Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

IMPLEMENTING MSER-5 IN COMMERCIAL SIMULATION SOFTWARE AND ITS WIDER IMPLICATIONS

Kathryn Hoad
Stewart Robinson

University of Warwick
Warwick Business School
Coventry, CV4 7AL, UNITED KINGDOM

ABSTRACT

Starting a model from an unrealistic state can lead to initialization bias in the simulation output. This, in turn, can produce bias in the results and lead to incorrect conclusions. One method for dealing with this problem is to run the model for a warm-up period until steady state is reached and remove the initialization bias by deleting the data within that warm-up period. Our previous research identified the MSER-5 algorithm as the best candidate warm-up method for implementation into an automated output analysis system, and for inclusion into existing DES software products. However, during an attempt to implement an automatable sequential version of the MSER-5 procedure into existing discrete-event simulation software several issues arose. This paper describes the framework and associated adaption of MSER-5 in order to automate it. It then discusses in detail the implementation issues that arose and some potential solutions.

1 INTRODUCTION

Starting a model running from an 'unrealistic' state can lead to the occurrence of initialization bias. This causes the output data collected at and near the beginning of the run to be uncharacteristic of the later and 'true' output steady state value. If this uncharacteristic data is included in the calculation of the overall response value it can produce a biased result and therefore incorrect conclusions. One method for dealing with this problem is to run the model for a warm-up period until steady state is reached and remove the initialization bias by deleting the data within that warm-up period.

Previous research (Hoad, Robinson, and Davies 2010) identified the MSER-5 algorithm (White, Cobb, and Spratt 2000; White and Robinson 2010) as the 'best' candidate warm-up method for implementation into an automated output analysis system, for inclusion into existing DES software products. Hoad, Robinson, and Davies (2010) identified MSER-5 as a good general method for automation since it is not "model or data type specific," "does not require estimation of any parameters," and can run with minimal user intervention. They also showed through quite extensive testing that MSER-5 runs quickly and performs "robustly and effectively" for a wide range of data series. However, our collaboration with a simulation software supplier in which we attempted to implement MSER-5 inside their discrete-event simulation software raised some practical issues that we had not previously considered in depth or at all. This paper will describe our devised framework and adaption of the MSER-5 method in order to automate it for potential inclusion in DES software (as previously outlined in Hoad, Robinson, and Davies 2010, 2011). We will then discuss in detail the implementation issues and possible solutions.

2 MSER-5 AND AN AUTOMATED FRAMEWORK

Put simply, MSER-5 is an algorithm that acts upon batched (batch size of 5) data to find the point in the data series where the standard error (test statistic) in the data is at a minimum when the data before that point is deleted. Therefore, for a finite stochastic sequence $X(j)$ of a specified simulation output of replication j, the optimal truncation point $d(j)^*$ selected by MSER-5 can be expressed as:

$$d(j)^* = \underset{n>d(j)\geq 0}{\arg\min}\left[\frac{1}{\left(n(j)-d(j)\right)^2}\sum_{i=d+1}^{n}\left(X_i(j)-\overline{X}_{n,d}(j)\right)^2\right]$$

where this expression is applied to a series of $b=\left\lfloor n/5 \right\rfloor$ batch averages instead of to the raw output data series, $d(j)$ is all possible truncation points for replication j and $n(j)$ is the total number of observations in replication j. Sometimes the last few batch values, through chance can be relatively close in value, thus producing a very low (and indeed minimum) standard error (test statistic) value, hence causing the algorithm to recommend a truncation point at the end of the data series. This is just an artefact of calculating standard error for a very small sample size of what is hopefully steady state data. In order to avoid this occurrence it is advisable to stop calculations before the end of the data series. Hoad, Robinson, and Davies (2011) stop calculating the test statistic at a default of 5 batches (5x5=25 data points) from the end of the data series. Figure 1 shows a working visual example of the MSER-5 method.

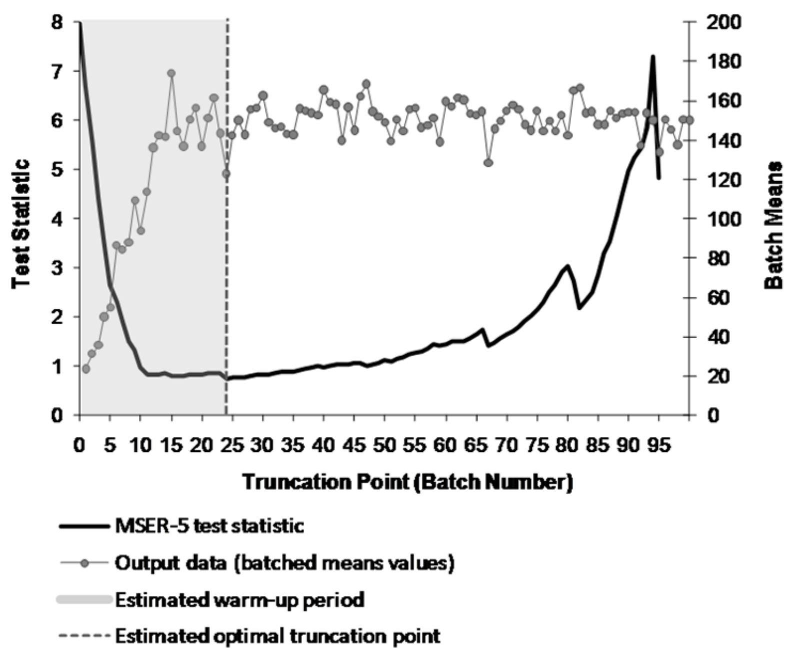

Figure 1: Example of the MSER-5 method at work

MSER-5 was originally devised as a fixed sample size method to estimate the truncation point, the point at which the warm-up period is judged to have ended, from a data series of set length. As White (1997) explains "…we propose to select a truncation point that minimizes the width of the CI about the truncated sample mean …Thus we will seek to mitigate bias by removing initial observations that are far from the sample mean, but only to the extent this distance is sufficient to compensate for the resulting reduction in sample size in the calculation of the confidence interval half width."

However, in an automated setting it is important that if insufficient data are first provided to the algorithm, that more data can then be produced and acted upon in an iterative fashion, as demonstrated in Figure 2. It may also be beneficial to continue to iteratively increase the data available to the truncation algorithm until a satisfactory degree of certainty about the warm-up decision has been reached. As a result, it is necessary to adapt MSER-5 to be a sequential, rather than a fixed sample size, procedure.

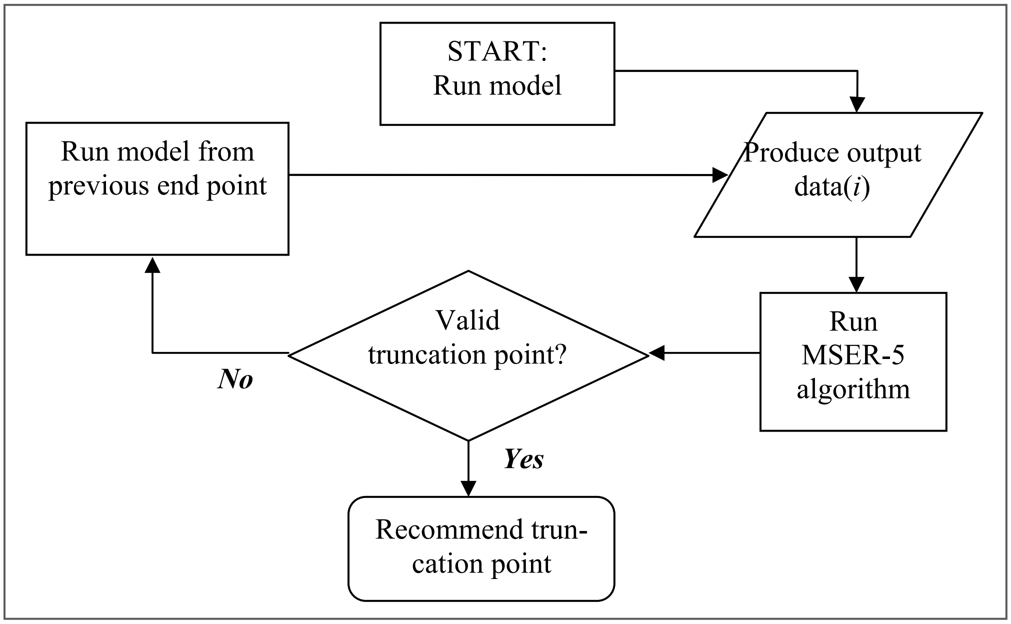

Figure 2: Overview of the framework for a sequential MSER-5 procedure, duplicated for each output variable of user interest (i = 1,…,k: k = the total number of output variables selected by user)

This iterative ability is particularly important in an automated setting because the output data first fed into the algorithm may not have yet reached steady-state or have too recently reached steady state for the algorithm to robustly calculate a truncation point, or is very highly auto-correlated (thus requiring more data to identify steady-state). In order to account for this possibility, if an estimated truncation point falls within the second half of the data series (see an example in Figure 3), this estimate is considered ‘invalid’, and more data are produced (default amount is 10% more batches) and analyzed until a truncation point within the first half of the data series is found. In order to protect against the possibility that no ‘valid’ estimate is found, the algorithm pauses after a set number of ‘invalid’ estimates are found, produces a graph of the data series for the user and requests guidance as to whether to continue or not. The user can also prompt the program to collect a larger amount of data than the default amount at this time, in order to quicken the process. For a more detailed discussion and description of the sequential MSER-5 automated algorithm please refer to Hoad, Robinson, and Davies (2011).

It is important to note that not all decisions in output analysis are currently possible to fully automate and it is arguable that not all decisions should be completely taken out of the user’s hands. Figure 4 shows a tree diagram that we believe captures the main decisions within DES output analysis, succinctly divided into two sets of decisions: “What to do?” and “How to do it?” The MSER-5 automated algorithm is an example of automating the “How.” The “What” decisions are not automated and they would prove more difficult to automate since they require an understanding of the model. We therefore expect the user to decide whether the simulation should be stochastic or deterministic, to understand whether the output of interest is in nature transient, steady state or steady state cycle, and therefore to decide whether warm-up analysis is required. In this context, we also expect the user to decide on the run type of the model (replications or one run), to decide whether a warm-up is appropriate (e.g., output is considered to be steady state), select output variables of interest that are appropriate for warm-up analysis (appropriateness discussed later), and to interact with the warm-up method’s interface in order to choose an appropriate warm-up period from the resulting recommendation(s). It is therefore important that within the sequential framework the algorithm can work with both multiple replications and one long run, dependent upon the user's run type preference. The algorithm should also be able to act upon multiple output variables (indicated in Figure 2), as selected by the user, and communicate with the user in an informative, timely and easy to understand manner.

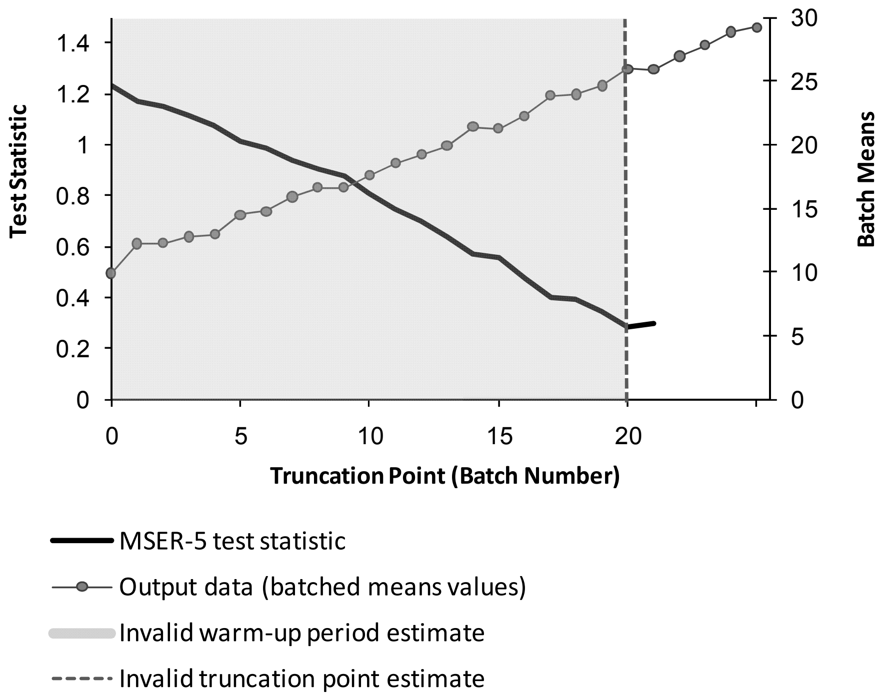

Figure 3: Example of the MSER-5 method producing an "invalid" truncation point estimate

3 IMPLEMENTING A SEQUENTIAL MSER-5 PROCEDURE IN DES SOFTWARE

During our attempt to implement an automatable sequential MSER-5 procedure into existing discrete-event simulation software several issues arose. We now discuss these and some potential solutions.

3.1 Issue 1: Sequential Collection of Replication Data

When running replications, the MSER-5 algorithm can act upon the average values across the replications:

Averaged data series $= \overline{X} = \left[\frac{\sum_{m=1}^{M} x_{m,1}}{M}, \frac{\sum_{m=1}^{M} x_{m,2}}{M}, \ldots, \frac{\sum_{m=1}^{M} x_{m,N}}{M} \right]$, where $x_{n,i}$ is the i^{th} output data value in the m^{th} replication of a total of M replications, each with a run length of N.

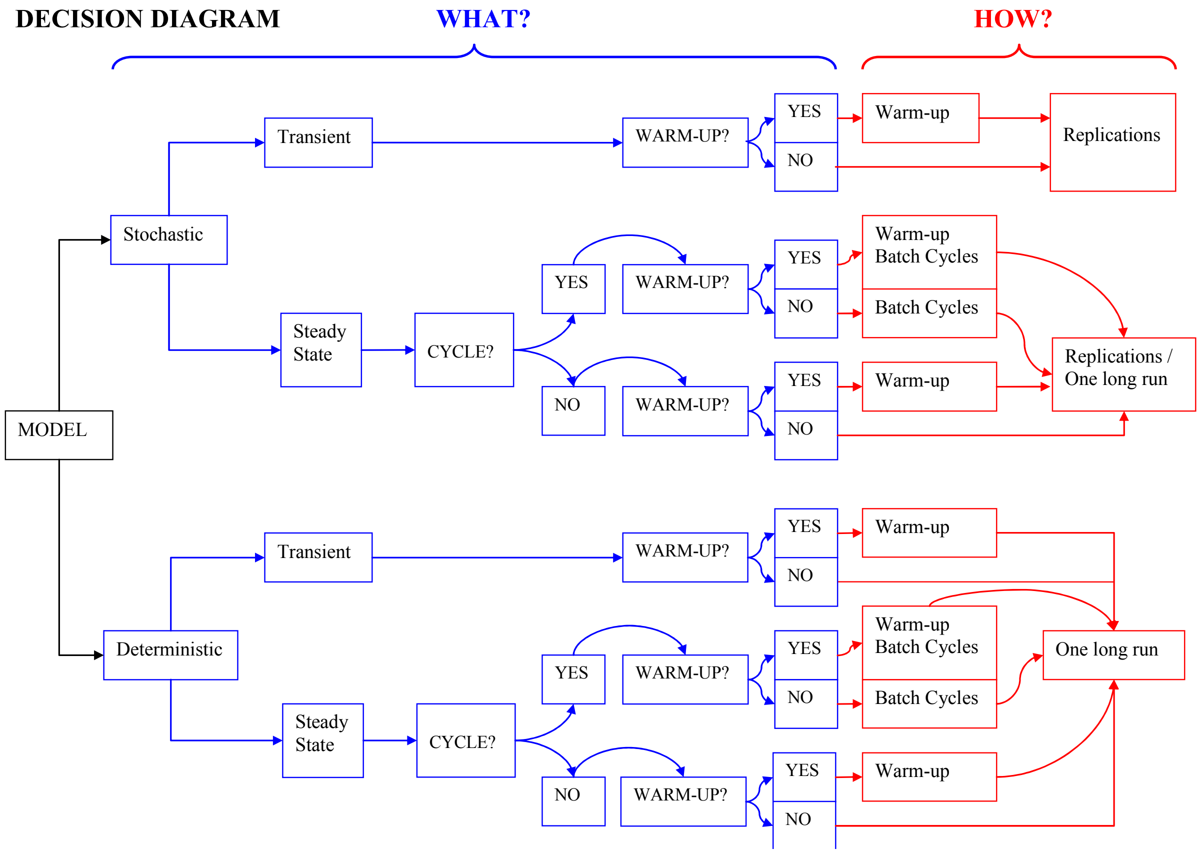

Figure 4: Decision tree outlining the main decisions to be made in DES output analysis

Not only is this the appropriate procedure when intending to run the model with multiple replications, but our previous testing (Hoad, Robinson, and Davies 2010) showed that the MSER-5 algorithm was more robust when applied to this kind of data series, due to the smoothing effect of averaging.

However, in order to run the automated warm-up method efficiently in an iterative fashion (as described in figure 2) it is necessary for the simulation to run on from where it previously terminated. This does not seem to pose a problem for one run, but can require far more effort and data storage when dealing with multiple replications since the end state of every replication needs to be stored so the model can be run on from this point. Multiple saves of each replication's end point were not routinely made in the software we were using and our experience suggests this is an issue with other simulation software as well. Such model status saves could easily be made, but this requires some overhead in storage.

3.2 Issue 2: Output Data Type

When automating the MSER-5 warm-up method, the issue of which type of output variables could sensibly be used for determining the warm-up period arose. Quite often we tell students/simulation users that they should warm-up every output that they are interested in (Robinson. 2004, Law 2009). However, quite often the performance indicators pre-programmed into DES software are not calculated or stored by the program in such a way as to be directly useful to a warm-up algorithm. We have identified two such output types that can suffer from this problem: cumulative values (e.g., average time in queue or average work centre use) and extrema (e.g., minima and maxima of various indicators such as work centre use or queuing time).

Extrema have a tendency to have reducing variance as the quantity of data increases. This is due to changes in the minimum or maximum values likely to become less frequent the longer the simulation runs. Cumulative results also suffer from a reduction in variance as the quantity of data increases. This produces three connected issues. Firstly, if no initialization bias exists, cumulative output data in particular will almost certainly look and behave as if a warm-up period does exist, and hence deleting this warm-up and starting the cumulative calculation again only wastes valid data.

Secondly, if initialization bias does exist, warming-up cumulative output data can lead to large over-estimation of the warm-up period length due to persistence of the bias in the cumulative calculation. These issues do not prohibit using a warm-up method (automated or otherwise) on this output type but suggest that it is not advisable.

Thirdly, this reduction in variation over the length of the output data series can have serious consequences for the functioning of the MSER-5 algorithm in particular. MSER-5 is designed to find the point in the data series where the standard error in the data is at a minimum when discarding all data before that point. Hence, if the variance is continually decreasing it is presumed that the algorithm will have a tendency to indicate a truncation point at or towards the end of the data series, and thus it will rarely produce a valid truncation point. In order to investigate this presumption, 100 artificial cumulative average data series, of length 1100 (chosen to allow the cumulative data to settle to its approximate mean value), were produced from 100 sets of normal random numbers with no initial bias imposed. These data sets were analyzed using the MSER-5 method and a histogram of the 100 subsequent truncation points is shown in Figure 5. The vast majority of the truncation estimates fell towards the end of the data series as predicted. Only in 7% of cases did the truncation point fall within the first half of the data series, and thus would have been considered by the automatic system to be a reportable (valid) estimate.

To avoid such problems when considering cumulative output it would be preferable to work directly with the disaggregated data and therefore it would be helpful if software provided this data. However, most software do not do this hence it may be necessary for the user to disaggregate cumulative time series data themselves, however, this can prove not to be a trivial task, as further discussed in the next subsection.

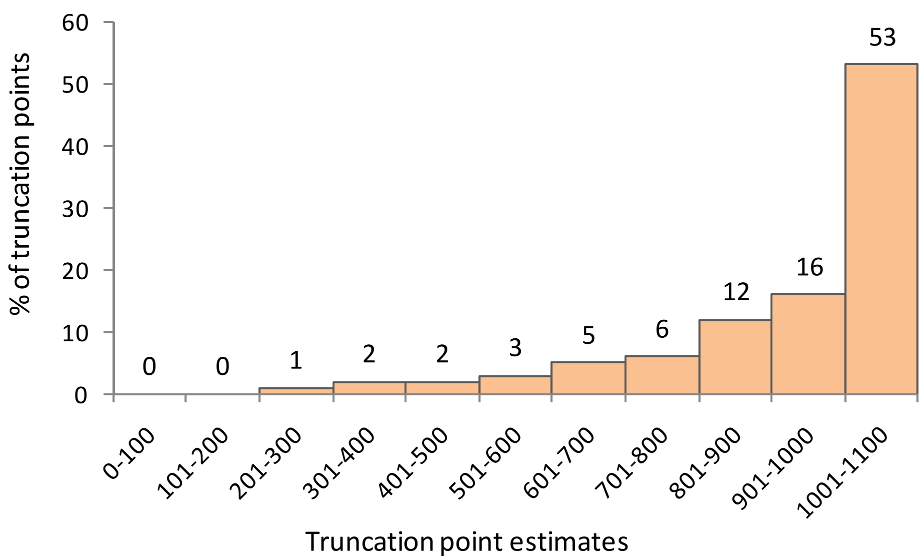

Figure 5: Histogram of the percentage of truncation point estimates falling within certain ranges of the 100 cumulative data series of length $n = 1100$

3.3 Issue 3: Capturing Entity or Time Based Data

In general, commercial simulation software automatically record outputs and define warm-up periods and run-length in terms of time. The user is then required to code for themselves any output that they wish to record by entity. It is interesting that many research papers record output data by entity which is not often supported by commercial simulation software. Sometimes it is useful to record output by entity as some output is entity based, e.g., time in system, time in queue, or queue size.

When considering the problem of disaggregating output for warm-up analysis, as suggested in the previous subsection, it is far more complicated to disaggregate based on time. An example of disaggregating time based data is shown in Table 1 and Figure 6. The user needs to record both the cumulative statistics and the number of entities involved in the cumulative values for each chosen time period, in order to be able to disaggregate the data. For example, if we let $D_i = i^{th}$ disaggregated average value, $y_i = i^{th}$ cumulative average value, and N_i = number of entities involved in y_i, then the i^{th} disaggregated average value can be calculated as

$$D_i = \frac{y_i N_i - y_{i-1} N_{i-1}}{N_i - N_{i-1}}, \text{ for } N_i - N_{i-1} > 0, \text{ otherwise } D_i = 0.$$

Table 1: Example of disaggregation of cumulative averaged time based data (the full data series can be seen in Figure 6).

Cumulative average time in system	Cumulative number of entities leaving system	Disaggregated averages		
1.714	5	V2	=	1.714
2.990	7	(V3*W3-V2*W2)/(W3-W2)	=	6.179
2.414	14	(V4*W4-V3*W3)/(W4-W3)	=	1.838
1.773	20	(V5*W5-V4*W4)/(W5-W4)	=	0.278
1.750	21	(V6*W6-V5*W5)/(W6-W5)	=	1.287
1.650	23	(V7*W7-V6*W6)/(W7-W6)	=	0.597
1.471	27	(V8*W8-V7*W7)/(W8-W7)	=	0.444
1.457	29	(V9*W9-V8*W8)/(W9-W8)	=	1.266
...	...	...		...

It would be useful if software was more flexible in this regard, allowing warm-up and run-length to be defined by number of entities arriving or leaving the model or an element in the model.

3.4 Issue 4: Choosing the Amount of Data to Analyze with MSER-5

Another issue arises when allowing the user to set the number of data points to be used in an automated warm-up analysis. As already mentioned, MSER-5 analyses the data provided and gives the truncation point as the point where the minimum standard error in the data occurs, when the data before that point is ignored. The premise of the algorithm (White 1997) implies that the estimated truncation point is dependent in some part upon the amount of data provided. In that, as the amount of steady state data (after the initial transient) is increased the impact of the initial bias is diluted until the point where, if you were to run the simulation for long enough, a warm-up period would no longer be necessary. Hence it could be advisable for the user not to use MSER-5 to analyze data of a run length much longer than the intended experimental run length. This suggests that the MSER-5 truncation point does not settle with increasing run length but tends to zero. However, during previous extensive testing of the MSER-5 and some current empirical testing specifically targeted to look at this phenomenon, it was not at all obvious that this is the case. For example, 100 sets of data were constructed from N(100,19) random numbers including an initial transient which decreased towards the mean at a gradient of -2 for the first 50 data points. Using MSER-5 to analyze these data sets with increasing run length (100 to 1100 in steps of 100), resulted in 75% of these data sets having the same estimated truncation point at a run length of 1100 as the first valid truncation point (estimated at run lengths of 100 to 300 data points). Of the remaining data sets, 18% had a decrease in their truncation point by an average of 14 data points (st.dev. = 12.7), and 6% had an increase in their truncation point by an average of 18 data points (st.dev. = 16). (This experiment was reproduced with artificially induced initial transients of varying severity and on artificial data sets with varying variance values. Results were found to be similar to that stated above.)

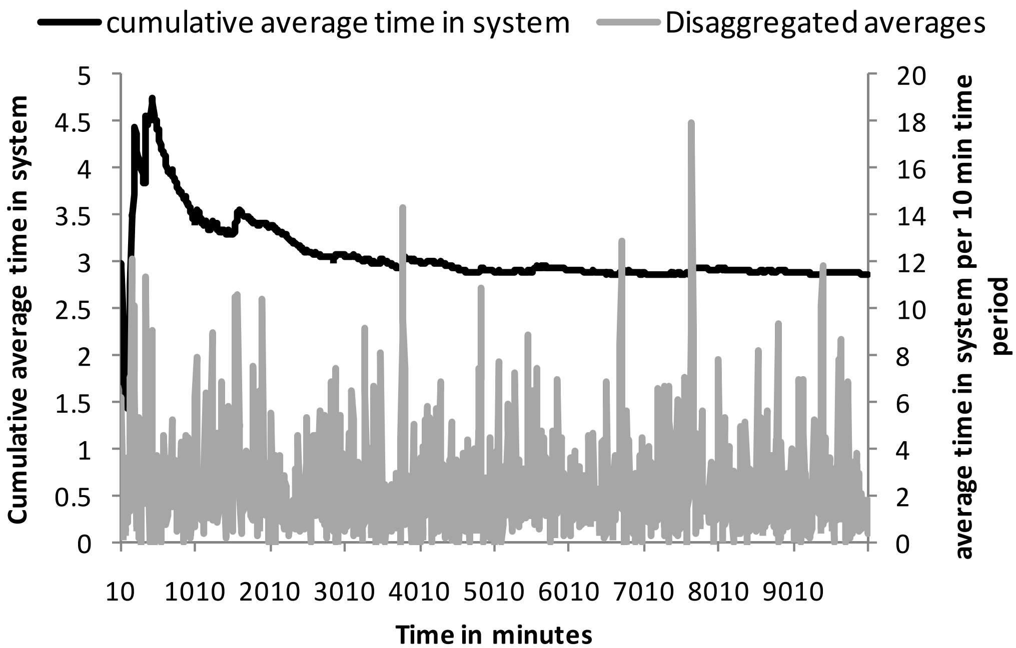

Figure 6: Plot of the full series of cumulative and disaggregated data of which an excerpt is displayed in Table 1

4 CONCLUSION

Through our collaboration with a simulation software supplier, we have identified a series of issues that need to be addressed to enable automation of a warm-up procedure; in this case MSER-5. In summary we would recommend that simulation software suppliers enhance their products to enable the following:

- Enable saves of the current end point for each replication in a simulation experiment so the run may be continued for each replication whenever required.
- Prevent the determination of warm-up on cumulative data.
- Provide cumulative output data in disaggregated form as well as in cumulative form.
- Allow warm-up and run-length to be defined by number of entities arriving or leaving the model, or an element in the model, as well as in terms of time.

In making these recommendations we recognize that some software products may already include some of these features.

REFERENCES

Hoad, K., S. Robinson, and R. Davies. 2010. “Automating Warm-Up Length Estimation.” *Journal of the Operational Research Society* 61:1389 –1403.

Hoad, K., S. Robinson, and R. Davies. 2011. “AutoSimOA: A Framework for Automated Analysis of Simulation Output.” *Journal of Simulation* 5: 9–24.

Law, A.M. 2009. *Simulation Modeling and Analysis*. 5th ed. New York: McGraw-Hill.

Robinson, S. 2004. *Simulation: The Practice of Model Development and Use.* Chichester, UK: Wiley.

White Jr., K. P. 1997. “An Effective Truncation Heuristic for Bias Reduction in Simulation Output.” *Simulation* 69(6):323–334.

White Jr., K. P., M. J. Cobb, and S. C. Spratt. 2000. “A Comparison of Five Steady-State Truncation Heuristics for Simulation.” In *Proceedings of the 2000 Winter Simulation Conference*, edited by J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick, 755–760. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

White Jr, K.P., and Robinson, S. 2010. “The Problem of the Initial Transient (Again), or Why MSER Works.” *Journal of Simulation* 4: 268-272.

AUTHOR BIOGRAPHIES

KATHRYN A. HOAD is an Assistant Professor in the Operational Research and Management Sciences Group at Warwick Business School. She holds a BSc in Mathematics and its Applications from Portsmouth University, an MSc in Statistics and a PhD in Operational Research from Southampton University. Her research interests cover statistical analysis/modeling and D.E.S. output analysis/experimentation. Her email address is kathryn.hoad@wbs.ac.uk.

STEWART ROBINSON is Professor of Operational Research and Head of the Operational Research and Management Sciences Group at Warwick Business School. Previously employed in simulation consultancy, he supported the use of simulation in companies throughout Europe and the rest of the world. He is author/co-author of four books on simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modeling, model validation, output analysis and alternative simulation methods (discrete-event, system dynamics and agent based). He is currently helping to lead a research project that is investigating the use of simulation with lean in healthcare. Professor Robinson is co-editor of the *Journal of Simulation* and Vice President of the United Kingdom Operational Research Society. For more details see: www.btinternet.com/~stewart.robinson1/sr.htm. His email is stewart.robinson@warwick.ac.uk.