



Exploring the relationship between simulation model accuracy and complexity

Stewart Robinson

To cite this article: Stewart Robinson (2022): Exploring the relationship between simulation model accuracy and complexity, Journal of the Operational Research Society, DOI: [10.1080/01605682.2022.2122740](https://doi.org/10.1080/01605682.2022.2122740)

To link to this article: <https://doi.org/10.1080/01605682.2022.2122740>



© 2022 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 20 Sep 2022.



Submit your article to this journal [↗](#)



View related articles [↗](#)



View Crossmark data [↗](#)

Exploring the relationship between simulation model accuracy and complexity

Stewart Robinson

Newcastle University Business School, Newcastle upon Tyne, UK

ABSTRACT

Little is known about the relationship between the accuracy of a discrete-event simulation model and its complexity. Despite this, authors propose that increasing model complexity leads to improved accuracy with diminishing returns. In this paper we explore whether this proposition is correct by applying successive simplifications, some in different sequences, to three simulation models and measuring the impact on the accuracy of the model output. The results show that while increased complexity often leads to improved accuracy, this is not always the case. Also, we see that even when there is a positive correlation between complexity and accuracy, the returns to accuracy from greater complexity can be diminishing, constant or increasing. So, we conclude that the proposition about the relationship between accuracy and complexity is not correct. However, it does remain a useful heuristic for guiding modellers when considering the scope and level of detail at which to model a system.

ARTICLE HISTORY

Received 25 January 2022

Accepted 2 September 2022

KEYWORDS

Discrete-event simulation;
model; accuracy;
complexity; simplification

1. Introduction

It would appear obvious that when presented with two simulation models of the same system, the one which incorporates more aspects of the real system will be the more accurate. In other words, higher fidelity, and by nature more complex, models are more accurate. Of course, there is a concern over the accuracy of more complex models arising from whether the requisite knowledge and data are available to populate those models; but should the knowledge and data exist, then surely the statement above holds. Indeed, so obvious is this statement, that there are very few studies that actually investigate the relationship between a model's accuracy and its complexity.

The statement above would appear to support a view that modellers should develop the highest fidelity models possible. In apparent contradiction to this, many authors in fact argue for the need to avoid overly complex models and so to develop simpler models that omit or simplify aspects of the real system (e.g. Robinson, 2008; Ward, 1989). Are these authors saying that simpler models are more accurate? No, this line of argument is not fundamentally driven by a belief that simpler models are more accurate, but by the view that the many benefits of simpler models (e.g. faster to build, quicker to run, easier to understand) outweigh the consequent loss in accuracy. Of course, a valid model is one that is sufficiently

accurate for its purpose and so as long as a simplification does not force a model below the desired threshold of accuracy, the validity criterion is still achieved.

More subtly, and underlying the argument for simpler models, is a sense that accuracy only reduces relatively modestly with initial reductions in complexity. However, if simplification is taken too far, accuracy will increasingly fall away, and a model will drop below the threshold required for validity. In other words, there are diminishing returns to accuracy from continuing to increase complexity. Again, this aspect of model accuracy and complexity has rarely been studied.

It is clear that we have only a limited understanding of the relationship between model accuracy and complexity. The purpose of this paper is to explore that relationship. Our aim is to determine whether increased complexity does lead to greater accuracy, and whether the improvements to accuracy diminish as complexity is increased. We investigate this relationship by applying different levels of simplification to three discrete-event simulation models and measuring the impact on the output from the model. In the next section we discuss the existing literature on model simplification and the relationship between accuracy and complexity. In section 3 we present the research approach as well as detailing the three models and the simplifications that we apply to each. Section 4 presents the results

from the models and in [section 5](#) we draw conclusions about what this tells us about the relationship between model accuracy and complexity. We conclude by discussing the limitations of this work and the potential for future research ([section 6](#)). The contribution of this paper is to provide detailed empirical evidence on the relationship between discrete-event simulation model accuracy and complexity.

2. Model simplification and the relationship between accuracy and complexity

The modelling literature emphasises the need to reduce complexity by simplifying models. For instance, Robinson (2014) calls for modellers to “build the simplest model possible to meet the objectives of the simulation study.” Pidd (1999) identifies six modelling principles, three of which focus on modelling simply: “model simple; think complicated”; “be parsimonious; start small and add”; “divide and conquer; avoid mega-models.” Powell (1995) also recommends that modellers seek to “divide and conquer” in order to create simpler models.

The line of argument is that simpler models provide many benefits, while only marginally reducing model accuracy, so long as simplification is not taken too far. Tako et al. (2020) identify the dangers of oversimplification leading to an invalid (insufficiently accurate) model. On the other hand, Brooks and Tobias (2000) argue that simplification does not always mean that a model’s accuracy is reduced.

Innis and Rexstad (1983), Fishwick (1988), Ward (1989), Sevinc (1991), Salt (1993), Brooks and Tobias (2000), Chwif et al. (2000), Lucas and McGunnigle (2003) and Urenda Moris et al. (2008) all discuss the benefits of simpler models, which include:

- More rapid model building
- Helps to ensure the focus is on the most important aspects of a system
- Fewer data requirements
- Easier to verify and validate
- Faster run-speed
- More flexible and so easier to change
- Easier to understand

On the last point Ward (1989) separates out the idea of constructive simplicity from model transparency. While simpler models may often be easier to understand, they can become less transparent if the simplifications lead to a model that appears to be very far removed (abstract) from the system it is representing.

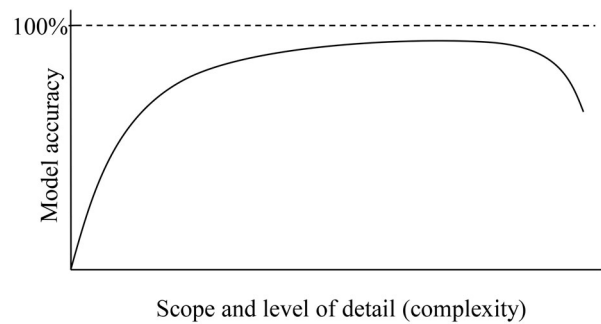


Figure 1. Simulation model accuracy and complexity (Robinson, 2008).

An underlying assumption in the literature on model simplification is that there is a positive correlation between a model’s accuracy and its complexity. This is illustrated by Robinson (2008) who proposes the relationship in [Figure 1](#) (an earlier version of this graph can be found in Robinson, 1994a). What the graph shows is that increasing the complexity of a model (its scope and level of detail) leads to a greater level of accuracy, but with diminishing returns. Ultimately, an over-elaboration of the model may lead to reduced accuracy (right-hand end of the graph) since there is insufficient knowledge or data to support the detail in the model. Whilst [Figure 1](#) is intuitively appealing, Robinson provides no empirical evidence to substantiate the proposed relationship between model accuracy and model complexity.

Leinweber (1979) proposes a very similar relationship when discussing the impact of model complexity on model error in the context of economic and policy modelling. He identifies two modelling errors: the error of specification (e_s) and the error of measurement (e_M). The former refers to the structure of the model and its relationships, and the latter to the input data for the model. As complexity increases, the error of specification reduces; at least it should, otherwise there is no sense in adding additional complexity to a model. However, the “compound error of measurement” increases with model complexity. Leinweber argues that errors in input data compound as more and more arithmetic operations are performed on them. He depicts these relationships and the resulting total error (e_T) as shown in [Figure 2](#). Although Leinweber briefly discusses policy models that predict coal consumption and air pollution emissions, like Robinson, he does not offer any empirical evidence for the relationships proposed in [Figure 2](#).

Both Robinson and Leinweber highlight the importance of input data to model accuracy. A number of studies identify the effect of input data uncertainty on the simulation output and propose means for addressing this. For instance, Ankenman and Nelson (2012) describe an approach for

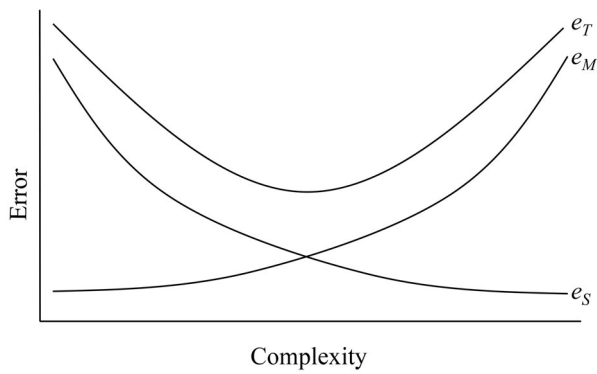


Figure 2. Model error and complexity (based on Leinweber, 1979).

assessing the effect of input data uncertainty. This is later developed by Song and Nelson (2015) who provide a simplified approach which requires only one diagnostic experiment. Meanwhile, both Zouaoui and Wilson (2004) and Barton et al. (2014) describe how to derive output data confidence intervals that account for uncertainty in the input data.

However, attempts to empirically explore the relationship between model accuracy and complexity are few and far between. Thomas and Charpentier (2005) demonstrate that reducing a manufacturing scheduling model from 275 “modules” to 79 has only a minimal impact on the output. Similarly, Hung and Leachman (1999), Brooks and Tobias (2000), Piplani and Puah (2004), Johnson et al. (2005) and Urenda Moris et al. (2008) implement simplification procedures on simulation models and all show that the output from the more abstract models correspond well to the output from the more detailed models; in Brooks & Tobias’ case the outputs correspond exactly. As such, these studies are demonstrating the flat part of the curve in Figure 1, that is, only marginal changes in accuracy with reduced complexity.

Only three publications explore model accuracy and complexity more extensively. Hood (1990) explores the impact of successive simplifications on the performance of a semiconductor manufacturing simulation model. Starting with a validated and detailed model, she implements a number of model simplifications in turn and measures the impact on the key outputs. The simplifications that are implemented are: ignoring rework, modelling rework by increasing processing times, no operator breaks, no operators, and no machine failures. Hood demonstrates that these reductions in model content degrade the model’s performance often by more than 20%, especially when the line is heavily loaded. By only testing one simplification at a time, what Hood does not explore is the incremental effect of successive simplifications.

Huber and Dangelmaier (2009) do explore the impact of successive simplifications on material flow simulations. Their motivation is to explore the automatic generation of simplified models by implementing

a simplification algorithm. They adopt two approaches to simplification:

- *Aggregation*: replacing a set of components with a smaller set
- *Omission*: removing components

They test their approach on three simulation models with 151 (“A”), 133 (“B”) and 61 (“C”) components and measure the impact of successive simplifications on the models’ output. The full set of simplifications leads to 45%, 60% and 20% “remaining complexity” for models A, B and C respectively. The “behavioural deviation” of the output is 5%, 10% and 13% respectively for the three models. Huber & Dangelmaier map the trajectory of successive simplifications by showing a graph of complexity and behavioural deviation for each step in their simplification procedure. What this shows is a tendency for greater behavioural deviation as the complexity of a model is reduced, but that at some steps a simplification can reduce the behavioural deviation. The graph is certainly a long way from the smooth curves proposed by Robinson (2008) and Leinweber (1979).

Finally, Lakshika et al. (2017) explore the relationship between model complexity and fidelity with three agent-based models of conversational group dynamics, sheepdog herding and traffic. Using four different measures of complexity, they demonstrate that for all three models fidelity improves with increased complexity, but with diminishing returns. Graphs for the three models showing the fidelity-complexity relationship are very similar to that proposed by Robinson (2008), but without demonstrating reduced fidelity through eventual over-elaboration.

3. Research approach

Our aim is to explore the relationship between the accuracy of a simulation model and its complexity. Specifically, we investigate the veracity of the propositions of Leinweber (1979) and Robinson (2008) that accuracy improves with diminishing returns as complexity is increased, up to the point that measurement errors start to dominate. We approach this by exploring the impact of successive simplifications with models of three different systems: a manufacturing model, a service model and a case study model. For the case study we use the Panorama Televisions model (Robinson, 2014). With the Manufacturing model we employ two types of simplification:

- *Aggregation*: replacing groups of machines and conveyors with a delay
- *Omission*: removing machines and conveyors from the model

We achieve the aggregation by replacing sections of the model with black-boxes (Robinson, 2014), or “blocks,” and generating empirical distribution data from the full model to estimate the time parts spend in each block. With the Service model we use a similar aggregation approach. For the Panorama Televisions case we simplify the model by successively removing detail (omission). In this case we are not approximating sections of the model from data generated by a more detailed version. As such, the simplest version of the Panorama Televisions model could be created without having to ever create the more detailed versions of the model. The same is true for the Manufacturing model when we employ the omission approach to simplification.

All models were developed in the Witness simulation software (Lanner, 2022). Full details of the three models are provided below, including the simplifications that were implemented and the run strategy for each model. First, we discuss how we measure the complexity of the models and the collection of the results.

3.1. Measuring the complexity of the models

The analysis requires a measure of model complexity. Unfortunately, there is no agreed complexity measure for discrete-event simulation models. Proposed measures have focussed on approaches based on specific model representations, for instance, condition specification (Wallace, 1987), graph theory (Schruben & Yücesan, 1993; Yücesan & Schruben, 1998) and computational complexity theory (Jacobson & Yücesan, 1999). Yavari and Roeder (2012) describe an “enrichment level” method for measuring a model’s complexity, taking into account factors such as scope, speed, variance and bias.

These methods are largely motivated by the desire to determine the complexity of a proposed model to help determine the resources required for the development of that model, and also for the purposes of comparing alternative models (Wallace, 1987). In our analysis we need to measure the complexity of an existing model and then determine how subsequent simplifications impact on its complexity. As such, we use two approaches to measure complexity.

First, we use a measure based on the work of Brooks and Tobias (1996, 2000). They define a model’s complexity as being a combination of the number of components, the pattern of connections, and the complexity of calculations; noting that simplification involves a reduction in these elements. It is straightforward to determine the number of components and number of connections in a model.

However, it is not so straightforward to determine the complexity of the calculations. Although the number of lines of code could be used as a proxy, this does not account for the number of times each code segment is executed during a run. A full analysis of computational complexity would require considerable effort to achieve, if it is possible at all, especially given the stochastic nature of many simulation models. Indeed, Brooks and Tobias (2000) only use the number of machines and number of connections to measure complexity. In Brooks (2010), he adds a simple high/medium/low measure for calculational complexity. For our purposes, we will adopt a simple integer measure of calculational complexity based on a judgement about the level of detail for each model component.

Second, taking advantage of the fact that we have fully executable models, we will use the time it takes to complete a run of the model (“run-time”) as a proxy measure for its complexity. This aligns with the approach of using running time as a measure of complexity in computer science (Arora & Barak, 2009).

3.2. Collection of results

For each model we collect data for a key output statistic described with the model details below. In presenting the results, we use these data to determine the error (e) between the result from the simplified model (R_S) and the base model (R_B) as follows:

$$e = \left| \frac{R_S - R_B}{R_B} \right|$$

The R_S and R_B values are calculated as the average from 10 replications as described in the run strategy discussion below. The absolute value is used since we are primarily interested in the size of the error and not its direction. In the presentation of the results, we will report the actual error in the tables, but use the absolute value in the graphical representation. Note that we measure the size of the error against the base model and not the real system. In the absence of a real system, we treat the output from the base model as the best estimate of system performance.

Given that we are simplifying the models, we expect errors to occur due to the change in the model specification. We are not exploring errors in measurement (or data) as discussed by Leinweber (1979). In other words, we are not exploring the right-hand end of Robinson’s (2008) accuracy vs. complexity graph. As such, if Leinweber’s and Robinson’s propositions are correct, we expect our results to resemble the curve in Figure 3.

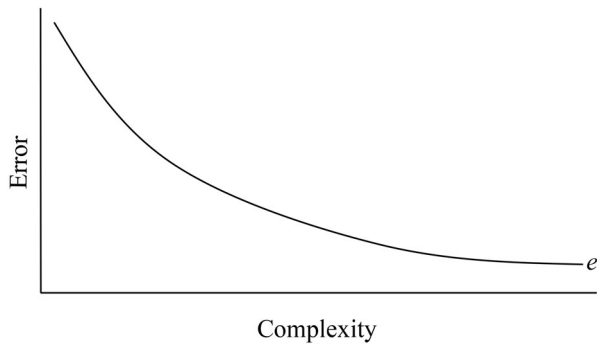


Figure 3. Expected Error vs. Complexity Graph.

Details of the Brooks & Tobias complexity measure for each model are provided in their descriptions below. In terms of recording the run-time, this was generated using the computer's clock time. In order to ensure consistency, the computer was disconnected from the Internet and all other tasks were shutdown for the duration of the runs. However, differences in the run-time for the same version of the model persisted. In order to address this, the run-time was averaged over 10 replications for each model version and 95% confidence intervals calculated for the mean run-time.

3.3. The manufacturing model

The Manufacturing model (Figure 4) consists of 100 machines connected by 100 accumulating conveyors. The aim of the model is to determine the throughput of the system. The machines work in series, so all parts are processed by all 100 machines. The first machine in the manufacturing line pulls in a part whenever it is available. All machines have a fixed cycle time of 1 min (all times are in minutes). Breakdowns occur based on the busy time of each machine. The time between failure follows a Weibull distribution with shape parameter 2. The scale parameter for each machine is sampled from a uniform distribution with a minimum value of 750 and a maximum of 1,500; the same sample value is used for every run of the model. The repair time is sampled at the time of the repair from a log normal distribution with mean 60 and standard deviation of 12. Set-ups occur according to the number of cycles performed by each machine. The number of cycles between set-ups is sampled from a uniform distribution with minimum 140 and maximum 300; the same sample value is used for every run of the model. The time to set-up each machine is sampled at the time of the set-up from a log normal distribution with a mean of 20 and standard deviation of 4. (Note: the parameters for the log normal distribution in the Witness software are the mean and standard deviation of the actual distribution and not

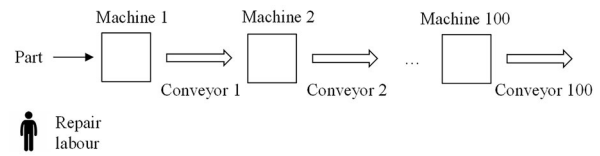


Figure 4. The manufacturing model.

the mean and standard deviation of the logarithmic values.)

All conveyors have a capacity of 5 parts and index time of 0.1 min. Conveyor breakdowns are again based on busy time, and the time between failure is sampled from a Weibull distribution with a shape of 2 and scale of 11,300 (giving a mean time between failure of just over 10,000 min). The repair time follows a log normal distribution with mean 30 and standard deviation of 6. Repair labour are required to perform repairs of machines and conveyors, and machine set-ups. There is a team of 10 repair persons. The output of interest for this model is the total throughput.

3.3.1. Model simplification

The first type of simplification approach applied to the manufacturing model is through the aggregation of machines and conveyors into blocks with randomly sampled delay times. We introduce blocks using two approaches: aggregation into groups of 10 and successively splitting the line into larger-and-larger blocks. In the aggregation approach machines and conveyors are blocked into groups of 10, that is, machines and conveyors 1-10, 11-20, ..., 91-100. However, the first block excludes machine 1 which needs to remain in the model so parts can continue to be pulled in whenever the machine is available. Each block has a capacity of 60 since it represents 10 machines and 10 conveyors (each with a capacity of 5). The simplified models are created by introducing the blocks in two ways. The "Blocks-down" (starting from the top and working down) version starts by replacing machines 2-10 and conveyors 1-10 with a block, then adding a block for machines and conveyors 11-20, and continuing until the whole model is represented by 10 blocks. The "Blocks-up" (starting from the bottom and working up) version starts by replacing machines and conveyors 91-100, then machines and conveyors 81-90, and continuing until the whole model is represented by 10 blocks. The two approaches are employed so we can determine if the ordering of the simplifications impacts on the model error and its trajectory as additional simplifications are applied.

In the splitting approach the line is initially split into 50 blocks of two machines and conveyors, then into 25 blocks of four machines and conveyors, continuing to 20 blocks of five, 10 blocks of 10, five blocks of 20, four blocks of 25, and finally two

blocks of 50 machines and conveyors. As with the aggregation approach, machine 1 is not included in the first block so it can continue to pull parts into the model. The capacity of each block is set to six times the number of machines it represents, e.g. when there are 20 blocks of five, the blocks' capacity is set to 30. We refer to this as the "Split Blocks" version of the model. We note that the Split Blocks model with 10 blocks is exactly equivalent to the Blocks-down and Blocks-up models with 10 blocks.

The time parts spend in a block is sampled from empirical distributions that are derived from a long-run of the full model (54,000 min) in which the time each part spends in the associated sections of the line is recorded. The blocks are modelled as a buffer with a delay time and a dummy machine that removes the part from the buffer and sends it to the next machine or block. Parts can overtake one another in a block based on the delay time sampled for each part. Although there is no overtaking in the full model, it is necessary to allow overtaking in the blocks so parts are not trapped behind other parts with very long delay times. This would have the effect of increasing the delays in the blocks which would reduce the throughput capacity of the whole system. Given our interest is in total throughput, and not in tracking individual parts, this is a reasonable way of simplifying the model.

The second simplification approach applied to this model is through omission which entails removing machines and conveyors from the model without replacement or compensation (Scope Reduction model). This is achieved by deleting groups of 10 machines and conveyors at a time, starting with machines and conveyors 91-100. This continues until only machines and conveyors 1-10 remain in the model.

For all of the above, the number of repair persons is adjusted as the number of individual machines and conveyors is reduced. In the Split Blocks model, at the first level of simplification (50 blocks), only a single repair person is required to service machine 1. In the Blocks-down, Blocks-up and Scope Reduction models, for each block that is employed, or each omission of 10 machines and conveyors, the number of repair persons is reduced by 1. As such, the final Scope Reduction model, which consists of 10 machines and conveyors, contains 1 repair person. However, the final Blocks-up and Blocks-down models, which consist of 10 blocks, do include logic to represent one repair person who is required to service machine 1.

These simplification approaches aim to mimic those used in practice for manufacturing simulation models. For instance, in semiconductor manufacturing, Sprenger and Rose (2011) simplify a model by representing bottleneck machine groups in detail,

but other sections of the line as a delay distribution. The aggregation into blocks approach adopted above similarly involves a mix of modelling some sections of the line in detail and others as a delay distribution, at least at the initial stages of simplification. The author has also used blocks on a number of occasions to represent sections of a manufacturing line or even whole plants (Robinson, 1994b). Both Hung and Leachman (1999) and Piplani and Puah (2004) simplify a semiconductor manufacturing model by eliminating workstations as for the omission approach above. Meanwhile, the Split Blocks model is employed to understand the impact of successively splitting the line into larger and larger blocks.

3.3.2. Model complexity

The Brooks & Tobias complexity of the Manufacturing model is calculated as follows:

- Number of Components = number of machines + number of conveyors + number of buffers + 1 (for repair labour)
- Number of Connections = Number of components - 1 (for labour) + 1 (for input of parts to the model)

(Connections are treated as the flow of parts between model components and so labour is not counted in the total connections.)

Calculational Complexity = number of machines + number of conveyors + number of buffers

(This assumes machines, conveyors and buffers have equivalent calculational complexity. Labour is excluded as the logic to call for labour is on the machines and conveyors.)

Total Complexity is calculated as a simple summation of the number of components, number of connections and calculational complexity.

Tables 1-4 summarise the complexity measure for each model version and level of simplification used in the experiment.

3.3.3. Run strategy

Given the model starts from an empty and idle state, the MSER algorithm (White, 1997; White et al., 2000) was used to determine an appropriate warm-up period for the model. Using hourly throughput data, MSER recommended a 13 h warm-up for the base model. A 24 h (1,440 min) warm-up was used for all model runs to allow for possible variation in the length of warm-up period needed

Table 1. Complexity of blocks-down manufacturing model.

Level of simplify-cation	Number of					Calculational complexity	Total complexity
	Machines	Conveyors	Buffers	Components	Connections		
Base	100	100	0	201	201	200	602
1 Block	92	90	1	184	184	183	551
2 Blocks	83	80	2	166	166	165	497
3 Blocks	74	70	3	148	148	147	443
4 Blocks	65	60	4	130	130	129	389
5 Blocks	56	50	5	112	112	111	335
6 Blocks	47	40	6	94	94	93	281
7 Blocks	38	30	7	76	76	75	227
8 Blocks	29	20	8	58	58	57	173
9 Blocks	20	10	9	40	40	39	119
10 Blocks	11	0	10	22	22	21	65

Table 2. Complexity of blocks-up manufacturing model.

Level of simplify-cation	Number of					Calculational complexity	Total complexity
	Machines	Conveyors	Buffers	Components	Connections		
Base	100	100	0	201	201	200	602
1 Block	91	90	1	183	183	182	548
2 Blocks	82	80	2	165	165	164	494
3 Blocks	73	70	3	147	147	146	440
4 Blocks	64	60	4	129	129	128	386
5 Blocks	55	50	5	111	111	110	332
6 Blocks	46	40	6	93	93	92	278
7 Blocks	37	30	7	75	75	74	224
8 Blocks	28	20	8	57	57	56	170
9 Blocks	19	10	9	39	39	38	116
10 Blocks	11	0	10	22	22	21	65

Table 3. Complexity of split blocks manufacturing model.

Level of simplify-cation	Number of					Calculational complexity	Total complexity
	Machines	Conveyors	Buffers	Components	Connections		
Base	100	100	0	201	201	200	602
50 Blocks	51	0	50	102	102	101	305
25 Blocks	26	0	25	52	52	51	155
20 Blocks	21	0	20	42	42	41	125
10 Blocks	11	0	10	22	22	21	65
5 Blocks	6	0	5	12	12	11	35
4 Blocks	5	0	4	10	10	9	29
2 Blocks	3	0	2	6	6	5	17

Table 4. Complexity of scope reduction manufacturing model.

Level of simplify-cation	Number of					Calculational complexity	Total complexity
	Machines	Conveyors	Buffers	Components	Connections		
Base	100	100	0	201	201	200	602
90 Machines	90	90	0	181	181	180	542
80 Machines	80	80	0	161	161	160	482
70 Machines	70	70	0	141	141	140	422
60 Machines	60	60	0	121	121	120	362
50 Machines	50	50	0	101	101	100	302
40 Machines	40	40	0	81	81	80	242
30 Machines	30	30	0	61	61	60	182
20 Machines	20	20	0	41	41	40	122
10 Machines	10	10	0	21	21	20	62

between model variants. The model was run for 54,000 min post warm-up, which is equivalent to 900 h of simulated time. For each level of simplification, the simulation run was replicated 10 times to provide independent samples for both total throughput and run-time. Initial runs showed that 10 replications were sufficient to obtain a narrow confidence interval. The confidence interval half-widths for total throughput are generally below one percent of the mean value, while for run-time there

is more variance, but no confidence interval half-width exceeds ten percent of the mean. The run-time was measured from the end of the warm-up period to the end of the simulation run using the computer's clock time.

3.4. The service model

The Service model (Figure 5) represents a 10-stage service process, with dedicated queues for each of the

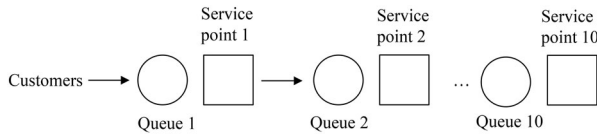


Figure 5. The service model.

10 service points. The aim of the model is to determine the average time customers spend in the system. The queues are constrained to a capacity of 9 customers. The model, therefore, represents a queuing network with finite capacities (Onvural, 1993) which could represent systems such as traffic or computer networks where buffers are constrained.

Customers arrive in the model according to a negative exponential distribution with a mean of 1 min. Customers balk if there is no spare capacity in the first queue. The service time at each service point is sampled from a negative exponential distribution with a mean of 0.85 min. The output of interest for this model is the average time customers spend in the system.

3.4.1. Model simplification

As for the Manufacturing model, this model is simplified through the aggregation of queues (Q) and service points (SP) into blocks. For the Service model this entails successively subsuming individual queues and service points into a single block. The block is represented by a buffer with a randomly sampled delay time and a dummy machine that moves the customers from the buffer to the next stage of the model (a queue or the exit). This is achieved in two ways:

- *Block-Down*: start by making Q1, SP1, Q2 and SP2 into a block, then add Q3 and SP3 to the block, and continue until all queues and service points are subsumed into the block with the exception of Q10 and SP10.
- *Block-Up*: start by making Q10, SP10, Q9 and SP9 into a block, then add Q8 and SP8 to the block, and continue until all queues and service points are subsumed into the block with the exception of Q1 and SP1.

The delay time in the buffer is sampled from an empirical distribution generated from a long run of the model (1,000,000 min). Individual empirical distributions are generated and used according to the size and coverage of the block. The capacity of the buffer is set at 10 times the number of service points represented by the block to represent the capacity constraints in the system. This capacity is reduced by 1 for the Block-down model, to allow for the capacity of the dummy machine. This is not necessary for the Block-up model as the dummy

machine can always exit the customer from the model and so does not lead to an effective increase in capacity. As for the Manufacturing model, and for the same reasons, overtaking is permitted in the block. If overtaking were not allowed, the time-in-system for individual customers would be inflated if they became trapped behind a slower customer, and these delays are already accounted for in the empirical distribution for the delay time.

This approach to simplification aims to recreate the situation where some parts of a service system are modelled in detail, while other sections are simplified and represented as a delay. Günel and Pidd (2011), for example, discuss how different levels of granularity can be used for representing different parts of the system in a whole hospital simulator.

3.4.2. Model complexity

The Brooks & Tobias complexity of the Service model is calculated as follows:

- Number of Components = number of service points + number of queues + 1 (for customers) + 2 (for buffer and dummy machine if block is in use)

(Customers are included as they actively arrive.)

Number of Connections = Number of components

Calculational Complexity = number of components

(This assumes service points, queues, the buffer and dummy machine, and customers have equivalent calculational complexity.)

Total Complexity is calculated as a simple summation of the number of components, number of connections and calculational complexity.

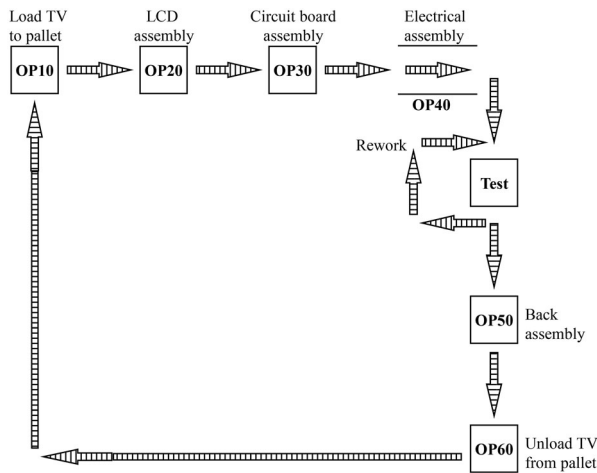
Table 5 summarises the complexity measure for each level of simplification used in the experiment. In this case the complexity is the same for the Block-down and Block-up models at all levels of simplification.

3.4.3. Run strategy

This model also starts from an empty and idle state, so a warm-up period is required before collecting any output statistics. Applying the MSER algorithm (White, 1997; White et al., 2000) on hourly data for time-in-the-system, a warm-up of 4 h (240 min) is recommended. We therefore decided to use a 1,000 min warm-up for all model runs. The model was run for 100,000 min post warm-up. For each level of simplification, the simulation run was replicated 10 times to provide independent samples for both average time-in-the-system and run-time.

Table 5. Complexity of block-down and block-up service models.

Level of simplification	Number of				Calculational complexity	Total complexity
	Service points	Queues	Components	Connections		
Base	10	10	21	21	21	63
Block of 2	8	8	19	19	19	57
Block of 3	7	7	17	17	17	51
Block of 4	6	6	15	15	15	45
Block of 5	5	5	13	13	13	39
Block of 6	4	4	11	11	11	33
Block of 7	3	3	9	9	9	27
Block of 8	2	2	7	7	7	21
Block of 9	1	1	5	5	5	15

**Figure 6.** The Panorama Televisions model (Robinson, 2014).

Initial runs showed that 10 replications were sufficient to obtain a narrow confidence interval. The confidence interval half-widths for average time in the system are generally below one percent of the mean value, while for run-time there is more variance, but no confidence interval half-width exceeds ten percent of the mean. The run-time was measured from the end of the warm-up period to the end of the simulation run using the computer's clock time.

3.5. The Panorama Televisions model

The Panorama Televisions case study is described in Robinson (2014), so the detail of the case will not be repeated here. The aim of the model is to determine whether Panorama's recently installed production line can achieve 500 units of throughput per day, which at present it is failing to do. The model represents the core of the production facility, which consists of a set of automated and manual operations, connected by conveyors (Figure 6). The output of interest for this model is the average daily throughput.

3.5.1. Model simplification

Our approach to the simplification of the Panorama Televisions model differs from that employed for the previous two examples. For this case the level of detail in the model is incrementally reduced. In

order to understand the impact of changing the sequence in which the simplifications are implemented, they are introduced in two different sequences described as "Forward" and "Reverse." The Forward Simplification sequence is as follows:

- *Step F1:* replace random cycle times for manual processes with fixed cycle times.
- *Step F2:* remove maintenance labour and all calls for labour to repair and set-up.
- *Step F3:* remove machine breakdowns and increase cycle times to allow for average losses due to machine failures.
- *Step F4:* remove machine set-ups and increase cycle times to allow for average losses due to set-ups.
- *Step F5:* remove the production schedule and modelling of different TV types.
- *Step F6:* remove rework and modelling of TVs failing in the test station. (Note: the model is now completely deterministic.)
- *Step F7:* replace conveyors with buffers of the same capacity, but zero dwell time.
- *Step F8:* remove modelling of pallets (used to transport TVs through the production line) and the return conveyors (buffers) between the unloading and loading stations.

The Reverse Simplification sequence adopts almost the complete inverse of these steps except that machine set-ups continue to be removed before taking out the production schedule, and maintenance labour continue to be removed before machine breakdowns. If the ordering of these pairs is reversed, the first deletion would effectively remove the second at the same time, e.g. removing the production schedule means there would be no set-ups. The Reverse Simplification sequence is as follows:

- *Step R1:* remove modelling of pallets (used to transport TVs through the production line) and the return conveyors between the unloading and loading stations.
- *Step R2:* replace conveyors with buffers of the same capacity, but zero dwell time.

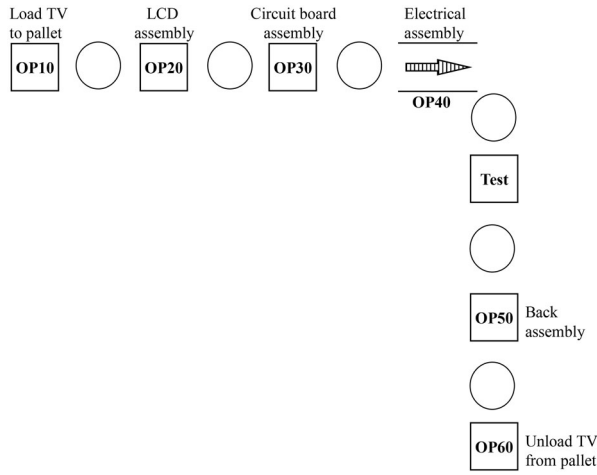


Figure 7. Fully simplified Panorama Televisions model (step F8 and R8).

- *Step R3:* remove rework and modelling of TVs failing in the test station.
- *Step R4:* remove machine set-ups and increase cycle times to allow for average losses due to set-ups.
- *Step R5:* remove the production schedule and modelling of different TV types.
- *Step R6:* remove maintenance labour and all calls for labour to repair.
- *Step R7:* remove machine breakdowns and increase cycle times to allow for average losses due to machine failures.
- *Step R8:* replace random cycle times for manual processes with fixed cycle times. (Note: the model is now completely deterministic.)

Figure 7 shows the layout of the fully simplified model following step F8. The final Reverse Simplification model is exactly the same at step R8.

3.5.2. Model complexity

The Brooks & Tobias complexity of the Panorama Televisions model is calculated as follows:

- Number of Components = number of machines + number of conveyors + number of buffers + 1 (for maintenance labour, until it is removed at step F2/R6)

(Electric Assembly consists of 5 manual stations and so is counted as 5 machines.)

Number of Connections = number of machines + number of conveyors + number of buffers + 1 (for output from Test machine to Rework, until rework is removed at step F6/R3)

(Connections are treated as the flow of parts between model components and so labour is not counted in the total connections.)

Calculational Complexity = 3 \times number of automated machines + number of manual operations + number of conveyors + number of buffers + 1 (for the production schedule, until it is removed at step F5/R5)

(Initially the four automated machines are assumed to have three times the calculational complexity of manual operations, conveyors and buffers. This is because automated machines also include logic for breakdowns and setup. The multiplier is reduced by 1 at step F3/R7 when breakdowns are removed, and again by 1 at step F4/R4 when set-ups are removed. Labour is excluded as the logic to call for labour is on the machines.)

Total Complexity is calculated as a simple summation of the number of components, number of connections and calculational complexity.

Tables 6 and 7 summarise the complexity measure for each level of simplification used in the experiment for the Forward and Reverse Simplification sequences respectively.

3.5.3. Run strategy

As per Robinson (2014), a warm-up period of 1,440 min was employed. However, the run-length was extended from the 30 days used in Robinson (2014) to 300 days (i.e. 432,000 min) post warm-up in order to get a reading for the run-time over a reasonable period, rather than around a second for some scenarios. As for the Manufacturing and Service models, for each level of simplification, the simulation run was replicated 10 times, and the run-time was measured from the end of the warm-up period to the end of the simulation run. Initial runs showed that 10 replications were sufficient to obtain a narrow confidence interval. The confidence interval half-widths for average daily throughput are generally below one percent of the mean value, while for run-time there is more variance, but no confidence interval half-width exceeds ten percent of the mean.

4. Results: Exploring the model accuracy and complexity relationship

We now present the results for each of the three models in turn. For each model version and level of simplification we report the mean of the output of interest from the 10 replications that are performed. We also report the half-width of the 95% confidence interval from the 10 replications. Similarly, we report the run-time of the model as the mean of the

Table 6. Complexity of Panorama Televisions model: forward simplification sequence.

Level of simplify-cation	Number of					Calculational complexity	Total complexity
	Machines*	Conveyors	Buffers	Components	Connections		
Base	12	11	0	24	24	32	80
Step F1	12	11	0	24	24	32	80
Step F2	12	11	0	23	24	32	79
Step F3	12	11	0	23	24	28	75
Step F4	12	11	0	23	24	24	71
Step F5	12	11	0	23	24	23	70
Step F6	11	8	0	19	19	19	57
Step F7	11	0	7	18	18	18	54
Step F8	11	0	6	17	17	17	51

*Includes 4 automated machines at every step.

Table 7. Complexity of Panorama Televisions model: reverse simplification sequence.

Level of simplify-cation	Number of					Calculational complexity	Total complexity
	Machines*	Conveyors	Buffers	Components	Connections		
Base	12	11	0	24	24	32	80
Step R1	12	9	0	22	22	30	74
Step R2	12	0	8	21	21	29	71
Step R3	11	0	6	18	17	26	61
Step R4	11	0	6	18	17	22	57
Step R5	11	0	6	18	17	22	57
Step R6	11	0	6	17	17	22	56
Step R7	11	0	6	17	17	17	51
Step R8	11	0	6	17	17	17	51

*Includes 4 automated machines at every step.

Table 8. Manufacturing blocks-down model results.

Level of simplify-cation	Total throughput		Run-time (min)		Error
	Mean	Half-width of confidence interval	Mean	Half-width of confidence interval	
Base	20,666.80	113.93	27.09	0.22	0.000
1 Block	20,680.00	129.97	27.86	0.09	0.001
2 Blocks	20,707.50	128.80	26.43	0.13	0.002
3 Blocks	20,730.80	125.20	23.65	0.10	0.003
4 Blocks	20,837.10	140.77	19.95	0.17	0.008
5 Blocks	20,874.40	132.25	15.63	0.15	0.010
6 Blocks	20,959.20	150.56	11.33	0.14	0.014
7 Blocks	21,202.00	147.50	7.24	0.06	0.026
8 Blocks	21,560.30	138.97	3.80	0.04	0.043
9 Blocks	22,596.50	165.65	1.44	0.02	0.093
10 Blocks	25,767.30	53.40	0.26	0.01	0.247

10 replications and the associated 95% confidence interval half-width. The half-widths are all small relative to the value of the mean, suggesting we have a high degree of precision in our estimates. It should also be noted that the models employ common random numbers and so this may help reduce the variance when comparing the results between different levels of simplification for the same model version (Law, 2015).

4.1. Manufacturing model

Table 8 presents the results for the Blocks-down version of the Manufacturing model with the results for absolute error against complexity shown graphically in Figure 8. The graphs are very similar for both the run-time complexity measure and for the Brooks & Tobias complexity measure. It is notable that the shape of the graphs is very close to that expected based on Leinweber's (1979) and Robinson's (2008) propositions, as depicted in

Figure 3. We clearly see that the accuracy of the model improves with each incremental increase in complexity, but with diminishing returns.

The pattern is very similar for the Blocks-up model (Table 9 and Figure 9), although the errors are greater for the lower levels of simplification ranging from between around 2-fold greater at the 7 Blocks level to 6-fold or more at the 1 Block level. Of course, the results for the 10 Blocks level are exactly the same for the Blocks-down and Blocks-up versions of the model as these models are identical. The greater errors from the Blocks-up model are likely to result from representing the end of the production line as a block even at the lower levels of simplification. This means that machine blocking effects due to failures and set-ups are not represented so accurately near the end of the line where the throughput data is collected. This is not the case for the lower levels of simplification in the Blocks-down model where the blocks are initially introduced at the start of the line.

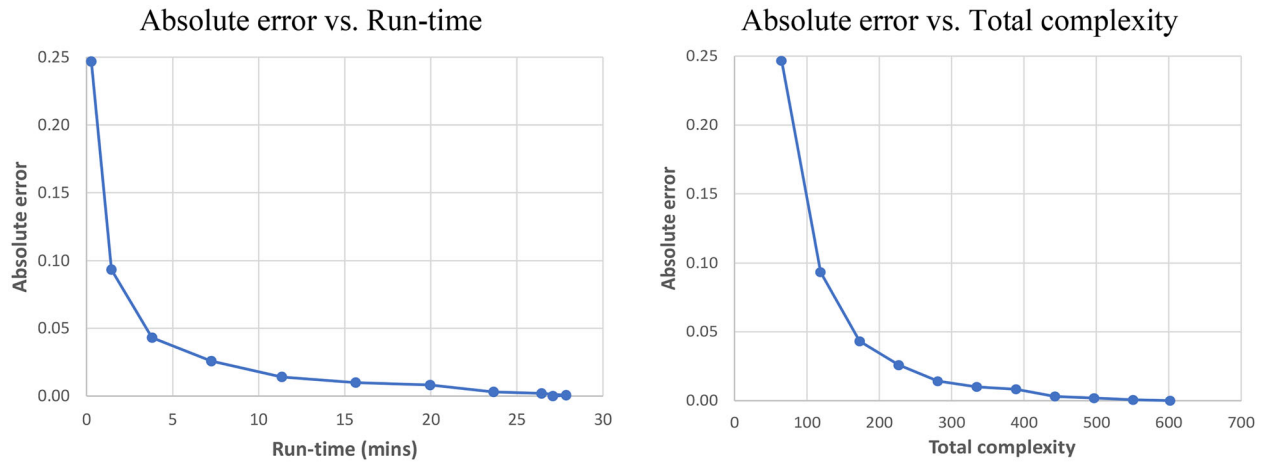


Figure 8. Manufacturing blocks-down model: error vs. run-time/complexity.

Table 9. Manufacturing blocks-up model results.

Level of simplification	Total throughput		Run-time (min)		Error
	Mean	Half-width of confidence interval	Mean	Half-width of confidence interval	
Base	20,666.80	113.93	27.09	0.22	0.000
1 Block	20,782.20	141.35	27.36	0.12	0.006
2 Blocks	20,940.50	118.51	25.68	0.12	0.013
3 Blocks	21,079.30	135.31	22.63	0.19	0.020
4 Blocks	21,248.70	141.43	18.33	0.16	0.028
5 Blocks	21,304.60	134.36	14.29	1.01	0.031
6 Blocks	21,421.40	94.49	8.86	0.12	0.037
7 Blocks	21,589.10	79.73	3.36	0.02	0.045
8 Blocks	21,657.80	89.31	2.15	0.01	0.048
9 Blocks	22,592.30	150.37	1.13	0.01	0.093
10 Blocks	25,767.30	53.40	0.26	0.01	0.247

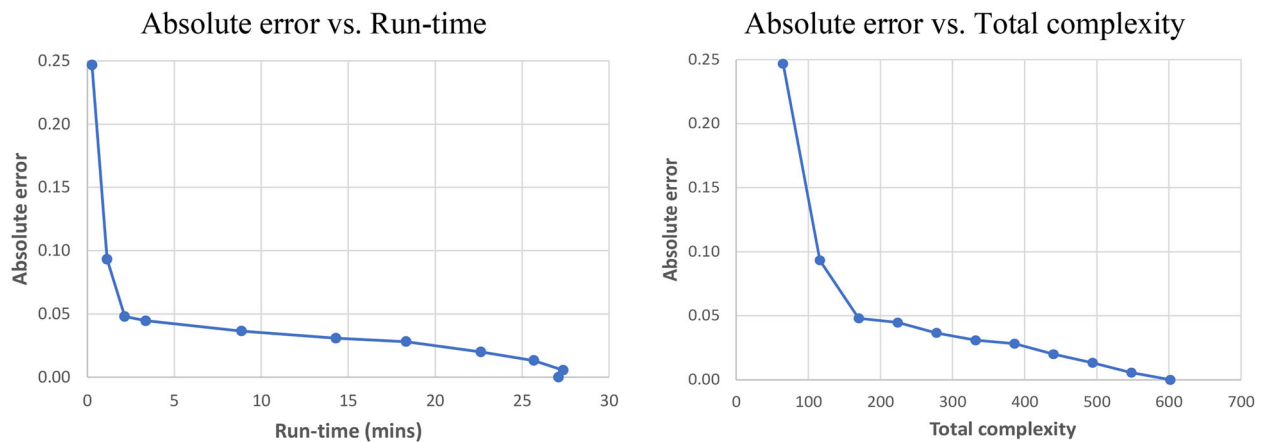


Figure 9. Manufacturing blocks-up model: error vs. run-time/complexity.

Table 10. Manufacturing split blocks model results.

Level of simplification	Total throughput		Run-time (min)		Error
	Mean	Half-width of confidence interval	Mean	Half-width of confidence interval	
Base	20666.80	113.93	26.59	0.22	0.000
50 Blocks	18187.60	39.60	1.00	0.00	-0.120
25 Blocks	23224.30	57.08	0.64	0.01	0.124
20 Blocks	23605.20	52.73	0.53	0.01	0.142
10 Blocks	25767.30	53.40	0.32	0.00	0.247
5 Blocks	27949.00	40.37	0.20	0.00	0.352
4 Blocks	28615.00	41.29	0.17	0.01	0.385
2 Blocks	30144.70	27.77	0.11	0.01	0.459

The successive simplifications in the Split Blocks model also produce a pattern similar to that predicted by Leinweber and Robinson (Table 10 and

Figure 10). This approach to simplification creates a much greater error at the first step than is the case for the Blocks-down and Blocks-up models. This is

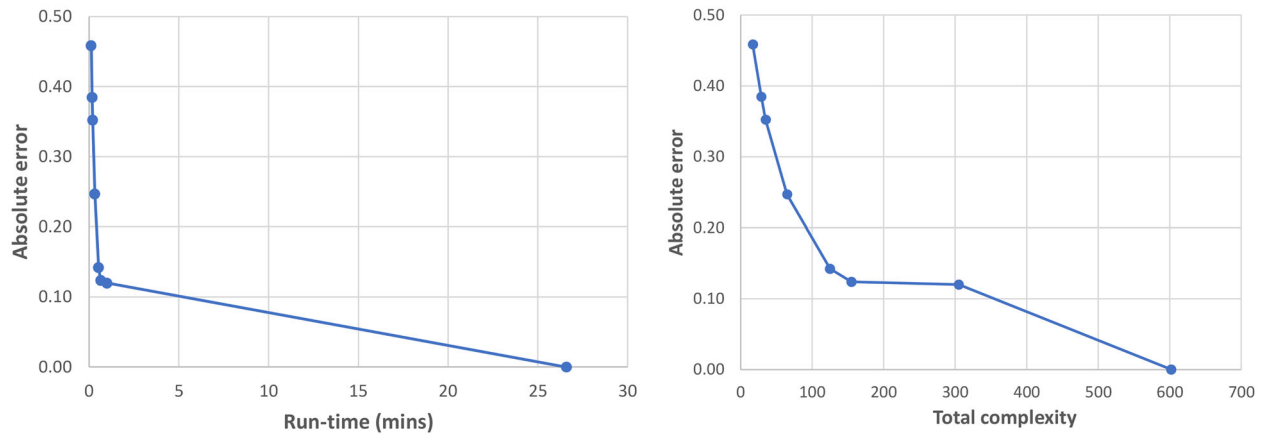


Figure 10. Manufacturing split blocks model: error vs. run-time/complexity.

Table 11. Manufacturing scope reduction model results.

Level of simplification	Total throughput		Run-time (min)		Error
	Mean	Half-width of confidence interval	Mean	Half-width of confidence interval	
Base	20,666.80	113.93	27.09	0.22	0.000
90 Machines	20,785.40	143.64	27.24	0.12	0.006
80 Machines	20,941.00	122.31	25.45	0.18	0.013
70 Machines	21,097.10	132.28	21.99	0.17	0.021
60 Machines	21,251.50	148.45	17.60	0.14	0.028
50 Machines	21,351.90	131.96	12.98	0.11	0.033
40 Machines	21,471.20	71.21	7.85	0.10	0.039
30 Machines	21,660.90	78.08	2.52	0.02	0.048
20 Machines	21,744.20	97.92	1.40	0.01	0.052
10 Machines	22,964.60	178.26	0.53	0.01	0.111

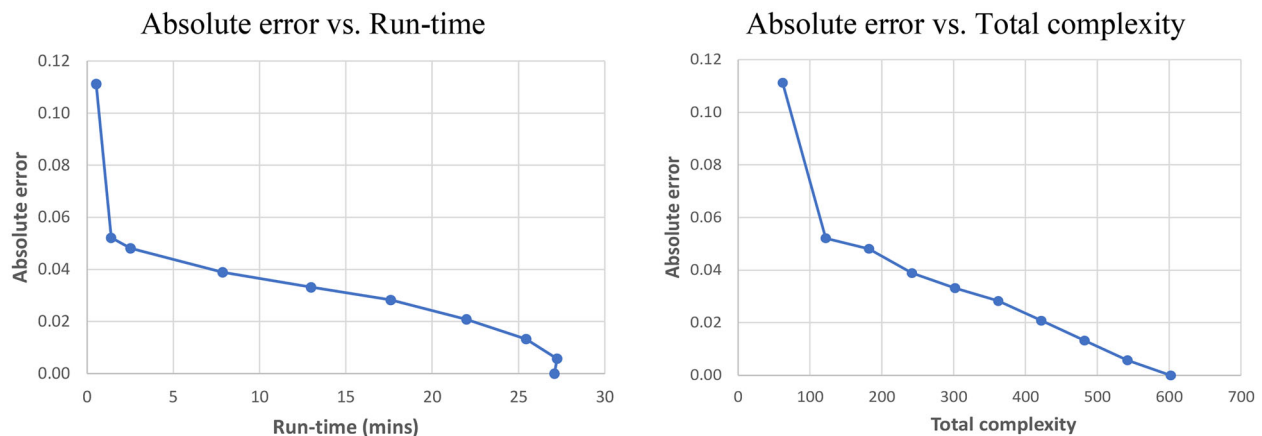


Figure 11. Manufacturing scope reduction model: error vs. run-time/complexity.

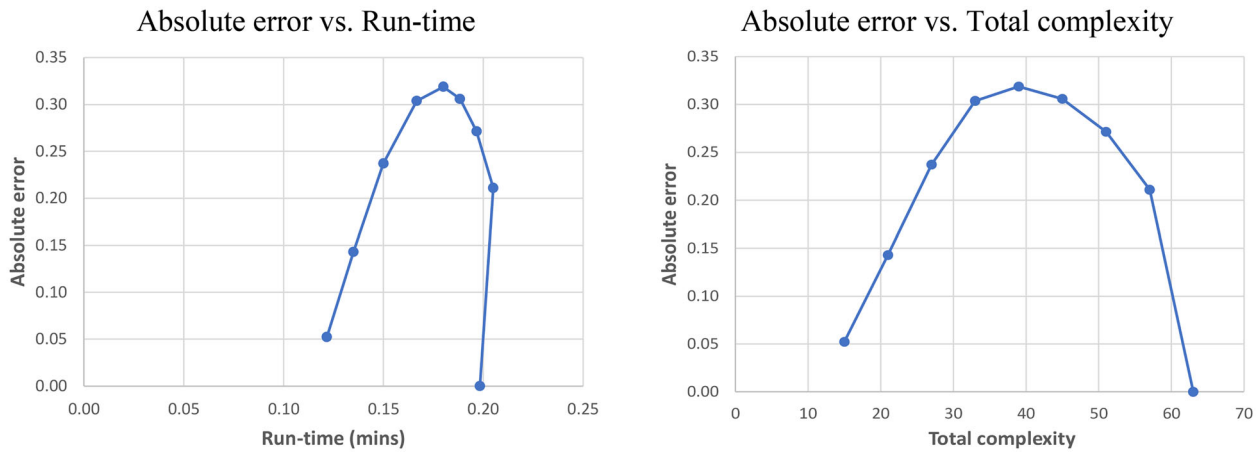
not surprising, since unlike for the above model simplifications, we immediately replace all individual conveyors and machines with blocks. As expected, the throughput result for the 10 Blocks level is exactly the same as for the 10 Blocks level in Tables 8 and 9, although the run-time information is slightly different in Table 10 as can occur when rerunning the model. The simplifications beyond 10 Blocks produce greater errors, increasing to just over 45% when the whole system is represented by two blocks. It is clear that the loss of detail at the greater levels of simplification, especially with respect to breakdowns, set-ups and the resultant blocking of the line, is leading to a significant over-estimate of the throughput.

The Scope Reduction model generates almost the same errors at each level of simplification as the Blocks-up version of the model (compare Table 11 and Figure 11 with Table 9 and Figure 9). This is probably because this model is removing the detailed modelling of machines in the same sequence as the Blocks-up version, albeit that they are not being replaced with blocks in the Scope Reduction model.

It is notable that for the Blocks-down, Blocks-up and Scope Reduction versions of the model, the first level of simplification causes the model to run slightly slower, although the difference in run-time is only statistically significant for Blocks-up and Blocks-down models ($\alpha = 0.05$). We assume this is

Table 12. Service block-down model results.

Level of simplification	Average time-in-system (min)		Run-time (min)		Error
	Mean	Half-width of confidence interval	Mean	Half-width of confidence interval	
Base	42.77	0.37	0.198	0.004	0.000
Block of 2	51.80	0.42	0.205	0.006	0.211
Block of 3	54.38	0.56	0.197	0.005	0.271
Block of 4	55.86	0.53	0.188	0.006	0.306
Block of 5	56.41	0.52	0.180	0.005	0.319
Block of 6	55.77	0.70	0.167	0.000	0.304
Block of 7	52.92	0.73	0.150	0.000	0.237
Block of 8	48.89	0.48	0.135	0.004	0.143
Block of 9	45.01	0.15	0.122	0.006	0.052

**Figure 12.** Service block-down model: error vs. run-time/complexity.**Table 13.** Service block-up model results.

Level of simplification	Average time-in-system (min)		Run-time (min)		Error
	Mean	Half-width of confidence interval	Mean	Half-width of confidence interval	
Base	42.77	0.37	0.198	0.004	0.000
Block of 2	41.12	0.30	0.188	0.006	-0.039
Block of 3	40.87	0.30	0.183	0.000	-0.045
Block of 4	40.75	0.21	0.173	0.006	-0.047
Block of 5	40.76	0.13	0.160	0.006	-0.047
Block of 6	41.03	0.09	0.147	0.005	-0.041
Block of 7	41.26	0.07	0.135	0.004	-0.035
Block of 8	41.63	0.05	0.125	0.006	-0.027
Block of 9	42.20	0.03	0.107	0.006	-0.013

because the benefit of reducing the model complexity is dominated by the increase in predicted throughput, meaning the model has to process more events. As greater levels of simplification are employed, the benefits of the reduced complexity start to dominate. However, for the Split Blocks model there is a large and immediate benefit in terms of run-time with a reduction from over 26 min for the full model to one minute or less for all the levels of simplification. This is primarily a result of removing the modelling of individual conveyors. This immediate improvement in run-speed means that the model complexity, as measured by the run-time, reduces significantly at the first level of simplification.

In terms of the relationship between accuracy and complexity, all four model versions demonstrate that accuracy reduces with greater levels of simplification. However, it is only for the Blocks-down and Split Blocks models that the relationship is non-

linear, that is, the accuracy of the model improves with each incremental increase in complexity, but with diminishing returns (although this is not the case for the highest two levels of complexity for the Split Blocks model when using the Brooks & Tobias complexity measure). For the other two model versions the relationship between accuracy and complexity appears to be nearly linear up until we reach the greater levels of simplification, that is, the last one or two steps before reaching the simplest model. This suggests that across the lower levels of simplification, the returns from increased complexity are roughly constant. With the exception of the slight increase in run-time at the first level of simplification, the patterns are very similar for both the run-time and Brooks & Tobias complexity measures for the Blocks-down, Blocks-up and Scope Reduction models. However, the Brooks & Tobias complexity measure does not reflect the immediate improvement in run-speed for the Split Blocks

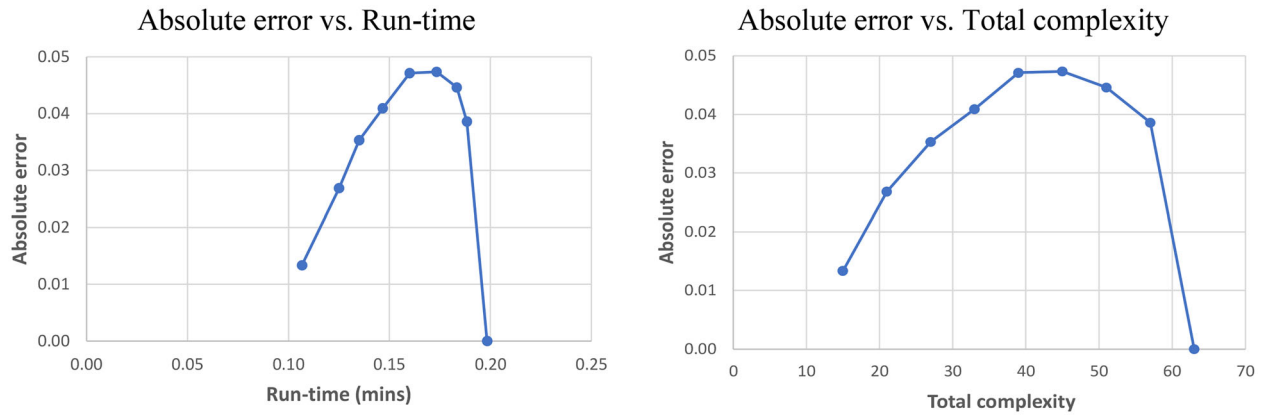


Figure 13. Service block-up model: error vs. run-time/complexity.

model, resulting in slightly different trajectories for the error vs. complexity curves (Figure 10).

4.2. Service model

Table 12 presents the results for the Block-down version of the Service model with the results for absolute error against complexity shown graphically in Figure 12. The graphs are quite similar for both complexity measures, but the Brooks & Tobias complexity measure gives a greater spread suggesting the gains in run-time are not as great as might be expected given the reduction in this complexity measure. The overall pattern is very similar for the Block-up version of the model (Table 13 and Figure 13). However, the maximum error is much greater for the Block-down model, at nearly 32% as opposed to less than 5% for the Block-up model. The actual errors are all positive for the Block-down model, showing that the simpler models overestimate the average time-in-the-system, while the actual errors are all negative for the Block-up model, showing that the simpler models underestimate the average time-in-the-system.

The Block-down model generates larger positive errors because the simplification approach leads to customers being unable to progress through the system more often than occurs in the full model. For example, when the block represents 5 service points (Block-down, Block of 5), the time spent in the block is sampled from the distribution for time-to-progress through the first five service points, which is generated from the full model. These times are randomly sampled and so autocorrelation effects (i.e. a slow customer will slow the progress of the next customer) are lost. As a result, once there is a build-up of customers in the block, there will be a more regular feed from the block to the next stage of the model as customers with shorter delay times overtake customers with longer delay times. This means the queue for the sixth service point is more congested and so prevents the block from always

being able to output customers. The same effect does not occur in the Block-up model as the block is always able to exit a customer from the model. However, because customers can also overtake in the block, the time-in-system is reduced (there are no delays beyond the sampled time), and as a result the Block-up model generates negative errors.

For both model versions, the absolute error initially increases as the size of the block is increased. As expected, greater simplification leads to greater error. However, once the block covers 6 or more service points, the error starts to reduce. This occurs as more-and-more of the service system is represented by sampling from a single empirical distribution that represents the time-to-progress through that part of the system. In turn, this block interacts with a smaller-and-smaller portion of the system that is modelled in detail. Indeed, the limit would occur if the full system were represented by sampling from a distribution for time-in-system. In this case the error would be zero, as we would simply be sampling from the output distribution of the full model.

The run-time of both model versions improves with greater levels of simplicity, with the exception of the first simplification step in the Block-down model. For this level of simplification the model runs marginally slower by $\frac{7}{1000}$ of a minute (the difference is statistically significant at $\alpha=0.05$). Although the introduction of a block of 2 reduces the number of components in the model by one queue and one service point, we suspect the increase in congestion in the first queue after the block (as described above), and the associated processing time to keep checking if a customer can be released from the block, causes the model to run slower. The effect is very marginal.

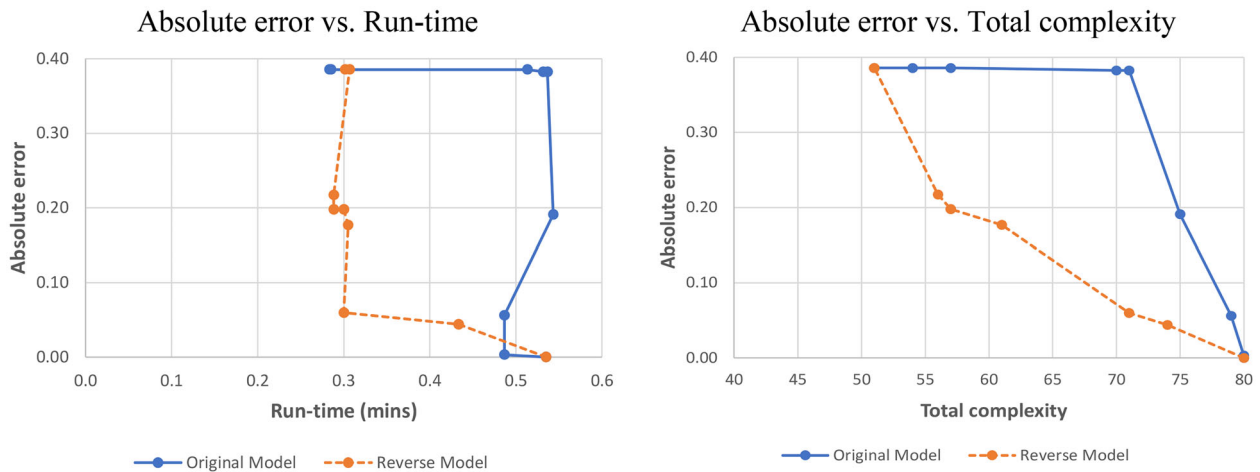
In terms of the relationship between accuracy and complexity, it is clear that none of the graphs bear much resemblance to Leinweber's (1979) and Robinson's (2008) propositions. For both model versions, up to the block of 5, there is a reduction in

Table 14. Panorama Televisions model results: forward simplification sequence.

Level of simplification	Daily throughput		Run-time (min)		Error
	Mean	Half-width of confidence interval	Mean	Half-width of confidence interval	
Base	414.05	1.30	0.535	0.027	0.000
Step F1	412.72	1.42	0.487	0.005	-0.003
Step F2	437.28	1.27	0.487	0.005	0.056
Step F3	493.11	0.71	0.543	0.006	0.191
Step F4	572.38	0.14	0.537	0.005	0.382
Step F5	572.38	0.14	0.532	0.007	0.382
Step F6	573.71	0.00	0.513	0.005	0.386
Step F7	573.71	0.00	0.283	0.000	0.386
Step F8	573.71	0.00	0.285	0.004	0.386

Table 15. Panorama Televisions model results: reverse simplification sequence.

Level of simplification	Daily throughput		Run-time (min)		Error
	Mean	Half-width of confidence interval	Mean	Half-width of confidence interval	
Base	414.05	1.30	0.535	0.027	0.000
Step R1	432.28	1.42	0.433	0.000	0.044
Step R2	438.79	1.55	0.300	0.000	0.060
Step R3	487.26	1.26	0.305	0.006	0.177
Step R4	495.90	1.53	0.300	0.000	0.198
Step R5	495.90	1.53	0.288	0.006	0.198
Step R6	503.98	1.65	0.288	0.006	0.217
Step R7	573.71	0.00	0.307	0.006	0.386
Step R8	573.71	0.00	0.302	0.004	0.386

**Figure 14.** Panorama Televisions model: error vs. run-time/complexity.

accuracy with greater levels of simplicity. However, the relationship demonstrates increasing returns as complexity is increased, and not diminishing returns as expected. Further, at higher levels of simplicity (block of 6 and more), accuracy actually improves.

4.3. Panorama Televisions model

Tables 14 and 15 present the results for the Forward and Reverse Simplification sequence versions of the Panorama Televisions model. The results are shown graphically in Figure 14. It is clear that the relationship between the error and model complexity is very dependent on the simplification sequence. For both simplification sequences, the error and complexity start and finish at the same point, although there is a slight difference in the run-time for the F8 and R8 runs (statistically significant at $\alpha = 0.05$). (As these models are identical, this will be a result of small changes in the

computer's processing activity.) However, the trajectory of the errors and complexity when moving from the base model to the fully simplified model is very different for the two simplification sequences.

The error at each simplification step is positive (overestimates daily throughput) with the exception of step F1, where there is a slight, statistically significant ($\alpha = 0.05$), underestimate. With the exception of moving from the base run to step F1, the change in error between steps is always positive or zero.

The largest improvements in run-time come from steps F7, R1 and R2. These steps all involve replacing the detailed modelling of conveyors with buffers with no delay. Removing the modelling of parts indexing on conveyors reduces the number of events that need to be processed. The only statistically significant ($\alpha = 0.05$) increases in run-time for the Forward and Reverse Simplification models occur when

breakdowns are removed at steps F3 and R7 respectively. The slowing of the model will be a result of the simplification leading to a higher daily throughput and so more events being processed.

With respect to the relationship between accuracy and complexity, both simplification sequences show a reduction in accuracy with greater levels of simplicity, but the trajectories are very different. For the Forward Simplification sequence there are large initial reductions in accuracy and then only marginal falls. For the Reverse Simplification sequence, initial gains are low and then the large gains occur with the final simplifications in the sequence. To all intents and purposes, the large reductions in accuracy are associated with the same simplifications for both sequences: the removal of breakdowns (F3 and R7) and the removal/reduction of set-ups (F4 and R3). While R3 removes rework from the Reverse Simplification model, since rework causes very frequent changes to the TV-type at the test area and beyond, the impact of removing R3 is to significantly reduce the number of set-ups on the Test machine and OP50. This effect does not occur in the Forward Simplification model, as set-ups have already been removed when the rework area is taken out of the model.

What we see from the Figure 14 is that the trajectory of accuracy with respect to the complexity of the model is very dependent on the sequence in which simplifications are implemented. The Reverse Simplification model has a trajectory that is not too dissimilar to that proposed by Leinweber (1979) and Robinson (2008). However, the Forward Simplification model demonstrates a very different trajectory where the greatest gains in accuracy come from the incremental changes at the highest levels of model complexity; in other words, increasing returns from greater complexity.

5. Conclusions: The relationship between model accuracy and complexity

What do the results presented above tell us about the relationship between discrete-event simulation model accuracy and complexity? In general, they demonstrate that increased complexity often leads to greater accuracy. However, there are occasions when the opposite is true and more complexity leads to less accuracy. In our investigation this is because of changes to the model specification and not measurement (data) errors. The most obvious example of this is seen in the Service model. When the block size covers most of the system, the model accuracy is high. Initially, as the block size reduces and more of the system is modelled in detail, the accuracy of the model falls. It is only once around half of the

system is being modelled in detail that the accuracy starts to improve.

When the run-time is used as a measure of model complexity, for all three models there are simplification steps for which reducing the complexity of the model actually increases its accuracy. For instance, with the Manufacturing model (Blocks-down, Blocks-up and Scope Reduction), the maximum run-time complexity occurs at the 1 Block and 90 Machines levels of simplification. Moving to the Base level reduces the run-time (complexity) while at the same time increasing accuracy. However, the same step is defined as an increase in complexity when the Brooks & Tobias complexity measure is used, so this result is dependent on the complexity measure that is employed. The effect occurs because taking detail out of the model leads to more events being processed (higher throughput in the Manufacturing model and Panorama Televisions model), or more complex interactions (greater congestion in a queue in the Service model). In effect, less detail increases the calculational complexity of the model, but this effect is only identified when the model is run and not through the static calculations used to evaluate the Brooks & Tobias complexity measure.

While increased complexity often leads to greater accuracy, the trajectories for the specific models and approaches to model simplification are very different. In summary:

- *Manufacturing Blocks-down model*: accuracy increases with complexity with diminishing returns.
- *Manufacturing Split Blocks model*: accuracy increases with complexity and generally with diminishing returns; the exception is for the highest two levels of complexity when using the Brooks & Tobias complexity measure.
- *Manufacturing Blocks-up and Scope Reduction models*: beyond the two or three simplest levels of the models, accuracy increases with complexity with near constant returns.
- *Service model*: accuracy initially reduces as complexity is increased, but then accuracy improves with increasing returns.
- *Panorama Televisions model*: the Forward Simplification sequence broadly demonstrates increasing returns from greater complexity, while the Reverse Simplification sequence broadly demonstrates diminishing returns.

Leinweber (1979) and Robinson (2008) both propose that accuracy improves with diminishing returns as complexity is increased, up to the point that measurement errors start to dominate. We have

not investigated the impact of measurement error and assumed throughout that the model data are accurate. Our expectation is, therefore, that the results will demonstrate the error to complexity relationship shown in Figure 3. However, we have found that only some of the models and simplification sequences demonstrate this relationship, in particular, the Manufacturing Blocks-down and Split Blocks models, and the Panorama Televisions model Reverse Simplification sequence. Quite different trajectories exist for the other models and simplification sequences. Similarly, Huber and Dangelmaier (2009) find quite different accuracy vs. complexity trajectories for the three models they investigate.

What does this mean for the Leinweber and Robinson propositions? In simple terms, the proposition that increasing complexity leads to improved accuracy with diminishing returns is not correct. The actual relationship between accuracy and complexity depends on the nature of the system, the model that represents the system and the modeller's choice of simplifications, including the sequence in which the simplifications are introduced. While increased complexity often leads to improved accuracy, this is not always the case. Even when there is a positive correlation between complexity and accuracy, the returns to accuracy from greater complexity can be diminishing, constant or increasing.

Despite the conclusion that the Leinweber and Robinson propositions are not correct in a strict sense, we believe that they remain useful heuristics for guiding modellers when determining the scope and level of detail at which to model a system. This is especially the case when starting from scratch and considering what to include and exclude in a new model.

6. Limitations and future work

In this paper we explore the relationship between discrete-event simulation model accuracy and complexity. Our approach is to implement a sequence of simplifications in three test models and to measure their impact on the accuracy of the model output. Our conclusions are contingent on the systems, models and simplifications that we have chosen, as well as the sequence in which the simplifications have been applied. Even within the limited set of models and simplifications that have been employed, quite different accuracy to model complexity trajectories have been observed. However, it would be useful to investigate other types of systems, models, simplifications and sequences using the same approach in order to develop a more comprehensive understanding of the relationship between accuracy and complexity.

The “block” simplifications we have used rely on being able to generate distribution data from a “full”

model of the system. While this is a valid approach to simplification, it is very specific to two of the models we have investigated, and it will not always be possible to generate data for simplifications in this way. Many models are developed from being simple to more complex, the “start small and add” principle (Pidd, 1999), and not by simplifying an existing model. The Panorama Televisions model investigates this circumstance, but further investigation is warranted with other models where scope and detail are successively added without data from a more complex model.

We have adopted two measures of complexity, run-time and the Brooks & Tobias complexity measure. Although the two measures were quite well correlated, they did lead to slightly different results. Future work should look at refining these measures and adopting alternative ways of measuring complexity. In particular, alternative methods for determining the calculational complexity of a simulation model need to be developed, especially so they can take account of the dynamic interactions in a model. A measure of cyclomatic complexity (McCabe, 1976) may be useful in this respect.

We have only explored the left-hand end of Leinweber's (1979) and Robinson's (2008) graphs, and not the right-hand where over-elaboration leads to reduced accuracy due to measurement (data) errors. As identified in section 2, the impact of input data uncertainty has been investigated (Ankenman & Nelson, 2012; Song & Nelson, 2015), but not so much from the perspective of how adding and removing model details with data uncertainty impacts on model accuracy. Future work should investigate the relationship between model accuracy and complexity in the face of data errors.

Finally, our focus has been on model simplification from the perspective of the relationship between model accuracy and complexity. Other impacts of simplification need to be explored beyond simple statements based on experience, both in terms of their advantages and disadvantages. For instance, what is the relationship between model build time and model complexity? How do simplifications impact on the users' understanding of the model? How do simplifications impinge on the ability to experiment with new scenarios as the understanding of the system improves?

Our contribution is to provide detailed empirical evidence on the relationship between discrete-event simulation model accuracy and complexity. In doing so, we have identified a range of trajectories that model accuracy can follow as complexity is increased. We encourage further work in this area so the simulation modelling community can better

understand the impact of increasing model complexity and its converse, model simplification.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

- Ankenman, B. E., & Nelson, B. L. (2012). A quick assessment of input uncertainty. In C. Laroque, J. Himmelsbach, R. Pasupathy, O. Rose, & A. M. Uhrmacher (Eds.), *Proceedings of the 2012 Winter Simulation Conference* (pp. 241–250). IEEE.
- Arora, S., & Barak, B. (2009). *Computational complexity: A modern approach*. Cambridge University Press.
- Barton, R. R., Nelson, B. L., & Xie, W. (2014). Quantifying input uncertainty via simulation confidence intervals. *INFORMS Journal on Computing*, 26(1), 74–87. <https://doi.org/10.1287/ijoc.2013.0548>
- Brooks, R. J. (2010). Complexity, level of detail and model performance. In S. Robinson, R. J. Brooks, K. Kotiadis, K., & D.-J. van der Zee, D.-J. (Eds.), *Conceptual modeling for discrete-event simulation* (pp. 31–56). CRC Press.
- Brooks, R. J., & Tobias, A. M. (1996). Choosing the best model: Level of detail, complexity and model performance. *Mathematical and Computer Modelling*, 24(4), 1–14. [https://doi.org/10.1016/0895-7177\(96\)00103-3](https://doi.org/10.1016/0895-7177(96)00103-3)
- Brooks, R. J., & Tobias, A. M. (2000). Simplification in the simulation of manufacturing systems. *International Journal of Production Research*, 38(5), 1009–1027. <https://doi.org/10.1080/002075400188997>
- Chwif, L., Barretto, M. R. P., & Paul, R. J. (2000). On simulation model complexity. In J. A. Joines, R. R. Barton, K. Kang, & P. A. Fishwick (Eds.), *Proceedings of the 2000 Winter Simulation Conference* (pp. 449–455). IEEE.
- Fishwick, P. A. (1988). The role of process abstraction in simulation. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1), 18–39. <https://doi.org/10.1109/21.87052>
- Günal, M. M., & Pidd, M. (2011). DGHPSIM: Generic simulation of hospital performance. *ACM Transactions on Modeling and Computer Simulation*, 21(4), 23:1–23:22. <https://doi.org/10.1145/2000494.2000496>
- Hood, S. J. (1990). Detail vs. simplifying assumptions for simulating semiconductor manufacturing lines. In IEMT Symposium, 9th IEEE/CHMT International (pp. 103–108). IEEE.
- Huber, D., & Dangelmaier, W. (2009). Controlled simplification of material flow simulation models. In M. D. Rossetti, R. R. Hill, A. Dunkin, & R. G. Ingalls (Eds.), *Proceedings of the 2009 Winter Simulation Conference* (pp. 839–850). IEEE.
- Hung, Y.-F., & Leachman, R. C. (1999). Reduced simulation models of wafer fabrication facilities. *International Journal of Production Research*, 37(12), 2685–2701. <https://doi.org/10.1080/002075499190473>
- Innis, G., & Rexstad, E. (1983). Simulation model simplification techniques. *SIMULATION*, 41(1), 7–15. <https://doi.org/10.1177/003754978304100101>
- Jacobson, S. H., & Yücesan, E. (1999). On the complexity of verifying structural properties of discrete event simulation models. *Operations Research*, 47(3), 476–481. <https://doi.org/10.1287/opre.47.3.476>
- Johnson, R. T., Fowler, J. W., & Mackulak, G. T. (2005). A discrete event simulation model simplification technique. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, & J. A. Joines (Eds.), *Proceedings of the 2005 Winter Simulation Conference* (pp. 2172–2176). IEEE.
- Lakshika, E., Barlow, M., & Easton, A. (2017). Understanding the Interplay of model complexity and fidelity in multiagent systems via an evolutionary framework. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(3), 277–289. <https://doi.org/10.1109/TCIAIG.2016.2560882>
- Lanner. (2022). *Witness simulation software*. www.lanner.com
- Law, A. M. (2015). *Simulation modeling and analysis* (5th ed.) McGraw Hill.
- Leinweber, D. (1979). Models, complexity, and error. A Rand note prepared for the US Department of Energy (N-1204-DOE), Rand, Santa Monica, CA.
- Lucas, T. W., & McGunnigle, J. E. (2003). When is model complexity too much? Illustrating the benefits of simple models with Hughes' salvo equations. *Naval Research Logistics*, 50(3), 197–217. <https://doi.org/10.1002/nav.10062>
- McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, SE-2(4), 308–320. <https://doi.org/10.1109/TSE.1976.233837>
- Onvural, R. O. (1993). Queueing networks with finite capacities. In L. Donatiello & R. Nelson (Eds.), *Performance evaluation of computer and communication systems*. Performance 1993, SIGMETRICS 1993, Lecture Notes in Computer Science (Vol. 729, pp. 411–434). Springer.
- Pidd, M. (1999). Just modeling through: A rough guide to modeling. *Interfaces*, 29(2), 118–132. <https://doi.org/10.1287/inte.29.2.118>
- Piplani, R., & Puah, S. A. (2004). Simplification strategies for simulation models of semiconductor facilities. *Journal of Manufacturing Technology Management*, 15(7), 618–625. <https://doi.org/10.1108/17410380410555862>
- Powell, S. G. (1995). Six key modeling heuristics. *Interfaces*, 25(4), 114–125. <https://doi.org/10.1287/inte.25.4.114>
- Robinson, S. (1994a). Simulation projects: Building the right conceptual model. *Industrial Engineering*, 26(9), 34–36.
- Robinson, S. (1994b). An introduction to visual interactive simulation in business. *International Journal of Information Management*, 14(1), 13–23. [https://doi.org/10.1016/0268-4012\(94\)90092-2](https://doi.org/10.1016/0268-4012(94)90092-2)
- Robinson, S. (2008). Conceptual modelling for simulation part I: Definition and requirements. *Journal of the Operational Research Society*, 59(3), 278–290. <https://doi.org/10.1057/palgrave.jors.2602368>
- Robinson, S. (2014). *Simulation: The practice of model development and use* (2nd ed.). Palgrave.
- Salt, J. (1993). Simulation should be easy and fun. In G. W. Evans, M. Mollaghasemi, E. C. Russell, & W. E. Biles (Eds.), *Proceedings of the 1993 Winter Simulation Conference* (pp. 1–5). IEEE.
- Schruben, L., & Yücesan, E. (1993). Complexity of simulation models: A graph theoretic approach. In G. W. Evans, M. Mollaghasemi, E. C. Russell, & W. E. Biles (Eds.), *Proceedings of the 1993 Winter Simulation Conference* (pp. 641–649). IEEE.
- Sevinc, S. (1991). Theories of discrete event model abstraction. In B. L. Nelson, W. D. Kelton, & G. M.

- Clark (Eds.), *Proceedings of the 1991 Winter Simulation Conference* (pp. 1115–1119). IEEE.
- Song, E., & Nelson, B. L. (2015). Quickly assessing contributions to input uncertainty. *IIIE Transactions*, 47(9), 893–909. <https://doi.org/10.1080/0740817X.2014.980869>
- Sprenger, R., & Rose, O. (2011). On the simplification of semiconductor wafer factory simulation models. In S. Robinson, R. Brooks, K. Kotiadis, & D.-J. van der Zee (Eds.), *Conceptual modeling for discrete-event simulation* (pp. 451–470). CRC Press.
- Tako, A., Tsiptsias, N., & Robinson, S. (2020). Can we learn from simple simulation models? An experimental study on user learning. *Journal of Simulation*, 14(2), 130–144. <https://doi.org/10.1080/17477778.2019.1704636>
- Thomas, A., & Charpentier, P. (2005). Reducing simulation models for scheduling manufacturing facilities. *European Journal of Operational Research*, 161(1), 111–125. <https://doi.org/10.1016/j.ejor.2003.08.042>
- Urenda Moris, M., Ng, A. H. C., & Svensson, J. (2008). Simplification and aggregation strategies applied for factory analysis in conceptual phase using simulation. In S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, & J. W. Fowler (Eds.), *Proceedings of the 2008 Winter Simulation Conference* (pp. 1913–1921). IEEE.
- Wallace, J. C. (1987). The control and transformation metric: Toward the measurement of simulation model complexity. In A. Thesen, H. Grant, & W. D. Kelton (Eds.), *Proceedings of the 1987 Winter Simulation Conference* (pp. 597–603). IEEE.
- Ward, S. C. (1989). Arguments for constructively simple models. *Journal of the Operational Research Society*, 40(2), 141–153. <https://doi.org/10.1057/jors.1989.19>
- White, K. P., Jr. (1997). An effective heuristic for bias reduction in simulation output. *Simulation*, 69(6), 323–334. <https://doi.org/10.1177/003754979706900601>
- White, K. P., Jr., Cobb, M. J., & Spratt, S. C. (2000). Comparison of contemporary truncation heuristics for steady-state simulation. In J. A. Joines, R. R. Barton, K. Kang, & P. A. Fishwick (Eds.), *Proceedings of the 2000 Winter Simulation Conference* (pp. 755–760). IEEE.
- Yavari, E., & Roeder, T. (2012). Model enrichment: Concept, measurement, and application. *Journal of Simulation*, 6(2), 125–140. <https://doi.org/10.1057/jos.2011.27>
- Yücesan, E., & Schruben, L. (1998). Complexity of Simulation models: A graph theoretic approach. *INFORMS Journal on Computing*, 10(1), 94–106. <https://doi.org/10.1287/ijoc.10.1.94>
- Zouaoui, F., & Wilson, J. R. (2004). Accounting for input-model and input-parameter uncertainties in simulation. *IIIE Transactions*, 36(11), 1135–1151. <https://doi.org/10.1080/07408170490500708>