Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

ON THE SIMULTANEOUS CONSTRUCTION OF SAMPLE PATHS

Enver Yücesan

INSEAD
Technology Management Area
Boulevard de Constance
77305 Fontainebleau Cedex, FRANCE

Sheldon H. Jacobson

Department of Industrial & Systems Engr.
Virginia Polytechnic Institute and
State University
Blacksburg, Virginia, 24061-0118 U.S.A.

ABSTRACT

Sensitivity analysis and optimization within stochastic discrete event simulation require the ability to rapidly estimate performance measures under different parameter values. One technique, termed "rapid learning," aims at enumerating all possible sample paths under different parameter values of the model based on the observed sample path under the nominal parameter value. There are two necessary conditions for this capability: *observability*, which asserts that every state observed in the nominal path is always richer in terms of feasible events than the states observed in the constructed paths, and *constructability*, which, in addition to observability, requires that the lifetime of an event has the same distribution as its residual life. This paper asserts that the verification of the observability condition is an NP-hard search problem. This result, in turn, implies that it is algorithmically not possible to find parameter values satisfying observability; hence, it encourages the development of heuristic procedures. Further implications are also discussed.

1 INTRODUCTION

Most techniques used in simulation output analysis are adopted from statistics and optimization. Comparing alternative system designs (Yücesan 1994), ranking and selection (Goldsman and Nelson 1994), sensitivity analysis, response surface methodology (Kleijnen 1987), optimization (Jacobson and Schruben 1989, Healy and Schruben 1991) are activities in simulation analysis where principles from statistics and mathematical programming are widely used.

These techniques require the ability to rapidly estimate performance measures under different parameter values. In naive simulation, this requirement is satisfied by conducting a set of replications under every possible parameter value, which, for complex models, can be a slow exercise. Considerable effort has been expanded to expedite the estimation process. On the one hand, parallel simulation offers one feasible approach. For example, Schruben (1992) runs parallel replications in SIGMA. In a parallel computing environment, Heidelberger (1988) describes the correct analysis of the replications, each executed on a separate processor.

On the other hand, techniques that take advantage of the special characteristics of a simulated environment have also been proposed for quick estimation of performance measures under different parameter values. Frequency Domain Methodology (Schruben and Cogliano 1987) modifies the values of the parameters during a run to assess their impact on the performance measure of interest. Such information can then be used for sensitivity analysis or as part of an optimization scheme. Infinitesimal Perturbation Analysis (IPA) (Ho et al. 1979) is another approach which, in one run, assesses the impact of a small change in the parameter values on the performance measure. The application of IPA to sensitivity analysis or to optimization is immediate. Neither technique can handle discrete parameter values; appropriate modifications have to be incorporated.

Another technique, termed "rapid learning," aims at enumerating (perhaps in parallel) all possible sample paths under different parameter values of the model based on the observed sample path under the nominal parameter value (Cassandras and Strickland 1989). This technique can be applied to the analysis of systems with discrete parameters. Moreover, the system under study may not necessarily be a simulation model, but a real system in operation. That is, the technique can be utilized on line.

There are two necessary conditions for this capability: *observability*, which asserts that every state observed in the nominal path is always richer in terms of feasible events than the states observed in the constructed paths, and *constructability*, which, in addition to observability, requires that the lifetime of an event has the same cumulative distribution function (cdf) as its residual life. The objective of this paper is to show that the *verification* of the observability condition is an NP-hard search problem. This result, in turn, implies that it is algorithmically not possible to find parameter values satisfying observability; hence, it encourages the development of heuristic procedures.

The theory of *computational complexity* provides a well-defined framework to assess the tractability of decision problems (Garey and Johnson 1979). A *decision problem* is one whose solution is either "yes" or "no."

In general, we are interested in finding the most "efficient" algorithm to solve a problem. Typically, the fastest algorithm is considered as the most efficient. We then define the *time complexity function* for an algorithm as the maximum amount of time needed by the algorithm to solve a problem instance of a particular size, which represents a worst-case performance criterion.

Note that the *size* of a problem instance is defined as the amount of input data needed to describe the instance. Various encoding schemes are possible to describe a problem instance. The most widely accepted scheme, which is the one adopted here, is the number of tape cells on a Turing machine.

A *polynomial-time algorithm* is one whose time complexity function is O(p(n)), where p is a polynomial function and n denotes the size of the problem instance. Any algorithm whose time complexity function cannot be so bounded is called an *exponential-time algorithm.* Given the explosive growth rates for exponential complexity functions, polynomial-time algorithms are much more desirable from a practical point of view. It is well accepted that a problem is not "well-solved" until a polynomial-time algorithm has been found for it (Garey and Johnson 1979; p. 8).

Decision problems in the class NP are those problems for which a potential solution can be verified in polynomial time in the size of the problem instance. The complete problems for this class (that is, NP-complete problems) are the hardest problems in NP such that, if one such problem could be solved in polynomial time, then all problems in NP could be solved in polynomial time. Moreover, NP-hard problems are search problems which are provably at least as hard as NP-complete decision problems.

Using the definitions in Jacobson and Yücesan (1995), we assume that the size of a discrete event simulation model specification is n. This is the number of tape cells on a Turing machine required to represent a model implementation of the model specification such that it can be executed. Note that model implementations are not unique, in that there are several possible model implementations associated with each model specification. Any event of the model specification is also assumed to be executable in polynomial time in n; that is, $p_1(n)$. This assumption restricts our work to a subclass of simulation models. Such a subclass, however, contains all of the relevant simulation models from a practical point of view, as models whose events could take an exponential amount of time to execute would not have much use in a simulation study. These assumptions will be used in all the subsequent theorems, unless it is otherwise stated.

To define the problem in a precise fashion, an appropriate modeling framework must be provided. This is done in Section 2, where the principal issues in modeling discrete event dynamic systems (DEDS) are also discussed. The conditions of observability and constructability are defined in Section 3. Section 4 discusses the implications of the results and draws further conclusions.

2 MODELING DISCRETE EVENT DYNAMIC SYSTEMS

Physical phenomena have been successfully characterized and analyzed using the Continuous-Variable Dynamic System (CVDS) framework. Within this framework, system states are depicted by continuous variables which change with respect to time. Hence, physical quantities such as velocity, temperature, acceleration, and flow can be modeled using differential or difference equations. In the absence of closed-form solutions, efficient numerical techniques exist for solving these equations.

Man-made systems such as flexible manufacturing systems or telecommunication networks do not fit within the CVDS framework. This is because system states are often depicted by discrete variables, and that they change with the occurrence of asynchronous events. For the characterization of such systems, a Discrete Event Dynamic System (DEDS) framework has been proposed. Finite-state automata (Hopcroft and Ullmann 1979, Chapter 2), Petri Nets (Peterson 1977), Generalized Semi-Markov Processes (GSMP) (Glynn 1989), and event graphs (Schruben and Yücesan 1993) have been proposed for performance analysis of DEDS. Although closely related, none of these approaches have received the universal acceptance enjoyed by the differential equations framework for CVDS.

For instance, a finite-state automaton can be modified to depict DEDS. To that end, a *state automaton* is defined as a five-tuple (E, X, Γ, f, x_0), where

E is a countable event set,
X is a countable state space,
$\Gamma(x)$ is the set of feasible or enabled events defined for all $x \in X$ and $\Gamma(x) \subseteq E$,
f is a state transition function, a mapping of the form $f: X \times E \rightarrow X$,
x_0 is an initial state, $x_0 \in X$.

This construct is also referred to as a Generalized Semi-Markov Scheme (GSMS) (Glasserman and Yao 1992). A state automaton can be further augmented through the incorporation of a clock structure, which is used to compute the clock sequences for all events. Every event $i \in \Gamma(x)$ has a residual lifetime y_i indicating the amount of time left until its occurrence. For event i, y_i initially assumes a lifetime value determined by an externally provided clock structure v_i. The resulting structure is called a *timed state automaton* and denoted by $(E, X, \Gamma, f, x_0, V)$, where $V = \{v_i : i \in E\}$. This structure can be made stochastic by associating transition probabilities with the state transition function and distribution functions with the clock structure. The resulting *stochastic timed state automaton* can be used to generate a GSMP.

Based on the stochastic timed state automaton framework, a DEDS can be viewed as a dynamic system where the input is a set of event lifetime sequences and the output is a sequence of time-stamped events. For example, the dynamic behavior of a single-server queueing system can be obtained by simply specifying a set of interarrival times and a set of service times. The state transition function of the automaton then generates a sequence of events with their corresponding times of occurrence. Suppose θ is a parameter that may affect the output either through the event lifetimes or through the state transition mechanism. For instance, θ could represent the arrival rate of the customers, or the queue capacity, or the queue discipline.

Adopting the notation of Cassandras (1993a, p. 683), let $\xi(\theta)$ be the output sequence of time-stamped events, $\{X_k(\theta)\}$ be the corresponding sequence of states, and $L(\xi(\theta))$ be a performance measure of interest. Suppose θ can assume values from a finite set $\Theta = \{\theta_1, \theta_2, ..., \theta_m\}$. From an optimization perspective, it may be interesting to select that value of θ that maximizes the expected value of the performance measure, $J(\theta) = E\{L(\xi(\theta))\}$. Alternatively, one might also be interested in conducting sensitivity analysis to determine the impact of different values of θ on the expected value of the performance measure, $J(\theta)$.

In a stochastic environment, these are difficult tasks (Law and Kelton 1991, p. 679). Simulation seems to be the most flexible approach. However, naive simulation, where the system is run under each parameter value, θ_i, $i=0,1,2,...,m$, is extremely time consuming so as to make this approach infeasible for complex models. More sophisticated techniques have been proposed to reduce the computational burden while increasing the reliability of the estimates (Jacobson and Schruben 1989). A recent idea, termed "rapid learning," (Cassandras and Strickland 1989) tries to reconstruct the sample paths (perhaps in parallel) under $\theta_1, \theta_2, ..., \theta_m$, based on a single sample path observed under θ_0. For example, in the analysis of a capacitated single-server queueing system, the nominal path could be observed under a system capacity of 5 and other sample paths can be reconstructed under system capacities of 4, 3, 2, and 1, respectively. While this example emphasizes the applicability of the technique in dealing with discrete parameters such as system capacity or size of buffers, one can in general think of such parameters as a set of discrete control actions to manage a DEDS.

More specifically, the *sample path constructability* problem is defined as follows: using the inputs together with the *nominal* sample path, $\xi(\theta_0)$, construct all other output sequences, $\xi(\theta_1), \xi(\theta_2), ..., \xi(\theta_m)$, under different system parameters, $\theta_1, \theta_2, ..., \theta_m$. In other words, is it possible to construct sample paths $\xi(\theta_1), \xi(\theta_2), ..., \xi(\theta_m)$ from the information contained in the observed (nominal) sample path $\xi(\theta_0)$? This is indeed possible under certain conditions.

3 NECESSARY CONDITIONS

Given the stochastic timed state automaton framework, the system evolves by the occurrence of events. In state x, the set of feasible events, $\Gamma(x)$ is scanned to determine the triggering event, the one with the smallest clock value. The execution of this event may force the system to change state and other clock values to be updated. There are therefore two important issues to verify for sample path constructability: the state structure of the system (or, more specifically, $\Gamma(x)$) and the clock values (or, more specifically, event lifetime distributions).

These issues are summarized under two conditions, *observability* and *constructability* (Cassandras and Strickland 1989). Let $\mathcal{E}(X_k(\theta))$ represent the set of feasible events (the future events list) in state $X_k(\theta)$, $k=0,1,2,...$. The conditions are then defined as follows:

Observability: For $\theta_m \neq \theta_0$, $\xi(\theta_m)$ is observable with respect to $\xi(\theta_0)$, if $\mathcal{E}(X_k(\theta_m)) \subseteq \mathcal{E}(X_k(\theta_0))$ for all $k=0,1,2,...$.

The sample paths are coupled in the sense that the same event sequence drives both of them. The condition asserts that every state observed in the nominal path is always "richer" in terms of feasible events than the states observed in the constructed path. For example, an M/M/1/2 system is observable with respect to an M/M/1/3 system, but not vice versa. To see this, let the state in this queueing system be described by Q, the number of customers in the system. In the M/M/1/3 system, $\mathcal{E}(Q=2)$ = {ARV, DEPT}, representing a new customer arrival and a departure, respectively. Within the M/M/1/2 system, however, $\mathcal{E}(Q=2)$ = {DEPT}, as no new arrivals are admitted into the system. Hence, the former system has a larger number of feasible events in this state.

Observability addresses the structural aspects of determining the triggering event in some state. It does not, however, address the issue of whether the clock values observed on the nominal path can be used on the constructed path. This, in turn, is done by the constructability condition.

Constructability: For $\theta_m \neq \theta_0$, $\xi(\theta_m)$ is constructable with respect to $\xi(\theta_0)$, if

(1) $\mathcal{E}(X_k(\theta_m)) \subseteq \mathcal{E}(X_k(\theta_0))$ for all $k=0,1,2,...,$

and

(2) $H(t, z_{i,k}(\theta_m); \theta_m) = H(t, z_{i,k}(\theta_0); \theta_0)$ for all $i \in \mathcal{E}(X_k(\theta_m))$, $k=0,1,2,...,$
where $H(t, z_{i,k}\theta); \theta)$ denotes the cdf of the clock value of event i, given its age $z_{i,k}(\theta)$.

The first part of the condition is observability. The second part imposes some restrictions on event lifetimes. More specifically, it requires the lifetime (or the clock value) of an event to have the same cdf as its residual lifetime. This is trivially satisfied when the lifetime densities follow the exponential distribution, which has the memoryless property. In this case, observability implies constructability. In general, it is a fairly restrictive requirement. We assert that the verification of the observability condition is a hard problem as well. To this end, we will first introduce the following search problem:

OBSERVABILITY

Instance: A discrete event simulation model specification with an associated model implementation,
a model parameter, θ_0,
a set of event lifetime sequences, and
a nominal sample path with a set of observed states, $X_1(\theta_0)$, $X_2(\theta_0)$, ..., $X_n(\theta_0)$.

Question: Find a parameter value θ_1 such that
$\mathcal{E}(X_k(\theta_1)) \subseteq \mathcal{E}(X_k(\theta_0))$ for all k=0,1,2,... .

In simple terms, the search problem seeks a model parameter $\theta_1 \neq \theta_0$ such that the sample path constructed under this parameter has always fewer feasible events in each state than the states in the nominal sample path. Therefore, $\xi(\theta_1)$ is observable with respect to $\xi(\theta_0)$.

THEOREM: OBSERVABILITY is NP-hard.
Proof: A polynomial Turing reduction is constructed to show that OBSERVABILITY is NP-hard (Yücesan and Jacobson 1995). □

4 CONCLUSIONS

Cassandras (1993a, p. 688) states that "the validity of observability is sometimes easy to check by inspecting the state transition diagrams of the system under θ and θ_m." We have shown that this is in general an NP-hard search problem. Our result encourages the study of special cases and easy instances of this problem and the design of heuristic algorithms for these cases.

One such technique is the standard clock approach (Vakili 1991) applied to birth-and-death processes. Within this class of models, the distributional requirements of constructability are readily satisfied due to the memoryless property of the exponential distribution. Observability, on the other hand, is achieved by forcing all events to be feasible in all states through the uniformization of the underlying process. This is analogous to modeling birth-and-death processes with a single event node in SIGMA (Schruben 1992, p. 219). One should note, however, that a uniformized process might generate a large number of fictitious events, which do not affect the state of the system, before a real event induces a change in the system state. This may compromise the speed of the technique.

The augmented system analysis (ASA) approach (Cassandras and Strickland 1989) is designed to avoid the generation of fictitious events. This method, which applies to Markovian systems, suspends the construction of the additional sample paths as soon as the observability condition is violated. The constructed path remains suspended until the nominal path enters a state in which the condition is satisfied again. The probabilistic structure of the constructed path is not disturbed by the suspension due to the memoryless property. Such an approach is referred to as event matching.

The Markovian requirement is relaxed through an age matching algorithm. In this approach, when the observability condition is violated, the construction of further sample paths are suspended and all events with nonexponential lifetime distributions are saved. When the nominal sample path enters a state where observability is once again satisfied, the construction resumes only if the clock values of the nonexponential events on the nominal path match the saved values on the constructed paths.

ASA seems to apply for discrete parameters that affect the state transition mechanism, but do not alter event times. The major drawback of this approach is that the system may remain suspended for long periods of time. Furthermore, the overhead needed to monitor the constructed paths for event or age matching algorithms may be prohibitive for complex systems.

A special application of ASA to the traffic smoothing problem in high-speed communication networks is discussed in Cassandras (1993b). In this case, ASA not only provides a convenient mechanism to investigate the structural properties of smoothing techniques such as the leaky bucket (LB) scheme (Turner 1986), but also allows the use of these properties in establishing key relationships for the LB scheme.

Adopting the computational complexity perspective enables several seemingly different problems in simulation modeling and analysis to be cast in a single unifying framework (Jacobson and Yücesan 1995). In particular, such problems are equivalent or equally difficult, from the computational complexity point of view. Using this perspective, a consequence of this result presented here is that algorithms that solve the observability problem are likely to be enumerative in nature. Such enumerative algorithms tend to execute in exponential time in the size of the problem instance. This, in turn, supports the development of polynomial-time heuristic procedures as well as the identification of special cases that are polynomially solvable. Further research is in progress to gain new insights from such problems as well as to identify other related problems that may have an impact on the way discrete-event simulation models are constructed and analyzed.

ACKNOWLEDGEMENTS

The research of the second author is supported in part by the National Science Foundation (DMI-9409266) and the Air Force Office of Scientific Research (F49620-95-1-0124).

REFERENCES

Cassandras, C.G. 1993a. *Discrete Event Systems: Modeling and Performance Analysis,* Irwin, Homewood, IL.

Cassandras, C.G. 1993b. Rapid learning techniques for DES: Some recent results and applications to traffic smoothing, *Proceedings of the 12th IFAC World Congress* 3:323-326.

Cassandras, C.G., and S.G. Strickland. 1989. Sample path properties of timed discrete event systems, *Proceedings of the IEEE* 77:59-71.

Garey, M.R., and D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman & Co., New York, NY.

Glasserman, P., and D.D.Yao. 1992. Monotonicity in generalized semi-Markov processes, *Mathematics of Operations Research* 17(1):1-21.

Glynn, P. 1989. A GSMP formalism for discrete event systems, *Proceedings of the IEEE* 77:14-23.

Goldsman, D., and B.L. Nelson. 1994. Ranking, selection, and multiple comparisons in computer simulation, *Proceedings of the 1994 Winter Simulation Conference,* eds. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 192-199.

Healy, K., and L. Schruben. 1991. Retrospective simulation response optimization, *Proceedings of the 1991 Winter Simulation Conference,* eds. B. L. Nelson, W. D. Kelton, and G. M. Clark, 901-906.

Heidelberger, P. 1988. Discrete event simulation and parallel processing: statistical properties, *SIAM Journal of Statistical Computing* 9.6:1114-1132.

Ho, Y.C., M.A. Eyler, and D.T. Chien. 1979. A gradient technique for general buffer storage design in a serial production line, *International Journal of Production Research* 17:557-580.

Hopcroft, J.E., and J.D. Ullmann. 1979. *Introduction to Automata Theory, Languages, and Computation,* Addison Wesley, Reading, MA.

Jacobson, S.H., and L. Schruben. 1989. Techniques for simulation response optimization, *Operations Research Letters* 8:1-9.

Jacobson, S.H., and E. Yücesan. 1995. Intractability results in discrete event simulation, *Recherche Operationnelle* 29.3:1-17.

Kleijnen, J.P.C. 1987. *Statistical Tools for Simulation Practitioners,* Dekker, New York, NY.

Law, A.M., and W.D. Kelton. 1991. *Simulation Modeling and Analysis,* 2nd edition, McGraw-Hill, New York, NY.

Peterson, J. L. 1977. Petri nets, *Computing Surveys* 9.3, 223-252.

Schruben, L. 1992. SIGMA: *A Graphical Simulation Modeling Program,* The Scientific Press, San Fransisco, CA.

Schruben, L., and V.J. Cogliano. 1987. An experimental procedure for simulation response surface model identification, *Communications of the ACM* 30.8:716-730.

Schruben, L., and E. Yücesan. 1993. Modeling paradigms for discrete event simulation, *Operations Research Letters* 13:265-275.

Turner, J.S. 1986. New directions in communications (or which way to the information age?), *IEEE Communications Magazine* 24.10:8-15.

Vakili, P. 1991. A standard clock technique for efficient simulation, *Operations Research Letters* 10:445-452.

Yücesan, E. 1994. Comparing alternative system configurations using simulation: a non-parametric approach, *Annals of Operations Research* 53:471-484.

Yücesan, E., and S.H. Jacobson. 1995. On the parallel generation of sample paths, Working Paper, INSEAD. Fontainebleau, France.

AUTHOR BIOGRAPHIES

ENVER YÜCESAN is an Associate Professor of Operations Research at the European Institute of Business Administration, INSEAD, in Fontainebleau, France. He holds an undergraduate degree in Industrial Engineering from Purdue University, and a M.S. and a Ph.D. in Operations Research from Cornell University.

SHELDON H. JACOBSON is an Assistant Professor in the Department of Industrial and Systems Engineering at Virginia Polytechnic Institute and State University (Virginia Tech). Before joining Virginia Tech, he served for five years on the faculty in the Department of Operations Research at Case Western Reserve University. He has a B.Sc. and M.Sc. in Mathematics from McGill University, and a Ph.D. in Operations Research from Cornell University. He has served as the Advanced Tutorial Track Coordinator at both the 1994 and the 1995 Winter Simulation Conferences. He also served as the Doctoral Colloquium Coordinator at both the 1993 and 1994 Winter Simulation Conferences. At present, he is the Treasurer for the INFORMS College on Simulation. His research interests include simulation optimization and sensitivity analysis, frequency domain approaches to analyzing simulation outputs, and issues related to the complexity of analyzing structural properties of discrete event simulation models.