introductory graduate simulation courses at Escola de Engenharia Mauá and FEI. His e-mail address is leonardo.chwif@maua.br

WILSON INACIO PEREIRA graduated in Electrical Engineering in 2000 at the Escola de Engenharia Mauá, São Paulo, Brazil and, in 2011, graduated as Higher Education Teaching Specialist at the Universidade Municipal de São Caetano do Sul, São Paulo, Brazil. He was a teacher at Escola de Engenharia Mauá for 17 years and now is working along with Simulate Simulation Technology in discrete-event simulation projects. His interest is in the area of discrete-event simulation mathematical programming and metaheuristics. His email address is wilson.in.pereira@gmail.com.

JOSE ARNALDO BARRA MONTEVECHI is a Titular Professor of the Production Engineering and Management Institute at the Federal University of Itajubá, in Brazil. He holds degrees of Mechanical Engineering from the Federal University of Itajubá, an M.Sc. in Mechanical Engineering from the Federal University of Santa Catarina, and a Doctorate of Engineering from the Polytechnic School of the University of São Paulo. His research interest includes Operational Research, Simulation, and Economic Engineering. His email address is montevechi@unifei.edu.br.