Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

APPLYING STATISTICAL CONTROL TECHNIQUES TO AIR TRAFFIC SIMULATIONS

Kirk C. Benson
David Goldsman
Amy R. Pritchett

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205, U.S.A.

ABSTRACT

While the literature contains several adaptive sampling techniques for statistical comparison of competing simulated system configurations and for embedded statistical computations during simulation run-time, these techniques are often difficult to apply to air traffic simulations because of the complexity of air traffic scenarios and because of the variety of model and data types needed to fully describe air traffic. Adaptive sampling techniques can be beneficial to the study of air traffic; for example, adaptive techniques can use ranking and selection methods to compare the relative worth of the competing configurations and calculate the number of observations required for rigorous statistical comparison, often dramatically reducing the run-time duration of simulations. In this paper, we will describe the implementation of such procedures in the Reconfigurable Flight Simulator for air traffic simulations. We also discuss implications for the coordination of simulation, analysis, and design activities.

1 INTRODUCTION

Increasing use of simulation for both design and analysis motivates models capable of increasingly realistic representations of air traffic systems. System performance can be inferred from simulation output in a variety of manners from simple queuing times to multifaceted compliance with regulations to continuous measures such as flight time, cost, and fuel burn.

Hybrid simulations, that is, simulations capable of simultaneously including discrete-event and continuous-time models, allow for cost-effective and detailed analysis of complex interactions between heterogeneous entities. Agent-based modeling is one method for describing such heterogeneous entities. Under this paradigm, each individual agent autonomously pursues a goal and also interacts with other agents inside the simulation. Agent-based modeling provides an inherently modular method for high-fidelity simulation of air traffic systems. This approach, however, requires the inclusion of a range of models with varied output data types such as discrete and continuous. For example, an appropriate discrete state variable may be the number of aircraft arrivals into a defined airspace, while a continuous variable of interest might be the minimum separation between two aircraft.

Detailed hybrid simulations, including agent-based simulations, require an increase in both size and run-time. Frequently, the amount of simulation output is determined by the availability of computational capacity. Subsequent data analysis, commonly done as a separate activity, often reveals either insufficient or excess (wasted) observations for the required statistical comparison. Therefore, embedding statistical estimators within a simulation can ensure computationally efficient sampling without requiring storage and post hoc analysis.

Incorporating an adaptive control technique, such as a Ranking and Selection (RS) method, offers an additional avenue for increased computational efficiency. RS methods calculate the number of required observations while ensuring statistically sound comparisons are made with modest computational expense. The methods discussed here are sequential and appropriate for general stationary output processes. A new adaptive control technique is used here that relies on embedded statistical estimators to calculate the number of required observations for each simulated configuration. Additionally, the control technique differentiates in an adaptive manner between competing simulated configurations by identifying which configurations do not warrant further analysis, potentially saving computational resources.

Bringing together hybrid simulation models, embedded statistical analysis, and adaptive control techniques improves the application of simulation to the analysis and design of air traffic systems. This improvement is realized in terms of computational reduction and statistically valid comparison of competing system configurations. Additionally, this method creates an environment conducive to Parallel and Distributed Simulation (PDS), although the control techniques employed here do not require the strict time management generally required in PDS. Instead, experiments can be implemented on a Network of Workstations (NOW) that coordinates observation sampling from complementary simulations.

As part of this research, the Reconfigurable Flight Simulator (RFS), a large-scale hybrid simulation, has been extended for embedded statistical computations and adaptive control techniques. The analysis of arrival routing configurations for Atlanta Hartsfield-Jackson International Airport (ATL) is presented as a demonstration.

The remainder of this paper is organized as follows. In section 2 we discuss current adaptive control techniques. Section 3 details embedded statistical analysis methods. Section 4 highlights PDS endeavors along with a distributed computing method. Section 5 presents a general discussion on their application to air traffic simulation. Preliminary results are shown in section 6. Section 7 concludes this work with a summary and discussion of future efforts.

2 ADAPTIVE CONTROL TECHNIQUES

The goal of any selection, screening, and multiple comparison problem is to determine the “best” of several competing configurations. In this context, a configuration implies that we have two or more competing systems that are compared by the mean value of some metric describing performance, where simulation is required to assess the value of this metric. Bechhofer et al. (1995) highlight several problem formulations appropriate to various experimental designs. Here, the focus is on the indifference-zone formulation where the objective is to select the system configuration with the highest/lowest (interpreted as “best”) expected value. In this realm, an expectation offers insight on long-term performance while enabling statistical rigor of the comparative method.

The experimenter provides (δ^*, P^*), where δ^* is the indifference-zone size and P^* denotes the threshold for the desired probability of correctly identifying a difference between system configurations. Note that the indifference-zone indicates some comparative region where the experimenter would not discriminate between competing system configurations. Also, the threshold probability, P^*, can be interpreted as a confidence level when configuration mean values do in fact differ by at least δ^*.

RS methods enable adaptive control of this multiple comparison problem. Ultimately, RS methods determine the number of required observations necessary for statistically rigorous comparison of competing simulated configurations. RS methods may be single or multistage. In this context, a stage denotes the execution of a simulated configuration for a specified number of observations. A single-stage method determines the number of required observations from parameters determined by the experimenter. Adaptive control is not possible with a single-stage RS method. However, a multistage RS method updates the required number of observations from simulated configuration output, thereby enabling adaptive control of the comparison process.

If the variance of a predetermined metric is unknown, then Rinott’s method (1978) provides a well known two-stage technique for comparing configurations. This method relies on the assumptions that obtained data are independent, identically distributed, and from a normal distribution. Goldsman et al. (2002) present an extended version of this procedure (*R*+) and the extended version of a procedure due to Kim and Nelson (*KN*+) (2001). *KN*+ is a multistage RS method relying on the same assumptions as *R*+. Note that batch means, i.e., sample means from contiguous batches of observations, are considered to be normally distributed under certain conditions.

Benson (2004) details a multistage extension, Benson/Goldsman/Pritchett 4 (*BGP4*), of the *KN*+ method incorporating embedded statistical analysis. Detailing this method is beyond the scope of this paper. However, the *BGP4* RS method has been shown to achieve the desired probability of correct selection with moderately to highly correlated data.

3 EMBEDDED STATISTICAL ANALYSIS

The methods discussed in section 2 require calculations on both individual and batched observation data. Benson (2004) describes an architecture that does not require use of historical experimental observation values, but instead maintains current state variables and certain summed values. This architecture allows for both estimator calculations and availability of these estimators at each simulation time step along with inherent reduction of memory usage. These time steps can be set to those required by the simulation of each particular configuration.

The computational overhead from using embedded estimators of this sort was assessed by running the same simulated configuration without embedded statistical analysis (NOSTAT), with embedded statistical analysis (STAT), and lastly with both the embedded statistical analysis and the distributed simulation client module (RFS Client) discussed in the following section. Figure 1 highlights the overall results. Addition of embedded statistical analysis increased the computational expense of obtaining a specified number of observations by less than 1% in this example. Here, the RFS client module increased the overall expense by less than 2% for the same number of observations. In practical terms, arrivals for an operational day at Atlanta Hartsfield-Jackson International Airport can be simulated on a single dual processor 2.2 GHz workstation with 512 megabytes of RAM in approximately 160 computer-minutes. An additional 3 minutes of workstation time allows for embedded statistical analysis in this example. Note the computational expense of embedded statistical analysis is inversely proportional to the expense of running the simulation. The impact of embedding statistical analysis is small when the computational requirements of the simulated configuration are large and vice versa.

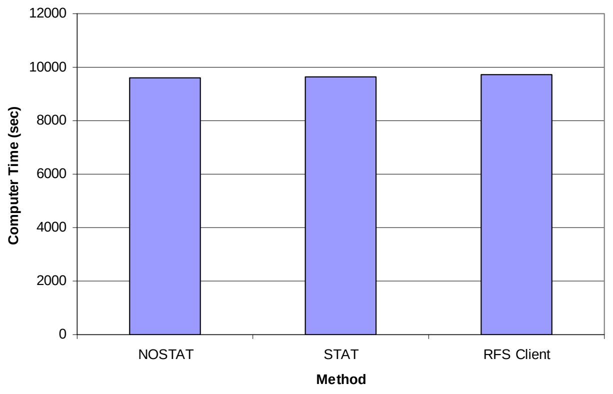

Figure 1: PDS Architecture

In order to analyze these results, users of an existing simulation generally record all data from a simulation for follow-on analysis. The computational cost of this output and storage is not analyzed here. Presumably, however, it can represent both an increase in runtime and a subsequent analysis process that is necessary only in the "NOSTAT" condition.

Embedding this architecture within a steady-state simulation whose output is neither independent nor identically distributed predicates assumptions on simulation initialization. Law and Kelton (2000) suggest truncating early data in a simulation as one method of avoiding initialization bias. Given $X_{i1}, X_{i2}, \ldots$ as the simulation output from a single replication of the ith alternative, then after appropriate initialization the following assumptions hold:

Stationarity: $X_{i1}, X_{i2}, \ldots$ forms a stationary stochastic process.

(Strong) Consistency: $\overline{X}_i(r) \to \mu_i$ as $r \to \infty$ with probability 1, where μ_i is the steady-state mean from system i and $\overline{X}_i(r)$ is the sample mean based on r observations from system i.

Functional Central Limit Theorem (FCLT): There exist constants μ_i and $v_i^2 > 0$ such that

$$\frac{\sum_{j=1}^{\lfloor rt \rfloor}\left(X_{ij} - \mu_i\right)}{\sqrt{r}} \Rightarrow v_i W(t) \qquad (1)$$

for $0 \le t \le 1$, where $\Rightarrow$ denotes weak convergence as $r \to \infty$ and *W(t)* is a standard Brownian motion (Weiner) process (see Billingsley 1968).

For the current article, comparisons will be made on steady-state means $\mu_1, \mu_2, \ldots, \mu_k$, which is reasonable due to the consistency assumption. The variance parameter, v_i^2, will be estimated by batch means. Note that variance estimation from a single long simulation run alleviates the issue of initialization bias. The following batch means technique provides an asymptotic estimator for the variance constant $v_i^2 \equiv \lim_{r\to\infty} r\mathrm{Var}\left(\overline{X}_i(r)\right)$.

If n observations $X_{i1}, X_{i2}, \ldots, X_{in}$ are divided into b batches of length m, then the jth batch mean from system i is:

$$\overline{X}_{i,j,m} \equiv \frac{1}{m}\sum_{p=1}^{m} X_{i,(j-1)m+p} \qquad (2)$$

The observations $X_{i,(j-1)m+1}, X_{i,(j-1)m+2}, \ldots, X_{i,jm}$ comprise the jth batch, $j = 1,2,\ldots,b$, for system i. For $b > 1$, the batch means variance estimator is:

$$mV_B^2 \equiv \frac{m}{b-1}\sum_{j=1}^{b}\left(\overline{X}_{i,j,m} - \overline{X}_i(n)\right)^2 \xrightarrow{D} \frac{v_i^2 \chi_{b-1}^2}{b-1} \qquad (3)$$

where χ_d^2 is a chi-squared random variable with $d = b-1$ degrees of freedom and $\xrightarrow{D}$ indicates convergence in distribution.

4 DISTRIBUTED COMPUTING METHOD

Parallel and Distributed Simulation (PDS) has been studied for many years in an effort to speed increasingly complex simulation models. PDS research has primarily focused on manipulating sequential simulations and unifying coupled simulation processes for discrete-event systems. Manipulation of sequential simulations is generally accepted for queuing systems or for the mass replication of a particular simulation configuration. Unification of separate but related simulations spans sophisticated synchronization algorithms and prescribed interfaces such as the DOD High Level Architecture (HLA). Fujimoto (2000) provides a comprehensive discussion of current PDS techniques.

The PDS architecture used here allows for heterogeneous processor contribution of observations for a given experiment. Specifically, contributing processors simulate different system configurations. Unlike temporal and spatial decomposition PDS methods that contain coupled dynamics, this approach incorporates comparatively independent execution of the simulations. The lack of coupled dynamics by this approach avoids causal synchronization issues. Beyond mere mass replication of a particular simulation, this method naturally acquires computational capacity for configuration comparison. Computational load sharing in this manner is related to previous efforts by Karatza and Hilzer (2002).

This PDS architecture can be implemented on a Network of Workstations (NOW). Simulation jobs need to be assigned by a central server, or controller. Jobs can be different system configurations or the replication of the same configuration. The controller coordinates NOW usage by using ranking and selection methods to determine which configurations, and how many observations are required from each, to distribute out to participating workstations.

Defining a job as a requirement for a specific number of simulated observations and a machine as a workstation highlights the scheduling problem inherent to this distributed simulation architecture. Typical scheduling problems are NP-hard (Hopp and Spearman, 2000). Assuming simulated configurations are similar, acquisition of first-stage observations requires approximately the same time when using homogeneous processors on the contributing workstations. However, heterogeneous workstation use and latter-stage observation requirements obtained from ranking and selection methods complicate the estimation of job duration.

With this distributed simulation architecture, job requirements can be dynamically resized using RS methods. The differing observational requirements, or job size, dramatically increase the difficulty of efficient job queuing. However, the decreased computational expense achieved through the deletion of jobs, i.e., simulated configurations that are no longer competitive, offers increased computational efficiency.

This distributed simulation architecture allows for job allocation in several manners. If the experimenter lacks knowledge of simulator job duration and believes combining simulated configuration output is inappropriate, then job allocation should be sequential. For example, if there are six configurations and three workstations then workstation one receives job A, workstation two job B, etc. If combining simulated configuration output is considered appropriate then all configurations can be distributed to each workstation. Note that combining simulation output relies on assumptions of sufficient and accurate variance estimation along with the integrity of combining such output.

The specific client-server architecture is shown in Figure 2. The server acts as the controller. Competing configurations such as systems A, B, and C are run on participating workstations providing observations using a simulation executable program that allows for external control in terms of run, pause, and terminate commands. Additionally, the simulation executable program is required to make embedded statistical calculations.

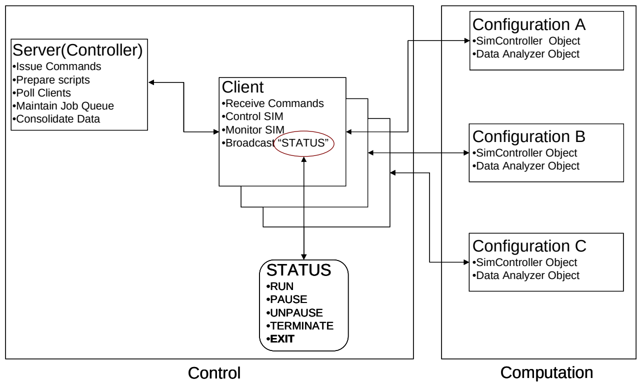

Figure 2: PDS Architecture

Participating workstations function as individual clients. Clients act as an interface between the server and the simulation executable program. Each client manages one or more simulation executable program and monitors the associated simulation status. It is assumed that more than one instantiation of the simulation executable program may execute on a participating workstation. Additionally, the client prepares simulation output for the server. For this implementation, if a client is tasked with more than one job then it must be differentiable. Also, job execution is sequential when more than one is assigned.

The server interacts with the clients as it compares different configurations' metrics in a statistical sense at appropriate intervals. Beyond monitoring status, the server also consolidates simulation output required for ranking and selection methods. Because these comparative intervals can be much greater than time steps within the simulations and because the comparisons are only used to start and end simulation runs, the distributed simulations are much more "loosely" coupled than in most PDS implementations. Therefore, strict time synchronization is not required.

The server uses RS methods to calculate the number of required observations and also to discriminate between competing system configurations. Here, if one particular configuration is deemed unworthy of further analysis, due to poor performance, then it is eliminated from further computational analysis. The server also maintains the status of participating clients, and manages job allocation.

Communication between the client and the existing simulation is accomplished through the use of text scripts. The server and clients communicate by the use of operating system managed TCP/IP Ethernet. This generalized approach is extensible. Additionally, it is easily reconfigured for varying experimental designs.

This distributed computing method can provide near linear computational performance increases for a small number of contributing workstations as shown by Benson (2004).

5 AIR TRAFFIC SIMULATION

Air Traffic Control (ATC) systems ensure the safe travel of an aircraft from one airport to another and Air Traffic Management (ATM) systems schedule and sequence aircraft to increase throughput and reduce delay. This section provides a general description of ATC and ATM simulations.

ATM simulation can determine the impact of flight restrictions on delay, throughput, and traffic congestion. Wieland (1998) describes the Detailed Policy Assessment Tool (DPAT) as a large-scale simulation capable of calculating traffic conditions for entire airspace regions, for example the continental United States. DPAT models the National Airspace System (NAS) as a sequence of capacitated resources in a parallel and discrete-event manner. Much of the data used within DPAT is obtained from external models. DPAT has successfully simulated NAS operations for the entire continental United States in faster than real time for specific models.

The Total Airspace and Airport Model (TAAM) simulation is a high-fidelity simulation modeling NAS components such as gates, terminals, taxiways, and airspace. Holden and Wieland (2003) incorporated simulation optimization methods with TAAM to optimize runway scheduling. For this particular analysis, the scheduling impact of adding a new runway was simulated. Potentially, this method could assist controllers with the allocation of aircraft to runways.

ATM simulation can also provide predictive insight on the impact of new equipment on airport throughput. For example, Schwartz et al. (1997) describe the use of simulation to evaluate the introduction of new Flight Management System (FMS) equipment in aircraft cockpits along with new routing procedures. They assumed that more sophisticated (that is, higher cost) FMS equipment corresponded to decreased controller-pilot verbal communication. Then, they simulated various combinations of traffic throughput and percentage of FMS equipped aircraft. Note that the capability of installed FMS equipment also varied in terms of acquisition cost. This method of sensitivity analysis provided insight that capacity could be increased by equipment fielding; and it also offered a cost-benefit element for determining the required sophistication in new FMS equipment.

Simulation of aircraft routing procedures has also been pursued as a method to increase capacity. Tofukuji (1993) provides an example in which various routing configurations were simulated to assess throughput. Results from this experiment included a relationship between throughput and required controller interventions. Additionally, this experiment compared existing route configurations along with proposed modifications.

The competing tradeoff between increased capacity and improved safety has also been investigated through the use of simulation. Zeghal and Hoffman (2000) explored model performance of ATC operations where the requirement of maintaining separation was delegated to individual aircraft. Here the sequencing of self-separating aircraft was simulated to predict future capacity and controller workload. Safety, in this case violation of a minimum separation threshold, was indirectly assessed using rules for sequencing aircraft that ensured safe separation.

Combining discrete-event and continuous-time models enables cost-effective and detailed analysis of complex systems such as ATC and ATM. Agent-based modeling is one method for describing the heterogeneous entities comprising these hybrid models. In contrast to many ATC and ATM simulations that often look exclusively at factors such as capacity or safety, hybrid models allow for analysis of both discrete and continuous-valued variables. Agent-based modeling provides an inherently modular paradigm for high-fidelity simulation.

Modeled ATC and ATM systems have been simulated in an effort to obtain predictive measures of performance by numerous agencies with varying fidelity. Common to all efforts is the need for metric assessment and computational efficiency. Application of adaptive control techniques within a distributed simulation architecture not only reduces the computational requirement but speeds experimental execution. Versatile, embedded data encapsulation methods enable these control techniques.

6 TEST CASE: ANALYSIS OF ARRIVALS TO ATLANTA

The Reconfigurable Flight Simulator (RFS) is a large-scale agent-based simulation. RFS is used as a test case for several reasons. First, it is hybrid in nature and models a complex system that cannot be simplified for an analytic solution without loss of fidelity. Second, it is a significant development effort in terms of personnel-hours as well as high-level software engineering. Minor modifications within the RFS software architecture enabled the application of adaptive control techniques. Also, as the name implies, RFS is easily initialized for alternative configurations of the NAS by the use of script files. Lastly, RFS supports analysis of both discrete and continuous state variables.

Pritchett and Ippolito (2000) discuss the Object-Oriented (OO) structure and capabilities of the RFS. Also, Lee, Pritchett, and Goldsman (2001) detail the RFS timing mechanisms and their application to a hybrid, agent-based simulation of air traffic. The OO structure of the RFS is extensible and modular. Instantiation of the base classes produces objects that compose the simulation; these objects can be configured by a script file during initialization. In this context, an object is also considered an agent if it can autonomously interact with other agents while pursuing a particular goal or set of goals. Note that each agent is also self-describing in terms of identity, performance parameters, and current state. Here, the combined agent behavior models complex system performance. Other objects in the simulation may not have two-way interactions with the agents, but instead serve other purposes such as graphic displays, date loggers, and analyzers.

An air traffic scenario with varied configurations provides an interesting large-scale simulation as a test case for the application of adaptive control and distributed simulation techniques. Specifically, different arrival routing density configurations for the Atlanta Hartsfield-Jackson International Airport (ATL) Macey-Two Standard Terminal Arrival Route (STAR) are compared. Figure 3 specifies the waypoints to be followed in the Macey-Two STAR.

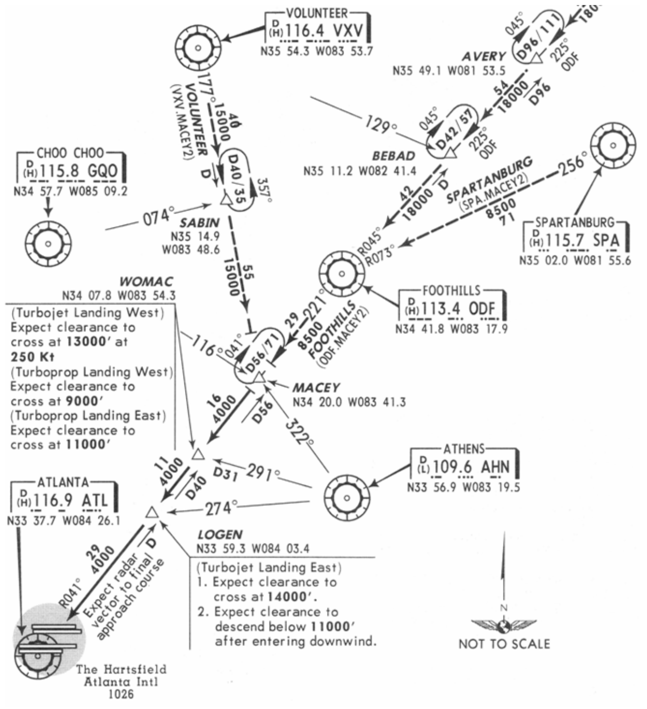

Figure 3: Atlanta Hartsfield-Jackson International Airport Macey-Two STAR

Of interest, the intersection at MACEY involves the incorporation or merging of traffic from the navigation aids Volunteer (VXV) and Spartanburg (SPA) and the “fix” AVERY. Arriving aircraft are assigned to one of these three entry points by an air traffic manager much earlier in the flight depending upon their location and expected aircraft density. Once on the route, an air traffic controller maintains spacing between the aircraft.

RFS waypoint following aircraft (WPT) agents model arriving aircraft for this scenario. Each WPT agent uses numerical integration routines to update continuously-varying state variables including speed, heading, latitude, longitude, and altitude. The trajectory of WPT agents is defined by a list of waypoints initialized at instantiation. In this test case, WPT agents adjust their internal dynamics to cross each waypoint at a specified speed.

WPT agents are instantiated by the RFS Random Plane Generator (RPG) agent with initial performance parameters. The RPG agent creates WPT agents as discrete events based on a random stationary Poisson process. The inter-arrival time for this Poisson process is set at RFS initialization. While actual arrivals to ATL are more closely modeled by a non-homogeneous Poisson process, this simplification still allows for relevant system analysis.

The RFS ATC agent provides a basic model of the air traffic controller. The ATC agent monitors waypoint following aircraft agents to ensure safe separation. The ATC agent maintains a list of WPT agents within a defined sector and provides calculated along route speed commands to the WPT agents. The ATC agent also determines WPT agent sequencing in merging arrival streams. Additionally, the ATC agent can model missed communication, communication delay, and misinterpreted command behavior (Lee, 2002).

Currently, approximately 615 aircraft arrive daily at ATL. The majority of these aircraft arrive between 6 am and 12 pm. This equates to an approximate 100 second inter-arrival time between aircraft although this can be much higher during banks of arriving aircraft. Varying the allocation of aircraft on the three merging arrival streams of the Macey-Two STAR approach defines comparable configurations for this test case.

Table 1 presents the three route density configurations under analysis. Configuration C1 is the base case with equal 300 second expected inter-arrival times on each of the three arrival streams for a system-wide expected inter-arrival time of 100 seconds. Configuration C2 involves a higher arrival density, i.e., a lower inter-arrival time, on the northern stream, resulting in a system-wide expected inter-arrival time of approximately 83 seconds. Lastly, configuration C3 involves higher arrival densities on the two southern streams with the same system-wide inter-arrival time as configuration C2.

Table 1: Test Case Configurations

Expected Inter-Arrival Time (sec)	
Configuration	**Description**
C1	VXV – 300 AVERY – 300 SPA – 300
C2	VXV – 150 AVERY – 375 SPA – 375
C3	VXV – 500 AVERY – 200 SPA – 200

Varying route densities in this manner addresses questions about efficient and safe allocation of aircraft to arrival streams. One potential safety metric is the average minimum separation distance between aircraft. A larger value for this metric is considered safer. Without modeling all the factors contributing to a near-miss or aircraft collision (NMAC) event, low average minimum aircraft separation provides good initial insight into potentially problematic system configurations. It can be inferred that a smaller average minimum separation implies a reduction in allowable reaction time from both pilots and controllers. The indifference-zone parameter, δ^*, is set to 1500 feet for this metric. This equates approximately to a six-second reaction time differential for pilots and controllers.

Several diagnostic tests are required before applying ranking and selection methods to these simulated configurations. First, the simulation, in this test case the Reconfigurable Flight Simulator, must be validated as adequately mimicking the real-world system. In this case, validation was performed subjectively; extensive validations are often conducted for such simulations, but are beyond the scope of this study.

The Reconfigurable Flight Simulator was set to mimic aircraft arrivals into ATL. Initial random number seeds were varied in an incremental manner controlled by the server module. Diagnostic tests allowed for the estimation of an appropriate simulation initialization period to avoid bias. A 30 simulation-minute warm-up period was deemed sufficient as aircraft had been generated on all arrival streams along with at least one arrival into ATL.

Next, a sampling rate must be found that provides observations with acceptable serial correlation. The performance of ranking and selection methods, such as *BGP4*, determines the level of acceptable correlation. Benson (2004) showed *BGP4* achieved the probability requirement with highly correlated unbatched observations, i.e., a correlation coefficient value of 0.95, when the means of selected performance metrics from competing simulated configurations are assumed to be Equally Spaced (ES).

For the k competing system configurations in this test case, it is assumed the ordered means of minimum average separation, $\mu_1, \mu_2, \ldots, \mu_k$, are equally spaced by a minimum of the indifference-zone factor, δ^*. Specifically, the minimal spacing of ordered configuration means is $\mu_1 = \delta^*, \ldots, \mu_{k-1} = (k-1)\delta^*$ and $\mu_k = k\delta^*$. This approach is conservative because it assumes simulated configurations are differentiated by at least some significant factor.

Arrival data, such as average minimum separation in this test case, is highly correlated due to its dependent nature. Varying the observation sampling rate within RFS from 30 to 120 simulation-seconds for raw or unbatched observations resulted in correlation coefficients decreasing from 0.95 to 0.80 respectively. Here, a larger sampling rate results in slower observation acquisition. Note the increased computational requirement for obtaining decreased unbatched observation correlation. Given the ES assumption, a 30 second sampling rate is appropriate for this test case given the performance of *BGP4*.

The batch size must be sufficiently large to ensure batched observations fit any normal distribution requirements of the ranking and selection method. Assuming a 30 second sampling rate is appropriate, the next simulation diagnostic involves determining the batch size for the Batch Means (BM) method. A sufficiently large batch size results in normally distributed batched observations that are approximately uncorrelated. The key tradeoff in selecting a batch size is the computational expense of acquiring batched observations versus the relative independence and normality of the batched observations.

A sampling rate of 30 seconds with a batch size of 100 obtains one batched mean observation for each 3000 simulation-seconds. With a dual 2.2 GHz Intel processor workstation, this equates to approximately 180 seconds of computer time. Normality tests indicated there is no evidence the batched observations are non-normal. For this reason, a batch size of 100 was selected for test case analysis.

Given unlimited computational capacity, the designer could execute independent simulation replications until able to make statistically valid conclusions. However, the goal of adaptive control techniques, such as *BGP4*, is to reduce the computational requirements. A single experiment using the *BGP4* RS method provides statistical selection of the "best" or "worst" configuration. Multiple experiments can subsequently strengthen the statistical argument if desired.

Table 2 presents the results of the application of *BGP4* to identify the "worst", i.e., the lowest average minimum separation, arrival route density allocation. Twenty-one experimental replications provided estimators of average minimum separation, standard error, and the average required number of unbatched observations, $\hat{\bar{T}}$. Typical experiment duration was four hours using one dual 2.2 GHz Intel processor workstation as the controller with three similar workstations contributing computational capacity.

Table2: Worst Case Arrival Routing Comparison

Average Minimum Separation			
	Configuration		
	C1	**C2**	**C3**
Mean (feet)	57569	52174	**49706**
S.E.	232	148	155
$\hat{\bar{T}}$	2700	5600	5600

Configuration C3, i.e., the highest route densities on the two southern paths in the Macey-Two STAR, is the "worst" performing simulated configuration. For the twenty-one experiments, the probability of selecting configuration C3 as the "worst" equaled 1.00. Also, the standard error indicates the competing configurations are statistically differentiable in post hoc analysis. Observe the early elimination of configuration C1 from further comparative analysis shown by the low average number of unbatched observations. Here, the computational savings from *BGP4* is exhibited. Configurations C2 and C3, the last two competing configurations under analysis, terminate at the same number of unbatched observations when one of them is selected as the "worst".

7 CONCLUSIONS AND FUTURE EFFORTS

This effort implements a combination of several techniques to enable analysis of complex, large-scale systems. Hybrid simulation, including agent-based simulation, provides the representation of system behavior. Embedded statistical analysis along with ranking and selection methods facilitate adaptive simulation control. PDS implementation enables relatively "fast" execution of otherwise computationally expensive simulations.

Relatively small modifications of current simulation architectures can allow for extended analysis without extensive recoding. One modification involves embedding statistical analysis along with a method for measuring relevant performance metrics within the simulation. Modifying the simulation architecture to allow for external control of the simulation executable in terms of run, pause, and terminate enables the application of PDS techniques.

RS methods ease the computational expense of comparative analysis by determining the number of required observations necessary for statistical analysis. A multistage RS method, such as *BGP4*, saves computational resources by eliminating simulated configurations from further contention for selection that are "poor" performers. For the test case presented here, the elimination of configuration C1 saved the computational resources necessary to obtain an estimated 2900 raw observations.

The use of existing computer systems (e.g., NOW) can provide computational resources to ease comparative analysis. Note the overhead of embedding statistical analysis is inversely proportional to the complexity of the simulation. Additional workstations provided near linear performance increases for a small NOW in this effort.

On-going and future efforts involve several pursuits. Exploration on the potential application of this method to other existing simulation domains is of particular interest. RS method incorporation of other test statistics may offer increased performance in certain conditions. Increasing the number of workstations contributing computational capacity to an experiment could potentially speed execution. Additionally, the incorporation of "job shop" algorithms may provide increased job allocation efficiency.

ACKNOWLEDGMENTS

This work was partly funded by the NASA Ames Research Center, under Grant NAG-2-1291, with Irv Statler and Mary Conners administering and serving as technical points of contact. The authors would also to thank those who have contributed to the development of the simulation, including Satchidanand Kalaver and Seungman Lee.

REFERENCES

Bechhofer, R. E., T. J. Santner, and D. M. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. New York: John Wiley & Sons, Inc.

Benson, K. C. 2004. *Adaptive Control of Large-Scale Simulations*. Ph.D. Dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia.

Billingsley, P. 1968. *Convergence of Probability Measures*, New York: John Wiley & Sons, Inc.

Fujimoto, R. M. 2000. *Parallel and Distributed Simulation Systems*. New York: John Wiley & Sons, Inc.

Goldsman, D., S.-H. Kim, W. S. Marshall, and B. L. Nelson. 2002. Ranking and selection for steady-state simulation: Procedures and perspectives. *INFORMS Journal on Computing* 14 (1):2-19.

Holden, T. C. and F. Wieland. 2003. Runway schedule determination by simulation optimization. In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, 1670-1676, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Hopp, W. J. and M. I. Spearman. 2000. *Factory Physics.* 2nd Ed. New York: McGraw-Hill.

Karatza, H. D. and R. C. Hilzer. 2002. Load sharing in heterogeneous distributed systems. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C. H. Cehn, J. L. Snowdon, and J. M. Charnes, 489-496, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available from <www.informs-cs.org> [full web address is <www.informs-cs.org/wsc02papers/061.PDF>; accessed June 23, 2003].

Kim, S. and B. Nelson. 2001. A fully sequential procedure for indifference-zone selection in simulation*. ACM TOMACS*, 11:251-273.

Law, A. M. and W. D. Kelton. 2000*. Simulation Modeling and Analysis,* 3rd. Boston: McGraw-Hill.

Lee, S. M. 2002. *Agent-Based Simulation of Socio-Technical Systems: Software Architecture and Timing Mechanisms.* Ph.D. Dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia.

Lee, S. M., A. R. Pritchett, and D. Goldsman. 2001. Hybrid agent-based simulation for analyzing the national airspace system. In *Proceedings of the 2001 Winter*

Simulation Conference, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 1029-1036, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available from <www.informs-cs.org> [full web address is <www.informs-cs.org/wsc01papers/138.PDF>; accessed June 23, 2003].

Pritchett, A. R. and C. Ippolito. 2000. Software architecture for a reconfigurable flight simulator. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference,* Denver, CO.

Rinott, Y. 1978. On two-stage selection procedures and related probability-inequalities. *Commun. Stat. – Theory and Methods* A8, 799-811.

Schwartz, J., A. Mundra, J. Broderick, and R. Nash. 1997. Some ATC implications of introducing flight management system based routes in the terminal airspace. *16th Digital Avionics Systems Conference*, AIAA/IEEE, 2:9.1-16-9.1-23, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Tofukuji, N. 1993. Engineering of complex systems with models. *IEEE Transactions on Aerospace and Electronic Systems,* 33(2):667-685.

Wieland, F. 1998. Parallel simulation for aviation applications. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Mediros, E. F. Watson, J. S. Carson, and M. S. Manivannan, 1191-1199, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Zeghal, K. and E. Hoffman. 2000. Design of cockpit displays for limited delegation of separation assurance. *18th Digital Avionics Systems Conference*, AIAA/IEEE, D:2-1-D2-8, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

KIRK C. BENSON recently received his Ph.D. from the School of Industrial and Systems Engineering at the Georgia Institute of Technology. He received a B.S. from the United States Military Academy and an M.S. degree from the Naval Postgraduate School. Currently, he is a lieutenant-colonel in the U.S. Army. His research interests include simulation, statistical selection techniques, and distributed computing methods. His email address is <kirk@bensco.com>.

DAVID GOLDSMAN is a Professor in the School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research interests include simulation output analysis and ranking and selection. He also studies applications arising in the healthcare field. Dave has been an active participant in the Winter Simulation Conference — he is currently on the Board of Directors, and was the 1995 Program Chair and 1992 Associate *Proceedings* Editor. His email and web addresses are <sman@isye.gatech.edu> and <www.isye.gatech.edu/~sman>.

AMY R. PRITCHETT is an Assistant Professor in the Schools of Industrial and Systems Engineering and Aerospace Engineering at the Georgia Institute of Technology. She is an area editor of *SIMULATION* for the areas of air traffic and aviation. She received S.B., S.M., and Sci.D. degrees from the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology. Her research specialties include cockpit design, air traffic control, flight simulation, and large-scale agent-based simulation of hybrid systems. Her email and web addresses are <amy.pritchett@isye.gatech.edu> and <www.isye.gatech.edu/people/faculty/Amy_Pritchett/>.