Procedia Computer Science

Volume 29, 2014, Pages 73–83

ICCS 2014. 14th International Conference on Computational Science

Optimal Run Length for Discrete-Event Distributed Cluster-Based Simulations

Francisco Borges[1], Albert Gutierrez-Milla[1], Remo Suppi[1], and Emilio Luque[1]

Department of Computer Architecture & Operating Systems
Universitat Autonoma de Barcelona, Bellaterra,08193, Barcelona, Spain
francisco.borges@caos.uab.es, albert.gutierrez@caos.uab.cat, remo.suppi@uab.es, emilio.luque@uab.es

Abstract

In scientific simulations the results generated usually come from a stochastic process. New solutions with the aim of improving these simulations have been proposed, but the problem is how to compare these solutions since the results are not deterministic. Consequently how to guarantee that the output results are statistically trusted. In this work we apply a statistical approach in order to define the transient and steady state in discrete event distributed simulation. We used linear regression and batch method to find the optimal simulation size. As contributions of our work we can enumerate: we have applied and adapted the simple statistical approach in order to define the optimal simulation length; we propose the approximate approach to normal distribution instead of generate replications sufficiently large; and the method can be used in other kind of non-terminating science simulations where the data either have a normal distribution or can be approximated by a normal distribution.

Keywords: Parallel and distributed simulation, Parallel discrete-event simulation, High performance distributed simulation, Output analysis, Run length, Transient state, Steady state

1 Introduction

Many researchers in Parallel and Distributed Discrete Event Simulation propose techniques and solutions to improve their simulations by using High Performance Computing (HPC). Generally, these researchers have to compare the previous solution with a new developed approach in order to show their scientific contributions. The challenging questions are: how to compare different models or approaches since simulations produce stochastic results? and do simulation results support the conclusions and generalizations? The solution is guarantee that the output results of each model be statistically reliable. It permits that the performance measures can be compared.

Such simulations need to have the run length statistically well-defined. In this regard, it is a requirement that the statistical method has a fast convergence in order to avoid extra computing. To define the optimal simulation run length is necessary to distinguish two phases

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2014

doi:10.1016/j.procs.2014.05.007

of simulation: transient stage and steady stage. In the first stage, simulation needs to initialize the variables and structures. Simulations are not steady in this stage, therefore the values generated must not be considered in output analysis. A common method to overcome this is execute the simulation for long time until the data generated in this phase become insignificant. In this way, the data of transient phase will not influence the output results. This method for realistic simulation is not appropriate because it requires a lot of computational resources. A more efficient solution is to truncate the initial data [1, 15, 18].

The steady stage starts immediately after the initial transient stage. The data produced in steady stage will be used for output analysis [7, 1, 6]. In this phase the researcher must know what is the appropriate size of the steady stage in order to produce results statistically trusted.

The run length of simulation is associate basically with the size of the transient stage; the time necessary to produce data statistically reliable; and the type of simulated distribution. This paper addresses our research problem as: how to define optimal run length to have output results reliable for realistic distributed simulation? In this work we will focus on non-terminating [6] simulations. Because we have been working in high performance distributed simulation by using a fish schooling simulator [13, 14] which has non-terminating behaviour. In order to find the optimal run length, we will use the practical statistical procedure based on Chung[5]. This procedure identifies the transient stage and steady stage by using linear regression and batch method. This method considered that the distribution of data is normally distributed. Therefore, we adapt it for our context which has binomial distribution. This method is simple and not required high statistical background. In addition it can be applied in any non-terminating simulation which distributions might be approximated by normal distribution.

This paper is organized as follow: we show some previous related works in section 2. We selected the number of created clusters during the simulation as performance measure to evaluate the output. Therefore we present an overview of cluster-based partitioning of our fish schooling simulator in section 3. The statistical method is explained and experimentally applied on our simulator in the section 4. Finally, in section 5 the conclusions and future work are proposed.

2 Related Work

One of the most important step in simulation is the output analysis. Since simulation deals with stochastic results it is a requirement to use statistical methods in order to check and make conclusions about the results obtained. Basically, two distinct phases must be analysed in simulations: transient and steady. In the first one the initial data must be discarded because these data produce an initialization bias. In the latter phase, when the simulation arrives into steady, the data must be produced in order to produce statistical estimator with appropriate confidence intervals. In transient phase, the problem is to figure out the number of observations to discard. In the steady state phase, the problem is find the run length which has trustworthy results.

There are three main pitfalls into output-data analysis [7]: underestimation of variances and standard deviations when the replication are not independent; failure to define warm-up period; and failure to determine the statistical precision of simulation output statistics by the use of a confidence interval.

Many output analysis methods are proposed, compared and analysed. In [8], they compare six methods to detect the transient phase length in a non-terminating simulation. They are the following: 1) Welch's method [17] is the simplest technique to determine the warm-up however this method has a high subjectively because it is a graphical method. As the simulation has stochastic behaviour, the graphical procedure requires multiple replications of the simulation

in a pilot study [7], [8]; 2) SPC method requires multiple replications and at least 20 batch means must pass for normality and correlation tests; 3) The Randomization Test verifies the null hypothesis where the mean is unchanged throughout the run. It is a statistical method that no requires the normality of data; 4) The Conway Rule method truncate the initial data of output in order to reduce bias. In this method the number and length of the replications must be specified; 5) Crossing of the Means Rule method establishes the truncation point counting the number of crossings of the mean until reach a pre-specified value; 6) Marginal Standard Error Rule-5 (MSER-5) defines the truncation point considering the batch size, run length and five as the number of batches. In [11] they suggest two new algorithms for using the MSER statistic and compare them by using mean squared error. They show that MSER is asymptotically proportional to the mean squared error and therefore they argue that it is a good solution for initial transient algorithms. The removal of the effect of the initial conditions is a challenging problem. In [1] it is suggests a method for eliminating the bias by truncating the initial observations. In [18] they compare the performance of five well-known truncation heuristics. Some authors ignore the first batches in order to reduce the potential effects of initialization bias [15].

In order to analyse the steady state phase, there are two common techniques used, which are: replications [7] or batch [15], [9]. Replications are independent simulations, which must be executed many times and their results are statistically analysed at the end. Batch technique is more applied in a non-terminating simulation. In this method, the observations of simulation are divided in batches, and each batch is considered as a replication. Methods of batch means and independent replications in the context of non-terminating simulation output analysis are compared by [2]. Batch methods improve the effects of initialization bias but produces batch means that often are correlated [2]. Consequently, the correlation between these batch must be checked. Determine the number, correlation and size of batch are a challenge in output analysis. In [12] he shows that number of batch need not be too large in order to decrease the risk of an invalid confidence interval. A vast literature about the use of batch size effect is available in [10]. Some methods in [1] and [16] assume approximately normally distributed because they consider sufficiently large run length. Other methods increase batch size until the batch means pass the Shapiro-Wilk test [15]. In addition, in [10] he observes that many batching algorithms tried to find a large, or even the largest, value of replications that also generate a valid confidence interval.

Several output analysis methods are not recommended for realistic simulations. Because they require large run length and consequently need huge computational resources. Depending on simulation parameters the total execution time of simulation to obtain the results to analysis can be not feasible. Therefore we must use statistical methods that neither demand many batch non large run length but that are statistically reliable.

3 Cluster-Based Partitioning

In order to develop the output analysis, we have to select a performance measure that would be appropriate to evaluate the simulation [9]. For that, we choose the number of cluster which are created during the simulation. One of the challenges in distributed simulation is how to distribute individuals on the distributed architecture in order to obtain the best scalability and efficiency. In [14] was implemented a partitioning algorithm using a cluster-based approach. This approach consists of assigning to each node a fixed set of individuals.

The partitioning method uses a hybrid partitioning method based on Voronoi diagrams and covering radius criterion. Voronoi diagram is a data structure in computational geometry where

given some number of objects in the space, their Voronoi diagram divides the space according to the nearest-neighbor rule [3]. In our case each object is represented by an individual (fish) and it is associated with the area closest to it. Covering radius criterion consists of trying to bound the area (c_i) by considering a sphere centered at c_i. These centroids (c_i) contain all the objects of the problem domain that lie in the area [4].

The individuals are associated with a position in a three-dimensional euclidean space and together with the euclidean distance generating a metric space. The distance between objects of this metric space is defined by a set of objects X subset of the universe of valid objects U and a distance function $d : X^2 \rightarrow R$, thus $\forall x, y, z \in X$, must be met the following conditions:

Positiveness: $d(x, y) \geq 0, d(x, y) = 0 \Rightarrow x = y$

Symmetry: $d(x, y) = d(y, x)$

Triangular inequality: $d(x, y) + d(y, z) \geq d(x, z)$

These conditions determine the visibility of individuals and allowing use of similarity or proximity within the distributed simulation. Therefore the partitioning method consists of two phases: the centroids selection by means of covering radius criterion which ensures it a set of centroids far away enough the others; and the space decomposition by means of voronoi diagrams which allow it define similar size areas with similar number of individuals. The aim of partitioning algorithm is create dense and not scattered clusters in order to simulate the strong cohesion and high level of synchronization of fish schooling.

Initially the individuals are uniformly distributed in space in the simulation. Therefore the partitioning algorithm consumes many simulations steps to run and converge into steady state. We consider that the stabilization occurs when the number of cluster and the distance among the individuals vary between a specific range(standard deviation). The next section (4.1) presents how we identify the steady state at distributed cluster-based partitioning simulations.

4 Statistical Method

The statistical method applied on our simulator is based on [5]. This method is divided in three parts: 1) the first one is to identify when the steady state starts; 2) the second one is to identify the run length; 3) and the last one is the replication analysis. This method was experimentally applied on our simulator by using 8192 individuals running in four cores.

4.1 Method to Identify the Steady State

In Figure 1 we can observe the behaviour of cluster-partitioning algorithm. Along the execution, the algorithm goes creating and deleting the clusters. The variation in number of clusters occurs because the individuals constantly change their position. We can verify clearly two phase: transient state and steady state. First, we will discuss about the transient state.

In the transient state occurs a fast increasing of number of clusters. It happens because the cluster-partitioning algorithm verify that the number of clusters is not enough to represent all individuals inside of the Voronoi diagram. In addition, the individuals are uniformly distributed in space. Consequently, the simulation spends some steps trying to converge. Identify this phase is important because these steps (warm-up) does not should be considered in output analysis.

We use the linear regression approach suggested by [5] to identify the end of transient state. This approach uses the least-squares method to determine if the linear regression slope

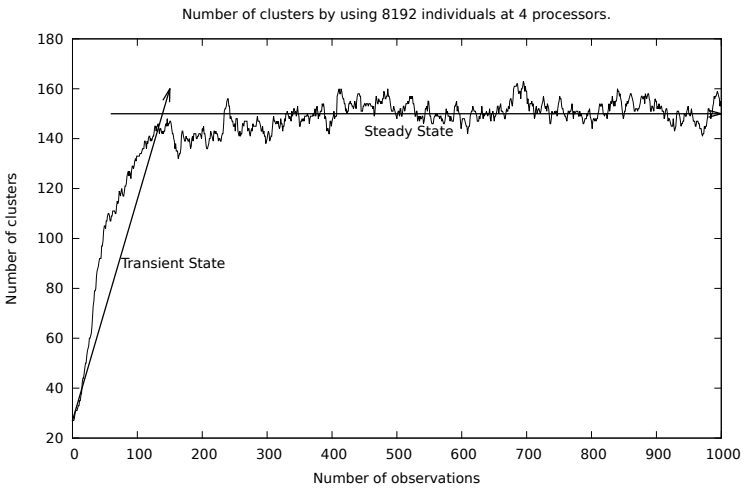

Figure 1: Transient and Steady State

coefficient is equal to zero for a specific range of observations. The range of observations must be advances if the slope is not zero or insignificant. We consider 30 observations for each range and we increment always by 5 observations forward ever that the linear regression does not find the approximate end transient state.

The transient state finish when the slope is equal to zero. Therefore the null hypothesis is that the slope of the data is zero. The null hypothesis is rejected if the *P-value* is less than 0.05. Because this α level is statistically significant. The Table 1 presents the *P-value* obtained for some ranges. In the range 126-155, the *P-value* is more than α (0.05) this imply that the coefficient is near of zero. Therefore we can assume that after the 126 steps of simulation the transient state is finished.

Range	Coefficient	P-value
1-30	1.36059	2.23263e-23
6-35	1.65911	3.32428e-24
11-40	1.90394	8.23758e-26
...	...	...
111-140	0.242365	3.50807e-07
116-145	0.174877	6.4216e-05
121-150	0.162562	0.000136116
126-155	**0.0600985**	**0.103787**
131-160	-0.118227	0.0540278
136-165	-0.373892	2.46027e-05
...	...	...
996-1025	-0.9	0.107547

Table 1: Linear regression for transient state

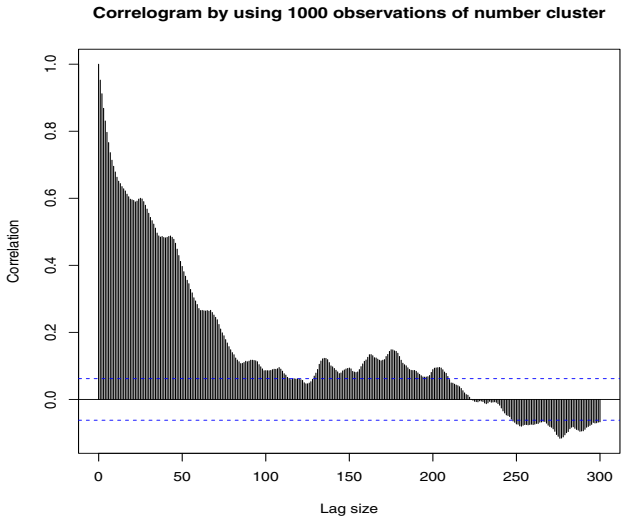

Figure 2: Correlogram

4.2 Method to Identify the Run Length

Our fish schooling simulator has a strong correlation in each step simulation. The cluster-partitioning algorithm passes the current partitioning state to next step simulation. In this way, we have to identify the autocorrelation in order to estimate the variance. It avoids that statistical study be affected negatively. We will use the batch method to finding the autocorrelation. The batch method consists of the following three steps [5]: 1) identify the non significant correlation lag size; 2) make a batch ten times the size of the lag; 3) and make the steady state replication run length ten batches long.

The non significant correlation lag size represents the interval between observations that have little correlation to each other. In order to define lag size we use the correlogram diagram by using 300 as observation size. As we can see in Figure 2, the non significant lag occurs at 223th observation where the correlation is equal to zero.

The next step is to figure out the correlation, which is obtained by batch ten times the size of the lag. First of all, we have to calculate the batch size by using the equation 1. Next, we have to obtain the time for a single observation, shown in equation 2. After, we calculate the transient length time by equation 3. And finally we have to calculate the execution time of batch in equation 4.

$$batch\_size = 10 * nonsignificant\_lag\_size \tag{1}$$

$$single\_observation\_time = \frac{total\_time\_execution}{number\_of\_observations} \tag{2}$$

$$transient\_length\_time = single\_observation\_time * transient\_phase \tag{3}$$

$$batch\_time = single\_observation\_time * batch\_size \tag{4}$$

In order to finish the correlation calculation, we have to calculate the run length by using the equation 5. The Table 2 summarizes these calculations.

$$length\_of\_run = transient\_length\_time + (10 * batch\_time) \tag{5}$$

Variable	Value
Total time execution	490.728017 s
Number of observations	1000
Nonsignificant lag size	223 observations
Batch Size	2230 observations
Single observation time	0.490728 s
Batch Time	1094.323477 s
Transient phase	126 observations
Length of run	**11005 s**
Number of simulation steps	22426

Table 2: Calculations of run length

The run length found is approximately 22426 observations. We have to discard the transient state and the rest must be divided by the number of batch size, in this case 2230. Then, these batch are the individual replications that should be used to output analyses. But before, we have to verify if these replications are statistically significant. Therefore, we will analyse and compare the means by using ANOVA and Duncan multiple-range test.

4.3 Replication Analysis

We have to ensure that the number of replication is enough. Then, we have to know the Standard error (equation 6) by using relative precision.

$$standard\_error = t_{1-\alpha/2,n-1} * s/\sqrt{n} \tag{6}$$

Where t is the t-distribution for $1 - \alpha/2$ and $n - 1$ degrees of freedom; s represent the standard deviation of the replication means; and n is number of replications, in our case 10. Relative precision give us a extensible model because we can use relative mean value. We will use 0.10 as relative precision (equation 7). Therefore our standard error can be just 10% of mean for our data.

$$relative\_precision = \frac{standard\_error}{\bar{\bar{x}}} \tag{7}$$

Where $\bar{\bar{x}}$ is the mean of the replication. The equation 8 finds out the required number of replications to specific level of relative precision.

$$number\_replication = \left[\frac{standard\_error}{vrp * \bar{\bar{x}}}\right]^{1/2} \tag{8}$$

Where vrp is the value of relative precision (0.09). The Table 3 shows the results after apply these equations.

As we can observe, in the Table 3, the Relative Precision is less than 0.10. It means that the current number of replications (10) is sufficient. In addition, in this experiment the number of replication required points out we need of one replication to obtain the relative precision

Variable	Value
s	6.105042
t	2.262157
Standard Error	4.367284
Relative Precision	0.026234
Required number of replication	0.849683

Table 3: Statistical summary of replications

wished. Since we have the enough replication number, then we have to determine if the mean are statistically significantly different from the others at a given α level. And therefore we will use analysis of variance (ANOVA) in order to verify which means are different.

ANOVA requires normal distribution but our replications have binomial distribution. Based on Central Limit Theorem, some author propose generate replications sufficiently large. But it is costly for realistic distributed simulation. To solve this problem we propose to approximate the binomial distribution to normal distribution, using equation 9.

$$\frac{number\_cluster - (np)}{\sqrt{np(1-p)}} \tag{9}$$

Where n and p come from the binomial distribution. For each number of cluster of each replications we approximate to normal distribution after we apply the ANOVA.

Groups	Count	Sum	Average	Variance		
B1	2230	211.509	0.09485	1.00444		
B2	2230	153.274	0.06873	1.00289		
B3	2230	84.1697	0.03774	1.00178		
B4	2230	413.192	0.18529	1.00553		
B5	2230	247.434	0.11096	1.00396		
B6	2230	52.6732	0.02362	1.00134		
B7	2230	205.19	0.09201	1.00329		
B8	2230	177.144	0.07944	1.00289		
B9	2230	329.878	0.14793	1.00460		
B10	2230	109.617	0.04916	1.00187		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F critical
Between Groups	49.6412	9	5.51569	5.49777	1.4E07	1.8803
Within Groups	22362.7	22290	1.00326			
Total	22412.3	22299				

Table 4: ANOVA result

In Table 4, the P-value is less than 0.05, thus we can conclude that the means are different. We can reject the hypothesis that all the replications have identical means. However, this does not mean that every average differs with every average. We will use the Duncan test to identify where the differences are. The Duncan test results are show in Table 5.

In Table 5, means with the same letter are not significantly different. Consequently the replications with the same letter cannot be used together to do output analysis. Because those replications must be significantly different for representing an overall population. In

Groups	Treatments	Means
a	B4	0.18530
ab	B9	0.14790
bc	B5	0.11100
bcd	B1	0.09485
bcd	B7	0.09201
cde	B8	0.07944
cde	B2	0.06873
cde	B10	0.04916
de	B3	0.03774
e	B6	0.02362

Table 5: Duncan test result

Table 3 the *Required number of replication* points out that about one replication is sufficient in this experiment. Therefore, we could choose the B1 replication for simulation result analysis. For this reason, the simulation does not need to execute more steps in order to obtain other replications.

But what we should do if the *required number of replication* was three, for example? In this case, we need six replications. Because, as we can observe in this experiment B1, B4 and B6 replications are significantly different. However the replications B2, B3 and B5 must be discarded because they are equal to B1. Therefore, in this hypothetical situation, the experiment must execute 13506 observations ($(6 * batch\_size) + transient\_phase$) to get reliable results. Suppose another situation where the required number of replication is four. More replications would be necessary. Because replications B7, B8, B9 and B10 are not significantly different. As a result, quantity of the number of replications in simulation depends on if the replications are significantly different. Consequently, the total run length of simulation can be variable.

5 Conclusions

The run length, number of replication and steady state of simulation should be well specified in order to produce results that must be statistically reliable. In addition, the statistical methods have to give optimal results when we are treating with distributed realistic simulation. In this work we have applied and adapted the simple statistical approach proposed by Chung[5] for discrete-event distributed cluster-based simulations. We propose the approximate approach to normal distribution instead of generate sufficiently large replications. The aim is to decrease the total simulation time but keeping the statistical confidence for output analysis. We observed that the run length depends on if the means represent statistically the overall population. The means that are not significantly different must be discarded and other replications must be used instead of them. In this way we can obtained experiments that are statistically trusted.

The main contributions and conclusions that can be extracted are:

- We have applied and adapted the simple statistical approach in order to define the optimal simulation length. We avoid the simulation runs steps of simulations beyond what is needed. Consequently, it save computational resources.

- In addition, we propose the approximate approach to normal distribution instead of generate sufficiently large replications. The method can be used in other kind of non-

terminating science simulations where the distribution of data is normal or it can be approximated by a normal distribution.

The main objectives for future work are: uses time series analysis techniques in order to define the optimal simulation and compare these techniques with the approach presented in this paper; and implement the statistical method shown in this paper in our parallel and distributed simulator.

6 Acknowledgments

- This research has been supported by the MICINN Spain under contract TIN2007-64974 and the MINECO (MICINN) Spain under contract TIN2011-24384.

References

[1] C. Alexopoulos. A comprehensive review of methods for simulation output analysis. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 168–178, 2006.

[2] Christos Alexopoulos and David Goldsman. To batch or not to batch? *ACM Trans. Model. Comput. Simul.*, 14(1):76–114, January 2004.

[3] Franz Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, September 1991.

[4] E. Chavez and G. Navarro. An effective clustering algorithm to index high dimensional metric spaces. In *Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00)*, pages 75–, Washington, DC, USA, 2000. IEEE Computer Society.

[5] Christopher A Chung. *Simulation modeling handbook: a practical approach*. CRC press, 2003.

[6] David Goldsman and G. Tokol. Output analysis procedures for computer simulations. In *Simulation Conference, 2000. Proceedings. Winter*, volume 1, pages 39–45 vol.1, 2000.

[7] Averill M. Law. Statistical analysis of simulation output data: The practical state of the art. In *Winter Simulation Conference*, pages 65–74, 2010.

[8] P.S. Mahajan and R.G. Ingalls. Evaluation of methods used to detect warm-up period in steady state simulation. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 1, pages –671, 2004.

[9] Marvin K. Nakayama. Output analysis for simulations. In *Proceedings of the 38th Conference on Winter Simulation*, WSC'06, pages 36–46. Winter Simulation Conference, 2006.

[10] B.L. Nelson. Thirty years of "batch size effects". In *Simulation Conference (WSC), Proceedings of the 2011 Winter*, pages 393–400, 2011.

[11] R. Pasupathy and B. Schmeiser. The initial transient in steady-state point estimation: Contexts, a bibliography, the mse criterion, and the mser statistic. In *Simulation Conference (WSC), Proceedings of the 2010 Winter*, pages 184–197, 2010.

[12] Bruce Schmeiser. Batch size effects in the analysis of simulation output. *Operations Research*, 30(3):556–568, 1982.

[13] Roberto Solar, Francisco Borges, Remo Suppi, and Emilio Luque. Improving communication patterns for distributed cluster-based individual-oriented fish school simulations. *Procedia Computer Science*, 18(0):702 – 711, 2013. 2013 International Conference on Computational Science.

[14] Roberto Solar, Remo Suppi, and Emilio Luque. High performance distributed cluster-based individual-oriented fish school simulation. *Procedia CS*, 4:76–85, 2011.

[15] Natalie M. Steiger, Emily K. Lada, James R. Wilson, Jeffrey A. Joines, Christos Alexopoulos, and David Goldsman. Asap3: A batch means procedure for steady-state simulation analysis. *ACM Trans. Model. Comput. Simul.*, 15(1):39–73, January 2005.

[16] A. Tafazzoli, J.R. Wilson, E.K. Lada, and N.M. Steiger. Skart: A skewness- and autoregression-adjusted batch-means procedure for simulation analysis. In *Simulation Conference, 2008. WSC 2008. Winter*, pages 387–395, 2008.

[17] P. D. Welch. The statistical analysis of simulation results. In S. Lavenberg, editor, *The Computer Performance Modeling Handbook*, pages 268–328. Academic Press, San Diego, California, 1983.

[18] Jr. White, K.P., M.J. Cobb, and S.C. Spratt. A comparison of five steady-state truncation heuristics for simulation. In *Simulation Conference, 2000. Proceedings. Winter*, volume 1, pages 755–760 vol.1, 2000.