*Proceedings of the 1999 Winter Simulation Conference*
*P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.*

# MULTITRAJECTORY SIMULATION PERFORMANCE FOR VARYING SCENARIO SIZES

John B. Gilmer, Jr.
Frederick J. Sullivan

Wilkes University
P.O. Box 111
Wilkes-Barre, PA 18766, U.S.A.

## ABSTRACT

Multitrajectory Simulation allows random events in a simulation to generate multiple trajectories, a technique called "splitting", with explicit management of the set of trajectories. The goal is to gain a better understanding of the possible outcome set of the simulation and scenario. This has been applied to a prototype combat simulation, "eaglet" which was designed to have similar, but simpler, representations of the features of the "Eagle" simulation used for Army analyses. The study compared the number of multitrajectory simulation trajectories with numbers of stochastic replications to experimentally determining the rate of convergence to a definitive outcome set. The definitive set was determined using very large numbers of replications to develop a plot of loss exchange ratio versus losses of one side. This was repeated with scenarios of from 40 to 320 units. While the multitrajectory technique gave superior results in general as expected, there were some anomalies, particularly in the smallest scenario, that illustrate limitations of the technique and the assessment method used.

## 1 BACKGROUND

The goal of multitrajectory simulation is to explore the outcome space of a simulation, that is, the set of all possible outcomes, more systematically and less expensively (for a given quality of understanding) than can be achieved with conventional stochastic simulation. This may be considered a variance reduction technique, but the analysis goals may be formulated not only in terms of better estimates of statistical properties of the outcome set, e.g. a mean and variance for Measures of Effectiveness (MOEs), but also representative instances of extreme behavior or other "interesting" cases (Al-Hassan, Gilmer, and Sullivan 1997).

The heart of the proposed method is to explicitly track each possible trajectory, as illustrated in Figure 1. When an event that would normally be stochastic occurs, instead of one outcome, multiple outcomes are generated, each constituting a trajectory having its own state. Because the trajectory bifurcates, this is also referred to as "splitting", with "cloning" of the state. In concept, such a multiple trajectory simulation is integrated with its support system in such a way that its use provides outcomes with probabilities associated with each, an accounting for the key events or circumstances leading to the differences, and some measure of confidence in these results.

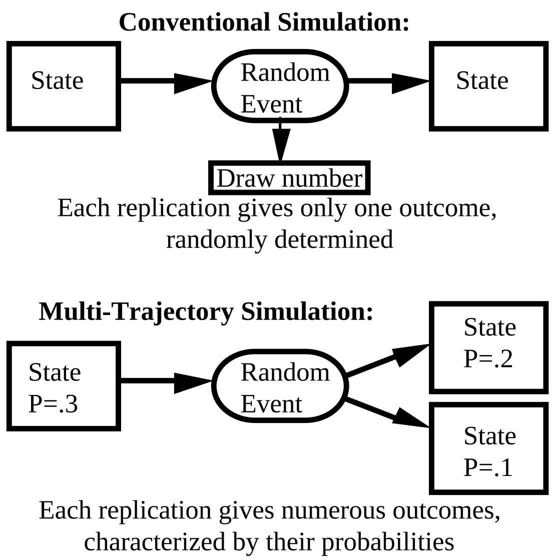


Figure 1: Concept of Explicitly Tracking Trajectories

Two techniques have been used which trade off coverage for the sake of keeping resources bounded. One is the "truncation" management technique that explicitly decides, for each event in some trajectory, whether to resolve the event in multitrajectory fashion (resulting in the creation of a new trajectory) or to instead resolve it deterministically or stochastically. The case of stochastic resolution with only one continuing state corresponds to the "Russian Roulette" used in conjunction with "splitting" as a variance reduction technique. This is illustrated in Figure 2. A second approach to reducing the number of states to look for and consolidate states that are "similar". That technique was not used in the work reported here.

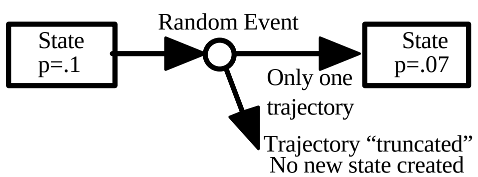


Figure 2: Trajectory Truncation

The software techniques for cloning the states and properly initiating the alternative trajectories can be tricky, especially if the stochastic event is deeply embedded in the functional model code. Several ways to do this have been found, but are beyond the scope of this paper (Gilmer and Sullivan 1998). The technique used to generate the results reported here is "Sullivan's Method", in which the objects in the simulation base classes are responsible for trajectory management, so that the modeler does not have to be concerned with the messy details of cloning and splitting. At the time the results were generated, the algorithm operated only in "breadth first" fashion, with all active trajectories brought forward with time staying consistent. Since then, a "depth first" technique and hybrid methods have been developed that allow the memory usage to remain manageable while producing large numbers of trajectories.

The most representative trajectory management technique used for the analysis to follow is illustrated in Figure 3. This "breadth first" approach uses multitrajectory resolution of events up to the point where a state limit is reached. From that point, the resolutions are random draws, with alternate trajectories truncated.

Other approaches included a method in which beyond a "soft" state limit only relatively high probability trajectories have events resolved in multitrajectory fashion. Deterministic rather than stochastic resolution beyond the state limit is another alternative, and in other studies the idea of using multitrajectory resolution only for "important" trajectories with respect to some Measure of Effectiveness has been explored.

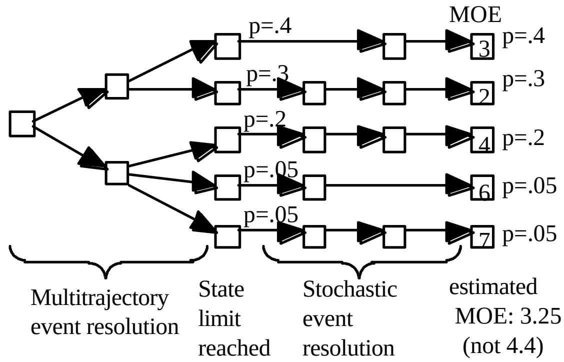


Figure 3: Hybrid Event Resolution (Method 4)

## 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, decisionmaking, and artillery support. Figure 4 illustrates the smallest scenario we have used with "eaglet". Two Blue units attack a Red unit. A second Red unit counterattacks from the flank. Note that the route objects show multiple paths; when there are multiple links from a given node, a multitrajectory event occurs for the choice of which path to follow. Figure 5 illustrates the process of creating a new trajectory when this happens. Multitrajectory attrition (variations in combat losses), decisionmaking (whether a decision rule fires), acquisition (whether a unit sees another) and acquisition loss have also been implemented. All except multitrajectory attrition were used in the results shown later. Multitrajectory attrition was found to be very expensive relative to its value, and so was not implemented in the "Sullivan's Method" version of "eaglet".

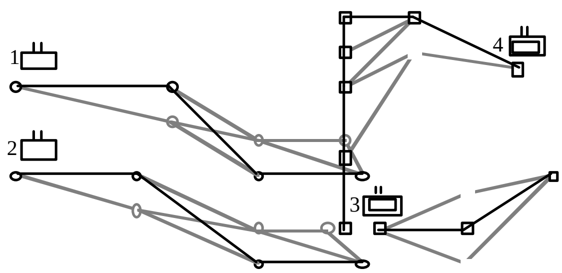


Figure 4: "Eaglet" Scenario with Multitrajectory Routes

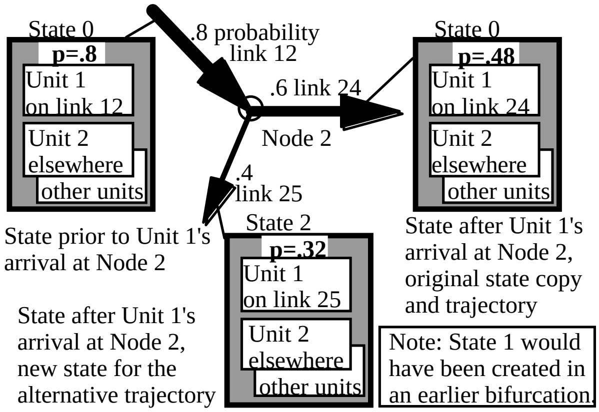


Figure 5: New Trajectory Creation for a Move Event

Because we were interested in scaling performance, there was a need to generate scenarios automatically at various sizes. To do so, a template based plan generator was developed that would, hierarchically, generate division and brigade plans for both sides. Templates were developed for attack (Blue) and defense (Red) operations at both levels, and instantiated for each division specified for a given size scenario. The smallest scenario in the analysis featured two divisions, one Blue and one Red, with a total of 40 resolution (battalion sized) units. Larger scenarios were generated by having multiple division sized battles, side by side. Units in adjacent divisions may encounter enemy units other than their direct opposites, so it is somewhat more complicated than just a series of disconnected battles. However, representing decisionmaking above division level was beyond the scope of what was possible in this effort.

Figure 6 below illustrates a brigade level template used in the planner. The tasks have associated template routes, which are illustrated in Figure 7. The unit initial locations, objectives, and route waypoints are varied by 25% with respect to the planning grid formed by the unit's sector width and distance to its objective. This gives variation in the battle configurations and prevents total synchronization across the scenario.

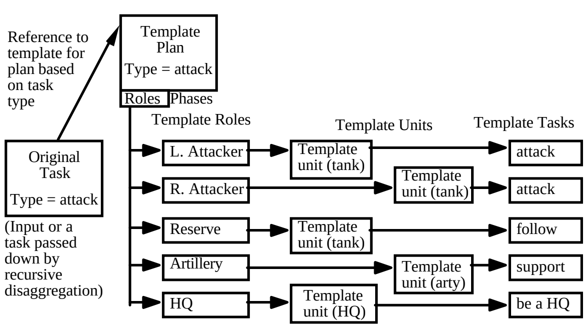


Figure 6: Brigade Attack Planning Template

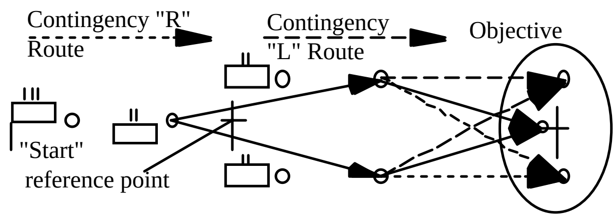


Figure 7: Attack Route Template with Contingency

Figure 8 shows a typical 2 division scenario. Some rear units are not shown. The grid lines are 5 Km apart. Only initial routes are shown. Note the two brigades, each with two battalions up and one in reserve, advancing left to right, with the division reserve being one large battalion.

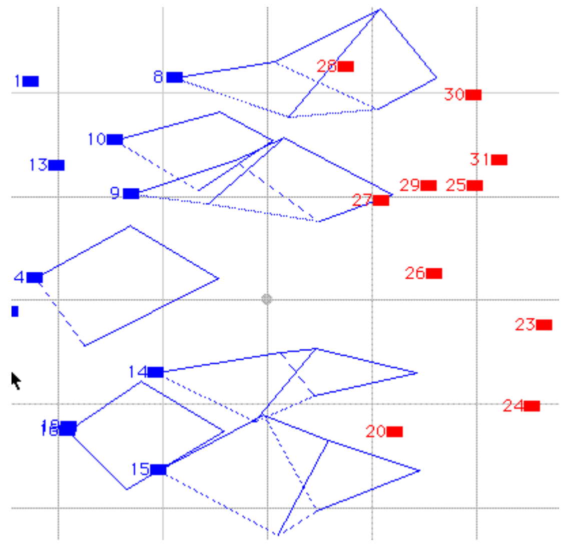


Figure 8: 2 Division Scenario with Random Variations

## 3 EXPERIMENTAL APPROACH

The analysis performed in this project was designed to compare the performance of the multitrajectory technique to that of traditional stochastic simulation. We have selected two Measures of Effectiveness of particular interest to form a surface onto which the outcome space generated by a given set of runs can be projected. We then can compare various outcome space subsets produced by different methods by comparing their "pictures" formed by this projection onto the two dimensional MOE space. In particular, we wished to compare sets of n stochastic runs with the outcome of multitrajectory runs with a state limit of m, and find the values of n and m that give equivalent performance in terms of the fidelity with which the MOE plot is given. Figure 9 illustrates. This was thought to be a useful measure of how well a given set of runs conveys the nature of the outcome space to the analyst.

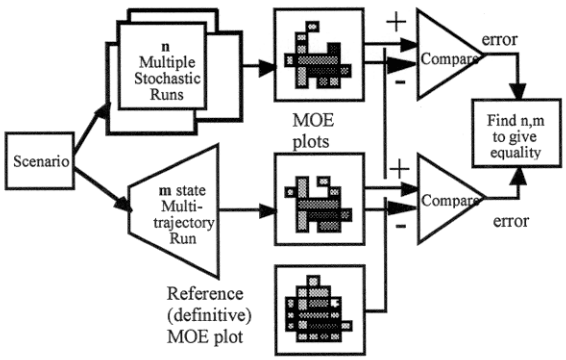


Figure 9: The Analysis Process Used to Compare Multitrajectory and Stochastic Runs.

This approach requires as a definitive reference an "outcome set" against which comparisons can be made. With the number of events in the hundreds, the size of the exhaustive outcome set is too large to produce even if memory limitations were overcome. We resorted to huge numbers of stochastic runs, generally 5 million. Figure 10 shows the reference 5M stochastic MOE plot for the two division case. The shading is logarithmic for this and all subsequent MOE plots.

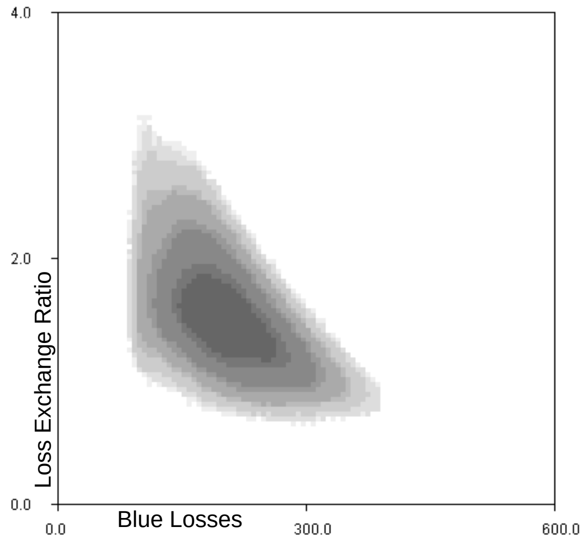


Figure 10: 2 Division Scenario Plot, 5M Replications

The number of points in an MOE plot (generally 100 x 100) is rather large considering the number of trajectories. Some degree of smoothing was thought necessary to dampen the random fine grained variations. We employed both the 3x3 and 11 x 11 center weighted smoothing functions, given in Figure 11. These were convolved with the both plots prior to making comparisons. The effect of smoothing can be seen by comparing unsmoothed Figure 12 and Figure 13, in which the 11x11 smooth has been applied.

$$\frac{1}{16}\begin{bmatrix}1, 2, 1\\ 2, 4, 2\\ 1, 2, 1\end{bmatrix}$$

$$\frac{1}{10}\begin{bmatrix}
.012 & .022 & .032 & .039 & .043 & .045 & .043 & .039 & .032 & .022 & .012\\
.022 & .044 & .061 & .075 & .084 & .087 & .084 & .075 & .061 & .044 & .022\\
.032 & .061 & .086 & .106 & .118 & .123 & .118 & .106 & .086 & .061 & .032\\
.039 & .075 & .106 & .130 & .145 & .150 & .145 & .130 & .106 & .075 & .039\\
.043 & .084 & .118 & .145 & .163 & .168 & .161 & .145 & .118 & .084 & .043\\
.045 & .087 & .123 & .150 & .168 & .174 & .168 & .150 & .123 & .087 & .045\\
.043 & .084 & .118 & .145 & .161 & .168 & .161 & .145 & .118 & .084 & .043\\
.039 & .075 & .106 & .130 & .145 & .150 & .145 & .130 & .106 & .075 & .039\\
.032 & .061 & .086 & .106 & .118 & .123 & .118 & .106 & .086 & .061 & .032\\
.022 & .044 & .061 & .075 & .084 & .087 & .084 & .075 & .061 & .044 & .022\\
.012 & .022 & .032 & .039 & .043 & .045 & .043 & .039 & .032 & .022 & .012
\end{bmatrix}$$

Figure 11: Smoothing Filters Used in Comparisons

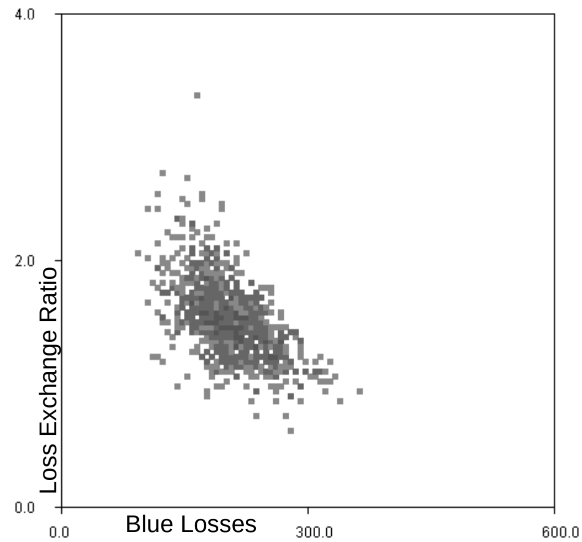


Figure 12: 2 Division Scenario Plot, 1K Stochastic

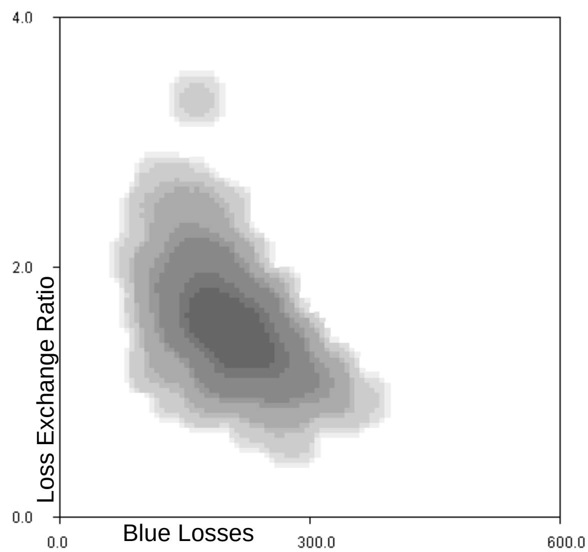


Figure 13: 2 Division 1K Stochastic plot, smoothed

Distances between histogram plots, either smoothed or unsmoothed, were taken by summing the absolute differences of the values for each pair of corresponding cells, then dividing by the number of cells.

As the number of runs increased, we checked to ensure that the differences in the MOE plot led to smaller and smaller changes. Figure 14 shows the convergence in average distance from the 5M replication case (shown in Figure 11) as the number of stochastic replications increases. (The comparison is to varying sized subsets of the large run, so the distance necessarily converges to zero at 5 Million.) The comparison is made with and without using a 3 x 3 point smoothing filter, that damps out the fine scale granularity when the number of replications (as few as 1K) is small compared to the resolution of the plot (100 by 100). For many runs an 11x11 smooth was used instead of 3x3, which is not shown, but would be lower on the left end of the graph.

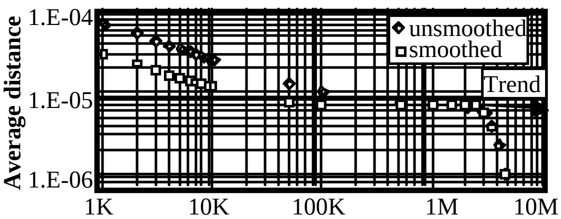


Figure 14: Convergence Toward 5M Replication Plot

Because of memory limitations, the Multitrajectory results are not for single multitrajectory runs, but for a collection of 1000 trajectory runs having different seeds. So, this really does not do full justice to the multitrajectory approach. With Multitrajectory Choice Policy 4, when the limited state budget is exhausted, the simulation runs in stochastic mode. Choice policy 6 differs from Policy 4 in that a lower "soft" state limit applies to low probability states, which go stochastic earlier than the "hard" state limit that applies to higher probability states. We did not find any significant differences between the two multitrajectory policies.

## 4 RESULTS, TWO DIVISION SCENARIO

The two division scenario includes 30 units. All of the maneuver units, and most of the other units, become involved in the battle, in contrast to the hand crafted 40 unit scenario that was used in earlier studies, in which somewhat less than half of the units enter combat. This data was produced using an 11x11 smoothing prior to making the comparisons. The unsmoothed and 3x3 smoothed data showed performance ratios between the two cases that were similar, with distances for the multitrajectory runs being about twice as large.. (Choice policy 4 was used on the multitrajectory runs rather than choice policy 6, which was used in later cases.)

It is illustrative to compare the actual outcome plots for this case. Figures 15 and 16 show the stochastic and multitrajectory plots for the 10K trajectory cases.

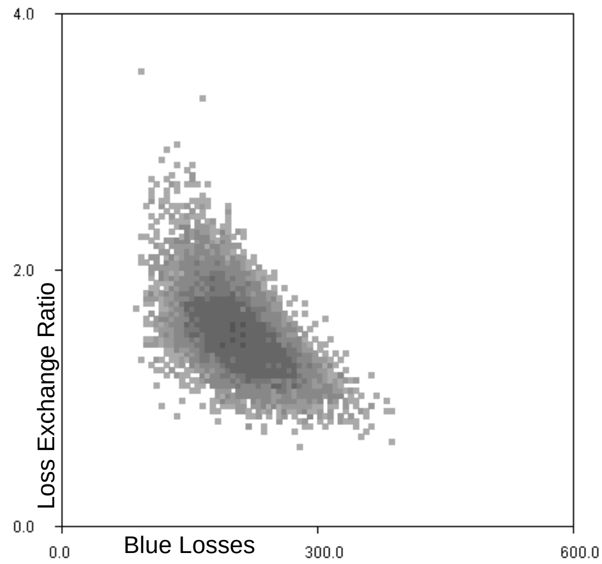


Figure 15: 2 Division Scenario, 10K Stochastic Plot

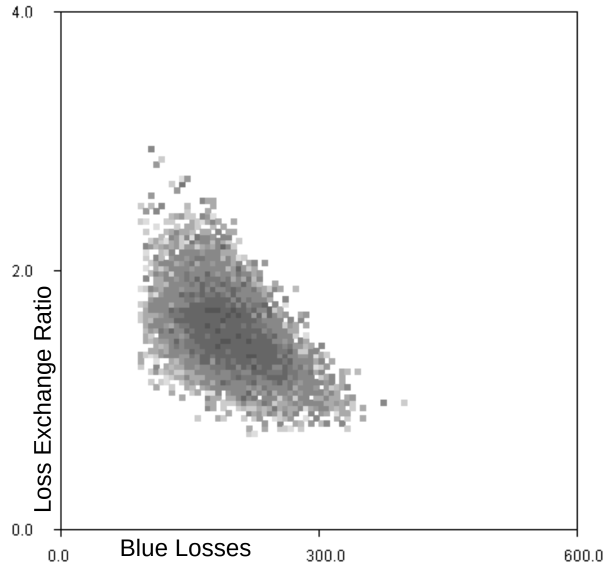


Figure 16: 2 Division Scenario, 10K Multitrajectory Plot

Distance data collected for the two division scenario are summarized in Table 1. For this scenario Multitrajectory Simulation does worse than Stochastic, at least using the analysis technique presented. The differences in distances are larger than the estimate of error in the definitive outcome set.

Table 1: Distance Results from 5M Stochastic, for Multitrajectory Runs for 2 Division Scenario, Smoothed

| Replications | All events stochastic | All events policy 6 |
|---|---|---|
| 1000 | 7.01E-06 | 1.52E-05 |
| 2000 | 6.36E-06 | 1.05E-05 |
| 3000 | 4.96E-06 | 1.00E-05 |
| 4000 | 3.89E-06 | 7.48E-06 |
| 5000 | 3.39E-06 | 5.90E-06 |
| 6000 | 2.72E-06 | 6.12E-06 |
| 8000 | 2.41E-06 | 5.94E-06 |
| 10000 | 1.99E-06 | 4.75E-06 |

The multitrajectory outcomes include some with smaller probabilities than are present in the stochastic case, reflecting the fact that some trajectories had less than the average probability at the time when the state limit was reached. This was expected to allow the multitrajectory mechanism to perform better. However, it is also the case that the multitrajectory outcome MOE plot shows more variability in the cells that are toward the middle of the MOE center of mass. It is this variability that seems to account for why the multitrajectory runs grade worse than stochastic. The greater fine scale variability toward the center of the figure is a side effect of the multitrajectory technique together with the smoothness of the distribution and the relatively high resolution of the MOE plot. In the multitrajectory case, trajectories toward the center of the distribution tend to be fewer but of greater weight, and toward the edges more numerous but of smaller weight. This gives more variability than in the stochastic case, where each outcome has the same weight, but more are toward the center, and thus more uniformly distributed.. In variance reduction, the greater importance of points away from the center would be expected to yield a benefit to the multitrajectory technique which is absent when using the analysis based on average histogram difference.

A possible perspective is that the analysis method attempts to use too high a resolution for the given numbers of runs. An analysis was made with 25 by 25 histograms instead of 100 by 100. Figure 17 shows the 5M stochastic reference, and Figures 18 and 19 the 10K cases. Table 2 gives histogram comparison results. The Multitrajectory technique does not do much better, except at 100K, a size well beyond what was considered earlier, and only for the smoothed data. This is not thought to be very significant.

As part of an investigation into the unexpected poorer multitrajectory performance, we chose policies to resolve some events in multitrajectory fashion, and others deterministically. The most interesting of these cases shows how the simulation and scenario perform-

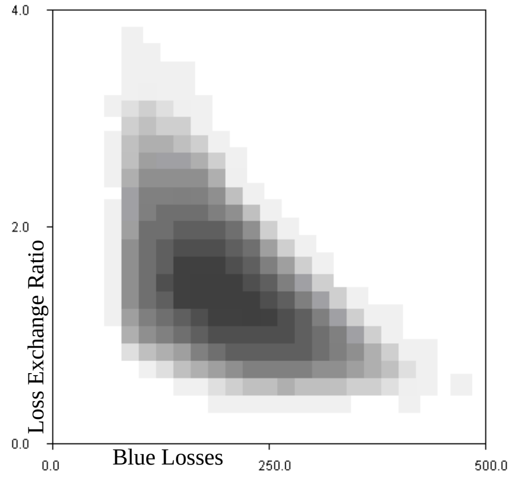


Figure 17: 2 Division Scenario, 5M Stochastic Plot, 25 x 25 Resolution

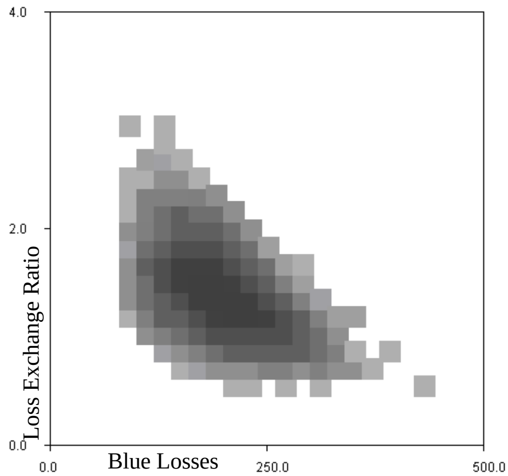


Figure 18: 2 Division Scenario, 5M Stochastic Plot, 25 x 25 Resolution

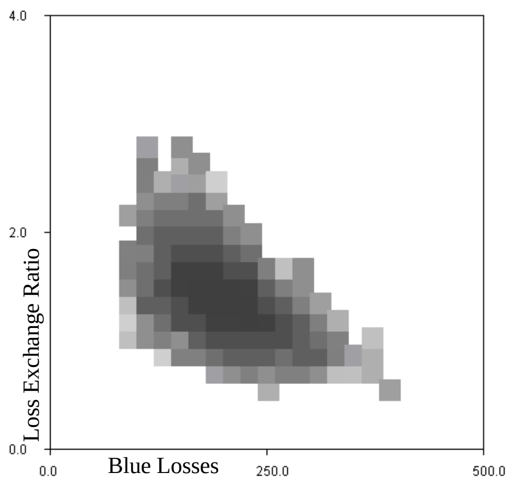


Figure 19: 2 Division Scenario, 5M Multitrajectory Plot, 25 x 25 Resolution

Table 2: Distance Results from 5M Stochastic, for 2 Division Scenario, 25 x 25 Plots, Smoothed

| | Stochastic | Multitrajectory 4 |
|---|---|---|
| 1000 | 2.61E-05 | 1.80E-04 |
| 5000 | 1.88E-05 | 6.81E-05 |
| 10000 | 6.89E-06 | 1.93E-05 |
| 20000 | 5.44E-06 | 1.07E-05 |
| 100000 | 6.74E-06 | 2.90E-06 |

with deterministic Decisionmaking, but other events stochastic or multitrajectory. Figure 20 shows a 2 million trajectory stochastic run (the largest we have for this case) used as a reference. Figures 21 and 22 show the plots for 10K runs. Table 3 gives numerical results.

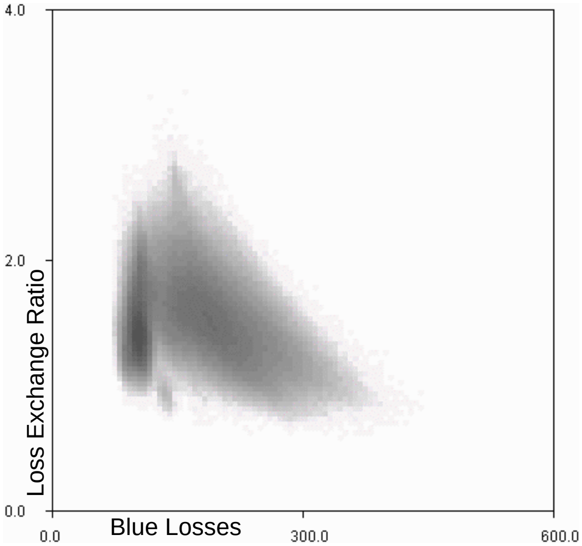


Figure 20: 2 Division Scenario, with Deterministic C2, 2M Stochastic

The response surface is no longer so simple; serious medium scale nonmonotonicity has been introduced. We still see small scale nonmonotonicity that can be attributed to random variations. Given smoothing, the multitrajectory algorithm does better, by almost a factor of two. Without smoothing, it does not do as well. The poorer performance of the multitrajectory algorithm seen earlier may have something to do with the decisionmaking events, which more recent analysis show are the most important. Even though the stochastic results seem to be smoother, the finer grained features are more recognizable in the multitrajectory plot. For example, the isolated island to the bottom left can be seen more clearly in Figure 22 than in Figure 21. Figure 22 allows two separate "tails" toward the bottom right to be distinguished, and gives a remarkable extreme at the top missing in Figures 21 and 20.

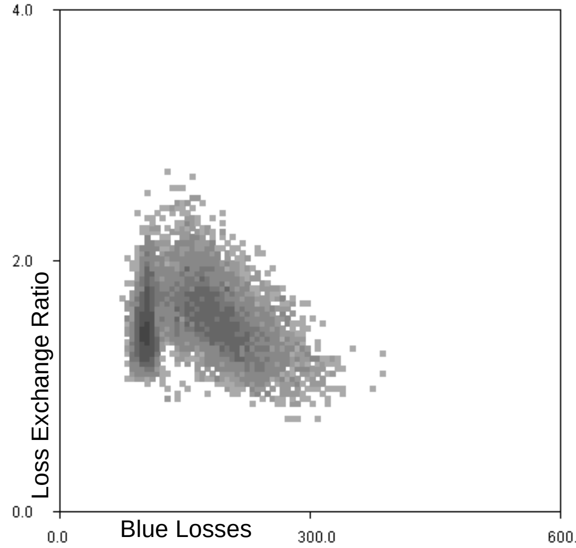


Figure 21: 2 Division Scenario, with Deterministic C2, 10K Stochastic

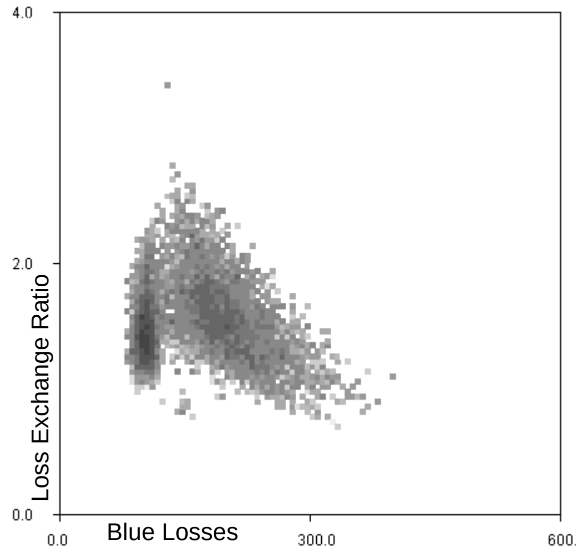


Figure 22: 2 Division Scenario, with Deterministic C2, 10K Multitrajectory

Table 3: Distances from 2M Stochastic Plot: for 10K Runs with Deterministic Decisionmaking

| | Stochastic | Multitrajectory 4 |
|---|---|---|
| unsmoothed | 2.38E-05 | 3.90E-05 |
| smoothed | 3.02E-06 | 1.62E-06 |

This excursion is of particular interest because of its irregularity. Recognition of such multimodal behavior is expected to be of particular interest to an analyst. If the multitrajectory technique does better in such cases, it may be justified even if it does worse on histograms that are roughly Gaussian in shape.

## 5 RESULTS: LARGER SCENARIOS

For the four division scenario, Table 4 gives the distance figures, and Figures 23 to 25 show the definitive 5M replication plot, and the 10K Stochastic and Multitrajectory MOE Plots. For this comparison, an 11 x 11 smooth was used. The Multitrajectory algorithm performs significantly better. For unsmoothed data, the Multitrajectory algorithm does slightly worse. The characteristic fine grained variability in the Multitrajectory runs that leads to this difference can be seen in these and other plots.

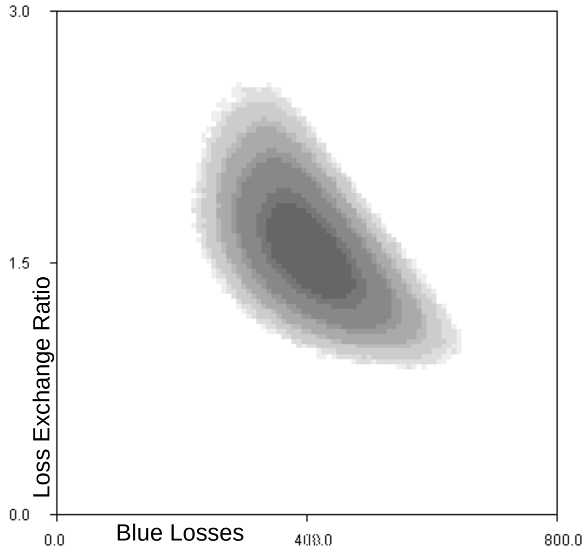

Figure 23: 4 Division Scenario 5M Stochastic Plot

Table 4: Four Division Scenario Results (smoothed)

| Replications | Stochastic | MT policy 4 |
|---|---|---|
| 1000 | 1.77E-05 | 1.78E-05 |
| 5000 | 1.90E-05 | 7.93E-06 |
| 8000 | 1.95E-05 | 5.94E-06 |
| 10000 | 1.95E-05 | 5.02E-06 |
| 50000 | 7.57E-06 | |
| 100000 | 3.15E-06 | |
| 500000 | 9.60E-07 | |
| 1000000 | 6.72E-07 | |

For the eight division scenario, Table 5 gives the results, also for smoothed data. MOE Plots for the 10K case are shown in Figures 26 and 27. For the 4 division smoothed case, it would take something over 100K stochastic replications to give the same distance as 10K multitrajectory replications. For the eight division scenario, even the 1M stochastic run did not have a distance as low as that of the 10K Multitrajectory run.

Sixteen division and thirty two division scenarios were run. We do not have large numbers of replications necessary to established the comparisons for the 32 division scenario. Table 6 gives MOE comparisons for the sixteen division case. Multitrajectory runs were in groups of 250. Plots are shown in Figures 28 and 29.

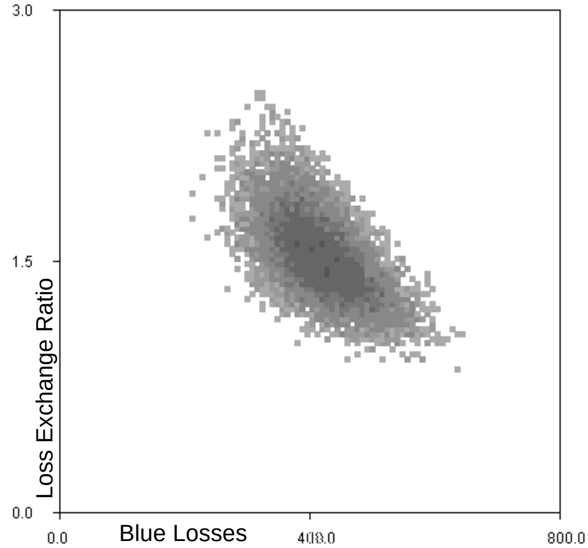

Figure 24: 4 Division Scenario 10K Stochastic Plot

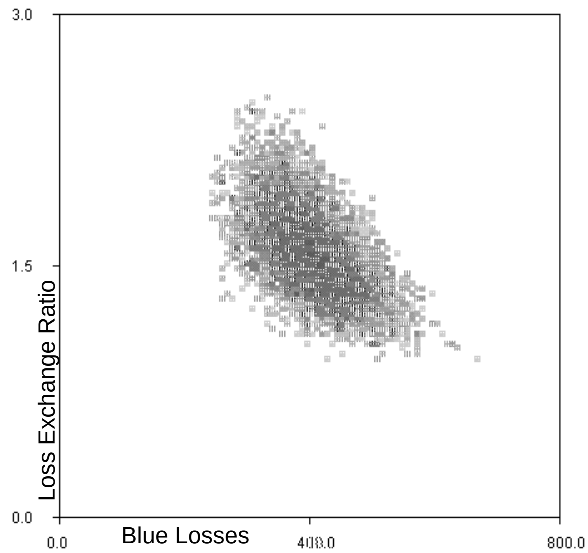

Figure 25: 4 Div. Scenario 10K Multitrajectory Plot

Table 5: Eight Division Scenario Results:

| Replications | Stochastic | MT policy 4 |
|---|---|---|
| 1000 | 2.58E-05 | 2.71E-05 |
| 5000 | 2.61E-05 | 1.23E-05 |
| 10000 | 2.60E-05 | 1.23E-05 |
| 50000 | 2.54E-05 | |
| 100000 | 2.54E-05 | |
| 500000 | 2.57E-05 | |
| 1000000 | 2.56E-05 | |

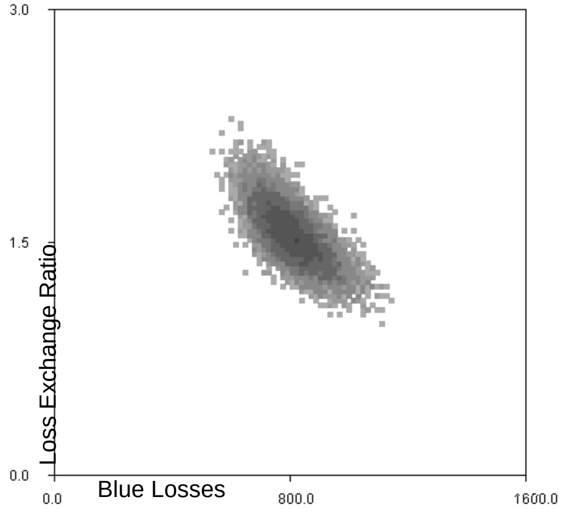


Figure 26: 8 Division Scenario, 10K Stochastic Plot

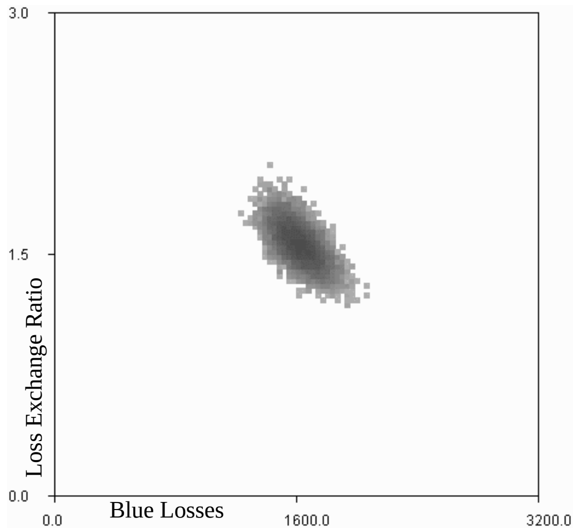


Figure 28: 16 Div. Scenario, 10K Stochastic Plot

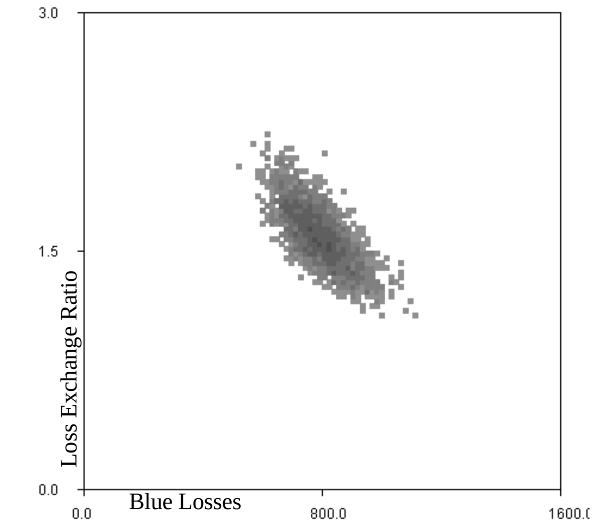


Figure 27: 8 Division Scenario, 10K Multitrajectory Plot

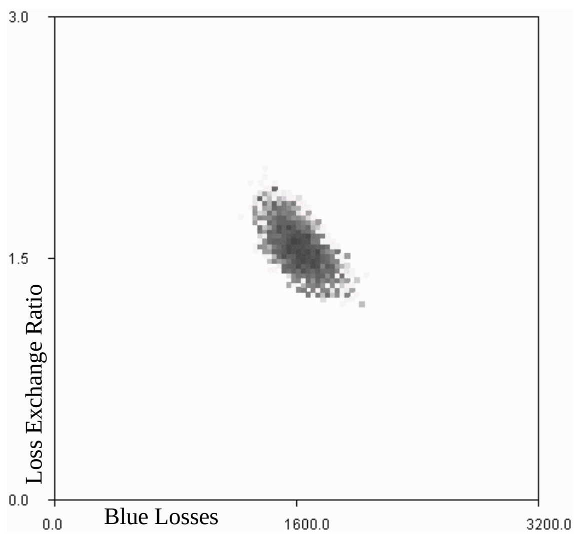


Figure 29: 16 Div. Scenario, 10K Multitrajectory Plot

Table 6: 16 Division Scenario Results

| | Stochastic | Multitrajectory |
|---|---|---|
| 1000 | 4.75E-06 | 1.46E-05 |
| 5000 | 1.46E-06 | 5.72E-06 |
| 10000 | 1.39E-06 | 6.72E-06 |
| 50000 | 3.16E-07 | |
| 100000 | 4.46E-07 | |
| 500000 | 9.75E-08 | |

## 6 OBSERVATIONS

The results obtained are unsatisfying, as the two division scenario, which seemingly should do best, shows better performance for the stochastic case than for multitrajectory. This extends even to simple statistics such as average blue losses. The figures are 198.00 for the 5M reference run, 198.61 for the 10K Multitrajectory run, and 197.75 for the 10K stochastic run, with other statistics also, on the average, being better for the stochastic case. The methodology and the simulation and analysis software have been checked extensively. There may yet be some sort of error that results in the multitrajectory and stochastic outcome sets being different, and thus explaining the unexpected two division results, but extensive and ongoing consistency checks have failed to uncover it.

If the results are taken at face value, then they seem to indicate that increasing scenario sizes make the Multitrajectory approach more advantageous up to a point, and then less advantageous. The smallest (2 division) and largest (16 division) scenarios showed a disadvantage. For the 4 division scenario, the Multitrajectory approach had a significant advantage. By interpolation, we would expect that a 1200 replication stochastic run would match the 1000 replication Multitrajectory run set in terms of achieving the same distance to the definitive outcome set. For the 8 division scenario, even 10,000 stochastic replications gives a greater distance than the 1000 replication Multitrajectory run. The rate of convergence is so slow that the number of stochastic replications needed for equivalence is huge.

The simple trajectory management used to generate these data results in the earliest events being given the preferred (multitrajectory) treatment, with later events resolved stochastically. Early events are mostly movement selection. A recent event importance analysis shows that movement events are the least important. Decisionmaking events, which tend to come late, are the most important, and are typically at least 10 times as important as movement events. These importance figures are the average distance in the (scaled) state vectors between trajectories having opposite outcomes for a given event, for the simplest (4 unit) scenario. If this relative importance is generally true, then the event management should be different, giving priority for multitrajectory mode to decisionmaking events. This will be examined in further work currently under way.

One unexpected result is the overall smoothness of the MOE plots. There is little evidence of the nonmonotonicity. It may be that the nonmonotonicity is generally due to a failure to use stochastic models for processes which, in the real world, are random. In the two division case, significant nonmonotonicity emerges only when one of the most important processes, decisionmaking, is resolved in deterministic fashion. This particular issue deserves additional study.

## 7 CONCLUSIONS

For some cases, the Multitrajectory simulation technique seems to have clear advantages over traditional stochastic simulation, requiring as much as an order of magnitude fewer runs to generate an equivalent quality histogram. However, in other cases the Multitrajectory technique does less well. In the two division case in which it was expected to do best, it failed to achieve the performance of the stochastic approach. These results and statistical behavior seem inconsistent with what we should expect from this technique. The analysis methodology, based on closeness of approach to a reference histogram, may be biased in favor of the stochastic technique. As a variation on the theme of variance reduction, the Multitrajectory technique thus cannot at this time be shown to offer a clear quantitative advantage. The multitrajectory technique does seem to be somewhat better at presenting structural features of the MOE plots which are more likely to be lost in stochastic plots. This is more applicable to simulations and scenarios which are nonmonotonic. It is thus unfortunate that the scenarios used to drive this study turned out, unexpectedly, to be so well behaved. How pervasive this kind of well behaved, almost Gaussian, performance is for a wider range of scenarios would seem worthy of a study independent of multitrajectory issues.

## ACKNOWLEDGMENTS

This project continues research that was funded by the US Army Research Office under Grants DAAH04-95-1-0350 and DAAG55-97-1-0360, with the sponsorship of the US Army Concepts Analysis Agency. Mr. Gerry Cooper, Col. Andrew Loerch, and Dr. Robert Alexander have been very helpful in advice and assistance.

## REFERENCES

Al-Hassan, Sadeq, John B. Gilmer Jr., and Frederick J. Sullivan, 1997. A Simulation State Management Technique Sensitive to Measures of Effectiveness. In *Proceedings of the 1997 Military, Government, and Aerospace Simulation Conference*, ed. Michael J. Chinni, 95-100. Society for Computer Simulation, San Diego, California.

Gilmer, John B. Jr., and Frederick 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. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Further data and MOE plots, working papers, and other documents, and other simulation / analysis screen shots, can be found at http://calvin.mathcs.wilkes.edu/mts.

## 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.

**FREDERICK J. SULLIVAN** teaches Computer Science at Wilkes University, and earlier did so at Rose-Hulman and SUNY Binghamton. His expertise is in operating systems and object oriented software. His Ph.D. is in Mathematics, from LSU.