Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

DIMENSIONALITY ANALYSIS OF A SIMULATION OUTCOME SPACE

John B. Gilmer, Jr.
Frederick J. Sullivan

Wilkes University
Post Office Box 111
Wilkes-Barre, PA 18766, U.S.A.

ABSTRACT

This paper investigates the dimensionality characteristics of the outcome space of a combat simulation. The independent state variables of all of the outcome states for a simulation run for given event management policies were analyzed using techniques based on Principal Component Analysis and Singular Value Decomposition, to give metrics for dimensionality. The number of dimensions in the outcome space is indicative of variety of possible outcomes, a property potentially important in hierarchical simulation. Events were managed using random choices, multitrajectory methods designed to give greater preference to high probability trajectories, and by various methods guided by analysis of the impact of the various events. The number of dimensions could not be increased greatly by the event management techniques used for selecting event outcomes for multitrajectory resolution. Beyond 1000 replication runs, the size of the state space did not even strongly influence the metrics.

1 BACKGROUND

Combat simulations are often used in a hierarchical manner, where a higher resolution simulation is used to produce a library of possible outcomes used in a more aggregated, higher scope simulation. This is illustrated in Figure 1 in extremely simplified form. At the Center for Army Analysis, for example, the higher resolution simulation of approximately brigade or division scope is used to develop killer-victim scoreboards for a theater scope simulation. For this purpose, the higher resolution simulation runs need to include a large variety of possible circumstances which may arise in the larger scope simulation. Variety in the outcome set is desirable, not so much because the total set would be indicative of the mean outcome as might usually be the case in simulation based analysis, but in order to produce as much variety as possible so that the results tables will be richer.

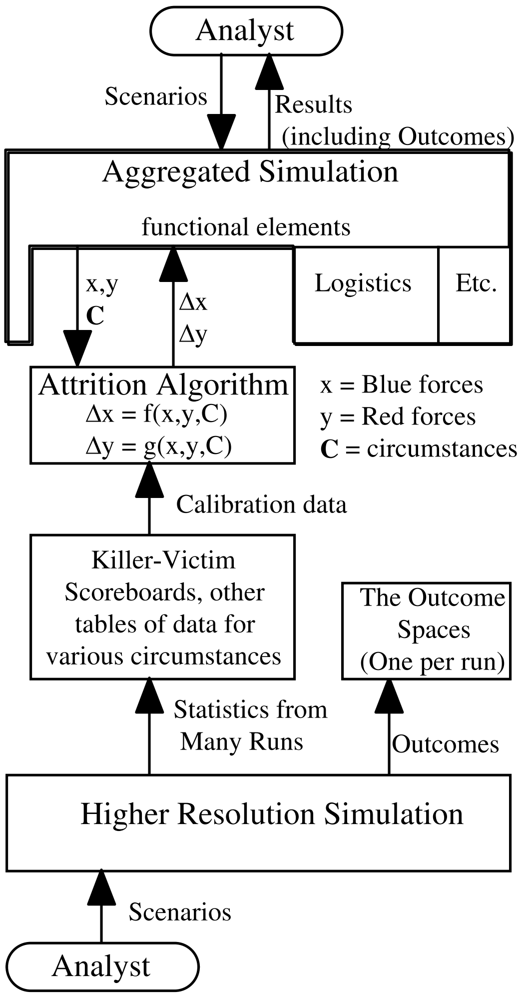


Figure 1: The Hierarchical Simulation Context

The hypothesis that a greater number of dimensions in the outcome space will benefit hierarchical simulation by producing better results was beyond the scope of this study. (By "outcome space," we refer to the collection of all state variables for all simulation trajectories.) The focus here is on the measurement of "dimensionality" of this outcome space as a metric for indicating variety, and examination of possible means to increase this variety through the management of how random event outcomes are chosen within individual trajectories.

The technique used for event management, "Multitrajectory Simulation", provides for "splitting" of states which encounter random events so that both (or multiple) outcomes of the event can be followed (Gilmer and Sullivan 1998). Figure 2 illustrates the concept. It is also possible to choose one outcome or the other for a given event based on some choice policy that reflects the interests of the analyst. Ultimately, some method is needed to decide when to use "splitting" and when to simply make random choices, since the potential for combinatorial explosion of the number of trajectories must be managed given finite resources. In our case, the interest is in greatest variety, as indicated by a large metric value used to assess the dimensionality of the outcome set. How such a choice might be guided is not obvious; in the work reported we tried several strategies ranging from the random choices normally used in stochastic simulation to choices guided by analyses of the importance of various events.

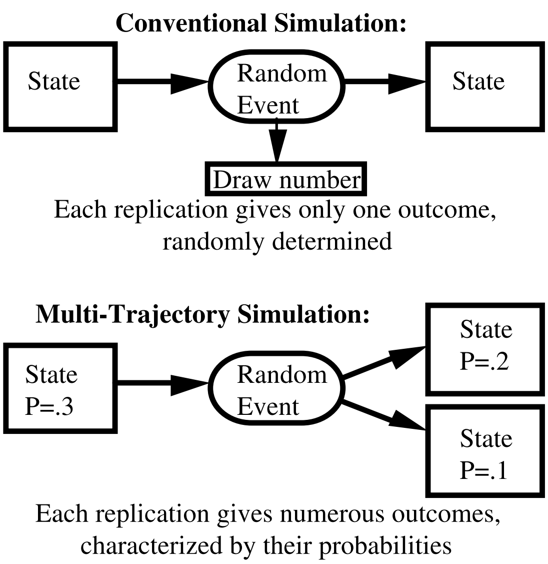


Figure 2: Multitrajectory Resolution of Random Events

2 THE MULTITRAJECTORY SIMULATION

This research has been conducted using a simplified, unclassified surrogate for the military simulations of interest. The simulation "eaglet" was designed to resemble the Corps level simulation "Eagle" in important respects, but to be of manageable simplicity. It includes Lanchester square law combat, movement by nominally battalion sized units along routes with multiple paths, acquisition and acquisition loss, decisionmaking, and artillery support. These processes can be selected for resolution in deterministic, random (stochastic), or multitrajectory methods. (The attrition resolution method had been found to be much less important than the others, so for all of the results given a deterministic attrition resolution was used.) In the version of the simulation used for the work reported here, units had multiple types of weapon systems. The "two division" scenario (referred to henceforth as "scenario 2"), used for this study is shown Figure 3. A small division of two Blue brigades and a large reserve battalion and supporting artillery attack a similarly configured, but somewhat smaller, Red force. Each of the brigades has two forward battalions, plus one in reserve. The multiple path options available to the advancing Blue maneuver battalions are shown. Decisionmaking logic for each HQ commits the reserves in support of the forward forces at some point in the battle.

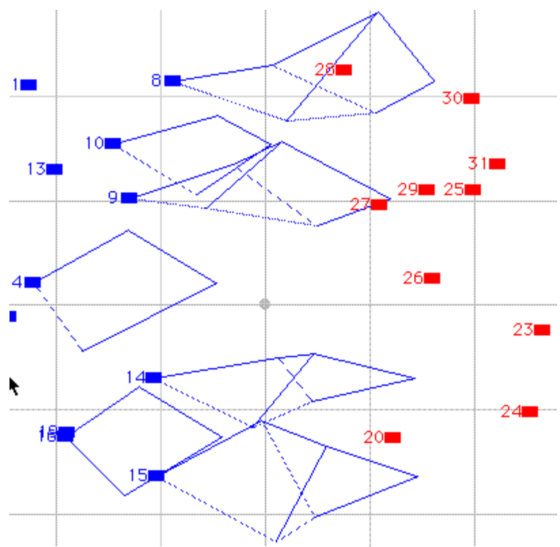


Figure 3: Two Division Scenario "Scenario 2"

In addition, some analysis was performed for a much smaller scenario of only 4 units, shown in Figure 4.

This scenario was constructed to present a very sensitive situation in which the battle can swing dramatically either way depending on whether Red reacts to reinforce the smaller force or not. The alternate paths are not shown in the figure.

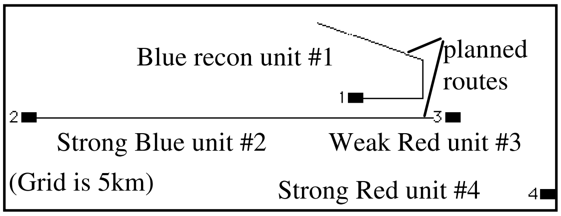


Figure 4: Small Scenario, "Scenario 0"

3 OUTCOME SPACE ASSESSMENT

An outcome of a simulation, if we think of a single replication (or, synonymously, trajectory), is the set of state variables for all of the objects in the simulation. This can simplified by eliminating all of the variables which can be directly derived from others, and by eliminating all state variables which have the same value in all states. The end point of a simulation trajectory can be thought of as a vector of these state variables. In the "two division" scenario, there were twenty variables per unit and thirty units, giving a total of up to 600 variables. These included such things as unit strengths for various weapons and other assets, location, speed, objective, type of operation and mission, and orientation. The number of variables after simplification never exceeded 300 since many of these were the same in all outcomes.

The set of all such state outcomes for multiple trajectories would be the outcome space for a set of trajectories generated in the course of a simulation execution, or "run". With conventional simulation one run might be thought of as corresponding to a single trajectory. With multitrajectory simulation, even in simple stochastic mode, many trajectories are executed simultaneously in one execution of the simulation, so the word "run" will be used for the execution of these many trajectories under a common regime of event management.

For statistical purposes, some variables were considered more important than others, and weighted accordingly, with weights ranging from 2 for force values (e.g. number of tanks) down to .25 (e.g. X and Y values of objective location). The zero state (which in some runs was the deterministic reference) was considered a reference, and all variables were expressed as differences from the zero state's value. The state variables were each normalized by subtracting the mean for that variable, after taking the difference from state 0, and dividing by the variable's measured estimate of standard deviation (limited to a floor of 1 to prevent extremes). The use of the zero state as a reference meant that the normalization process eliminated that state, so that the largest state space order for a 100 trajectory run would be 99. Figure 5 illustrates the normalization concept, and Figure 6 gives a simple example

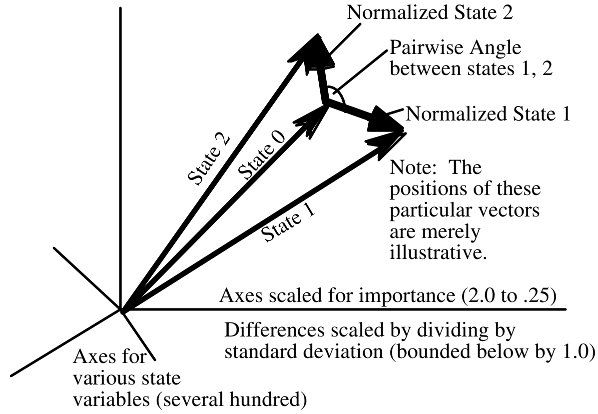


Figure 5: Normalization of State Vectors

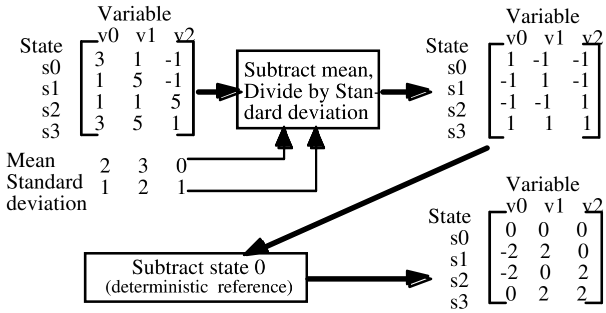


Figure 6: Normalization Example

A central issue in this project was how to evaluate the "variety" of outcomes for such an outcome space. We are interested in the degree to which states are divergent or orthogonal to others. But this is subject to different possible meanings, and no one metric seemed to tell the whole story. Several approaches to measuring outcome space variability were considered, ranging from simple (counting the number of vectors left after performing Gaussian elimination) to quite complex. It was expected that any one metric might not characterize the outcome space in all of the most important ways, so ultimately several metrics were used, and are described below:

1. Sum of sines: The angle each vector makes to another can be derived from a dot product of the two, which in turn will indicate if the vectors or orthogonal (sine of 1 or -1) or parallel (sine of 0). As each state is considered, the product of the sines of that state to all preceding states will remain with an absolute value of 1 if it is orthogonal to all others, but near zero if aligned in parallel with any. The sum of these (absolute values of) products of sines therefore is a count of orthogonal states, with a partial credit for states which are not orthogonal but are also not parallel with any of the earlier ones. Figure 7 shows how this is done using the four state, three variable state set used above.

As the number of states becomes large, fewer add very much value to this metric at all. Unlike other methods, a state coplanar with two others will get some credit for not being parallel to either, although it does not really add another dimension in the usual sense. This metric proved to be subject to quite a large variability: some small runs had quite large values, while some large runs had small values.

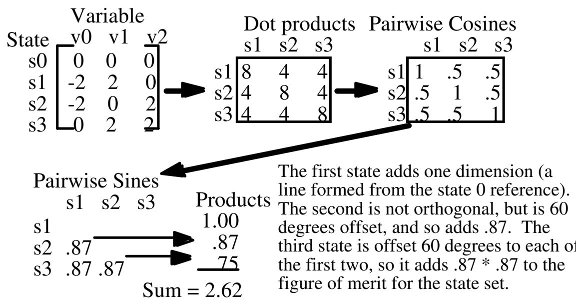


Figure 7: Sum of Sines Method Example

2. Number of independent variables: When the outcome space is simplified by eliminating any variables which are constant over all of the states, the number of remaining variables is a metric directly indicative of outcome variety in a useful sense.

3. Principal Component Analysis (PCA): The intent of this method is to measure the relative contributions of the various independent variables. This is done by finding the Eigenvalues of the Correlation Matrix for those variables. If the variable having the largest Eigenvalue is normalized to "1", as contributing a full dimension, then other variables which are somewhat dependent on others contribute less. The normalized sum of these Eigenvalues this gives a metric that can be thought of as the equivalent number of independent dimensions in this statistical outcome space, though in fact there are many more dimensions with varying degrees of dependency. (Examples: A circular distribution for two variables would give a value of 2. An oval distribution of two variables would give 1.5 if the correlation coefficient is .5. A football shaped distribution for 3 variables would give approximately 2.) Figure 8 illustrates this process using the simple state data shown earlier.

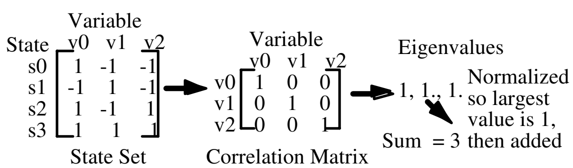


Figure 8: Illustration of Metric Based on Principal Component Analysis

In this case, the variables happen to be uncorrelated, so the sum of the Eigenvalues is three, the same as the number of independent variables. More typically the sum is much smaller than the number of variables, usually from 15 to 20 for state sets having 100 to 300 or so variables.

Note that the correlation matrix is calculated at the same time as the standard deviations, so the variables are not normalized to set states to have values relative to state zero as in the earlier case.

4. Singular Value Decomposition (SVD): The number of basis vectors in the outcome space itself is a useful measure of outcome variety (Press, et.al.). These can be found using Singular Value Decomposition, and the corresponding "Singular Values" that measure the contributions of each. Unlike the PCA technique which is derived from variable statistics (the correlation matrix), the Singular Values derive from the state set itself. As with PCA Eigenvalues, the largest Singular Value can be considered the norm for a unit contribution to a scalar dimension metric, with smaller values contributing proportionally less, to give an aggregate metric characterizing the sense of the dimensionality of the spanned space. Figure 9 illustrates this process:

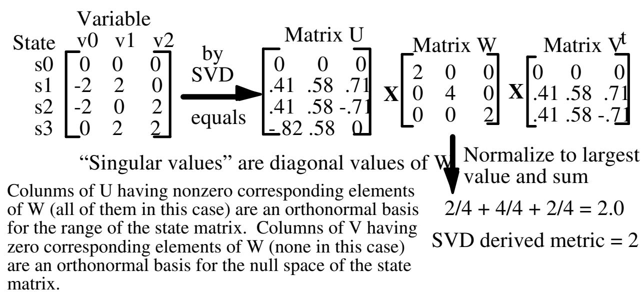


Figure 9: Illustration of Metric Based on Singular Value Decomposition

5. Number of Nonzero Singular values: The number of nonzero singular values could also be considered a useful metric, but in the presence of noise the threshold of significance that distinguishes a value from zero is a bit arbitrary. Any value greater than .00001 was considered nonzero. Generally the number was the same or very close to the number of states or the number of independent variables, whichever was less. For the simple example above, this count would be 3.

As an example of the above metrics, see Table 1, which gives the values derived from a number of runs for varying numbers of stochastic replications. Each of the three runs in a set used a different random number generator. Notice that the scalar metrics derived by the sum of sine products, PCA, and SVD show considerable variation. Indeed, the smallest value for the Sum of Sine Products method occurs for one of the largest runs. Although the PCA values here seem more consistent than the others, that metric also had occasional extreme values. Note also that the impact of increasing the number of replications begins to be more marginal in its effect.

Table 1: Scenario 2 Stochastic Outcome Space Metrics

N of Trajectories	Sum Sines	N variables	PCA	Nonzero SV's	SVD
100	5.683	257	13.27	99	6.581
100	12.42	270	13.66	99	9.928
100	5.101	262	13.07	99	7.069
1000	11.698	279	16.74	275	11.94
1000	7.276	285	17.5	277	10.05
1000	13.006	289	17.12	278	13.48
10000	10.308	292	17.66	292	12.69
10000	2.197	292	17.74	292	6.379
10000	6.438	298	17.81	294	11.06

One reason that considerable variation in these metrics is seen is the sensitivity to the value of the largest Eigen or singular value. The two sequences below give the largest several ordered Eigenvalues for the smallest and largest PCA based metrics respectively from Table 1. The remainder of the 256 or 290 (respectively) Eigenvalues taper off toward zero.

Sum = 1.97: 130.08, 11.30, 7.74, 6.64, 5.73, 5.32, 5.16..

Sum = 30.99: 6.77, 12.12, 9.73, 8.94, 7.95, 6.37, 6.61..

A similar run with a different seed or random number generator selection might have a quite different array of Eigenvalues without having the extreme initial value. An extreme value for one metric did not always correlate with an extreme for another. Indeed, the metrics seldom vary together, although the second 10000 trajectory of Table 1 shows much lower than expected values for both SVD and Sum of Sine Products, but not PCA. As can be seen, these variations seemed as likely to affect the metric as did choice of simulation event management method or event he number of trajectories. Thus, averages of a number of runs were used to compensate somewhat. This occasional large variation may be indicative of potential for better event management techniques that might more consistently generate large metric values, but this potential has not been explored yet.

4 EVENT MANAGEMENT TECHNIQUES

The intent of event management is to select the event outcome for a random event in a manner that will enhance the value of the outcome set to the analyst. The methods used in the work reported here include those listed below. (In addition, other methods have been developed which were not used in this analysis.)

mt0. Deterministic: For each event type, there is a deterministic default resolution method:

a. Movement: Choose the first exit link from a node
b. Acquisition: Units inside the deterministic radius are seen, those outside unseen. (This radius is half-way between the bounds where a unit might be seen.)
c. Acquisition Loss: Units inside a deterministic radius retained, those outside lost.
d. Decisionmaking: Rules satisfied under 2 or 3 of 3 sets of alternative criteria fire; if satisfied under only one or fewer sets, it doesn't. (Variable criteria are for rule thresholds, e.g. 60%, 65%, 70% effectiveness.)

mt1. Stochastic: A random number is drawn, and used to resolve the event:

a. Movement: Choose the exit link from a node according to their probabilities
b. Acquisition: Units within the "possibly acquired" zone acquired with a 50% chance.
c. Acquisition Loss: Units within "possible loss" zone lost with a 50% chance.
d. Decisionmaking: Rules satisfied for 1 or 2 sets of criteria have increasing probabilities.

mt4. Multitrajectory with stochastic limit: Up to a given state limit, slitting is used to generate new trajectories. Beyond that limit, stochastic resolution is used. This is a useful blend of techniques. The disadvantage, though, is that trajectories will have a very large range of probability values.

mt6. Multitrajectory with "soft" stochastic state limit: Up to a "soft" state limit all events are multitrajectory. Beyond that, up to a hard state limit, the trajectory probability multiplied by the state limit is compared to a numeric criterion, usually about 1.0 or a bit higher. Trajectories that have a sufficiently high probability pass this test, and their events are resolved in multitrajectory manner (splitting the probabilities of the outgoing trajectories). Events for trajectories with lower probabilities are resolved stochastically by random number draw. As the state limit is approached, fewer trajectories satisfy the criterion, so that most trajectories will have about the same probability at the end. This is perhaps the best method to use when no information about the relative importance of the various events is available.

mt9. Multitrajectory up to soft state limit, then stochastic if event importance and probability are sufficiently low. This is similar to Method 6, but adds another consideration: event importance. If Method 9 is used for any event, event importance data from an earlier run is read into the simulation. There is an event category for each event type, unit, and data (movement node, unit to be seen, or rule number). Each such category has an importance score, all of which are normalized such that they sum to one. If the trajectory probability times event importance divided by the state limit exceeds a numerical criterion, the event is resolved in multitrajectory fashion. Otherwise, it is resolved by random draw. This allows the management of events in a manner sensitive to event importance. There are actually five different sets of importance numbers in a typical event importance file. These are derived using different methods for assessing event importance. The analyst must perceive which of the five is to be used.

One strategy that can be used for obtaining a wider variety of outcomes is based on the notion of a limited event tree. The simplest example is a "leftist tree" run in which the initial trajectory is entirely deterministic. For this reference trajectory, at each event, a new trajectory is initiated for the alternative outcome. But these alternative trajectories do not branch further; they continue to termination as deterministic branches. Figure 10 illustrates.

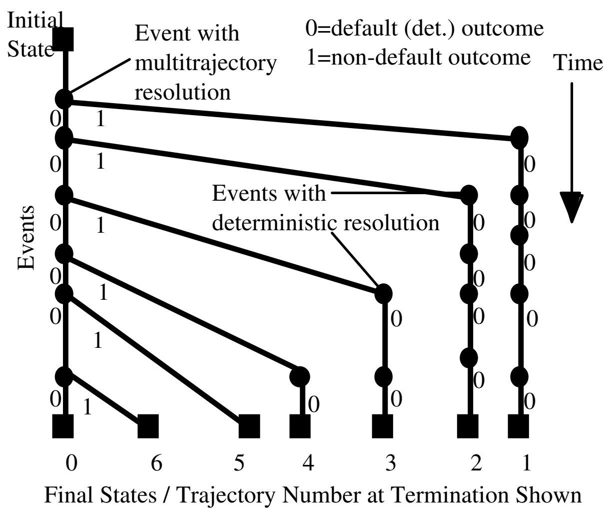
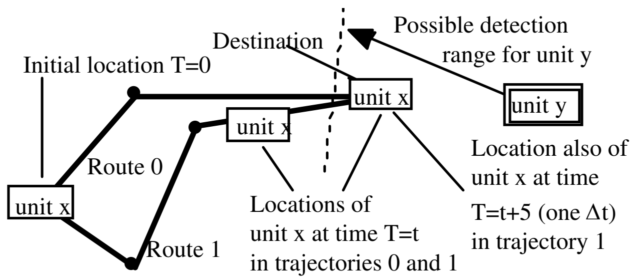


Figure 10: Leftist Tree Multitrajectory Execution

The total number of uniquely identifiable events is typically much larger than the number in any given run, since events in different trajectories that one might otherwise consider the same may happen at different times. For scenario 2, a typical trajectory might have 300 events, but a 500,000 trajectory run typically has about 17,000 uniquely identified events across all of the trajectories.

The advantage of a tree run is that there are two trajectories that differ only by the impact of a single event. Differences in their final states can be attributed to that event. However, the existence of an event in a particular trajectory may depend upon the outcome of a previous event. For example, an acquisition event may depend upon an earlier movement selection event that affects the time of arrival at the target unit's destination. If the unit is later arriving in one trajectory, the acquisition event against it would not occur in that trajectory while it might occur in another. Figure 11 shows such a case.



Trajectory 0: Event 0: unit x chooses Route 0.
Event 1: unit y attempts to detect unit x
Event 2: unit x attempts to detect unit y
Event 3: unit y attempts to detect unit x (again)
.... (events 1,2,3 spawn yet more trajectories)
Trajectory 1: Event 0: unit x chooses Route 1.
Event 1: does not occur
Event 2: does not occur
Event 3: unit y attempts to detect unit x (first time)

Figure 11: Event Dependence on an Earlier Event

All of the multitrajectory methods that use information on events, designated as Multitrajectory Method 9 (or "mt9") depend on estimation of event importance. Several methods of event analysis were explored, each of which results in a different set of event importance values. Space does not allow these methods to be described in detail. The methods are summarized below:

1. Average event effect: The (normalized) sets of state vectors associated with the default and non-default outcomes are both averaged, and the difference is taken. The magnitudes of the difference vectors associated with the various events are then normalized so that they sum to one. The effects for similar events are aggregated, and included in the event importance output. In the case of events for which only one outcome occurs, we cannot really find a difference in the sense just described. However, we can take a difference between the average for states having the one event outcome, and the others where the event does not occur at all. This metric may most directly reflect the impact of an event in terms of making a difference in outcome, and may be the most obvious and straightforward metric. However, there is no good reason to suppose that the use of this metric will also effect much of an improvement on outcome space variety metrics.

2. Average sine of angles: This metric is closely related to the sum of product of sines metric for state analysis. The interest is in identifying events that cause state vectors which are more nearly orthogonal. Angles are measured between each state with the default outcome and all states with the opposite outcome. The average absolute value of the sine of these angles is then taken as the metric associated with the event. These are then grouped and normalized to add to one as for the other metrics.
3. Average and minimum distances: For each state, the distance to the nearest state, and the average distance to all other states, is calculated. If the state results from some outcome of event E, the sum of nearest and average distances for this state are averaged in with the same values for other states having an outcome for this event. This metric gives weight to both nearest neighbor and average distance; it does not care what the event outcome is, only that it occurred.
4. Linear Event Impact Model: This method attempts to model the event set as having a linear effect on the outcome of the simulation, and find the weights such that the linear model gives a close approximation to the actual outcome space. Ultimately, we would hope to have a fairly close convergence toward the actual outcome set. But, the events are neither linear nor independent, and the approximation was observed to be rather poor, and does not even converge. Undoubtedly this algorithm might have been better constructed, with perhaps better event effects models sensitive to dependencies, but that remained beyond what could be achieved.
5. Single event difference event effect model: This method is a less ambitious version of number 4 above, in that event impacts are only calculated where there are two trajectories that differ by only that event. (An event occurring in one but not the other of a trajectory pair is considered not disqualifying for this purpose.) In a normal stochastic or multitrajectory run, this would happen very seldom, so that this method would only rarely produce a viable event impact distance. But tree runs, as described earlier, are designed to produce just such coincidences. A multiple tree run, say a two way run of simultaneous rightist and leftist trees, will produce two such pairs in some cases, and four way trees may be even richer in useful trajectory pairs.

A few observations about these importance measures are worth noting. In a sense, the first metric is the most robust as well as straightforward, in that it returns some value for every event whether both outcomes occur or not. Method 3 is also quite robust, but tends to generate importance values which do not vary widely, suggesting that all events are more equally important than would be reasonably expected. The others tend to miss some events. Method 5, which requires event pairs to register a nonzero importance, is fairly sparse. Method 4, however, tended to produce event sparser importance values. To summarize, Method 4 importances cannot be regarded as trustworthy, and there are reasons to consider Method 3 suspect. Method 5 misses getting data on quite a few events due to the coincidence requirements, and thus may miss some important events. Method 1 would seemingly be the most trustworthy, though method 2 is better targeted on the dimensionality goal.

5 RESULTS AND OBSERVATIONS

The sets of runs and methods of analysis were different for the two scenarios. For scenario_0, a much greater variety of run sizes were made, but in many cases only one run was made for a given size. In order to avoid statistical aberrations, comparisons for comparable mixes of run sizes are made. This is not as much a problem as might be thought, since the metrics collected seem rather insensitive to run size for this scenario. For scenario_2, on the other hand, only a few run sizes were used (100, 1K, and 10K) but three runs each with different random number methods were used. Even so, aggregate comparisons based on a mix of run sizes are useful.

Averages for various methods for different techniques have been collected in Table 2. These averages are taken across a number of different trajectory set sizes ranging from 100 to 500,000, with the different sizes weighted similarly for each set of metric values reported in the table. The standard deviations calculated were for individual runs. The "Sum of Sines" metric was not available.

Table 2: Scenario 0 Result Averages

	nC	PCA	nW	SVD
mt1	57	3.69	54	3.84
mt4	57	4.24	55	3.84
mt6	57	3.67	55	4.11
mt9 (1)	58	3.97	56	3.61
mt9 (2)	58	3.94	56	3.86
mt9 (3)	58	3.94	56	3.86
mt9 (4)	56	4.16	53	3.11
mt9 (5)	57	4.85	54	3.41
std dev	2.2	0.65	4.3	0.43

If we are willing to assume that the numbers in Table 2 are meaningful, we can observe that almost every method does better than stochastic by the PCA metric. But the standard deviations are large enough that we can only say that mt4 and, more so, mt9 method 5 are significantly better. (The estimated standard deviations shown are for single runs and we are dealing with averages of several runs, so the standard deviations for table entries can be considered smaller by a factor of about 2.) On the other hand, mt9 method 5 does worse than stochastic by the SVD metric, and mt4 is no better. Only mt6 does better, and by barely enough to be significant.

The two primary metrics, PCA and SVD, seem to be pulling in opposite directions. A correlation coefficient taken over all mt9 runs turned out to be -.054. Whatever we may mean by "dimensionality" is clearly different by these two metrics. One, PCA, concerns statistics about the state set, and the other measures properties of the state set itself. Which is more important? Whichever it is, it would seem there is some method that will improve on stochastic methods. It may well be that better methods of assessing event importance will improve on these results.

Table 3 summarizes the results by averaging all runs having the same event control method and state limit. (Actually, the number of states may be smaller than the state limit when the criterion parameter is set high in mt9 runs.)

Table 3: Scenario 2 Result Averages
(for 1000 or more trajectory runs)

	Sin	n	PCA	n	SVD	runs
mt1	8.59	291	18.17	288	11.24	9
mt4	6.46	291	17.55	288	9.07	9
mt6	8.73	290	17.58	287	10.08	9
mt9 (1)	8.21	292	18.13	288	9.77	13
mt9 (2)	9.94	290	18.63	289	10.62	11
mt9 (3)	12.29	291	17.60	288	11.49	13
mt9 (4)	9.60	290	17.69	287	9.80	9
mt9 (5)	9.20	282	16.04	274	9.99	12
tree runs	16.27	248	4.74	211	7.769	7
std dev	3.76	3.93	2.15	7.43	1.87	

For this larger, less sensitive scenario, the choice of trajectory management technique seems to have made less difference than it did for scenario 0. Except for the tree runs themselves, no technique seems to perform significantly better or worse than any other evaluated by the PCA criterion. The estimates of standard deviation (for individual runs) are based on the 68 individual runs of 1000 trajectories or more. For the average statistics shown above, each from several individual runs, one would expect a standard deviation of about 1/3 that for individual runs. The only technique which, by this criterion, may slightly outperform stochastic simulation is multitrajectory technique 9 with evaluation method 2, which gives a marginally better PCA. Method 5 was significantly lower, but many of these runs were small because of the limitations of the event evaluation technique, which means that the cases being compared are not really equivalent. (For example, both of the nominally 50,000 trajectory limit runs never got beyond 1000 trajectories.)

By the SVD technique, only Multitrajectory policy 9 with event evaluation method 3 was higher, and not significantly so. This same case registered remarkably higher by the Sum of Products of Sines method, which did tend to correlate somewhat with SVD scores. This was the method that used minimum and average distances between states in the outcome set as the criterion for evaluating event importance. Other methods perform significantly below stochastic simulation by the SVD metric.

For this scenario, the results of the tree runs is also given. These are very small runs compared to the others. Yet the Sines method registers a very large score. This may be an indication that this metric is suspect. All of the tree runs included many duplicate states, typically 5% to 10% of the total; no other technique produced any for this scenario. (Only a handful of duplicates occurred outside of tree runs occurred for the smaller scenario 0.)

6 CONCLUSIONS

Perhaps the most important conclusion we can make is that, on the average, the performance of the various methods, and event sizes, are not very different. We do not see dramatic differences of factors of 2 or more; the numbers seem to be pretty close. On the other hand, individual runs show considerable variation. There are a number of alternative hypotheses:

1. With at least 100 or so trajectories, we are learning about all we can expect to about the outcome space, at least as far as dimensionality is concerned. The wide variations among individual runs is due to the sensitivity to the largest Eigen or Singular value and is a defect in the metric more than an indication of variation, since this occurs with both large and small runs.
2. The number of dimensions (taking into account their importances as is done with the PCA and SVD metrics) is a number that varies only slowly with the number of runs. Just as it takes a doubling of the number of runs or symbols to add one bit to an entropy metric, it may take many runs to increase dimensionality significantly. If that is so, then methods that show even a small increase in these metrics may be valuable alternatives to enormous numbers of runs.

These two hypotheses are not mutually exclusive. The metrics chosen are indeed sensitive. Perhaps a better metric more directly reflecting the operational characteristics important to hierarchical simulation could be developed that is less sensitive. Improvement here has been beyond the scope of what could be accomplished.

ACKNOWLEDGMENTS

This project and those immediately preceding it were funded by the US Army Research Office under Grants DAAH04-95-1-0350, DAAG55-97-1-0360, and DAAG55-98-1-0451 with the sponsorship of the US Army Center for Army Analysis, particularly Mr. Gerry Cooper, Col. Andrew Loerch, and Dr. Robert Alexander. The authors are indebted to these persons for their encouragement and support. They would also like to thank Kosmo Tatalias for his advice and help.

REFERENCES

Press, William H., Saul A. Teukolsky, William T. Vettering, and Brian P. Flannery. 1992. *Numerical Recipes in C, 2nd edition*. New York: Cambridge University Press.

Gilmer, John B. Jr. and F. J. Sullivan. 1998. Alternative implementations of multitrajectory simulation. In *Proceedings of the 1998 Winter Simulation Conference,* ed. D.J. Medeiros, F. J. Watson, J. S. Carson, and M. S. Manivannan. 865-872. San Diego, California: Society for Computer Simulation.

AUTHOR BIOGRAPHIES

JOHN B. GILMER JR. worked in the development of combat simulations, with a focus on C2 representation and parallelism, at BDM, Inc. He was the chief designer of the CORBAN combat simulation. He has a Ph.D. in EE from VPI and currently teaches Electrical and Computer Engineering at Wilkes University. His e-mail address is `jgilmer@wilkes.edu`.

FREDERICK J. SULLIVAN is Dean of Technology and teaches Computer Science at Wilkes University, and earlier taught at Rose-Hulman and SUNY Binghamton. His expertise is in operating systems and object oriented software. His Ph.D. is in Mathematics, from LSU. His e-mail address is `sullivan@wilkes.edu`.