STOCHASTIC LANCHESTER AIR-TO-AIR CAMPAIGN MODEL

MODEL DESCRIPTION AND USERS GUIDES—2009

REPORT PA702T1

Robert V. Hemm Jr.
David A. Lee
Jeremy M. Eckhause
John A. Dukovich

LMI
GOVERNMENT CONSULTING

JANUARY 2009

LMI

Stochastic Lanchester Air-to-Air Campaign Model:
Model Description and Users Guides—2009

PA702T1/JANUARY 2009

Executive Summary

This report documents the latest version of the Stochastic Lanchester Air-to-Air Campaign Model (SLAACM), developed by LMI for the Tactical Air Division, a component of the Office of the Secretary of Defense, Program Analysis and Evaluation (OSD PA&E). During the past year, optimized offensive air campaigns, including suppression of enemy air defense and the impact of electronic warfare and low observable technology, have been added to the basic air defense SLAACM to produce a new "attack" version that includes both defense and offense scenarios. The "classic" air defense SLAACM has been retained as a separate version.

SLAACM is a fast and flexible model designed for analysts who need to consider many combat scenario options quickly and who wish to have indications of the uncertainties in military outcomes. SLAACM models both the defensive and offensive counter-air battle, including order of battle optimization by both attackers and defenders. The current version can address issues of fighter combat effectiveness, defense battle management capabilities, time-phasing of offensive and defensive forces, use of indigenous defensive forces, and basing options for deployed forces. Unique, efficient mathematical algorithms and fast integer programming tools allow SLAACM to generate optimized results, including statistical uncertainties for numerous types and large numbers of combatants for campaigns of many days, with sufficient speed to allow statistical variation of campaign scenarios and deployment options. Included tables and charts display the day-by-day development of the campaign. In addition, the Microsoft Excel, Visual Basic implementation of the model allows wide flexibility for storing and displaying results in user-preferred formats.

This report includes descriptions of the fundamental mathematics and analytical structure of SLAACM and users guides both the classic and attack versions.

The authors wish to thank Mr. Bradley Berkson, the director of OSD PA&E, for his support of the development of SLAACM.

Contents

Figures

Tables

Chapter 1
Introduction

The Stochastic Lanchester Air-to-Air Campaign Model (SLAACM) was developed by LMI for the DoD Office of the Director, Program Analysis and Evaluation, Tactical Air Division (TACAIR). This parsimonious probabilistic air engagement and optimized campaign model is designed to be a flexible tool for analysts who need to consider many cases quickly in the PC environment and who wish to have indications of the uncertainties in military outcomes. SLAACM combines stochastic Lanchester engagement modeling with integer programming campaign optimization.

SLAACM is an analytic, probabilistic model, as compared to a Monte Carlo simulation, and provides complete and repeatable traceability between input and output variables. The stochastic Lanchester engagement method used in SLAACM was developed to address shortcomings in deterministic Lanchester analysis when modeling campaigns involving multiple few-on-few engagements of heterogeneous forces, i.e., involving many flights of many types of aircraft. Stochastic Lanchester analysis can directly model large engagements consisting of multiple few-on-few engagements and provides full statistics of the outcomes. LMI has developed a unique, high-speed algorithm for engagement calculations that is key to practical stochastic Lanchester campaign analysis.

SLAACM is implemented in Microsoft Excel and Visual Basic for Applications (VBA) and typically runs a fully optimized 10-day campaign on a standard PC in less than 2 minutes. SLAACM's speed and flexibility are due to use of the fast engagement analysis algorithm, the capability of modern PCs and the use of either a fast heuristic, or a fast commercial IP solver for optimization.

SLAACM outputs in benchmarking tests compare closely with those of some large-scale simulation models, but SLAACM is intended to complement, rather than replace, such models. The basic SLAACM is unclassified. Analysis of classified scenarios is performed by loading the SLAACM on classified computers and running it with classified weapon system and scenario input data. The defensive air campaign capabilities of SLAACM been developed in a boot-strap fashion over the past 8 years, adding features in response to analysis requirements. The "classic" defensive air campaign version of SLAACM has been used to model and analyze several large defensive campaign scenarios of interest to DoD.

During the past year, we have expanded SLAACM to model offensive air campaigns. We have expanded the offense to include typical U.S. attack packages, including suppression of enemy air defense (SEAD) escort aircraft. We have added surface-to-air missiles (SAMs) to the defense. The model expansion has been accomplished through a major transformation of the VBA code structure from procedural to object-oriented, allowing straightforward inclusion of new offense and defense packages. The new structure supports the ultimate goal of modeling an integrated, optimized, campaign using both offensive and defensive air power in order to inform acquisition, deployment, and employment decisions.

This report describes the analytical and statistical methods used to generate SLAACM outputs, including the new offensive capabilities. It also gives detailed instructions for operating SLAACM, with illustrative examples.

SLAACM OVERVIEW

SLAACM treats campaigns between two opponents, called Attack and Defense.[1] The Attack side uses fighter-escorted bombers to attack assets defended by the Defense side's fighters and SAMs. In each day's operations, the Attack side determines optimal allocations of its fighters and bombers to attack packages, maximizing a payoff function that considers the bombers' payload and weapon-delivery accuracy, the value to Attack of destroying Defense fighters, and the penalty to Attack of losing fighters and bombers.

Defense's fighters respond. Each Defense fighter can be "smart" or not. Smart fighters can determine the makeup of oncoming attack packages before engaging and can coordinate their operations to make optimal defenses, maximizing a payoff function that considers the value to Defense of destroying Attack fighters and bombers, and the penalty to Defense of losing a fighter.

Defense fighters that are not smart encounter attack packages randomly. An adjustable parameter varies the effectiveness of Defense's battle management for not-smart fighters, reducing the probability of engagements below that implied simply by the numbers of attacking packages and defending flights.

SAMs are included in the new Attack SLAACM scenarios, but not in the Air Defense SLAACM scenario. Some Attack SLAACM scenarios include SEAD aircraft to suppress the SAMs.

In Attack SLAACM scenarios, each attack package engages one target, which is defended by one SAM site. If not destroyed by SEAD aircraft, the SAM site is capable of firing a user-input number N_m of missiles at the bombers. SAMs engage SEAD aircraft and bombers, but not attacker escort fighters. Defending fighters engage attacker escorts and bombers, but not SEAD aircraft.

[1] In previous SLAACM documentation, friendly forces are identified as Blue and Green, and enemy forces are Red. With the ability of friendly forces to defend or attack, we have changed the nomenclature in this report to "Attack" and "Defense." The previous version has been called "Red SLAACM," "Air Defense SLAACM," and "Classic SLAACM," and the new SLAACM version incorporating attack has been called "Blue SLAACM" and "Attack SLAACM." Despite the names, the new version includes both defensive and offensive scenarios and is a superset of the older version.

SLAACM does not use iterative simulation. Rather, it calculates the outcome probabilities using analytic probability algorithms, including the full loss statistics of each day's combat. The mean losses are propagated and used, along with any replenishment aircraft, to determine Attack and Defense orders of battle (OOBs) for the next day's operations.[2]

SLAACM tracks the destructive effectiveness of the bombs delivered, the total tons of bombs delivered, as well as the tons of smart bombs (guided munitions with higher destructive capability per ton). SLAACM also tracks which defense aircraft killed which attack aircraft and vice versa, and standard deviations of Defense and Attack losses.

SLAACM's worksheets and analytic structure provide complete audit trails for the input parameters used in the calculations. The kill-rate ratios used in the calculations may be obtained in several ways. The kill-rate ratios may be derived from probabilities of detection and kill ($P_D - P_k$ probabilities) of the combinations of combatants involved. They may also be derived from loss ratios found from simulations, including man-in-loop simulations, or from data on actual air combat, or from combinations of all these sources. This flexibility of inputs is an important strength of SLAACM.

Obtaining the kill-rate ratios from simulation results gives SLAACM results a basis of consistency with the simulations. (Again, we point out that SLAACM is intended to complement, not supplement, large-scale simulations.) Because many SLAACM exercises have been based on loss ratios, the input loss ratios are displayed in the model. In classified versions of SLAACM, the loss ratios are based on documented historical data, simulations, and military expertise. The analytic nature of the model ensures that the effects of parameter changes are consistently and individually reflected in the output and are not convolved with other parameter effects as happens in iterative simulations. These features are quite helpful to understanding the implications of different scenario options and weapons parameters.

REPORT ORGANIZATION

This report is organized as follows:

- Chapters 2 through 5 describe how SLAACM works. Specifically, those chapters, respectively, describe how the model is used to
  - calculate air-to-air, SEAD, and SAM engagement results,
  - calculate bomber effectiveness,

[2] The term "day" here refers to a single set of campaign sorties. There could be more than one such set on a single calendar day, or there could be calendar day gaps between sets.

- specify the Attack and Defense payoff logic and calculate optimal attacks and defenses, and
- calculate campaign results.

- Chapters 6 and 7 are users guides for operating Classic SLAACM and Attack SLAACM, respectively.

The appendixes contain supplemental information of interest for combat modeling:

- Appendix A discusses the mathematics of the fast engagement analysis algorithm.
- Appendix B addresses the use of NASA-developed Markov tools for auxiliary engagement analysis and scenario development.

Chapter 2
How SLAACM Works: Engagement Models and Calculations

SLAACM does not simulate; rather, it makes complete calculations of outcome probability distributions for specific probabilistic engagement models. That approach is significantly different than the approach used in other campaign models, so it is appropriate to begin our discussion by explaining the SLAACM approach.

Basic M vs. N Probabilistic Engagement Model

The heart of SLAACM's calculations is a set of probabilistic models of air-to-air engagements. The simplest engagement model used in SLAACM is the M vs. N probabilistic engagement model. We describe that model in considerable detail, to clarify some fundamental aspects of SLAACM's operation.

In previous reports, defenders were friendly and designated as Blue or Green aircraft, while attackers were hostile and identified as Red aircraft. With the modeling of offensive operations by friendly forces, this nomenclature no longer applies; instead, whenever we deal with specific attack/defense scenarios, we will refer to Attack and Defense aircraft. However, for discussions of basic engagement modeling, we retain the Red and Blue nomenclature.

In the basic M vs. N probabilistic model, M Blue aircraft engage N Red aircraft. Throughout the engagement, the time between kills by each Blue aircraft is taken to be identically and independently distributed exponentially with parameter k_b, and the time between kills by each Red aircraft is assumed to be identically and independently distributed exponentially with parameter k_r. Specifically,

$$\tau_r \sim k_r e^{-k_r \tau_r} \quad \text{[Eq. 2-1]}$$

and

$$\tau_b \sim k_b e^{-k_b \tau_b} \quad \text{[Eq. 2-2]}$$

where "~" means "distributed as" and τ_r and τ_b are, respectively, the time between kills made by each Red aircraft and the time between kills made by each Blue aircraft.

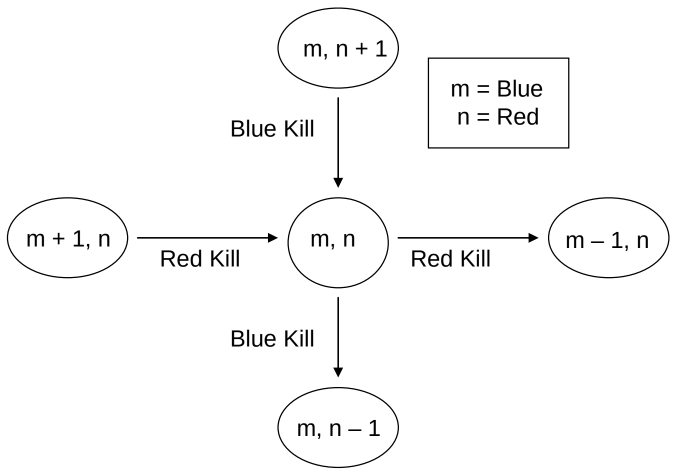
If the time between kills by an engaged aircraft is exponentially distributed with parameter k, then the probability that the aircraft makes a kill in a time interval

$(t, t + \delta t)$ is $k\delta t + O(\delta t^2)$,[1] independent of total elapsed time, t. (This property is sometimes called the "memoryless" property of exponential distributions.)

If two such aircraft are present, then by virtue of the assumption that their times between kills are statistically independent, the probability that the pair makes a kill in the interval $(t, t + \delta t)$ is $k\delta t + k\delta t + O(\delta t^2) = 2k\delta t + O(\delta t^2)$. If m such aircraft are present, the probability that the set of them makes a kill in the interval $(t, t + \delta t)$ is $mk\delta t + O(\delta t^2)$.

If, at a given time during the engagement, M Blue aircraft and N Red aircraft are present, we will say that the engagement is in state (m, n). Figure 2-1 diagrams the possible transitions that the engagement can make into and out of state (m, n).

Figure 2-1. Engagement State Transition Diagram



Now let $P_{m,n}(t)$ be the probability that the engagement is in state (m, n) at time t. Let us consider the probability $P_{m,n}(t + \delta t)$. The engagement can reach state (m, n) at time $(t + \delta t)$ in just three ways:

- The engagement was in state (m, n) at time t, and no kills happened during the interval $(t, t + \delta t)$.
- The engagement was in state (m, n + 1) at time t, and the set of Blue aircraft made a kill during the interval $(t, t + \delta t)$.
- The engagement was in state (m + 1, n) at time t, and the set of Red aircraft made a kill during the interval $(t, t + \delta t)$.

[1] $O(\delta t^2)$ signifies "higher order terms" of order δt^2 and higher, which are very small and can be ignored.

By the assumptions that all the kill events are independent, these three events are statistically independent. Accordingly, the probability that one of them occurs is the sum of their individual probabilities, so that, neglecting terms of $O(\delta t^2)$,

$$P_{m,n}(t+\delta t) = P_{m,n}(t)\left[1 - mk_b\delta t - nk_r\delta t\right] + P_{m,n+1}(t)\left[\left[mk_b\delta t + P_{m+1,n}(t)nk_r\delta t\right]\right]. \quad \text{[Eq. 2-3]}$$

Subtracting $P_{m,n}(t)$ from each side of Equation 2-3, dividing by δt, and taking the limit as $\delta t \to 0$, gives

$$\dot{P}_{m,n}(t) = -(mk_b + nk_r)P_{m,n} + mk_b P_{m,n+1} + nk_r P_{m+1,n}. \quad \text{[Eq. 2-4]}$$

With the result (Equation 2-4), we can write down a general initial value problem describing the evolution of the statistics of an M vs. N engagement, in which the Blue forces break away when they have fewer than b_{min} aircraft, and the Red forces break away when they have fewer than r_{min} aircraft:

$$\begin{aligned}
&\dot{P}_{m,n} = -(mk_b + nk_r)P_{m,n} + mk_b P_{m,n+1} + nk_r P_{m+1,n}, \\
&\qquad \forall\, b_{min} \le m \le M \text{ and } r_{min} \le n \le N; \\
&\dot{P}_{0,n} = nk_r P_{b\min,n},\ \ r_{min} \le n \le N \\
&\dot{P}_{m,0} = mk_b P_{m,r\min},\ \ b_{min} \le m \le M \\
&P_{m,n}(0) \equiv 0 \text{ except } P_{M,N}(0) = 1
\end{aligned} \quad \text{[Eq. 2-5]}$$

(The probabilities that the system has more than M Blue aircraft, or more than N Red aircraft, are of course always zero.)

The initial value problem (Equation 2-5) presents a system of linear ordinary differential equations. The number of equations is equal to $(M - b_{min} + 1)(N - r_{min} + 1) + (M - b_{min} + 1) + (N - r_{min} + 1)$. The equations for the probabilities of the absorbing boundary states, $P_{0,n}$ and $P_{m,0}$, decouple from those for the transient states (those whose probabilities tend to zero for long times), however. Therefore, Equation 2-5 may be treated by solving only the $(M - b_{min} + 1)(N - r_{min} + 1)$ equations for the transient states and computing the probabilities of the absorbing boundary states by integration.

In any case, Equation 2-5 offers no difficulties to numerical solution. Moreover, introducing the nondimensional time parameter $\tau \equiv k_r t$ allows us to write Equation 2-5 in a form that involves only the nondimensional parameter Kill-Rate Ratio = KRR = $\kappa = k_b/k_r$:

$$\begin{aligned}
&\dot{P}_{m,n} = -(m\kappa + n)P_{m,n} + m\kappa P_{m,n+1} + nP_{m+1,n}, \\
&\qquad \forall\, b_{min} \le m \le M \text{ and } r_{min} \le n \le N; \\
&\dot{P}_{0,n} = nP_{bmin,n}, \quad r_{min} \le n \le N \\
&\dot{P}_{m,0} = m\kappa P_{m,rmin}, \quad b_{min} \le m \le M \\
&P_{m,n}(0) \equiv 0 \text{ except } P_{M,N}(0) = 1
\end{aligned} \qquad \text{[Eq. 2-6]}$$

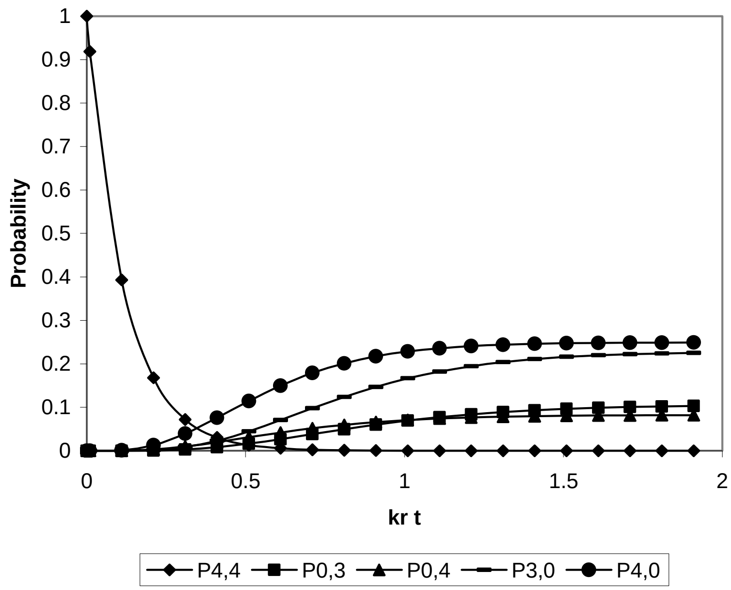
The state probabilities in Equation 2-6 are functions of τ, not t, and so are different functions from those giving the state probabilities as functions of t. Also the derivatives in Equation 2-6 are with respect to τ, not t. In the interest of uncluttered notation, we will not introduce new symbols for these new functions; rather, in Equation 2-6 and in all following parts of this chapter, the probabilities are functions of τ, and dots denote differentiation with respect to τ.

Writing the initial value problem for the M vs. N probabilistic engagement in nondimensional form gives useful insights into the nature of the problem's solutions. First, because the only parameter appearing in Equation 2-6 is κ, we see that the probabilities $P_{m,n}$, which in general depend on time t and the two parameters k_r and k_b, can always be written as functions of the nondimensional time τ and just one parameter, the kill-rate ratio κ.

Second, the fact that the solutions of the initial value problem depend only on κ and τ leads to the conclusion that the limiting values of the probabilities as time tends to infinity (which of course is also the limit as τ tends to infinity) are functions only of the single parameter κ.

Figure 2-2 illustrates these ideas. It shows the evolution of certain probabilities, obtained by numerical integration for a 4 vs. 4 engagement with $k_b/k_r \equiv \kappa = 1.46$, and in which the combatants fight to annihilation.

Figure 2-2. Evolution of State Probabilities



To find the value of one of the probabilities shown in Figure 2-2 at a given time t for a given engagement, one would find the value of τ corresponding to that t by computing $\tau = k_a t$ and reading the desired value from the figure. For example, at t = 20 time units in an engagement in which the mean time for the Red aircraft to make a kill is 180 time units, the value of τ is 20/180 ≈ 0.11; thus the desired value of $P_{4,4}$ is 0.4.

We will be more interested in the long-time limiting values of the probabilities. Figure 2-2 indicates that for $\tau > 2$, the absorbing boundary states have essentially reached their long-time limiting values. The numerical solution confirms that for $\tau > 2$, the total probability of the absorbing boundary states is larger than 95 percent. The physical time required to reach the long-time limit will of course vary with the value of the dimensional parameter k_a. But the values of the probabilities depend only on the single, nondimensional parameter κ.

Relating Loss Data to Kill-Rate Ratios

The engagement models underlying SLAACM, like all Lanchester-type models, employ kill rates for the opposing forces. In deterministic Lanchester models, the parameters are deterministic kill rates, i.e., kills per firer per unit time. In stochastic Lanchester models like SLAACM, the parameters are the reciprocals of mean times between kills by a single firer. For simplicity, we still refer to parameters of stochastic Lanchester models as “kill rates.”

Observed engagement data, from simulations or from observations of actual combat, are rarely, if ever, rate data. Rather, they are typically loss ratios for specific M vs. N engagements. To apply empirical loss ratio data to rate-based models, we need a method to convert from loss ratio data to kill-rate ratio data. To do this, we make the significant assumption that the long-time limits of SLAACM's engagement models are appropriate models for the observations.

Thus, with the basic M vs. N engagement model, we assume that the observed engagements end only with one side's reaching its specific breakaway condition—a certain number of losses, or annihilation—rather than with a clock-time-dependent condition such as fuel exhaustion, or an event-dependent condition such as exhaustion of munitions. More advanced engaged models, described in Appendix B, do account for missile use.

As noted above, the nondimensional kill-rate ratio parameter, κ, is the single parameter of interest in the basic engagement model. To infer a value of κ from loss ratio data for an M vs. N engagement, we find the value of κ for which the long-time limiting value of the ratio of expected Blue loss to expected Red loss in the M vs. N probabilistic model that models the engagement is equal to the observed loss ratio. In this way, we infer a nondimensional quantity, κ, from a nondimensional quantity, the observed loss ratio.

Suppose, for example, that the available data are for a 2 vs. 2 engagement to annihilation and that they show a loss ratio of 1 Blue to 10 Reds. Calculations show that, for the basic 2 vs. 2 probabilistic engagement model, a kill-rate ratio κ of 8.28 makes the ratio of expected Blue loss to expected Red loss equal to 1:10.[2]

Although we usually use a finite-step iterative method to compute long-time limits directly, it is of course possible to find long-time limits with time-based methods, such as numerical integration. In time-based Markov calculations, the time required to reach the long-time limit will depend on the value chosen for k_r, where k_b is equal to κ times k_r. For example, analysis shows that the kill-rate ratio of 8.28 results in a 1:10 loss ratio in approximately 0.22 time units for $k_r = 1$, in roughly 2.2 time units iterations for $k_r = 0.1$, and in about 22 time units for $k_r = 0.01$. The 0.2, 2, and 22 time units represent adequate model time to bring the solution to the long-time limit, for their corresponding kill rates. Thus, in a time-unit-based model, selection of the individual rates is arbitrary as long as the ratio κ is preserved and the calculations continue to a time on the order of twice the inverse of the sum of the parameters k_b and k_r. (Appendix A describes the general analytic method that produces these numerical results.)

Another major assumption in our engagement modeling is that the kill-rate ratio is independent of the Blue:Red aircraft ratio. We assume that kill-rate ratios derived from data on 4 vs. 8 engagements can be applied directly to 4 vs. 4 and other engagement combinations. A corollary assumption is that kill-rate ratios are independent of the state of an engagement. These are significant and powerful

[2] The kill-rate ratio reflects the performance of individual Blue vs. Red pairs and is numerically equal to the loss ratio for a 1 vs. 1 engagement. A 10:1 kill-rate ratio would be required to produce a 1:10 loss ratio for a 1 vs. 1 engagement, but only an 8.28:1 kill-rate ratio is necessary for the same loss ratio in a 2 vs. 2 engagement, because all available aircraft are contributing to each kill.

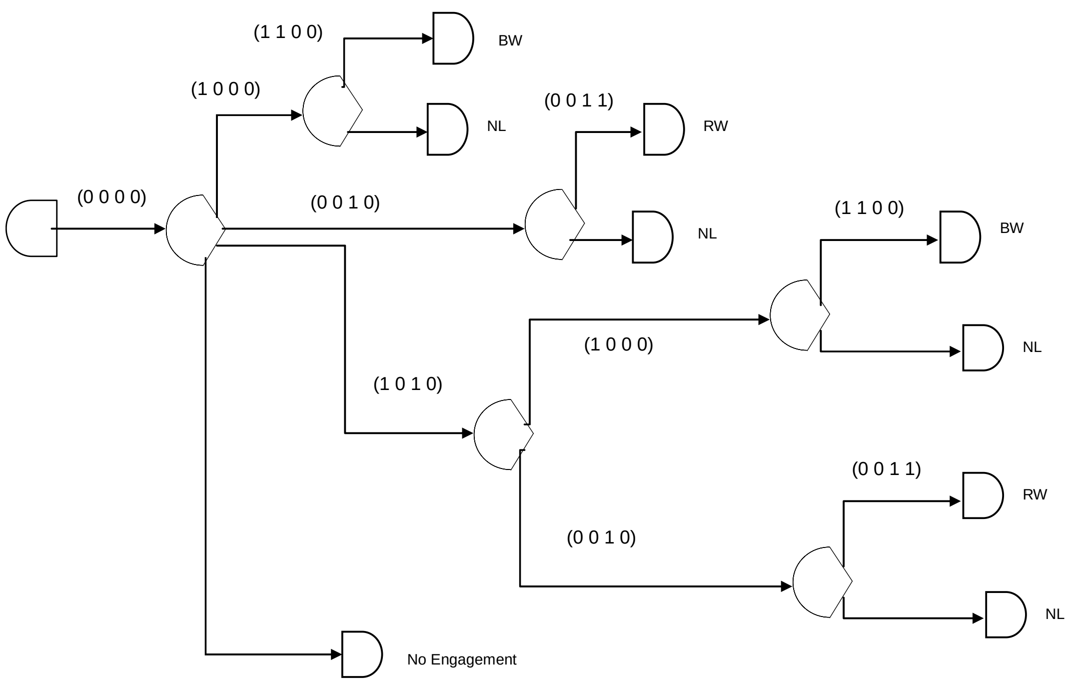
simplifying assumptions, which we are always eager to test with new combat data.

Available data from TAC BRAWLER engagement simulations give expected values of Red losses and Blue losses. These data, some of them classified, are available for several Red and Blue aircraft types of interest. Government subject matter experts have scaled these results to model additional aircraft combinations. A SLAACM utility uses bisection iteration and the long-time limit algorithms (discussed below and in Appendix A) to calculate the values of κ from the loss ratios for each of the Red vs. Blue combinations. The calculations are repeated until successive iterations agree to four significant figures.

Relating Probabilities of Detection and Probabilities of Kill Data to Kill-Rate Ratios

Air combat performance data are often expressed in terms of probabilities of detection, Pd, and probabilities of kill, Pk. These can be used directly to calculate loss ratios and KRRs.[3] Figure 2-3 diagrams a model that can be used for this purpose.

Figure 2-3. Pd, Pk Engagement Diagram



The model described is a 1 Blue vs. 1 Red engagement in which each aircraft can detect the opponent and fire one missile. Each side has an initial probability of detection plus a separate probability of detecting the opponent after the opponent has fired his missile. Outcomes include Blue kills Red, Red kills Blue, each kill

[3] Loss ratios equal kill-rate ratios for 1 vs. 1 engagements.

the other, neither detect, and neither kill. This model can be easily programmed in Microsoft Excel to develop KRR values for SLAACM.

Table 2-1 contains the state and state transition information for the model.

Table 2-1. Pd, Pk Transition Probability Table

State transition	From	To	Transition probability
Initial detection			
Blue detects, Red does not	(0,0,0,0)	(1,0,0,0)	Pbd * Qrd
Red detects, Blue does not	(0,0,0,0)	(0,0,1,0)	Prd * Qbd
Both Red and Blue detect	(0,0,0,0)	(1,0,0,0)	0.5Pbd * Prd
		(0,0,1,0)	0.5Pbd*Prd
Neither Blue nor Red detect	(0,0,0,0)	No Eng't	Qbd * Qrd
Consequences: Blue detects, Red does not			
Blue kills Red	(1,0,0,0)	BW	Pkb
Blue misses, Red detects and kills Blue	(1,0,0,0)	RW	Qkb * Prr *Pkr
Blue misses, Red detects and misses	(1,0,0,0)	NL	Qkb * Prr * Qkr
Blue misses, Red does not detect	(1,0,0,0)	NL	Qkb * Qrr
Consequences: Red detects, Blue does not			
Red kills Blue	(0,0,1,0)	RW	Pkr
Red misses, Blue detects and kills Red	(0,0,1,0)	BW	Qkr * Prb * Pkb
Red misses, Blue detects and misses	(0,0,1,0)	NL	Qkr * Prb * Qkb
Red misses, Blue does not detect	(0,0,1,0)	NL	Qkr * Qrb
Consequences: Both Blue and Red detect			
Blue kills, Red misses	(1,0,1,0)	BW	Pkb * Qkr
Red kills, Blue misses	(1,0,1,0)	RW	Pkr * Qkb
Neither Blue nor Red kill	(1,0,1,0)	NL	Qkb * Qkb

The following definitions apply to Table 2-1:

BW = Blue win state

RW = Red win state

NL = No-loss state

P_{BW} = Probability of Blue win

P_{RW} = Probability of Red win

P_{NL} = Probability of no loss

Px = Probability of event x

Qx = (1-Px) Probability of failure of event x

Pdb = Probability that Blue detects Red

Pdr = Probability the Red detects Blue

Pkb = Probability that Blue kills Red

Pkr = Probability that Red kills Blue

Pbr = Probability that Blue detects Red after Red fires and misses

Prr = Probability that Red detects Blue after Blue fires and misses Red.

The model of Figure 2-1 and Table 2-1 avoids the unrealistic outcome of simultaneous destruction of both adversaries by immediately transitioning the "both detect" case to either the Blue detect and shoot or the Red detect and shoot case, with probability one-half for each.

Now let us generate KRR values from the results of the model. Please recall the equations for a 1 vs. 1 stochastic Lanchester equation,

$$\begin{aligned} \dot{P}_{11} &= -(kb + kr)P_{11} \\ \dot{P}_{10} &= kbP_{11} \\ \dot{P}_{01} &= krP_{11} \end{aligned} \qquad \text{Eq. 2-7]}$$

which, with the initial condition of $P_{10}(0) = P_{(01}(0) = 0$; $P_{11}(0) = 1$, leads to

$$\begin{aligned} P_{11} &= \exp(kb + kr)t \\ P_{10} &= \frac{kb}{kb + kr}\left[1 - \exp(kb + kr)t\right] \\ P_{01} &= \frac{kr}{kb + kr}\left[1 - \exp(kb + kr)t\right] \end{aligned} \qquad \text{[Eq. 2-8]}$$

These equations imply

$$P_{10} = \frac{kb}{kb + kr}\left[1 - P_{11}\right] \text{and } \; P_{01} = \frac{kr}{kb + kr}\left[1 - P_{11}\right]. \qquad \text{[Eq. 2-9]}$$

Thus for the stochastic Lanchester model, at time t the probability of no loss is P_{11}, the probability of a Blue win is P_{10}, and the probability of a Red win is P_{01}.

For the Pd−Pk model, the corresponding probabilities are P_{NL}, P_{BW}, and P_{RW}.[4] To infer KRR ≡ kb/kr, we may evaluate P_{11} as equal to P_{NL}, and then solve

$$P_{10} = P_{BW} = \frac{kb}{kb + kr}\left[1 - P_{11}\right] = \frac{KRR}{1 + KRR}\left[1 - P_{11}\right] \qquad \text{[Eq. 2-10]}$$

for KRR.

[4] One may choose to divide the Pd − Pk probabilities P_{BW}, P_{RW}, and P_{NL} by 1 − QdrQdb, taking the position that the stochastic Lanchester model presumes contact between the adversaries.

Table 2-2 gives two examples of Pd−Pk data and the resulting values of kill-rate ratio.

Table 2-2. Parameter Values and Kill-Rate Ratios

Parameter	Value, Case 1	Value, Case 2
Pbd	0.8	0.95
Prd	0.4	0.1
Pbk	0.7	0.7
Prk	0.6	0.3
Pbr	0.9	0.9
Prr	0.8	0.8
KRR	2.2	27.0

Calculation Method for the Long-Time Limit

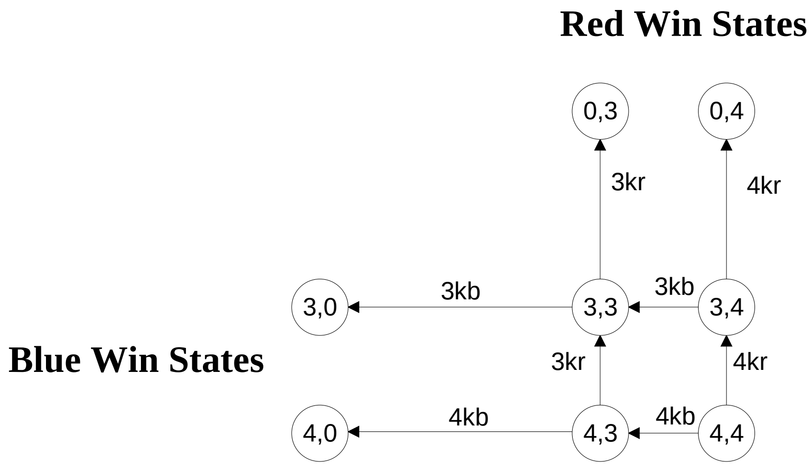
If the engagement continues to completion—that is, until the probabilities that the system is not in one of the absorbing boundary states (0, n) or (m, 0) are negligibly small—then the probabilities of these states are functions only of κ and the initial values M and N. These conditions allow significant simplification in the calculation of state probabilities. One can obtain values of the long-time limits of the probabilities of absorbing boundary states by a simple, finite-step iteration. This speeds up the calculations tremendously, compared to time-based Markov methods, and is a key to SLAACM utility.

An Example

As an example of the use of the initial value problem (Equation 2-5), we consider 4 Blue aircraft engaging 4 Red aircraft. Both sides break away on sustaining 2 losses, that is, $b_{min} = r_{min} = 3$.

Figure 2-4 illustrates the states in the engagement and the transitions between them. In that figure, the absorbing boundary states are labeled as having 0 aircraft for the side that has broken away.

Figure 2-4. Engagement Diagram



For this case, the general initial boundary value problem (Equation 2-6) takes this specific form:

$$
\begin{aligned}
\dot{P}_{4,4} &= -(4\kappa + 4)P_{4,4} \\
\dot{P}_{4,3} &= -(4\kappa + 3)P_{4,3} + 4\kappa P_{4,4} \\
\dot{P}_{3,4} &= -(3\kappa + 4)P_{4,3} + 4P_{4,4} \\
\dot{P}_{3,3} &= -(3\kappa + 3)P_{3,3} + 3P_{4,3} + 3\kappa P_{3,4} \\
\dot{P}_{4,0} &= 4\kappa P_{4,3} \\
\dot{P}_{3,0} &= 3\kappa P_{3,3} \\
\dot{P}_{0,3} &= 3P_{3,3} \\
\dot{P}_{0,4} &= 4P_{3,4} \\
P_{m,n}(0) &\equiv 0, \text{ except } P_{4,4}(0) = 1
\end{aligned}
\qquad \text{[Eq. 2-11]}
$$

where $\kappa = k_b/k_r$ and the (0) states correspond to the break condition of two aircraft.

The Red side wins the engagement if all the Blues are gone, that is, if the engagement is in a state (0, n), $r_{min} \le n \le N$. The Blue side wins if the engagement is in a state (m, 0), $b_{min} \le m \le M$.

Note that the first four equations in Equation 2-11 describe the evolution of the probabilities of transient states, whose probabilities will tend to zero at long times. The remaining four equations describe the variation of the absorbing boundary states, or outcome states. At long times, the system will be found in one of these states.

The equations for the transient states decouple from those of the outcome states. That is, the equations involve only the four transient-state probabilities. The evolution of the transient states can thus be determined independently of the outcome states.

The outcome states' evolution is, by contrast, completely determined by the transient states and the initial conditions. For example, given $P_{4,3}(t)$, the entire history of $P_{4,0}(t)$ follows from

$$\int_0^t \dot{P}_{4,0}(\tau)d\tau = P_{4,0}(t) - P_{4,0}(0) = P_{4,0}(t) = 4\kappa\int_0^t P_{4,3}(\tau)d\tau \qquad \text{[Eq. 2-12]}$$

where we have used the initial condition $P_{4,0}(0) = 0$.

Letting $t \rightarrow \infty$ in Equation 2-12 we see that the long-time limiting value of $P_{4,0}(t)$ is given by

$$\lim_{t \rightarrow \infty} P_{4,0}(t) = 4\kappa\int_0^\infty P_{4,3}(\tau)d\tau\,. \qquad \text{[Eq. 2-13]}$$

We now show how to exploit characteristics of the engagement we are considering, shown by Figure 2-4 and reflected in properties of Equation 2-11, to allow calculation of long-time limiting probabilities of outcome states with a simple, finite iterative scheme. This engagement is an example of what has been called a "pure death" process: the states always transition to states with fewer participants. Reflecting this, Figure 2-4 is an "acyclic" graph. That is, the graph gives no path by which to return to a previously occupied state.

A consequence of the acyclic character of Figure 2-4 is found in the structure of Equation 2-11. Defining the vector x as (P_{44}, P_{43}, P_{34}, P_{33}), we see that the four equations for the transient states can be written as

$$\dot{x} = Ax \qquad \text{[Eq. 2-14]}$$

where the matrix A is given by

$$A = \begin{pmatrix} -(4\kappa+4) & 0 & 0 & 0 \\ 4k & -(4\kappa+3) & 0 & 0 \\ 4 & 0 & -(3\kappa+4) & 0 \\ 0 & 3 & 3\kappa & -(3\kappa+3) \end{pmatrix}. \qquad \text{[Eq. 2-15]}$$

The acyclic character of Figure 2-4 is reflected in the lower-triangular character of the matrix A. This lower-triangular feature allows calculation of the integrals from 0 to ∞ of each of the transient-state probabilities with a simple iterative scheme. Defining the vector $\hat{x}$ by

$$\hat{x} \equiv \int_0^\infty x(t)\,dt\,, \qquad \text{[Eq. 2-16]}$$

we see, on integrating Equation 2-14 from t = 0 to t = ∞ and remembering that $P_{4,4}(0) = 1$ and all other transient-state probabilities are zero at t = 0, that

$$-e_1 = A\hat{x} \qquad \text{[Eq. 2-17]}$$

where e_1 is the vector (1, 0, 0, 0). Thus $\hat{x}$ is determined by the solution of a system of linear algebraic equations. Moreover, by virtue of the lower-triangular character of A, that system is easily solved with an obvious iterative scheme. On writing out Equation 2-17 in detail, we find

$$\begin{aligned} -1 &= -(4\kappa + 4)\hat{x}_1 \\ 0 &= 4\kappa\hat{x}_1 - (4\kappa + 3)\hat{x}_2 \\ 0 &= 4\hat{x}_1 - (3\kappa + 4)\hat{x}_3 \\ 0 &= 3\hat{x}_2 + 3\hat{x}_3 - (3\kappa + 3)\hat{x}_4 \end{aligned} \qquad \text{[Eq. 2-18]}$$

To solve those equations, one may obtain $\hat{x}_1$ from the first equation, then $\hat{x}_2$ and $\hat{x}_3$ follow from the second and third equations, and finally, $\hat{x}_4$ follows from the last equation. The lower-triangular character of the matrix A guarantees that the n^{th} equation in the system of Equation 2-18 involves $\hat{x}_n$ and does not involve $\hat{x}_j$ for any j > n. Another way to view the benefits of the lower-triangular character of the matrix A is to observe that this makes Equation 2-17 have the form of a system of equations after the forward course of Gaussian elimination has been performed, so that only the rapid back course remains to be done.

Why were we justified in integrating the $\hat{x}$ from 0 to ∞? Texts on differential equations[5] show that solutions of equations such as Equation 2-14 are always linear combinations of functions $\exp(\lambda_j t)$, where λ_j is an eigenvalue of the matrix A, when these eigenvalues are distinct. The eigenvalues of a lower-triangular matrix are the elements on the main diagonal. Inspection of Equation 2-15 shows that these elements are distinct, and negative. Thus solutions of Equation 2-11 decrease exponentially with time, and the integrals we use exist.

In Appendix A, we show that the approach to calculating long-time limiting probabilities illustrated here can be applied to many engagement models. We use this method, which from now on we will call “the method of Appendix A,” frequently in the present SLAACM and in planned extensions to it.

[5] W.E. Boyce and R.C. DiPrima, *Elementary Differential Equations*, Third Edition (New York: Wiley, 1976), Sections 7-6 through 7-9.

For SLAACM, the key result from engagement calculations is the discrete bivariate loss distribution. This gives the loss statistics that are propagated day-by-day in a campaign. Figure 2-5 shows an example of the discrete bivariate loss distribution, for a 4 vs. 4 engagement in which both sides fight to annihilation (breakpoint = 0), and κ = 1.46. In the figure, blue bars indicate Blue win states (states with 4 Red losses), and red bars indicate Red win states (states with 4 Blue losses). The figure shows that, as should be the case when κ > 1, the Blues are likely to win the engagement and the total probability of the blue bars is distinctly larger than the total probability of the red bars. There is significant dispersion in both Red and Blue losses.

Figure 2-5. Bivariate Loss Distribution

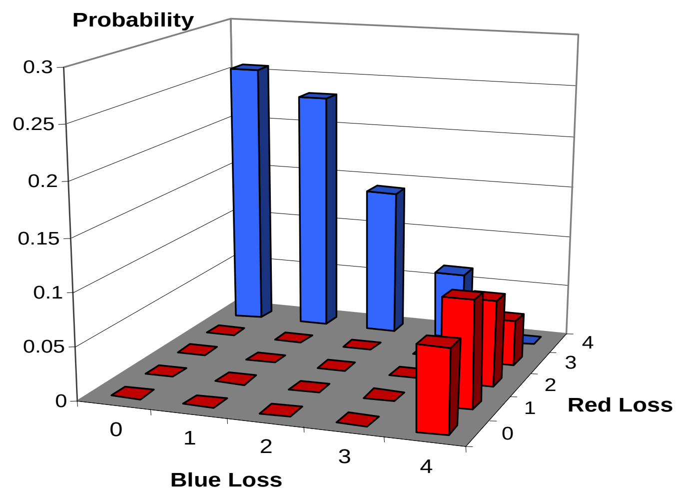
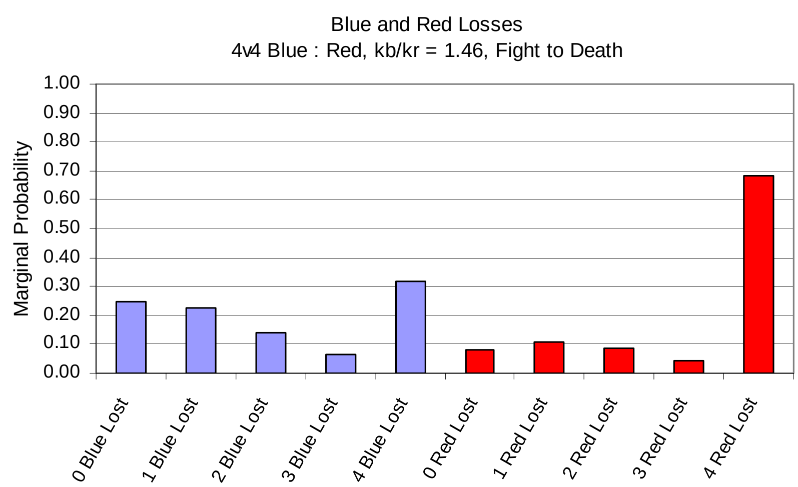


Figure 2-6 shows the marginal distributions of Blue and of Red losses and the total win probabilities. The marginal distributions show that the marginal probabilities are not necessarily monotone. They also show that, in this case, the probability of 4 losses is the largest Blue probability.

Figure 2-6. Marginal Loss Distributions with KRR=1.46

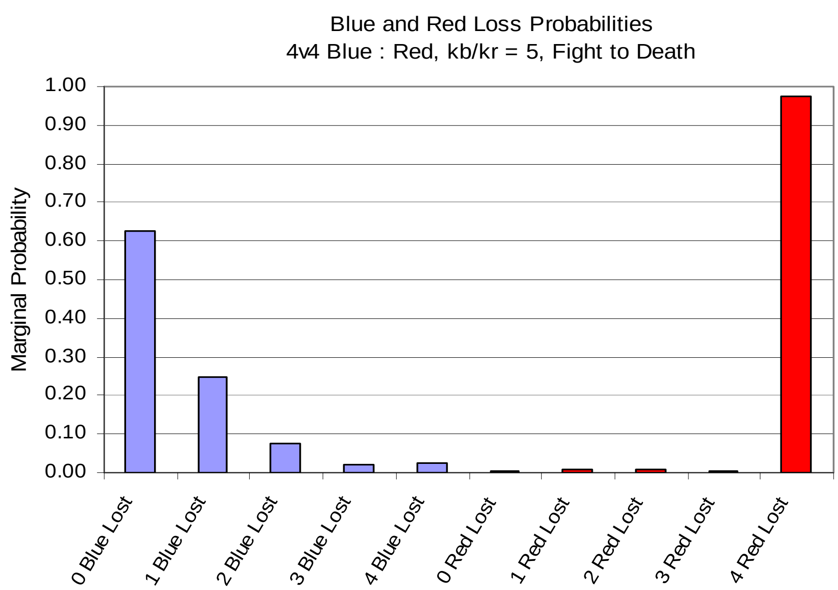


Such a "toe-up" at the greatest possible loss may seem counterintuitive. In fact, that feature is certain to happen for kill-rate ratios κ sufficiently close to 1. To see this, consider a 4 vs. 4 engagement to annihilation, with κ = 1. The probability of a Blue win must equal the probability of a Red win, and these probabilities must both equal 0.5. The probability of a Red win is the probability of 4 Blue losses, so the probability of 4 Blue losses must be 0.5.

The sum of the marginal probabilities of Blue losses must be 1. Therefore, none of the marginal probabilities of 0, 1, 2, or 3 Blue losses can exceed 0.5. If, as in fact is the case, all the marginal probabilities of 0, 1, 2, and 3 Blue losses are positive, each of them must be less than 0.5, so that, in this case of equal strength forces (κ = 1), the marginal probability of 4 Blue losses must be the largest marginal loss probability.

The marginal loss probabilities are continuous functions of κ (they are in fact rational functions of κ, as may be seen by applying the method of Appendix A), so that, as κ increases from 1, at least for a while, the most probable Blue loss must be 4. Figure 2-7 shows that the Blue toe-up is negligible for κ = 5.

Figure 2-7. Marginal Loss Distribution



SCALABILITY AND COPING WITH LARGE-DIMENSIONED PROBLEMS

The examples illustrate that initial value problems (Equations 2-5 and 2-6) are tractable, and that their solutions give results that military planners may find useful. These examples also show why computational probabilistic models have not been widely used: Even moderate values of M and N generate a lot of equations. For example, an engagement of 4 Defense aircraft with 8 Attack aircraft, not at all uncommon, leads to a system of 44 differential equations. The more sophisticated models that we consider in Appendix B can lead to systems of thousands of equations. Computational probabilistic models have significant scalability concerns.

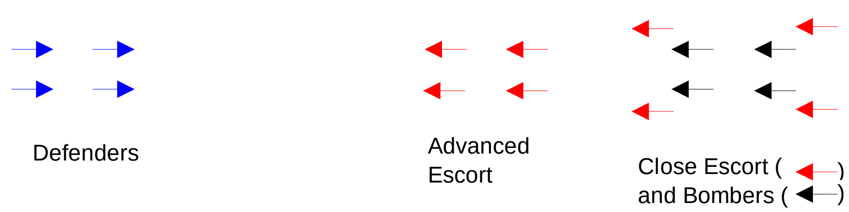
Until fairly recently, it was not practical to develop models to deal with such large state spaces. Advances on four fronts have, however, changed that. First, the memory and speed of widely available PCs now match or exceed those of the supercomputers of only a decade or so ago. Second, recently available applications packages make it possible to build and analyze large-dimensioned probabilistic models. In the development of SLAACM, and for offline analyses, we have used Markov analysis tools developed by NASA for analysis of high-reliability electronics to set up and solve the large-dimensioned initial value problems to which computational probabilistic models often lead. Appendix B discusses how we have applied these tools to air combat analysis. Third, the high-speed algorithm discussed above allows us to calculate engagement results extremely fast. Fourth, commercial packages are now available to solve the many-variable integer programming problems, to which computational probabilistic models lead.

Composite Engagements

In the SLAACM air-defense scenario, engagements are always composite engagements of a particular kind, in which a flight of 4 defending fighters engages an attack package made up of 4 advanced escorts, 4 close escorts, and 4 bombers. We discuss that scenario in detail here and then introduce the additional engagements we have added for the attack scenarios.

The defenders first engage the advanced escorts. If the defenders win that engagement, the surviving defenders engage the close escorts. If the defenders again prevail, defenders who survive the engagement with close escorts engage the bombers. Results of this final engagement determine the distribution of the number of bombers "leaking" through the defense. Figure 2-8 illustrates the opposing forces for this composite engagement.

Figure 2-8. Opposing Forces in SLAACM Composite Engagement



Sixteen final outcome or absorbing states are possible when both sides fight to annihilation. The outcome states may be described by the ordered quadruple (A, C, B, D), where A is the number of surviving attack advanced escorts, C is the number of attack close escorts that survive, B is the number of surviving attack bombers, and D is the number of surviving defense fighters. Table 2-3 lists the

16 possible absorbing quadruples for the SLAACM composite engagement, and Table 2-4 shows the 64 total transition and absorbing states possible for the scenario.

Table 2-3. Absorbing States, SLAACM Composite Engagement

A Attack AE	C Attack CE	B Attack B	D Defense
4	4	4	0
3	4	4	0
2	4	4	0
1	4	4	0
0	4	4	0
0	3	4	0
0	2	4	0
0	1	4	0
0	0	4	0
0	0	3	0
0	0	2	0
0	0	1	0
0	0	0	4
0	0	0	3
0	0	0	2
0	0	0	1

Table 2-4. Transition and Absorbing States, SLAACM Composite Engagement

A Attack AE	B Attack CE	C Attack B	D Defense
4	4	4	4
4	4	4	3
4	4	4	2
4	4	4	1
4	4	4	0
3	4	4	4
3	4	4	3
3	4	4	2
3	4	4	1
3	4	4	0
2	4	4	4
2	4	4	3
2	4	4	2
2	4	4	1
2	4	4	0

Table 2-4. Transition and Absorbing States, SLAACM Composite Engagement

A Attack AE	B Attack CE	C Attack B	D Defense
1	4	4	4
1	4	4	3
1	4	4	2
1	4	4	1
1	4	4	0
0	4	4	4
0	4	4	3
0	4	4	2
0	4	4	1
0	4	4	0
0	3	4	4
0	3	4	3
0	3	4	2
0	3	4	1
0	3	4	0
0	2	4	4
0	2	4	3
0	2	4	2
0	2	4	1
0	2	4	0
0	1	4	4
0	1	4	3
0	1	4	2
0	1	4	1
0	1	4	0
0	0	4	4
0	0	4	3
0	0	4	2
0	0	4	1
0	0	4	0
0	0	3	4
0	0	3	3
0	0	3	2
0	0	3	1
0	0	3	0
0	0	2	4
0	0	2	3
0	0	2	2
0	0	2	1
0	0	2	0

Table 2-4. Transition and Absorbing States, SLAACM Composite Engagement

A Attack AE	B Attack CE	C Attack B	D Defense
0	0	1	4
0	0	1	3
0	0	1	2
0	0	1	1
0	0	1	0
0	0	0	4
0	0	0	3
0	0	0	2
0	0	0	1

The first engagement is 4 defenders vs. 4 advanced escorts. Eight outcome states are possible from this engagement: four Attack win states—(4, 4, 4, 0), (3, 4, 4, 0), (2, 4, 4, 0), and (1, 4, 4, 0)—in which the advanced escorts defeat the four defenders with 0, 1, 2, or 3 losses. Four Defense win outcomes are possible: (0, 4, 4, 4), (0, 4, 4, 3), (0, 4, 4, 2), and (0, 4, 4, 1), when Defense defeats the advanced escorts with 0, 1, 2, or 3 losses. We use the method of Appendix A to determine the probabilities of these states. If the advanced escorts win this engagement, the composite engagement ends.

The Defense win states cause the engagement to continue into engagements between 1, 2, 3, or 4 defenders and the 4 close escorts. Four possible outcome states are Attack win states: (0, 4, 4, 0), (0, 3, 4, 0), (0, 2, 4, 0), and (0, 1, 4, 0), corresponding to close escort losses of 0, 1, 2, or 3 aircraft.

The engagement between 1 defender and the 4 close escorts has one Defense win state: (0, 0, 4, 1). The engagement between 2 defenders and 4 close escorts has two Defense win states: (0, 0, 4, 1) and (0, 0, 4, 2). The engagement between 3 defenders and 4 close escorts has three Defense win states—(0, 0, 4, 1), (0, 0, 4, 2), and (0, 0, 4, 3)—and the engagement between 4 defenders and 4 close escorts has four Defense win states: (0, 0, 4, 1), (0, 0, 4, 2), (0, 0, 4, 3), and (0, 0, 4, 4). Again, we use the method of Appendix A to evaluate the probabilities of all these outcomes.

Thus the engagements with close escorts lead to four Attack win states and to four cases of engagements between defenders and bombers: 1 vs. 4, 2 vs. 4, 3 vs. 4, and 4 vs. 4. These engagements lead to the Attack win states (0, 0, 1, 0), (0, 0, 2, 0), (0, 0, 3, 0), and (0, 0, 4, 0) and to the Defense win states (0, 0, 0, 1), (0, 0, 0, 2), (0, 0, 0, 3), and (0, 0, 0, 4). Using the method of Appendix A to evaluate the probabilities of the outcome states in the defenders vs. bomber engagements completes determination of the probabilities of all 16 outcome states in Table 2-3.

SLAACM contains individual kill-rate ratios for each Defense-Attack aircraft type-pair in the Defense and Attack inventories. SLAACM uses these to calculate probabilities for each of the 16 possible outcomes for all the feasible 4 vs. 4+4+4 combinations. In a campaign, Attack and Defense use the resulting probabilities to construct optimal attack packages and optimal defenses.

MODIFICATIONS FOR ATTACK SLAACM

New Scenarios

New engagement scenarios added to SLAACM are intended to represent potential U.S. attack package options using both normal and low-observable aircraft.. These are shown in Table 2-5.

Table 2-5. New SLAACM Scenarios

Attack	Defense
4 escorts + 2 bombers	8, 4, or 2 defenders + 1 SAM with 3 missiles
4 escorts + 2 bombers + 2 SEADs	8, 4, or 2 defenders + 1 SAM with 3 missiles
2 escorts + 1 bomber	8, 4, or 2 defenders + 1 SAM with 3 missiles
2 escorts + 1 bomber + 1 SEAD	8, 4, or 2 defenders + 1 SAM with 3 missiles

In these scenarios SEAD aircraft engage the SAM site (the SAM has unlimited missiles for this duel). The SAM site does not engage the escort fighters, and the defending fighters do not engage the SEAD aircraft. If the SEAD aircraft are destroyed by the SAM, the SAM site (with up to three missiles) engages the bombers. In scenarios with no SEAD aircraft, the SAM site always engages the bombers with three missiles. The limitation to three missiles accounts for the limited time interval on which the bombers are on their bomb run, rather than for any limit to the number of missiles at the SAM site.

New Options

Several options have been added to SLAACM to more accurately represent engagements and campaigns and to allow the analyst to change parameters that were previously hardcoded. These new options include the following:

- *Escaping pure bombers complete mission.* SLAACM has been revised so that, in the case that all escorts are killed, the pure bombers either escape and go home or escape and complete their mission

- *Escaping fighter-bombers complete mission.* Fighter-bombers fight through as before, and optionally go home or complete their missions.

- *User-definable breakpoints.* The analyst can now select loss breakpoints for both offensive and defensive forces. These breakpoints are used in the engagement calculations to determine the bivariate loss probabilities.
- *Attacker and Defender loss values.* The analyst now has the option to assign the payoff function values the attacker and defender place on their own loss and on killing an opponent. As before, the LAD (local area defender) buttons associated with each defender aircraft type, can be used to force that defender type to ignore his own losses.
- *Bomb payload value.* The bomb payload values, calculated from the bomber tonnage and CEP payloads, are hardcoded, but are now displayed on the “RedBombs” worksheet, along with the Attacker and Defender loss values to give the analyst insight into the Attacker and Defender payoff values.

SAM and SEAD Features

The following subsections describe the engagement models used in the models of SLAACM’s new features.

SAM–SEAD Engagements

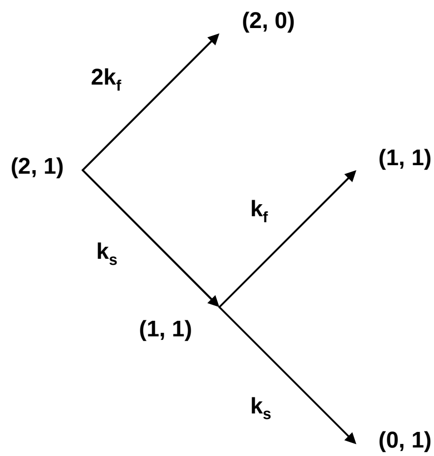
When attack packages include SEAD aircraft, 2 SEADs engage the SAM site defending the package’s target. This engagement takes place before any of the package’s bombers reach the target. If the SEADs destroy the site, then the site does not engage the bombers.

Two options are available for the SEADs breakaway behavior: the SEADs may engage to annihilation, or they may disengage on sustaining 1 loss.

We model the SAM–SEAD engagement as a stochastic Lanchester engagement, in which the times between kills by the two adversaries each are exponentially distributed. We denote the parameter of the SAM site’s time-to-kill distribution by k_s and call that of the SEAD’s distribution k_f. The parameter k_f is equal to the mean time between kills by the SEADs; k_s is the mean time between kills by the site.

We denote the state of the engagement by the ordered pair (f, s), where f denotes the number of fighters present. The parameter s has the value 1 if the site is active, 0 if it is destroyed. Figure 2-9 diagrams the engagement when the SEADs fight to annihilation. The labels on the arrows indicating state transitions are the rates of the transition indicated.

Figure 2-9. SAM–SEAD Engagement Diagram, SEADs Break at 0



Let $P_{ij}(t)$ denote the probability that the system is in state (i, j) at time t. The forward Chapman-Kolmogorov equations for the engagement can be written by inspection of Figure 2-9:

$$\begin{aligned}
\dot{P}_{21} &= -(2k_f + k_s)P_{21} \\
\dot{P}_{20} &= 2k_f P_{21} \\
\dot{P}_{11} &= -(k_f + k_s)P_{11} + k_s P_{21} \\
\dot{P}_{10} &= k_f P_{11} \\
\dot{P}_{01} &= k_s P_{11}
\end{aligned} \qquad \text{[Eq. 2-19]}$$

The system begins in state (2, 1), so that at t = 0, all P_{ij} are zero except P_{21}. This initial condition and equations 2-19 determine the complete evolution of P_{ij}.

Consistent with SLAACM's treatment of other stochastic engagements, we use the long-time limiting values P_{ij}^{∞} of the P_{ij} for the outcome probabilities of the SAM–SEAD engagement. The method of Appendix A allows us to write these values explicitly. In the long-time limit P_{21} and P_{11} are zero, and

$$\begin{aligned}
P_{20}^{\infty} &= \frac{2\rho}{2\rho + 1} \\
P_{10}^{\infty} &= \frac{\rho}{(2\rho + 1)(\rho + 1)} \\
P_{01}^{\infty} &= \frac{1}{(2\rho + 1)(\rho + 1)}
\end{aligned} \qquad \text{[Eq. 2-20]}$$

We use the symbol ρ for the ratio k_f/k_s. The probability that the site is up is P_{01}^{∞}, the probability that the site is down with 0 SEADs lost is P_{20}^{∞}, and the probability that the site is down with 1 SEAD lost is P_{10}^{∞}.

Similar considerations show that, for a SAM–SEAD engagement in which the SEADs break on sustaining 1 loss, the probability that the site is down is P_{20}^{∞}, which is also the probability that 0 SEADs are lost. For this engagement, the probability that the site is up and 1 SEAD is lost is $1 - P_{20}^{\infty} = 1/(2\rho + 1)$.

SLAACM incorporates the results of this section to treat the outcomes of SAM–SEAD engagements.

SAM–Bomber Engagements

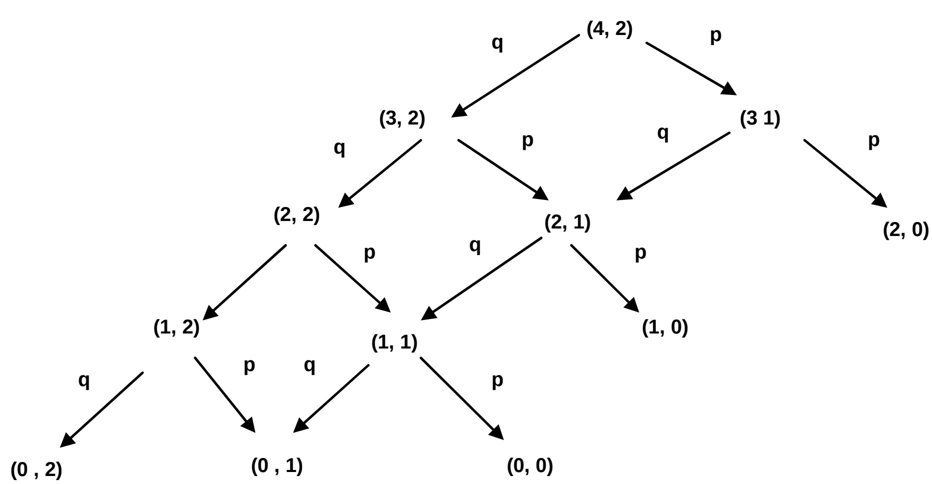
We are interested in the probability $S_m(j,k)$ that j of k attacking bombers survive when a SAM site fires m shots at the bombers. We assume that the site's battle management is sufficiently sophisticated that the outcomes of all the shots fired are statistically independent, with single-shot kill probability p. Then when $m \le k$, obviously

$$S_m(j,k) = B(k-j, m, p),\ 0 \le j \le k \qquad \text{[Eq. 2-21]}$$

where B(i, j, p), the binomial probability distribution function, represents the probability of i successes in j trials, when the single-trial probability of success is p.

When $m > k$, $S_m(j,k)$ may be reckoned with the following considerations. Let us begin with a simple example, when m = 4 and k = 2. Introducing a system state variable (i, j), where i is the number of shots remaining and j the number of aircraft remaining, the system evolves as shown in Figure 2-10. In that figure, $q \equiv 1 - p$.

Figure 2-10. Four SAM Shots vs. Two Bombers



The diagram of Figure 2-10 proceeds in the same way as the diagram for the binomial distribution, until two shots have been fired. For three shots, since the state (2, 0) is an absorbing boundary state (the engagement ends when both targets are destroyed), the left-going arrow from that state to the state (1, 0) is absent. Thus the probability of state (1, 0) is not B(2, 3, p), but, rather, B(2, 3, p) − qB(2, 2, p). Similarly, the probability of state (0, 0) is B(2, 4, p) − qB(2, 3, p).

The probabilities of states (0, j), $1 \le j \le 2$ may be read from the bottom row of the diagram of Figure 2-10. Since each of these states has all the predecessors of the binomial distribution, they are binomial, specifically, B(2 − j, 4, p).

Thus for the example of Figure 2-10,

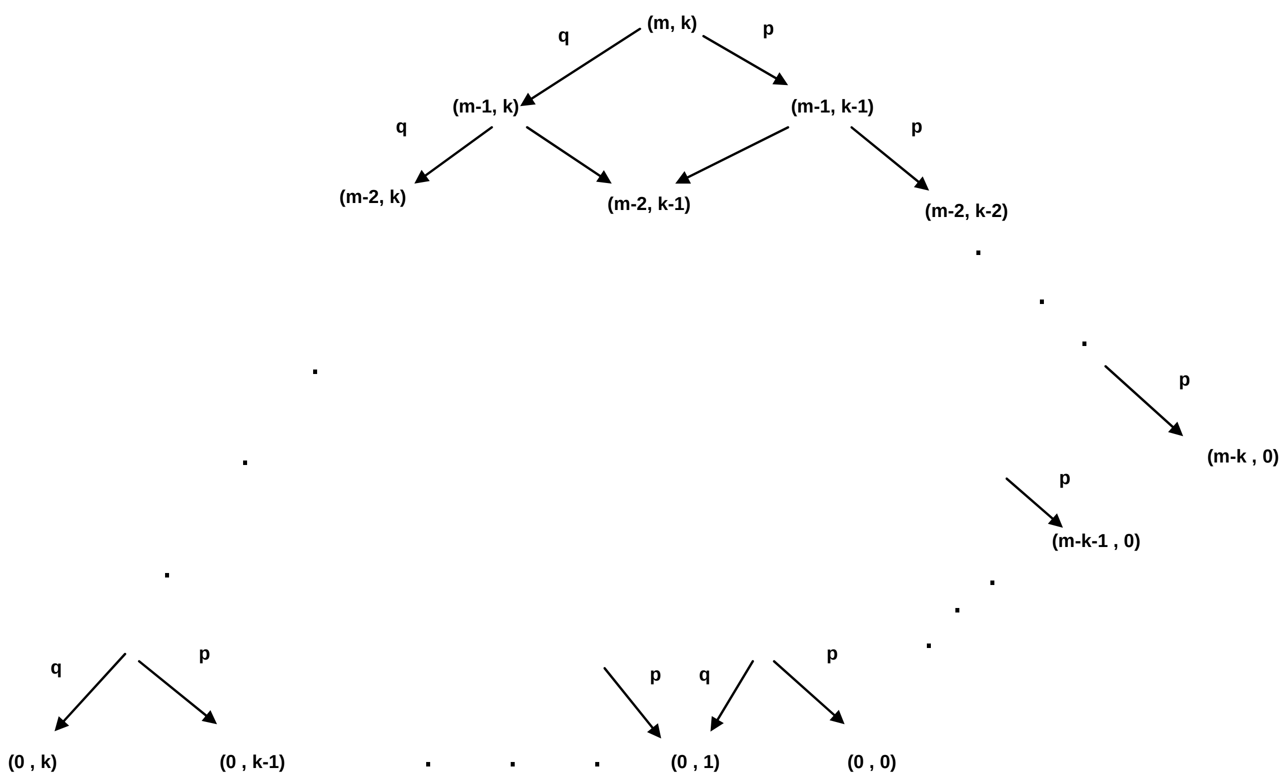
$$S_4(j,2) = B(2-j,4,p),\ 1 \le j \le 2 \qquad \text{[Eq. 2-22]}$$

and

$$S_4(0,2) = \sum_{n=0}^{1}\left[B(2,4-n,p) - qB(2,4-n-1,p)\right] + B(2,2,p)\,. \qquad \text{[Eq. 2-23]}$$

We may evaluate $S_m(j,k)$ in general with an extension of the diagram of Figure 2-10. The extended diagram begins in state (m, k), as shown in Figure 2-11.

Figure 2-11. Extended Diagram



The diagram of Figure 2-11 proceeds in the same way as the diagram for the binomial distribution, until k shots have been fired. For k + 1 shots, since the state (m − k, 0) is an absorbing boundary state (the engagement ends when all k targets are destroyed), the left-going arrow from that state to the state (m − k − 1, 0) is absent. Thus the probability of state (m − k − 1, 0) is not B(k, k + 1, p), but, rather, B(k, k + 1, p) − qB(k, k, p). As more shots are fired, the probability of state (m − k − n, 0) is similarly seen to be B(k, k + n, p) − qB(k, k + n − 1, p), until all m shots have been fired, bringing us to the bottom row of the diagram. The probability of state (0, 0) is B(k, m, p) − q B(k, m − 1, p).

The probability of ending states (0, j), read from the bottom row of the diagram of Figure 2-11, is B(k − j, m, p), 1 ≤ j ≤ k. Consequently

$$S_m(j,k) = B(k-j,m,p), 1 \le j \le k \qquad \text{[Eq. 2-24]}$$

and

$$S_m(0,k) = \sum_{n=0}^{m-k-1}\left[B(k,m-n,p) - qB(k,m-n-1,p)\right] + B(k,k,p)\,. \qquad \text{[Eq. 2-25]}$$

Now let us consider the expected value of the number of bombers destroyed by SAMs, for the cases involving SAMs. We consider attack package 4 as a representative case. In this package, 2 SEAD aircraft precede 4 escorts accompanying 2 bombers. Each package's target is defended by a SAM site.

The SEADs engage the site. The defending fighters first engage the escorts; survivors engage the bombers. If SLAACM II's user has chosen options permitting victorious fighter-bombers, or escaping bombers, to continue to the package's target, and if the site survives the SEADs' attack, bombers reaching the target are attacked by the SAM site.

When bombers do reach the target, either one or two arrive. A viable site then fires a number m of shots at the bombers; m is a user option in SLAACM II.

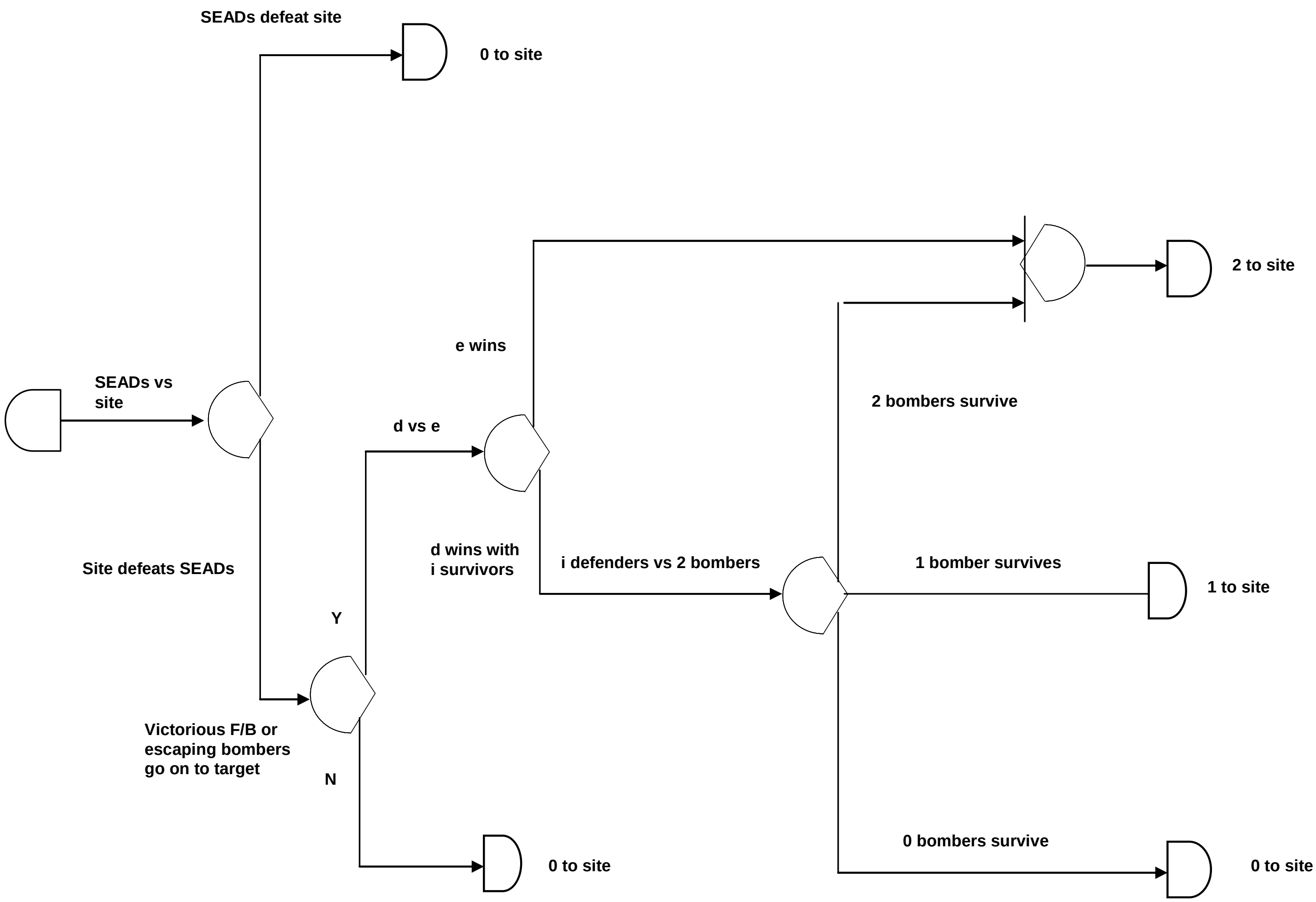
Thus the expected number S_k of bombers destroyed by the site is

$$S_k = PB(1) \times S_m(0,1) + PB(2) \times (2S_m(0,2) + S_m(1,2)) \qquad \text{[Eq. 2-26]}$$

where PB(j) denotes the probability that j bombers reach the site. Calculation of S_k thus requires PB(j), and the ability to calculate $S_m(j,k)$. Calculating $S_m(j,k)$ is readily done using Equations 2-24 and 2-25. To evaluate PB(j), one may consider the activity network diagram of Figure 2-12.

Please note that the sequence of events diagrammed in Figure 2-12 is appropriate for calculating the probabilities of the number of bombers reaching the site, but it is not necessarily appropriate for other probability calculations.

Figure 2-12. Activity Network Diagram



Referring to Figure 2-12, we see that, when escaping bombers or victorious fighter-bombers do not continue to the target, the only way in which bombers reach the target is when the SEADs have not defeated the SAM site, and the escorts have defeated the defenders. Thus, in this case,

$$PB(2) = P(\text{site up}) \sum_{j=e\,min}^{4} Ares(d\,min - 1, j)$$

$$PB(1) = 0 \qquad \text{[Eq. 2-27]}$$

where Ares(i, j) denotes the probability that i defenders and j escorts survive the defender-escort engagement, dmin denotes the minimum number of defenders willing to continue the engagement, and emin denotes the minimum number of escorts willing to continue. Let PS(j) be the probability that j SEADs survive the SEAD-site engagement; then when the SEADs fight to annihilation, P(site up) = PS(0). If the SEADs break at 1 loss, then P(site up) = PS(1).

If escaping bombers and/or victorious fighter-bombers do continue to their targets, then when bombers are fighter-bombers, PB(j) is given by

$$
\begin{aligned}
PB(2) &= P(\text{site up})\left[\sum_{j=e\min}^{4} Ares(d\min-1, j) + \sum_{k=d\min}^{ndef} Ares(k, e\min-1)Bres(kb,2)\right] \\
PB(1) &= P(\text{site up})\left[\sum_{k=d\min}^{ndef} Ares(k, e\min-1)Bres(kb,1)\right]
\end{aligned}
\quad \text{[Eq. 2-28]}
$$

In Equation 2-28, Bres(i, j) denotes the probability that i defenders and j bombers survive the defender-bomber engagement, and kb = dmin − 1 when the bombers are fighter-bombers, while kb = k when the bombers are escaping bombers.

Calculations of the expected number of bombers killed by sites proceed similarly for attack packages 5, 6, and 7, which are modifications of attack package 4.

Escaping Bomber Characteristics

A class of engagement models in which bombers never kill defenders, but may escape from defenders, is discussed in “TACAIR Analytical Support, Interim Mathematical Report 1,” published in January 2008. For completeness, we include the basic discussion of the “escaping bomber” engagement models here.

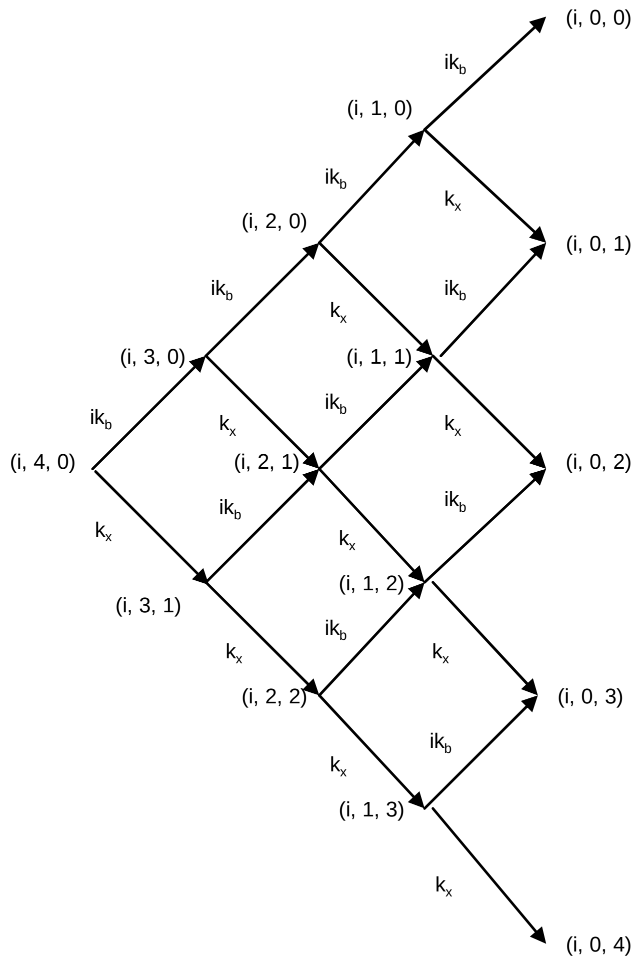
Typically, bomber aircraft, other than fighter-bomber aircraft used as bombers, are quite weak compared to defending fighter aircraft. Specifically, the mean time between kills of fighters by bombers is much longer than the mean time between kills of bombers by fighters. This is a realistic model of what would happen if the fighters actually engaged the bombers.

But the bombers might be able to escape the fighters, instead of engaging with them. Consequently, a better model of engagements between bombers and defenders should not allow the bombers to kill a defender, but should allow for the possibility that bombers escape from the engagement. The following paragraphs develop a model of that situation and explain how the new model has been integrated into an extended version of SLAACM.

We consider an engagement of i defending fighters with a flight of 4 bombers, in which the bombers do not attempt to destroy the fighters, but, rather, attempt to evade them. Consistent with SLAACM’s general modeling approach, we assume that the times between kills made by the fighter aircraft are exponentially distributed, identically and independently, with mean time $1/k_b$. We further assume that the times between escapes made by the bomber aircraft are identically and independently exponentially distributed, with mean time $1/k_x$.

Let the state of the engagement be described by the triple (i, j, k), where i denotes the number of fighters present, j the number of bombers present, and k denotes the number of bombers that have escaped from the engagement. Then the engagement's state transition diagram is as shown in Figure 2-13.

Figure 2-13. State Transition Diagram



The state transition diagram of Figure 2-13 implies the following set of 15 evolution equations (forward Chapman-Kolmogoroff equations) for the probabilities $P_{i,j,k}(t)$ that the system is in state (i, j, k) at time t:

$$\dot{P}_{i,j,k} = -(ik_b + jk_x)P_{i,j,k} + ik_b P_{i,j+1,k} + (j+1)k_x P_{i,j+1,k-1}, \;\; 4 \geq j \geq 0, \;\; 0 \leq k \leq 4-j \quad \text{[Eq. 2-29]}$$

where $P_{i,j,k}(t) \equiv 0$ if j > 4 or k < 0.

Initially, the system is in state (i, 4, 0), so that

$$P_{i,4,0}(0) = 1, \;\; P_{i,j,k}(0) \equiv 0, j \neq 4, k \neq 0 . \qquad \text{[Eq. 2-30]}$$

It is convenient to introduce the nondimensional time $\tau \equiv k_x t$. For the independent variable τ, Equations 2-29 and 2-30 are

$$P'_{i,j,k} = -(i\kappa + j)P_{i,j,k} + i\kappa P_{i,j+1,k} + (j+1)P_{i,j+1,k-1}, \quad 4 \geq j \geq 0, \quad 0 \leq k \leq 4-j \quad \text{[Eq. 2-31]}$$

and

$$P_{i,4,0}(0) = 1, \quad P_{i,j,k}(0) \equiv 0, \; j \neq 4, k \neq 0 \,. \qquad \text{[Eq. 2-32]}$$

In Equation 2-31 the prime denotes differentiation with respect to τ, and $\kappa \equiv k_b / k_x$. The parameters k_b and k_x appear in Equation 2-31 only in the single parameter κ . (Strictly speaking, the probabilities as functions of τ are distinct from the probabilities as functions of t, but for clarity and simplicity, we will not introduce distinct notation for them.)

The initial value problem of Equations 2-31 and 2-32 may be solved analytically or numerically by standard methods to give the time evolution of the state probabilities. Alternatively, one may consider the long-time limit of the state probabilities. This limit is relevant if engagements are likely to last for times much longer than the mean time for a bomber to escape.

The long-time limiting probabilities may be calculated in finitely many steps. Integrating Equations 2-31 with respect to τ from 0 to infinity and using the initial conditions of Equations 2-32 leads to a system of 10 linear algebraic equations for the integrals from 0 to infinity of the transient-state probabilities, and to a system of 5 expressions for the long-time limiting values of the probabilities of the absorbing boundary states, that is, of states (i, 0, k), 0 ≤ k ≤ 4, as linear combinations of integrals of the transient state probabilities.

The 10 equations for the integrals $\phi_{i,j,k}$ of the transient-state probabilities are

$$\begin{aligned} \phi_{i,4,0} &= \frac{1}{i\kappa + 4} \\ \phi_{i,j,k} &= \frac{i\kappa}{i\kappa + j}\phi_{i,j+1,k} + \frac{(j+1)}{i\kappa + j}\phi_{i,j+1,k-1}, \quad 1 \leq j \leq 3, \quad 0 \leq k \leq 4-j \end{aligned} \qquad \text{[Eq. 2-33]}$$

Equations 2-33 may be solved recursively, speeding up the solution process considerably. With the solutions, the long-time limiting probabilities $P_{i,j,k}^{\infty}$ for the absorbing boundary states are given by

$$\begin{aligned} P_{i,0,0}^{\infty} &= i\kappa\phi_{i,1,0} \\ P_{i,0,1}^{\infty} &= i\kappa\phi_{i,1,1} + \phi_{i,1,0} \\ P_{i,0,2}^{\infty} &= i\kappa\phi_{i,1,2} + \phi_{i,1,1} \\ P_{i,0,3}^{\infty} &= i\kappa\phi_{i,1,3} + \phi_{i,1,2} \\ P_{i,0,4}^{\infty} &= \phi_{i,1,3} \end{aligned} \qquad \text{[Eq. 2-34]}$$

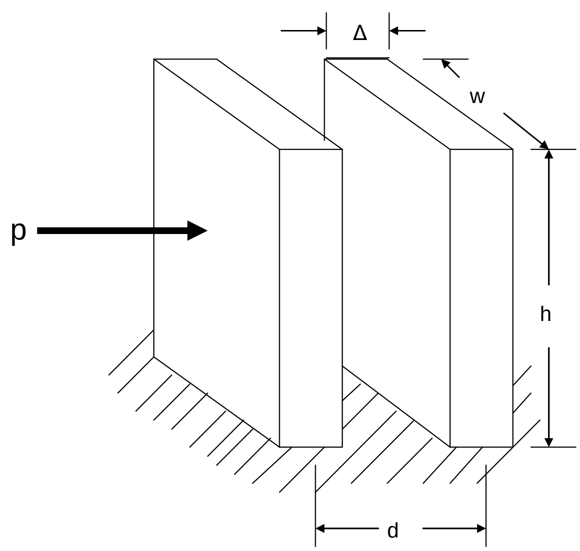
Equations 2-33 and 2-34 give means of extending SLAACM to model bomber-defender engagements in which the bombers may escape. Like the equations for the M vs. N stochastic Lanchester equations of SLAACM's present engagements, "escaping-bombers" engagements that go to the long-time limit have only one parameter, κ. These models are available in current versions of SLAACM.

Chapter 3
How SLAACM Works: Bomber Effectiveness Parameter

The basic campaign modeled in SLAACM is a bomber attack against defended targets. The attacker's bombers have different levels of effectiveness, based on the attack package. The offensive effectiveness of a bomber depends on two factors: the payload of bombs carried, and the accuracy with which those weapons can be delivered to targets. We developed a measure of bomber effectiveness based on these factors.

We considered first the relation between blast overpressure p and the properties of a structure designed to resist that overpressure. We used the simple model of reinforcing structure shown in Figure 3-1.

Figure 3-1. Reinforcing Structure Model



In this simple model, two columns reinforce the structure. Their thicknesses are Δ, their heights are h, and the spacing between their centers is d. Their dimension into the page is w.

The reinforcing columns act as cantilevers against lateral loads. A pressure p acting on the left face of this simple structure will induce axial loads $\frac{pwh^2}{2d}$ in each column, tension in the left and compression in the right. The pressure will also induce shear loads $\frac{pwh}{2}$ in each column. Since both loads are proportional to p, the maximum stress induced in the columns will also be proportional to p. If the columns' dimensions are adjusted so that a material of given maximum allowable stress can support the loads induced by the overpressure, the area of the columns' bases w∆ will be proportional to the maximum stress and, therefore, also to p.

Thus the volume of the reinforcing structure, 2w∆h, will be proportional to p. Assuming that the cost of the reinforcing structure is proportional to its volume leads to the conclusion that the cost of reinforcement will be proportional to overpressure p.

Now let us consider how blast overpressure varies with explosive yield E. Dynamic overpressure p is proportional to the square of the air velocity v immediately behind the blast wave produced by the explosion. Dimensional analysis shows how v varies with E.[1] In addition to E, the parameters affecting v are the undisturbed air density ρ and the time t required for the blast wave to reach the locations of interest.

According to the principles of dimensional analysis, v can be expressed as the product of a term $E^{\alpha}\rho^{\beta}t^{\gamma}$ and a dimensionless function of all possible dimensionless combinations of E, ρ, and t. If $E^{\alpha}\rho^{\beta}t^{\gamma}$ is equal to a velocity, that term must have the dimension L/T, where L represents dimension length and T dimension time. Now, energy E has dimension ML^2/T^2, where M denotes dimension mass; ρ has dimension M/L^3, and, of course, t has dimension T. Thus

$$\left(\frac{ML^2}{T^2}\right)^{\alpha}\left(\frac{M}{L^3}\right)^{\beta}T^{\gamma} = \frac{L}{T}. \qquad \text{[Eq. 3-1]}$$

Equation 3-1 implies that

$$\begin{aligned} \alpha + \beta &= 0 \\ 2\alpha - 3\beta &= 1 \\ -2\alpha + \gamma &= -1 \end{aligned} \qquad \text{[Eq. 3-2]}$$

The unique solution to Equation 3-2 is $\alpha = 1/5, \beta = -1/5, \gamma = -3/5.$ Similar arguments to those leading to Equations 3-1 and 3-2 show that there are no nondimensional combinations of E, ρ, and t, other than a constant. Therefore

$$v \propto \left(\frac{E}{\rho}\right)^{\frac{1}{5}} t^{-\frac{3}{5}} \qquad \text{[Eq. 3-3]}$$

[1] L. Sedov, *Similarity and Dimensional Methods in Mechanics* (New York: Academic Press, 1959).

and the dynamic overpressure p will satisfy

$$p \propto \rho v^2 \propto E^{\frac{2}{5}} \rho^{\frac{3}{5}} t^{-\frac{6}{5}}. \qquad \text{[Eq. 3-4]}$$

Similar dimensional considerations lead to the conclusion that the radius r of the blast wave satisfies the proportionality

$$r \propto \left(\frac{E}{\rho}\right)^{\frac{1}{5}} t^{\frac{2}{5}}. \qquad \text{[Eq. 3-5]}$$

From proportionalities (Equations 3-4 and 3-5), it follows that the radius r_L at which the overpressure falls to p_L satisfies

$$r_L \propto \left(\frac{E}{\rho}\right)^{\frac{1}{3}} p_L^{-\frac{1}{3}}. \qquad \text{[Eq. 3-6]}$$

Thus the lethal radius of a blast wave increases as the cube root of the blast's energy and decreases as the cube root of the lethal overpressure.

Since in our model the value of a structure is proportional to the overpressure required to destroy it, proportionality (Equation 3-6) implies that the lethal radius of a blast with energy E varies directly with the cube root of E, and inversely with the cube root of the value of the structures it affects.[2]

We now use the information about the variation of lethal radius with explosion energy and structure value to see how bombs' single shot kill probabilities (sspk's) vary with explosion energy and value of structures affected.

The sspk of a bomb with lethal radius r_L, delivered with circular error probable (CEP), is

$$\text{sspk} = 1 - \left(\frac{1}{2}\right)^{\frac{r_L^2}{\text{CEP}^2}}. \qquad \text{[Eq. 3-7]}$$

Introducing the variation of r_L with explosive energy E and value V obtained above, we have

$$\text{sspk} = 1 - \left(\frac{1}{2}\right)^{\frac{kE^{2/3}V^{-2/3}}{\text{CEP}^2}} \qquad \text{[Eq. 3-8]}$$

---

[2] These conclusions, obtained with simple dimensional analyses, agree with those from the exact Taylor-Sedov solution [Sedov, l. c. ante] for values immediately behind the shock.

where k is a constant.

Now,

$$kE^{2/3}V^{-2/3} = kE_{ref}^{2/3}V_{ref}^{-2/3}\left(\frac{E}{E_{ref}}\right)^{\frac{2}{3}}\left(\frac{V_{ref}}{V}\right)^{\frac{2}{3}}\left(\frac{CEP_{ref}}{CEP}\right)^{2}. \qquad \text{[Eq. 3-9]}$$

Defining the parameter k_1 by

$$k_1 \equiv kE_{ref}^{2/3}V_{ref}^{-2/3}, \qquad \text{[Eq. 3-10]}$$

we have

$$sspk = 1-\left(\frac{1}{2}\right)^{k_1\left(\frac{E}{E_{ref}}\right)^{\frac{2}{3}}\left(\frac{V_{ref}}{V}\right)^{\frac{2}{3}}\left(\frac{CEP_{ref}}{CEP}\right)^{2}}. \qquad \text{[Eq. 3-11]}$$

Let E_{ref} be the energy of a 500 lb bomb. Let V_{ref} be the value of a target hardened to withstand a 500 lb bomb. We will consider 500 lb, 1,000 lb, and 2,000 lb bombs and assume that bombs are always matched to targets, so that a bomber on a mission against 500 lb targets carries 500 lb bombs, and one on a mission against 1,000 lb targets carries 1,000 lb bombs, and that one going against 2,000 lb targets carries 2,000 lb bombs.

With those assumptions, the product of the term involving E and the term involving V in Equation 3-11 is always 1, and

$$sspk = 1-\left(\frac{1}{2}\right)^{k_1\left(\frac{CEP_{ref}}{CEP}\right)^{2}}. \qquad \text{[Eq. 3-12]}$$

We arbitrarily set the constant k_1 to 1; we only want a systematic way to compute the value to Red of a bomber with given payload P that is capable of delivering bombs with relative accuracy CEP/CEP_{ref}.

With this assumption, the value of the expected number of targets killed by such a bomber carrying 500 lb bombs is

$$V_1 = \left\lfloor \frac{P}{500} \right\rfloor sspk_0 V_{500} \qquad \text{[Eq. 3-13]}$$

where

$$\mathrm{sspk}_0 = 1 - \left(\frac{1}{2}\right)^{\left(\frac{\mathrm{CEP}_{\mathrm{ref}}}{\mathrm{CEP}}\right)^2} \quad \text{[Eq. 3-14]}$$

and where V_{500} is the value associated with a target hardened to withstand 500 lb bombs.

When the bomber is loaded with 1,000 lb bombs, the value of the expected number of targets killed is

$$V_2 = \left\lfloor \frac{P}{1000} \right\rfloor \mathrm{sspk}_0 (2)^{2/5} V_{500}\,, \quad \text{[Eq. 3-15]}$$

since the value of a target goes as the 2/5 power of the explosive yield required to destroy it.

Finally, if the bomber is loaded with 2,000 lb bombs, the value of the expected number of targets killed is

$$V_2 = \left\lfloor \frac{P}{2000} \right\rfloor \mathrm{sspk}_0 (4)^{2/5} V_{500}\,. \quad \text{[Eq. 3-16]}$$

Using this information, SLAACM assigns the average of V_1, V_2, and V_3, divided by V_{500}, as the value of the bomber.

The value for the CEP/$\mathrm{CEP}_{\mathrm{ref}}$ ratio is set by the analyst for each bomber type on the "ExRatios" worksheet. We use the ratio of 1 if the bomber cannot deliver guided (smart) bombs and a lower value, such as 0.1 to 0.4, if it can. Bomber destructive power is used in the optimizer payoff functions, but it has been useful to report the bombs delivered in terms of tonnage in the output tables and graphs. The user can display the tonnage of smart bombs delivered by setting set a CEP/$\mathrm{CEP}_{\mathrm{ref}}$ ratio display value on the "RedBombs" worksheet.

As mentioned in Chapter 2, the destructive power of the bomber payloads are now displayed on the "RedBombs" worksheet to give the analyst insight into the payoff values used in the optimizer.

Chapter 4
How SLAACM Works: Campaign Logic and Optimization

This chapter discusses the attacker and defender payoff logic and optimization algorithms used in SLAACM.

SPECIFYING ATTACKER AND DEFENDER PAYOFF FUNCTIONS

SLAACM determines optimal attacker attacks and optimal defender responses using payoff functions for each side. This section describes in detail the calculations used for the basic air defense scenario. Other scenarios use similar logic.

Attacker's Payoff Function

Attacker values an attacker package by considering the benefit of getting bombers through and of downing defender aircraft. The attacker may also consider penalties for losing its own aircraft, assigning different values to losing a fighter and to losing a bomber. Specifically, for the basic air defense scenario, a package's value to the attacker is determined in this way: An attacker package is defined by the triple (a, c, b), where a is the advanced escort type, c the close escort type, and b the bomber type (the bombers may be fighter-bomber aircraft). In Attack SLAACM, attack packages can differ in composition from the basic air defense (a, c, b) triple. Nevertheless, the attacker's expected payoff is calculated using the same method. The outcome statistics for an engagement between the package and the "planning defender" aircraft (the user identifies a defender type for the attacker's planning) are determined, and, from these, the expected number B_s of surviving bombers, the expected number RF_k of lost fighters, and the expected number D_k of downed defenders are computed. Then V_R, the attacker's value for the package, is given by

$$V_R = c_1 B_s + c_2 D_k - c_3 RF_k - c_4 (4 - B_s) . \qquad \text{[Eq. 4-1]}$$

The positive parameter c_1 is currently set at the value of a bomber computed by the method described in Chapter 3. Values of c_2 and c_3 are inputs from the user. When the attacker believes the defenders to be loss-averse, the value of c_2 is typically set at 20, reflecting the fact that the attacker believes the defender is likely to abandon the campaign if their side incurs many losses. For an attacker not overly concerned about losses, the value of c_3 is typically set at zero; also, if the attacker faces defenders that are strong in comparison with his aircraft (values of κ significantly greater than one), even values of c_3 that are small compared with one may cause the attacker not to dispatch any packages at all. Currently, c_4 is set equal to c_3.

Defender's Payoff Function

The defender values engaging a given attack package by considering the benefit of downing the attacker's fighters and bombers, and the penalty of losing its own aircraft. Specifically, the value to the defender of engaging an attacker package with fighters of a given type is calculated by computing the outcome statistics for that composite engagement (described in Chapter 3) and, from them, the expected value B_k of bombers killed, the expected value RF_k of attack fighters killed, and the expected value D_k of defenders killed. Then the value V_B of the engagement to the defender is evaluated as

$$V_B = d_1 B_k - d_2 D_k + d_3 RF_k . \qquad \text{[Eq. 4-2]}$$

Currently, d_1 is set to the attacker's value for the bomber type in the package, and d_2 and d_3 are user inputs. (A prime reason for allowing user assignment of the payoff values is that they are likely to be different for friendly defense and friendly attack campaigns.)

DETERMINING ATTACKS AND DEFENSES

A key part of SLAACM's operation is determining the engagements that occur each day. This is done by finding optimal sets of attacker packages dispatched by the attacker and by determining the defender's response in two ways: the response of smart defenders that know the makeup of the attacker packages before intercepting them and make optimal defenses, and the response of not-smart defenders that encounter attack packages randomly. This section explains these calculations.

Determining the Attacker Packages

As the aggressor, the attacker always plans its attacks. This leads to an integer programming (IP) problem, which SLAACM treats in one of two ways: a classic "greedy" heuristic, or an exact solution using the commercial general optimization package LINGO™. This section describes those calculations.

ATTACKER'S INTEGER PROGRAMMING PROBLEM

Each day, the attacker has a set of fighter, bomber, and fighter-bomber aircraft to deploy. In some scenarios, the attacker also has SEAD aircraft. In SLAACM, the attacker chooses the set of attacker packages to maximize the total of the expected payoff of the packages dispatched (evaluating payoff as described above in "Attacker's Payoff Function"), constrained by the available forces.

Here we formalize the IP problems performed in the LINGO™ model within SLAACM. The main idea behind using an IP tool is to solve the following optimization problem: the attacker wishes to come with optimal attacker packages based on a determined payoff function (this is the first optimization). Then, the defender aircraft that are sophisticated enough to have a priori knowledge of what attacker packages are coming may optimize their defending packages accordingly (the second optimization).

A basic defense scenario attacker package consists of 4 advanced escorts, 4 close-in escorts, and 4 bombers. Scenarios for Attack SLAACM also include SEADs and varying numbers of escorts and bombers.

Since some fighters may be bombers, for the basic air defense SLAACM scenario, we could have a set of T possible attacker packages, where, given R types of fighters and bombers, we could have $|T| = R^3$. In reality, however, most mathematically feasible combinations of packages are unrealistic for warfare (one would not, for example, dispatch a package of heavy bombers escorting fighters). Since the number of reasonable attacker packages is much smaller than the complete enumeration, we assume those undesirable packages are removed in advance of the optimization; then, we limit ourselves to those remaining, reasonable possibilities in the integer program. The degree to which one wishes to limit the number of potential packages depends primarily on the size of the integer program, which is a function of the size of R.

In Attack SLAACM, an attack package may consist of SEAD, escort, and bomber aircraft. If the package has S types of SEAD aircraft, E types of escort aircraft, and B types of bomber and fighter-bomber aircraft, then $|T| = SEB$. Of course, here too not all combinations are feasible militarily.

We define a set of attacker packages as $I \subseteq T$. We define variable r_{ij} as the number of aircraft of type j used in attacker package i. To calculate its payoff, the attacker plans its attack assuming each package will be confronted by one specific defender aircraft type, usually the most numerous of the defender aircraft. Based on the payoff functions described previously, each attacker package has a certain payoff, denoted as p_i. We denote the number of attack aircraft of type j as n_j.

At this point, the attacker can solve for its optimal attack strategy, denoted by the following integer optimization problem:

$$\max \sum_{i=1}^{I} p_i x_i$$

$$\text{s.t.} \sum_{i=1}^{I} r_{ij} x_i \le n_j \quad \forall j \in J$$

$$x_i \in \{0,1,2,...\}.$$

We solve for the values of x_i, which represent the optimal number of packages of type i sent. This problem is a variant of the classic knapsack problem. The solution to this integer programming problem provides the optimal set of attacker packages dispatched by the attacker.

HEURISTIC SOLUTION

The exact integer programming problem can be solved using an IP software tool such as LINGO. In SLAACM, the user can specify that the exact solution technique be used, provided the user's PC has a LINGO license. Although utilizing an optimization package is the ideal circumstance—it can guarantee optimal solutions if run to completion—many analysts will not have LINGO available on their PCs. For this reason, SLAACM includes a greedy heuristic, using the attacker and defender payoff functions, that attempts to approximate the optimum solution. The heuristic has been shown to yield the optimum solution when the attacker and defender aircraft types and numbers are limited. However, as the choices grow, and particularly when the attacker has options of using aircraft as fighters or fighter-bombers, the packages selected by the heuristic will deviate significantly from an IP solution. This difference will be compounded when defenders are smart and also optimize. (When defender aircraft are dominant, the different package choices may produce only small differences in the outcome statistics.)[1]

The greedy heuristic calculates what might be considered the "obvious" answer. It is considered greedy because it takes, without more holistic considerations, packages with the largest payoffs first. This approach can be suboptimal because it does not consider (or mathematically eliminate from consideration) all combinations of package types.

As we noted earlier, the payoff for attacker package i is denoted as p_i. The heuristic first orders the packages from greatest to least. Without loss of generality, we denote the largest payoff as package p_1, the second largest as p_2, and so on. Thus, we have $p_1 \le p_2 \le ... \le p_I$. Again, we define n_j as the number of attack aircraft of type j. The heuristic can then be described by the following procedure.

Step 0: Set $\tilde{n}_j = n_j$ for all j.

Initialize $x_i = 0$ for all i.

Set $i = 1$.

[1] For further information on integer programming and greedy heuristics, the interested reader should consider one of the many textbooks on integer programming and optimization, such as L. Wolsey and G. Nemhauser, *Integer and Combinatorial Optimization* (New York: Wiley, 1988).

Step 1: If $\tilde{n}_j - 1 \geq 0$ for all j such that $r_{ij} > 0$, then {

$\tilde{n}_j = \tilde{n}_j - 1$ for all j such that $r_{ij} > 0$

$x_i = x_i + 1$

Go to Step 1

} Else {

Go to Step 2 }

Step 2: $i = i + 1$.

If $p_i \leq 0$, quit.

If $i = I + 1$, quit.

Else, Go to Step 1.

In essence, this heuristic keeps selecting the package with the greatest payoff until it exhausts one of the package's aircraft types (i.e., "hits a constraint"). It then looks for the package with the next greatest payoff (assuming all its aircraft types are still available). It keeps selecting that package until one of its aircraft types is exhausted, and so forth. It continues this procedure until there are no more packages with positive payoffs for which the advanced escort, close-in escort, or bomber aircraft types have not been exhausted.

It is important to reemphasize that this approach will not necessarily yield an optimal solution. Indeed, for sample campaigns modeled in SLAACM, the above heuristic typically does *not* yield the same solution for the attacker as the IP solution solved to optimality in LINGO. For the purposes of SLAACM, it appears that it is rare for the heuristic solutions to differ by more than a few percent from optimality. Nevertheless, theoretical bounds on such a heuristic for a *general* IP problem can be rather far from optimality, and those rare instances of significant difference may be militarily significant.

Exact Solution

The attacker's IP problem may be solved exactly using an integer programming solver. LINGO is a commercially available general linear, nonlinear, and integer programming solver tool.[2] The version embedded in SLAACM is Extended LINGO™ 9.0, which does not limit the number of variables (integer or otherwise) or constraints. Of course, for a large enough problem, even sophisticated IP solvers will have trouble producing the optimal solution in reasonable amounts of time. The example problems run in SLAACM, but rarely take more than a couple of seconds on a modern desktop PC.

[2] Lindo Systems, Inc., http://www.lindo.com/.

The IP model below is written in LINGO's modeling language and implemented in SLAACM for the attacker's optimization. The data and variables are defined elsewhere in SLAACM. For sake of clarity, we omit portions of the data, variable definitions, and input/output declarations.

```
MODEL:
SETS:
RPACKAGE/1.rposize/: rpo, X;
REDAC/1.rac/: rob;
RMATRIX(RPACKAGE,REDAC): rcnst;
ENDSETS
! INTEGER PROGRAMMING PROBLEM;
MAX = VALUE;
VALUE = @SUM(RPACKAGE: rpo*X);
@FOR( REDAC(J):
@SUM(RPACKAGE(I): rcnst(I,J)*X(I)) <= rob(J) );
@SUM(RPACKAGE(I): X(I)) <= (redLim/12);
@FOR( RPACKAGE: @GIN(X));
END
```

Determining the Defender's Response

Certain defender aircraft are identified by the user as smart. They are assumed to know the composition of the attacker packages before intercepting them and to coordinate their defense to maximize its total value (value computations are described above in the section "Defender's Payoff Function"). The remaining defender aircraft encounter attacker packages randomly. The following subsections describe how SLAACM determines the set of engagements made by smart defenders and by the remaining defender aircraft.

DEFENDER'S INTEGER PROGRAMMING PROBLEM

Given the set of attacker packages, the smart defenders determine which types to engage by maximizing the total defender payoff, subject to constraints imposed by their order of battle. As stated previously, the defender responds with optimal defenses of the attacker if a given aircraft is smart enough to have a priori knowledge of the attacker package composition. We assume the attacker has obtained its solution before the defender calculates its optimal response. We denote the optimal solution for the attacker—that is, $x^* = (x_1^*, x_2^*, ..., x_I^*) = (a_1, a_2, ..., a_I)$—as constraints for the defender's optimization. In other words, a_i is the number of attacker packages of type i.

The defender has his own payoff function for each type of potential attacker package. We denote $\hat{p}_{ik}$ as the payoff to the defender for intercepting an attacker package i with the defender n-ship package of aircraft type k (in the traditional air defense scenario, $n = 4$). Again, the components of the payoff function have been described previously; they include a loss-aversion factor for the defender, along with positive payout for expected attacker kills and bomb damage prevented. We denote the set of defender aircraft types with a priori knowledge of attacker packages as $K' \subseteq K$. We denote the number of defender aircraft of type $k \in K'$ as m_k. We then formulate and solve the following IP problem for the defender. In short, we want to solve for the optimal number of defender n-ship package of type k that intercepts attacker package i, denoted by y_{ik}.

$$\max \sum_{i=1}^{I} \sum_{j=1}^{K'} \hat{p}_{ik} y_{ik}$$

$$\text{s.t.} \sum_{i=1}^{I} n y_{ik} \le m_k \quad \forall k \in K'$$

$$\sum_{k=1}^{K'} y_{ik} \le a_i \quad \forall i \in I$$

$$y_{ik} \in \{0,1,2,...\} \qquad \text{[Eq. 4-3]}$$

The IP model written in LINGO's modeling language is embedded into SLAACM. LINGO solves the IP problem and returns the solution to variables defined in SLAACM. At this point, the remaining not-smart defender aircraft, denoted by $K'' \subseteq K$, where $K' \cup K'' = K$ and $K' \cap K'' = \emptyset$, engage the attacker packages that were not selected by the smart defenders. That is, the defender aircraft without a priori knowledge engage randomly the attacker packages that remain after the above IP problem is solved; in other words, those defender aircraft engage a set of attacker packages $(a'_1, a'_2, ..., a'_I)$, where $a'_i \equiv \max\left(a_i - \sum_{k=1}^{K} y_{ik}, 0 \right)$. Those encounters are described in a later section.

Heuristic Solution

As stated previously, SLAACM allows the user to specify whether to use LINGO to solve the IP formulation or to use a greedy heuristic. If SLAACM uses the heuristic (or solves the IP problem in LINGO), it must do so for both the attacker's and defender's optimization.

The greedy heuristic for the defender works similarly to the heuristic for the attacker. Following the notation from the IP formulation, we denote $\hat{p}_{ik}$ as the payoff to the defender for intercepting an attacker package i with the defender n-ship package of aircraft type k. We then order the $\hat{p}_{ik}$ from greatest to least over all i and k. We denote that new set of payoffs as $\hat{p}_1 \geq \tilde{p}_2 \geq ... \geq \tilde{p}_B$, and note their associated i and k.

It is worth noting that defenders that are not members of the local air defense (LAD) are presumed to be loss averse, and it is not uncommon for some $\hat{p}_{ik} < 0$; in other words, intercepting certain attacker packages may have negative payoffs for the defender. This result occurs because loss-averse defenders suffer a large penalty for their expected losses. Neither the heuristic nor the LINGO model uses any aircraft whose payoff values result in $\hat{p}_{ik} \leq 0$, since selecting those combinations would of course reduce the defender's objective function.

Again, we denote the number of attacker packages of type i sent as a_i. As before, we denote the number of defender aircraft of type $k \in K'$ as m_k. The set of decision variables (number of defender n-ship packages of type k that intercept attacker package i) is denoted by y_{ik}. As with the attacker, we wish to solve a multidimensional knapsack problem.

We can then describe the optimization heuristic we employ by the following algorithm:

Step 0: Set $\tilde{m}_k = m_k$ for all k.

Set $\tilde{a}_i = a_i$ for all i.

Initialize $y_{ik} = 0$ for all i, k.

Begin with $b = 1$.

Step 1: Let $\hat{i}, \hat{k}$ be the values for $\hat{p}_{\hat{i}\hat{k}}$ such that $\hat{p}_{\hat{i}\hat{k}} = \hat{p}_b$

If $\tilde{m}_k - n \geq 0$ and $\tilde{a}_i > 0$, Then {

$\tilde{m}_k = \tilde{m}_k - n$

$\tilde{a}_i = \tilde{a}_i - 1$

$y_{ik} = y_{ik} + 1$

Go to Step 1

} Else {

Go to Step 2}

Step 2: $b = b + 1$.

If $p_b \leq 0$, quit.

If $b = B + 1$, quit.

Else, Go to Step 1.

In essence, this heuristic finds the solution with the largest payoff to the defender; notes the associated attacker and defender package types (i.e., i and k, respectively); checks to see if n defender aircraft of that type are available; checks to see if the attacker sent any packages of that type; and chooses those defender aircraft. It sends the lesser of the number of defender aircraft of that type still available or the number of attacker packages of that type that were sent. Then, the heuristic looks for the next highest payoff and repeats the above process until all smart defender aircraft with positive payoffs are assigned or until there are no more attacker packages to which to assign a smart defender 4-ship package.

This process is similar to the one for the attacker heuristic, but it needs to be restated that this method does not guarantee optimality for the defender, because it does not consider the optimization of the entire defender fleet over all possibilities. Instead, it looks for a greedy local solution. That said, both the defender's and the attacker's heuristics are computationally easy to implement. Indeed, without the ability to solve moderately complex integer programs, these techniques can form good approximations. However, neither heuristic is able to provide tight bounds on its performance, at least for a general knapsack problem. Therefore, the preferred method, given the choice of the two methods, is to solve both optimizations exactly using an IP solver.

Exact Solution

The defender's IP problem may be solved exactly using LINGO. In general, the defender's optimization is a little more complex than the attacker's, because it requires assigning defender packages to attacker packages. While the attacker simply creates optimized attacker packages, the defender is effectively optimizing over two indices (which defender packages should be sent, and which attack packages should be intercepted). In most cases, however, the number of variables in the defender's optimization is not larger than the number of decision variables in the attacker's optimization, because the attacker sends only a small fraction of its potential package types. In both cases, LINGO obtains the optimal solution to the IP problem in a few seconds on a standard, modern desktop PC for most sample problems in SLAACM.

The IP model written below in LINGO's modeling language solves the defender's optimization problem. It is similar in syntax to the attacker's LINGO model. Again, the data and the variable names are defined elsewhere in SLAACM. For the sake of clarity, we omit most data and variable definitions, as well as input/output calls.

```
MODEL:
SETS:
PPACKAGE/1.bsize/: sboob;
REDAC/1.rsize/: rsent;
PMATRIX(REDAC, PPACKAGE): bpo, X;
ENDSETS
```

```
! INTEGER PROGRAMMING PROBLEM;
! Objective Function;
MAX = VALUE;
VALUE = @SUM(PMATRIX: bpo*X);
! subject to;
@FOR( PPACKAGE(J): @SUM(REDAC(I): X(I,J)*ndef) <= sboob(J) );
@FOR( REDAC(I): @SUM(PPACKAGE(J): X(I,J)) <= rsent(I) );
@FOR(PMATRIX(I,J): @GIN(X(I,J)));
END
```

Random Engagements by Not-Smart Defender Aircraft

The optimal engagements made by smart defender aircraft may leave some attacker packages unengaged. For determining the packages that need to be intercepted by not-smart defenders, we assume, optimistically, that an attacker package intercepted by a smart attacker flight does not continue the engagement. This assumption is reasonable when the smart defenders are much stronger than any of the attacker's fighters and bombers.

Other SLAACM versions have considered the opposite case, which may be implemented in SLAACM with simple reprogramming, in which the attacker forces that survive the smart defense reorganize into optimal attacker packages and continue the assault.

The defender aircraft not identified as smart encounter attacker packages randomly. They do not conduct optimized defensive encounters. There is still an element of battle management for these aircraft, however, and that is the efficiency with which they can locate and engage the attacker packages.

With perfect battle management, if the number of defending flights is at least as large as the number of attacker packages, every attacker package would be intercepted. The intercepting aircraft types would be random, however, rather than optimal. If the number of attacker packages is larger than the number of defending flights, every defending flight would engage an attacker package.

With less-than-perfect battle management, not every attacker package would be intercepted in the former case, and not every defending flight would be engaged in the latter case. SLAACM's battle management feature allows the user to enter a "goodness" parameter that characterizes the effectiveness of battle management for defender aircraft that are not smart. The following paragraphs explain how that parameter affects SLAACM's calculations.

Let K distinct types of defender flights deal with an attack by J distinct types of attacker packages. The defenders are assumed not to know the makeup of individual attacker packages before interception, so that the type of attacker package engaged by a given defending flight is the result of random selection from the set of attacker packages.

Let m_i be the number of defender flights of type i, and let n_j be the number of attacker packages of type j. Then the total number of attacker packages, N, and the total number of defending flights, M, are given respectively by

$$N = \sum_{1}^{J} n_j ; \quad M = \sum_{1}^{K} m_i . \qquad \text{[Eq. 4-4]}$$

First we consider the case of perfect battle management. If $M \geq N$, then every attacker package will be intercepted. Not all defending flights engage; in this simple analysis, we assume that the fraction N/M of each defending flight type engages.[3]

Then, a measure of $\overline{E}_{ij}$, the central tendency of the number E_{ij} of m_i vs. n_j engagements, is

$$\overline{E}_{ij} = \frac{N}{M} m_i \frac{n_j}{N} = \frac{m_i n_j}{M} . \qquad \text{[Eq. 4-5]}$$

For this simple analysis, we take $\overline{E}_{ij}$ to be the number of m_i vs. n_j engagements. Summing $\overline{E}_{ij}$ over j shows that the fraction of each defending flight type engaged is N/M, as it should be; summing over i shows that all attacking flights of each type are engaged.

When $M < N$, every defending flight engages, but not all attacker packages can be engaged. The estimate for $\overline{E}_{ij}$ analogous to the one given in Equation 4-5 is

$$\overline{E}_{ij} = m_i \frac{n_j}{N} = \frac{m_i n_j}{N} . \qquad \text{[Eq. 4-6]}$$

Summing this estimate for $\overline{E}_{ij}$ over i shows that the fraction M/N of each attacker package type is engaged; summing over j shows that every defending flight of each type is engaged.

THE BATTLE MANAGEMENT EFFICIENCY FACTOR

In some cases, M and N are both $O(10^2)$. For such large M and N, providing this "perfect" defender's battle management could exceed the capabilities of available systems. We account for the limitations of the defender's battle management with a simple adaptation of the "perfect" case. We make the adaptation by introducing "ghost" attacker packages or defender flights. A defender flight that engages a

[3] We recognize that this simple analysis treats a complex combinatoric problem crudely.

ghost attacker package does not actually engage; an attacker package engaged by a ghost defender flight is not actually intercepted.

When $M \geq N$, we introduce m_g defending flights of type "ghost," and proceed as in the above analysis (where the defender has perfect battle management). This gives us a new estimate for $\overline{E}_{ij}$:

$$\overline{E}_{ij} = \frac{m_i n_j}{M + m_g} = \frac{m_i n_j}{M} b_1 \qquad \text{[Eq. 4-7]}$$

where

$$b_1 \equiv \frac{M}{M + m_g}. \qquad \text{[Eq. 4-8]}$$

When $M < N$, we introduce n_g attacker packages of type "ghost," and find

$$\overline{E}_{ij} = \frac{m_i n_j}{N + n_g} = \frac{m_i n_j}{M} b_2 \qquad \text{[Eq. 4-9]}$$

where

$$b_2 \equiv \frac{N}{N + n_g}. \qquad \text{[Eq. 4-10]}$$

Noting the similarity of Equations 4-8 and 4-10, we introduced into SLAACM the simple model

$$\overline{E}_{ij} = \frac{m_i n_j}{M} b \qquad \text{[Eq. 4-11]}$$

and allow the user to choose the "battle management efficiency factor" $b, 0 < b \leq 1$.

Once seen, Equation 4-11 is so simple, and seemingly obvious, that it is fair to ask why we did not simply introduce it into SLAACM without analysis. The reason is that we wanted to understand what that simple choice implied about how the battle proceeded. The analysis given here supplies that understanding.

Chapter 5
How SLAACM Works: SLAACM Campaign Calculations

This chapter describes the calculation protocols used by SLAACM for campaign analysis.

DETERMINING AND PROPAGATING LOSS AND DESTRUCTION STATISTICS

The calculations described in Chapter 4, under “Determining Attacks and Defenses,” generate the sets of engagements that take place in each day’s fighting. There are then specified numbers of engagements between a specific attacker package (with its specified aircraft types for the advanced escorts, close escorts, SEAD aircraft, and bombers) and a flight of 2, 4, or 8 defenders of a specific type, and SAMs. SLAACM uses the method described in Chapter 2, under “Composite Engagements,” to generate probabilities for all outcome states of each kind of engagement that occurs.

Then, to evaluate attacker losses, SLAACM calculates expected values of the losses for advanced escorts, close escorts, SEADs, and bombers. These expected values are accumulated for each attacker aircraft type, for engagements between smart defenders and the attacker packages, and for engagements between non-smart defenders and the attacker packages not engaged by smart defender aircraft.

The total expected losses for each attacker type are rounded to integers and subtracted from the attacker order of battle. SLAACM then generates the attacker order of battle for the following day using the surviving attackers plus any reinforcements scheduled to arrive for that day. Reinforcements are a user input to SLAACM.

Expected values of surviving bombers are multiplied by bomber payloads and rounded to the nearest integer, to give tons-of-bombs results. Tons delivered by bombers with relative circular errors probable (CEPs) less then a user-definable level are accounted for separately from tons delivered by other bombers and identified as “smart bombs.”

In treating defender losses, SLAACM considers measures of dispersion as well as measures of central tendency. The probability distributions for outcome states of the engagements occurring each day are used to calculate both means and variances of the losses of each type of defender aircraft, for each day. Mean losses, rounded to the nearest integer, are subtracted from the defender's order of battle and, after increments from any scheduled reinforcements, are posted as the defender's order of battle for the next day.

Variances in the losses of each type of defender aircraft are accumulated day-by-day, and the total variances are reported to give a measure of dispersion in defender losses.

This procedure neglects a potentially important source of dispersion, the effect of propagating losses other than the means of both attacker and defender losses. The present dispersion calculations are used for checking the validity of using only mean values. Propagating even a limited number of results, such as maximum, mean, and minimum losses, leads to explosive growth of computations when all combinations are considered. There may value to propagating only the strings of worst cases and best cases to establish bounds for the mean results, especially when forces are evenly matched. For the cases in which defender aircraft are always much stronger than the attacker aircraft, there is little dispersion in attacker losses—nearly all attacker aircraft that are engaged are lost, nearly all the time—so the significant dispersion in each day is the dispersion in defender losses. But, again by virtue of the defender's much greater strength, the dispersions never lead to significant probabilities of the defender losing substantial fractions of his forces, except near the ends of campaigns in which the defender is losing. The converse is true for strong attackers.

SLAACM's Sequence of Calculations

This section describes SLAACM's step-by-step calculations. It explains how SLAACM uses the methods described above to model air-to-air campaigns.

User Inputs

The SLAACM user inputs attacker and defender orders of battle for the first day of fighting and lists any scheduled reinforcements.[1] The following subsections summarize the inputs, focusing on their meanings and use. (Chapter 6 details the methods for entering data into SLAACM.)

Inputs for the Defender Force

The user may identify certain defender aircraft as smart. Smart defender aircraft can determine the composition of attacker packages before intercepting them, and can coordinate their defensive response among themselves to provide optimal defenses.

[1] "Day" is used to signify a set of attack and defense sorties. There could be more than one set of sorties on a single calendar day, or there could be gaps of calendar days between attacks.

Defender aircraft based long distances from the battle space will have to be rationed and rotated to maintain continuous defensive coverage. The user may select a combat air patrol (CAP) factor for any defender aircraft type, so that only a specified fraction of the inventory of that defender type is available to meet an attack.

The user may identify certain defender aircraft as local air defenders (LADs). SLAACM treats LADs differently from other defender types when considering defender dispatch options. Normally, the defender values the life of its forces and will not dispatch a defender type if engaging a attacker package gives a negative value of the defender's payoff function. Defender types designated as LADs, however, will be dispatched in all cases.

INPUTS FOR THE ATTACKER FORCE

The user inputs the bomb payload of each attacker aircraft and specifies whether the aircraft is to be used as a fighter only, or as a bomber only, or as either a fighter or a bomber. The user also inputs the ratio of the CEP with which bomber aircraft can deliver bombs, to the reference CEP. As discussed in Chapter 3, the default ratio is 1.0 for dumb bombs and 0.1 to 0.4 for smart bombs. If delivery of cruise missiles is of interest, the user inputs the number of cruise missiles that can potentially be carried by each aircraft.

The attacker's order of battle may be so large that airfield capacity and battle management capability may preclude using all forces. The user may specify a maximum number of attacker aircraft that can be dispatched in a given attack.

The attacker optimizes its dispatch choices with the assumption that each package encounters a specified defender fighter type. The user identifies this type.

INPUT FOR RELATIVE CAPABILITIES OF ATTACKER AND DEFENDER AIRCRAFT

Currently, SLAACM models engagements with the M vs. N probabilistic engagement model, and it assumes that outcome probabilities have their long-time limiting values as described in Chapter 2 under "Calculations for the Long-Time Limit." Thus only one aircraft parameter—the ratio κ of the parameter of the defender time-between-kill distribution to that parameter for the attacker time-between-kill distribution—affects outcomes.

Values of κ parameters may not be readily available, and SLAACM includes a utility to calculate κ from the defender-to-attacker "loss ratio," that is, the generally available output from simulations or combat. (Some simulations provide exchange ratios, which are ratios of defender relative losses to attacker relative losses. Multiplying an exchange ratio by the ratio of the initial number of defenders to the initial number of attackers gives the loss ratio.) The user inputs the loss ratios and the source scenario, such as 4 defenders vs. 8 attackers with defenders breaking away after 2 losses. SLAACM then determines the value of κ that gives the observed loss ratio and uses it in subsequent calculations. SLAACM uses bisection to find κ values correct to four significant figures.

With all these inputs completed, the user enters the number of days to run the campaign, specifies the battle management efficiency parameter, as described in Chapter 4 under “The Battle Management Efficiency Factor,” and instructs SLAACM to use either heuristic optimization or exact optimization with LINGO. SLAACM then performs the sequence of calculations described below.

Steps in SLAACM’s Calculations

SLAACM first calculates values of the attacker payoff function, as described in Chapter 4 under “Attacker’s Payoff Function,” for all possible choices of advanced escorts, close escorts, and bombers that can make up an attacker assault package. There will often be hundreds of such choices. SLAACM stores the payoff values for use in later optimizations.

SLAACM then begins the day-by-day calculations of outcomes in the campaign. Using either the heuristic or the exact solution, as specified by the user, SLAACM determines the set of attacker packages that maximizes the total attacker payoff function, subject to constraints imposed by the attacker’s available forces and to the maximum number of attacker aircraft that can be dispatched in 1 day as specified by the user. These calculations are described in Chapter 4 under “Determining Attacks and Defenses.”

SLAACM considers this set of attacker packages to determine the optimal defense by the smart defender aircraft. After calculating the defender payoff, as described in Chapter 4 under “Defender’s Payoff Function,” for each possible combination of smart defender defense with an attacker package type actually dispatched, SLAACM determines the defender dispatch option that maximizes the total defender payoff, constrained by the available defender forces and the CAP values input by the user. SLAACM uses either the heuristic or the exact solution of the resulting IP problem, as directed by the user.

As described above under “Determining and Propagating Loss and Destruction Statistics,” SLAACM then evaluates the losses to all forces from engagements between the attacker packages and the smart defenders. Then SLAACM calculates the results of engagements between the attacker packages that were not engaged by smart defender aircraft. Only those attacker packages not engaged by smart defenders are available for engagements with the not-smart defender aircraft.

SLAACM calculates the defender payoff for each combination of not-smart defender aircraft and a attacker package type that was not intercepted by a flight of smart defenders. Then, following the procedure described in Chapter 4 under “Random Engagements by Not-Smart Defender Aircraft,” and considering the user’s input of battle management efficiency factor, SLAACM determines the numbers of engagements between the several types of not-smart defenders and the attacker package types available to them. As described above under “Determining and Propagating Loss Statistics,” SLAACM updates the losses of all aircraft types, calculates the tons of smart and not-smart bombs delivered, and, in view of any scheduled reinforcements, updates attacker and defender orders of battle for the next day of the campaign.

After calculating outcomes for the number of days specified by the user, SLAACM produces daily records of attacker and of defender orders of battle, of attacker losses, of defender losses and their variances due to engagements, of tons of smart and of not-smart bombs delivered, and of cruise missiles delivered. As described in Chapters 6 and 7, SLAACM also produces graphs of attacker and of defender drawdowns, the quantities and types of bombs delivered, and information on “who shot whom.”

Chapter 6
Users Guide to Classic SLAACM

Classic SLAACM is housed in a Microsoft Excel workbook with extensive Visual Basic for Applications code. The current version of Classic (Defender) SLAACM was updated to use object-oriented programming principles. In addition, it has an option to conduct integer programming by linking to an external optimization program called LINGO.[1] Using Excel to house SLAACM has several advantages. Most analysts are familiar with the Excel environment and will be able to navigate it easily. Other than acquiring the SLAACM workbook, users do not need to install additional software unless they want to run integer programming optimizations in LINGO. Finally, users can easily copy and paste output values and charts into their preferred presentation formats for displaying results to others.

This chapter describes how to operate Classic SLAACM. It begins with a description of the model inputs. It then describes how to run the model once all of the input parameters are set up. Next, the chapter describes the outputs reporting SLAACM's results. The last section addresses how the outputs are analyzed to determine the effects of the various engagement scenarios.

INPUTS

SLAACM's input parameters are defined in five worksheets:

- BlueSupply, which is used to specify the quantities of Blue aircraft available on each day of the campaign.
- RedSupply, which is used to specify the quantities of Red aircraft available on each day of the campaign.
- BlueOOB, which is used to define Blue's order of battle and other pertinent parameters. BlueOOB is the main sheet from which SLAACM runs are conducted. (SLAACM also has a RedOOB worksheet, but it is purely an output worksheet and does not receive any user inputs.)
- ExRatios, which is used to specify engagement parameters such as loss ratios and breakpoints.
- CM, which is used to specify the number of cruise missiles that a Red aircraft could carry.

We discuss each sheet below and describe the proper way to complete each one.

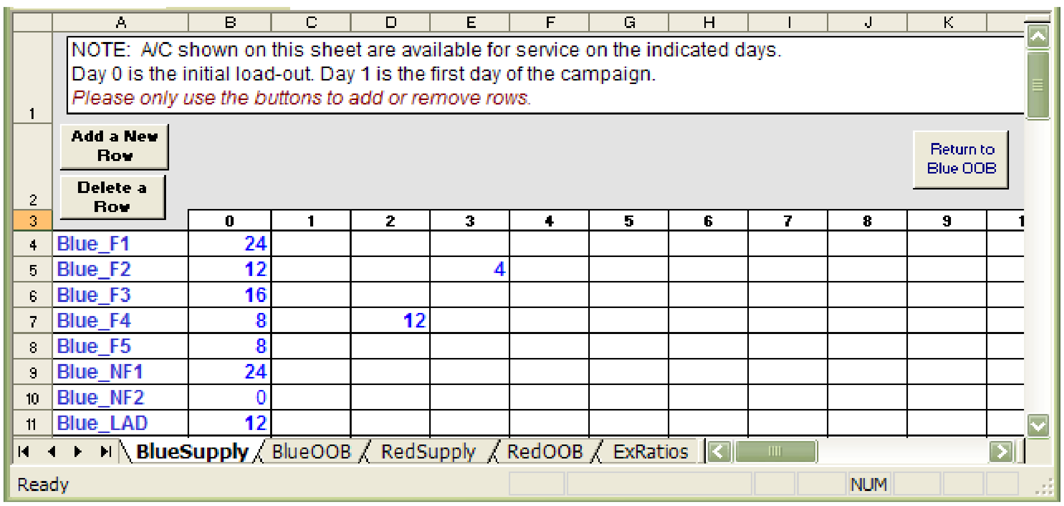
[1] For more information on LINGO, see http://www.lindo.com/products/lingo/.

BlueSupply and RedSupply Worksheets

For each force, Blue and Red, the user will need to set up the initial quantities of aircraft available and the reinforcements available on each day of the campaign. The user will need to type those quantities into the appropriate worksheets, called BlueSupply and RedSupply.

Figure 6-1 is an example of the BlueSupply worksheet. As shown in the figure, the worksheet has a row for each Blue aircraft type and a column for each day of the campaign. Initial aircraft quantities are entered into the column labeled 0. Reinforcements may be provided for subsequent days of the campaign. In the figure, we see that reinforcements are entered for Day 2 and Day 3.[2]

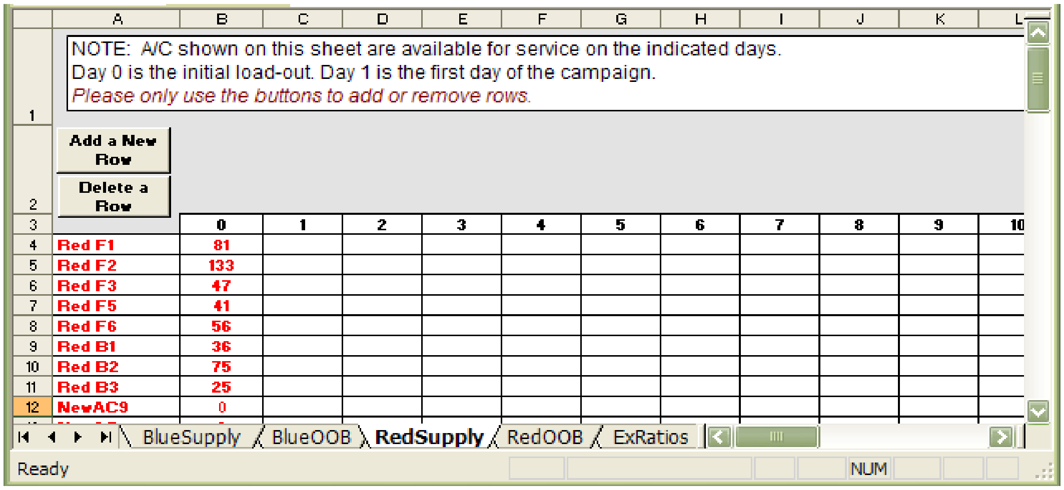
Figure 6-1. Classic SLAACM: BlueSupply Worksheet



The RedSupply worksheet is designed in the same way as the BlueSupply sheet. Figure 6-2 is an example. In the figure, we see that only initial aircraft quantities are provided and no reinforcements are specified.

[2] Note that a day is used in the model as a convenient designator to represent an individual attack/defense sortie. In an actual campaign, there may be more than one sortie on a particular calendar day, or, conversely, sorties may be distributed among several calendar days.

Figure 6-2. Classic SLAACM: RedSupply Worksheet



The BlueSupply and RedSupply worksheets can accommodate up to 20 aircraft types each. If new rows are needed to accommodate additional aircraft types, they can added using the “Add a New Row” button.

Caution!

Any time a row is added or deleted, it is imperative that the buttons on the BlueSupply and RedSupply worksheets are used to add or delete the rows.

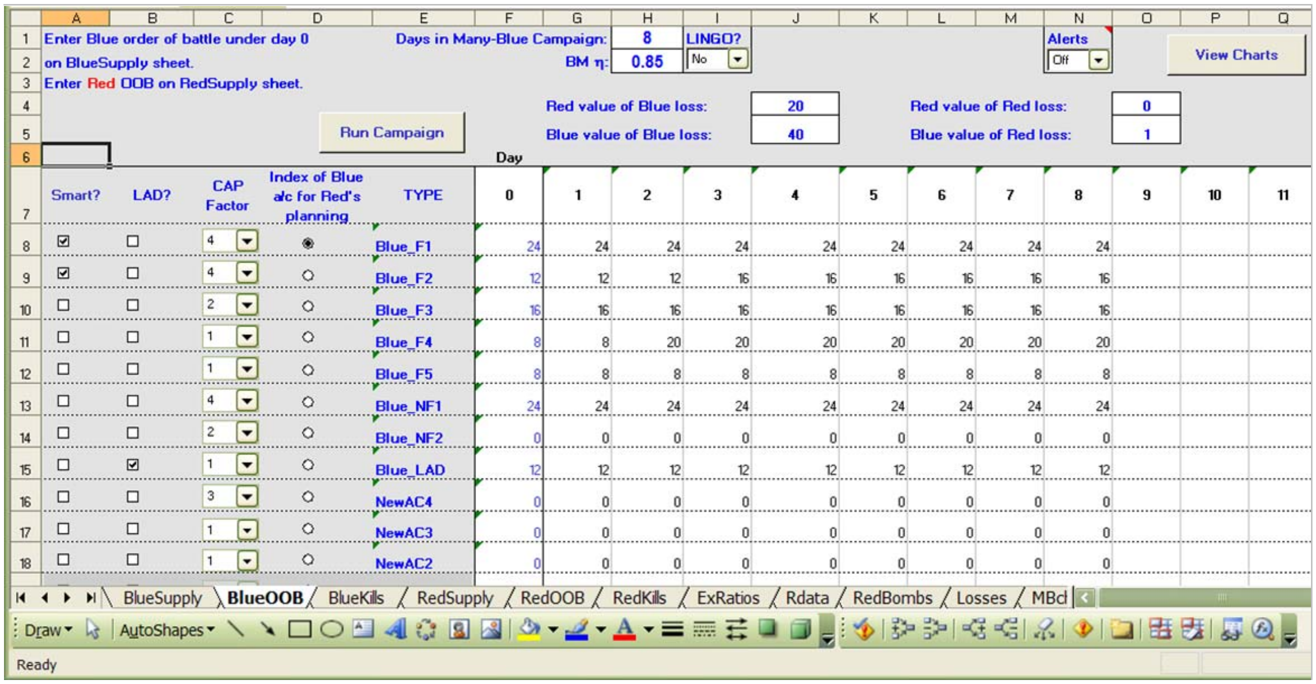
Do not simply insert or delete rows and columns.

Several of SLAACM’s worksheets reference the aircraft types. To keep the worksheets synchronized, the “Add a New Row” and “Delete a Row” buttons must be used. Using typical worksheet approaches to inserting and deleting rows will likely result in model errors or misinterpretation of results.

BlueOOB Worksheet

The worksheet named BlueOOB (which stands for Blue order of battle), is the main worksheet for input parameters for the Blue aircraft. It is also the worksheet used to launch campaign analyses. BlueOOB also serves as an output worksheet, as does the RedOOB worksheet; the output portion of BlueOOB will be discussed in the output section of this chapter. Figure 6-3 shows the BlueOOB worksheet.

Figure 6-3. Classic SLAACM: BlueOOB Worksheet



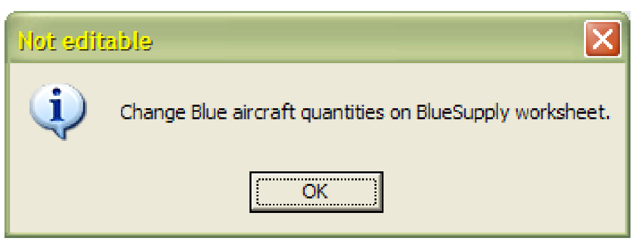
Smart?	LAD?	CAP Factor	Index of Blue a/c for Red's planning	TYPE	0	1	2	3	4	5	6	7	8	9	10	11
☑	☐	4	◉	Blue_F1	24	24	24	24	24	24	24	24	24			
☑	☐	4	○	Blue_F2	12	12	12	16	16	16	16	16	16			
☐	☐	2	○	Blue_F3	16	16	16	16	16	16	16	16	16			
☐	☐	1	○	Blue_F4	8	8	20	20	20	20	20	20	20			
☐	☐	1	○	Blue_F5	8	8	8	8	8	8	8	8	8			
☐	☐	4	○	Blue_NF1	24	24	24	24	24	24	24	24	24			
☐	☐	2	○	Blue_NF2	0	0	0	0	0	0	0	0	0			
☐	☑	1	○	Blue_LAD	12	12	12	12	12	12	12	12	12			
☐	☐	3	○	NewAC4	0	0	0	0	0	0	0	0	0			
☐	☐	1	○	NewAC3	0	0	0	0	0	0	0	0	0			
☐	☐	1	○	NewAC2	0	0	0	0	0	0	0	0	0			

Caution!

Rows or columns should never be deleted from this worksheet!

The aircraft listed in the BlueOOB worksheet are linked to those that were shown in the BlueSupply worksheet (Figure 6-1). The Day 0 quantities are also linked to those in the BlueSupply worksheet. The user who attempts to click the aircraft names or Day 0 quantities on the BlueOOB sheet will receive the warning shown in Figure 6-4.

Figure 6-4. Warning: Do Not Alter Aircraft of BlueOOB Worksheet



The first four columns on the BlueOOB worksheet define aircraft-specific input parameters for Blue:

- *"Smart?" column.* The user may use this column to identify specific Blue aircraft as smart. Smart Blue aircraft can determine the composition of Red attack packages before intercepting them and can coordinate their defensive response among themselves to provide optimal defenses. This coordination and optimization is discussed in Chapter 5. Multiple Blue aircraft types may be designated as smart. In Figure 6-3, the first two aircraft are selected to be smart.

- *"LAD?" column.* The user may use this column to identify certain Blue aircraft as local air defenders. SLAACM treats LADs differently from other Blue types when considering Blue dispatch options. In general, a non-LAD Blue has a penalty for its own losses that is significantly greater than a LAD Blue, which presumably is less loss averse due to defending its own territory. Some non-LAD Blues may have a negative payoff function for engaging certain Red packages dispatched. Blue will not dispatch that type. LADs, however, are dispatched in all cases. Multiple Blue aircraft types may be designated as LADs. In Figure 6-3, the eighth aircraft is designated as such. Note that SLAACM also has a separate aircraft type labeled Blue_LAD, which is intended to represent an indigenous "Green" defender. In earlier versions of SLAACM, this Blue_LAD was the only aircraft in the model that placed no value on its own survival. The current version allows all aircraft to be optionally "fearless." This facilitates the analysis of supplying different types of aircraft to the indigenous forces.

- *"CAP Factor" column.* This factor addresses the situations in which Blue fighters are stationed at remote airfields and must divide their forces to maintain continuous combat air patrol coverage over the battle space. The user may select a CAP factor for any Blue aircraft type, so that only a specified fraction of the inventory of that Blue type is available. CAP factor values are chosen from a drop-down box for each aircraft type. Values range from 1 to 10. A CAP factor of 1 indicates that no CAP limitation is placed on that aircraft type. In Figure 6-3, several aircraft have CAP factors greater than 1. For example, the first aircraft has been assigned a CAP factor of 4. That aircraft, Blue_F1, also begins with an initial quantity of 24 aircraft. Therefore, Blue_F1 has only 24/4 = 6 available aircraft on the first day, which means that only one 4-ship package of Blue_F1s can be deployed. In addition, if Blue_F1 losses on subsequent campaign days were to exceed 8 aircraft, Blue could no longer dispatch Blue_F1s, since 15/4 < 4. Thus, Blue_F1 would be unable to maintain a 4-ship package at all times. This assumption is fairly conservative: if there are not enough aircraft to fulfill the combat air patrol at all times, those flights are not dispatched.

- "*Index of Blue a/c for Red's planning*" *column*. Only one aircraft type can be selected for this parameter. In Figure 6-3, the first aircraft type is designated as Red's "planning aircraft." This aircraft type is used by Red to compute its expected payoff function and optimize its dispatch choices. Experience has shown that Red optimization is relatively insensitive to the selection of the Blue type, so normal practice is to select the most numerous Blue type for Red planning.

In addition to the aircraft-specific parameters, several other input parameters must be selected on the BlueOOB worksheet.

At the top of the worksheet, the number of days in the campaign is specified. In Figure 6-3, the number of campaign days is 8. The longer the campaign, the longer it will take SLAACM to run, since each campaign day requires the full breadth of SLAACM computations. Initially, the user may need to start out with a longer campaign and then decrease its length as a steady state in aircraft losses is reached. Eight is a good initial number of days to use since some campaigns will complete earlier and some later, depending on the mixes and capabilities of Blues and Reds specified.

Beneath the number of campaign days is a parameter labeled "BM η." This parameter is the battle management efficiency factor. BM η is a value greater than 0 and less than or equal to 1. Simply stated, a BM η less than 1 indicates the percentage of time that a Blue defender will be unable to find a Red attack package. In Figure 6-3, the efficiency factor is shown as 0.85, which means that 15 percent of the not-smart Blue defenders will chase "ghost" attackers. No losses will result, but in those cases, Blue will not be confronting an actual attacker, so its efficiency is reduced. Since smart Blues know Red's attack package compositions in advance, we assume they have perfect management.

Next to the campaign days and the efficiency factor is a drop-down box labeled "LINGO?" If "No" is selected, as shown in Figure 6-3, SLAACM will use its internal heuristic optimization to calculate engagement results. If "Yes" is selected, LINGO will be used to perform integer optimization (a licensed copy of LINGO software is required to be installed on the PC for this feature to work). Both the heuristic and the exact integer programming solution are described in detail in Chapter 5. If LINGO is used, there are two LINGO input worksheets: "RedOpt" and "BlueOpt." However, those sheets are already complete and should not be altered by the user.

Below the parameters just described are four value boxes:

- Red value of Blue loss
- Blue value of Blue loss
- Red value of Red loss
- Blue value of Red loss.

These parameters establish the value of killing an opponent, as well as the penalty a side takes for losing one of its own. These values, which are applied to fighter aircraft only, are used in the optimization objective function. The Red value of Blue loss represents the benefit to Red of destroying a Blue aircraft, while Blue value of Blue loss is the penalty to Blue for losing one of its own aircraft. Values can be between zero and 100.

In the case shown in Figure 6-3, we see that the value of a Blue loss to Red is 20, while the penalty to Blue for losing its own aircraft is 40. Therefore, Blue's aversion to a loss is twice the value of Red's benefit for obtaining the Blue loss. Conversely, we see in this example that Red is not penalized for a Red fighter loss (the value is zero) and Blue barely achieves a benefit for a Red fighter loss (the value is one).

Another parameter, shown on the far right side of Figure 6-3, is labeled "Escaping bombers complete mission?" The drop-down box allows for either a yes or no selection. If yes is chosen, then bombers that escape attack go on to complete their mission and deliver bombs to their target. If no is chosen, then escaping bombers do not complete their missions.

The final input selection on BlueOOB is a drop-down box labeled "Alerts." The default setting for this feature is "Off." However, if "On" is selected, message boxes will become visible as the analysis runs.

ExRatios Worksheet

The ExRatios worksheet contains a row for each Red aircraft that has been designated on the RedSupply worksheet. The worksheet has several parameter columns, discussed below, and a column for each Blue aircraft labeled "F v. Blue_x" where "F" indicates that Red is a fighter and "x" corresponds to the label provided on BlueSupply. Columns are repeated for each Blue aircraft, except they are labeled "B v. Blue_x" where "B" indicates that Red is a bomber and "x" is as just described. The default setting in SLAACM assumes that loss ratios are for engagements between 4 Blues and 8 Reds, with the Reds and Blues fighting to annihilation. However, those assumptions can be changed on this sheet under "Red Start," "Blue Start," "Red Quit," and "Blue Quit." Red Start and Blue Start correspond to the engagement type, e.g., 8 and 4, respectively. Red Quit and Blue Quit denote the breaking points, e.g., 0 and 0 means that both sides will fight to annihilation. The user will want to specify values that correspond to the conditions under which the loss ratios were obtained.

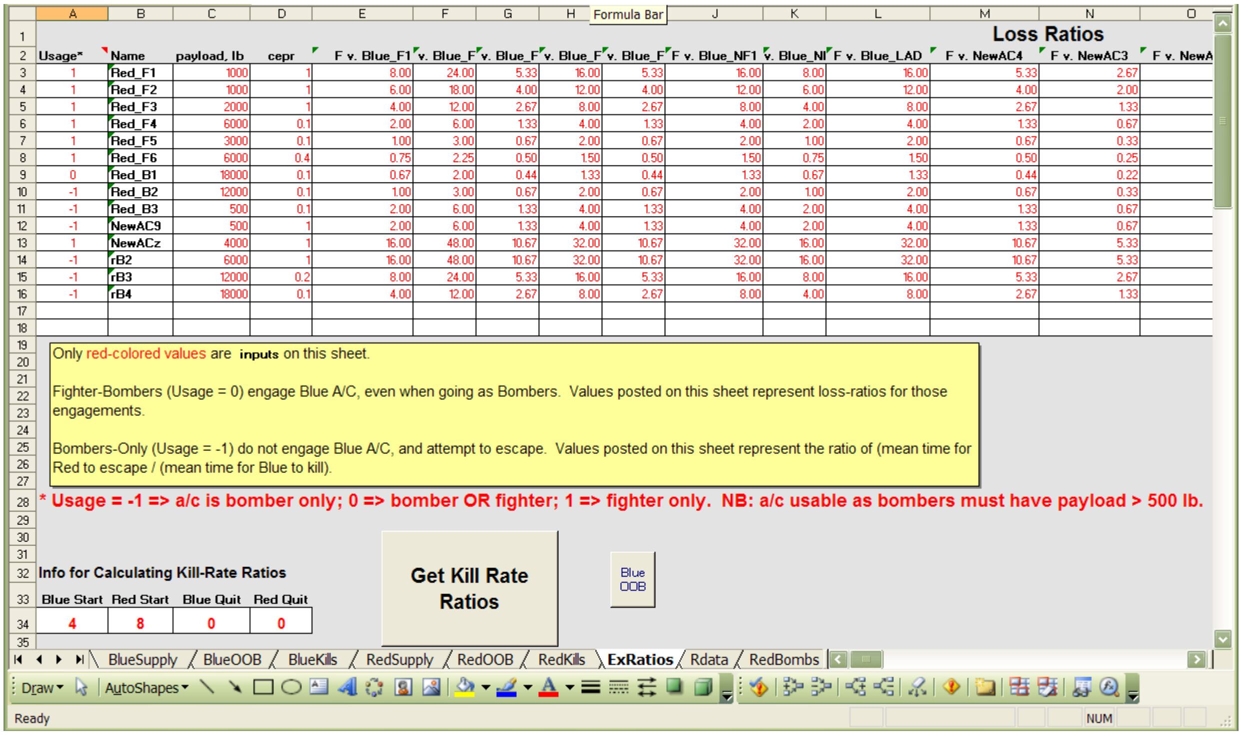
To keep the rows and columns synchronized with the BlueSupply and RedSupply worksheets, it is imperative that no rows or columns on ExRatios be inserted or deleted.

Caution!

As stated earlier, use the add and delete buttons on the BlueSupply and RedSupply worksheets to add or delete rows.

Figure 6-5 is a screenshot of the ExRatios worksheet.

Figure 6-5. Classic SLAACM: ExRatios Worksheet



Usage*	Name	payload, lb	cepr	F v. Blue_F1	v. Blue_F	v. Blue_F	v. Blue_F	v. Blue_F	F v. Blue_NF1	v. Blue_NI	F v. Blue_LAD	F v. NewAC4	F v. NewAC3	F v. NewA
1	Red_F1	1000	1	8.00	24.00	5.33	16.00	5.33	16.00	8.00	16.00	5.33	2.67	
1	Red_F2	1000	1	6.00	18.00	4.00	12.00	4.00	12.00	6.00	12.00	4.00	2.00	
1	Red_F3	2000	1	4.00	12.00	2.67	8.00	2.67	8.00	4.00	8.00	2.67	1.33	
1	Red_F4	6000	0.1	2.00	6.00	1.33	4.00	1.33	4.00	2.00	4.00	1.33	0.67	
1	Red_F5	3000	0.1	1.00	3.00	0.67	2.00	0.67	2.00	1.00	2.00	0.67	0.33	
1	Red_F6	6000	0.4	0.75	2.25	0.50	1.50	0.50	1.50	0.75	1.50	0.50	0.25	
0	Red_B1	18000	0.1	0.67	2.00	0.44	1.33	0.44	1.33	0.67	1.33	0.44	0.22	
-1	Red_B2	12000	0.1	1.00	3.00	0.67	2.00	0.67	2.00	1.00	2.00	0.67	0.33	
-1	Red_B3	500	0.1	2.00	6.00	1.33	4.00	1.33	4.00	2.00	4.00	1.33	0.67	
-1	NewAC9	500	1	2.00	6.00	1.33	4.00	1.33	4.00	2.00	4.00	1.33	0.67	
1	NewACz	4000	1	16.00	48.00	10.67	32.00	10.67	32.00	16.00	32.00	10.67	5.33	
-1	rB2	6000	1	16.00	48.00	10.67	32.00	10.67	32.00	16.00	32.00	10.67	5.33	
-1	rB3	12000	0.2	8.00	24.00	5.33	16.00	5.33	16.00	8.00	16.00	5.33	2.67	
-1	rB4	18000	0.1	4.00	12.00	2.67	8.00	2.67	8.00	4.00	8.00	2.67	1.33	

The first column on the ExRatios worksheet is labeled "Usage." For each Red aircraft type, a value of 1, −1, or 0 needs to be entered. A value of 1 indicates a fighter. When bombers are designated by the Usage = −1 code, they are assumed to try to escape rather than engage the Blue fighters. Because the bombers are escaping and not engaging in combat, the values for fighter vs. bomber encounters in the ExRatios spreadsheet no longer represent loss ratios. Rather, they represent the ratio of the mean time for Red to escape to the mean time for Blue to kill. Bombers with the Usage = 0 code are assumed to be fighter bombers and will engage; their encounter values in the ExRatios spreadsheet represent kill-rate ratios. (A note reminding the user of this difference has been added to the ExRatio spreadsheet and is shown in Figure 6-5.)

The second column indicates the name of the Red aircraft type; it is linked to the RedSupply worksheet. The third column, "Payload, lb," is used to enter the number of pounds of bombs that the associated Red aircraft can carry.

The fourth column is labeled "cepr," which stands for the circular error probable ratio (CEP/CEP_{ref}). The cepr is a value greater than 0 and less than or equal to 1, where 1 indicates that the bomber cannot deliver smart bombs (guided munitions) and 0.2 (or less) indicates that it can. The smart bomb threshold of 0.2 can be adjusted if necessary on the worksheet named "RedBombs." (We do not include the smart bomb threshold on the ExRatios input worksheet because we recommend that the default value of 0.2 be used.)

The remaining "F v. Blue_x" and "B v. Blue_x" columns are filled with loss ratio values for Usage values of 1, kill-rate ratios for Usage values of 0, and the ratio of the mean time for Red to escape to the mean time for Blue to kill. Those values are generally determined from experience or simulation and are beyond the scope of this users guide. It is expected that users will have access to relevant loss ratio data. Once the loss ratios are completed, the large button at the bottom of the screen labeled "Get Kill Rate Ratios" needs to be clicked. That button will run a macro that uses the loss ratios to compute kill-rate ratios. The kill-rate ratios are automatically entered onto the worksheet labeled "Rdata." As an added safeguard, every time SLAACM is run, the kill-rate ratio macro is run to ensure that the most recent values are entered onto Rdata.

SLAACM will not run if the ExRatios worksheet contains any blank values. If a loss ratio is left blank on ExRatios, the error message shown in Figure 6-6 will appear.

Figure 6-6. Classic SLAACM: Loss Ratio Error

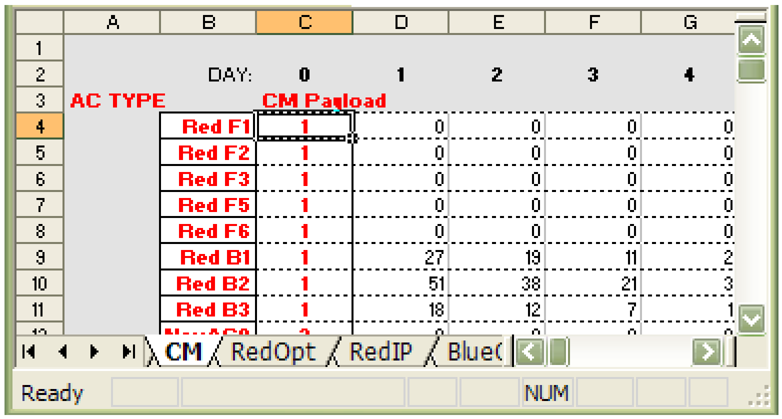
Then the user will be taken to the ExRatios worksheet. Any blank values will be highlighted as shown in Figure 6-7.

Figure 6-7. Classic SLAACM: Blank Loss Ratio Highlighted

CM Worksheet

The final user inputs are on the worksheet named “CM,” which stands for cruise missiles. On this worksheet, the user can specify the number of cruise missiles a Red aircraft could carry. Figure 6-8 shows that Column C is used to input the cruise missile capacity of each aircraft.

Figure 6-8. Classic SLAACM: CM Worksheet



No tradeoff between cruise missiles and bomb payload is required because the cruise missile computation is a secondary analysis that is meant to provide “what-if” information to the user. In this computation, the number of Red aircraft or each type that get past Blue defenses each day is multiplied by the number of cruise missiles specified on the CM worksheet. This analysis merely gives an idea of the potential number of cruise missiles that Red could launch if its aircraft were armed to do so.

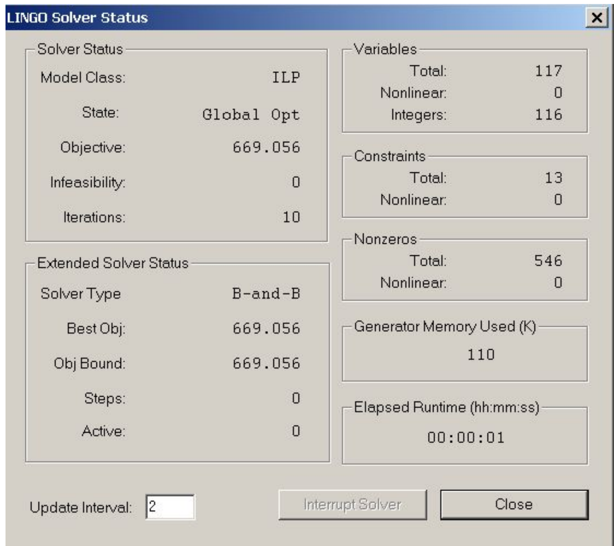
RUNNING SLAACM

Once all of the inputs have been entered, the user needs to return to the BlueOOB worksheet. To launch the analysis, the user clicks the “Run Campaign” button.

If the heuristic is being used, no other action is required on the part of the user.

If LINGO is being used, then a series of message boxes will appear to let the user know what day is being considered and which optimization is taking place. The user simply needs to click “OK” to proceed. The optimization in LINGO usually takes only a couple of seconds, and the screen often clears automatically. If it does not clear by itself, the user may need to click “Close,” provided LINGO has solved for the globally optimal solution (noted by “Global Opt” in the “State”). This case is shown in Figure 6-9.

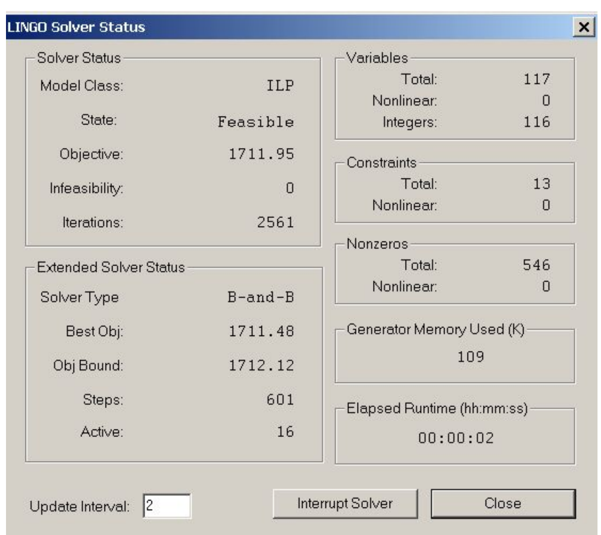
Figure 6-9. Classic SLAACM: LINGO Solution for Red Optimization



If the “State” shows “Feasible” as its status, then the user should wait until the global optimal solution is reached. If the solver is running for a long time, and the difference between the “Best Obj” and “Obj Bound” is small (e.g., less than 0.5 percent), then the user may click “Interrupt Solver”; the best feasible solution at that point will be entered into SLAACM. This event occurs rarely in SLAACM, since LINGO is usually able to solve these integer programming problems very quickly.

Figure 6-10 shows LINGO before it has reached optimality; it has reached a feasible—but not necessarily optimal—solution. In this example, the difference between the best objective and the objective bound is less than 1/1,712 = 0.05 percent, so one could certainly interrupt the solver at this point.

Figure 6-10. Classic SLAACM: Feasible LINGO Solution



OUTPUTS

Once the analysis has been completed, the user has several output results to review. SLAACM produces daily records of Red and Blue orders of battle, of Red and Blue losses, of their variances due to engagements, of tons of smart and not-smart bombs delivered, and of cruise missiles delivered.

Six worksheets report SLAACM's output results:

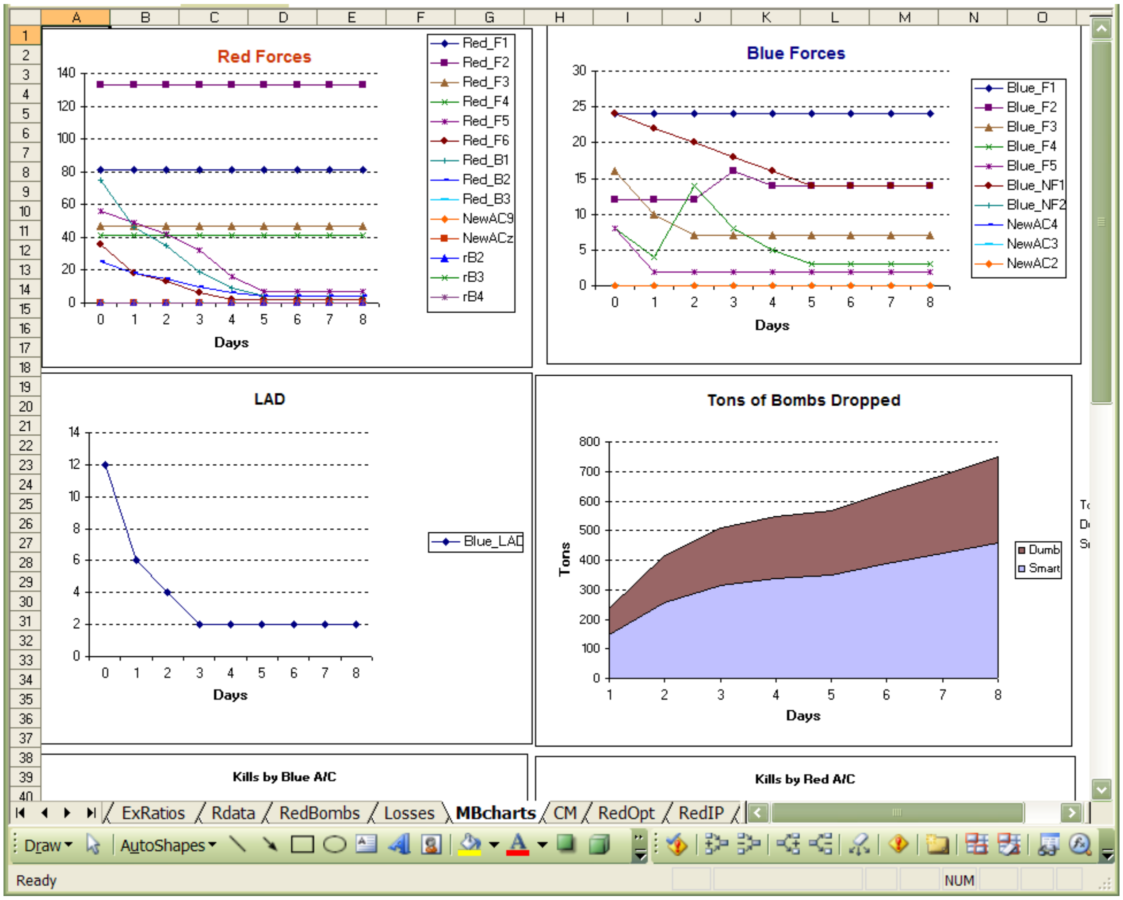
- BlueOOB
- RedOOB
- RedBombs
- Losses
- MBCharts
- CM.

We discuss the outputs—charts and worksheets tables—below. By reviewing the charts and tables, the user can gain an understanding of specific campaign outcomes.

Charts

The main results are presented graphically. From the BlueOOB worksheet, the user can click the "View Charts" button, or simply proceed to the worksheet tab labeled "MB Charts." The MB Charts worksheet contains eight charts. Figure 6-11a shows the quad chart "dashboard" of the four main SLAACM results.

Figure 6-11a. Classic SLAACM: Output Charts



The "Red Forces" chart in the upper left corner of Figure 6-11a shows the quantity of Red aircraft on each day of the campaign. In the example, most Red aircraft are annihilated by Day 4. The aircraft that have some quantity remaining, but which reach steady-state quantities, are most likely not being sent. For example, Red F1 and Red F2 are weak fighters, so they are never sent; instead, Red is choosing to escort its bombers with more capable aircraft. A quick check of the "Tons of Bombs Dropped" chart in the lower right corner shows that it looks like the bomb levels were decreasing by Day 5, indicating that fewer Red bombers are getting through Blue defenses beyond that day; however, after Day 5 we see a steady increase in bombs dropped. We can see that no additional Blue fighters are sent after Day 5 and a low level of Red bombers continue to reach their target. That is because we had set up Blue to be highly averse to losing aircraft. If the Blue value of Blue Loss parameter is set to 10 (half of the Red value), then the Blue aircraft are sent and the campaign is over by Day 5.

The "Blue Forces" chart in the upper right corner of Figure 6-11a shows the daily quantities of Blue aircraft. We see that Blue sustains some losses, although not nearly as many as Red; therefore, we can assume that Blue's forces are superior to Red's. In addition, in the first few days, we see increases in some quantities of Blue aircraft. That is due to Blue receiving reinforcements. (See the BlueSupply worksheet in Figure 6-1.) Blue_LADs are broken out separately from the Blue forces and are shown in the chart in the lower left corner of Figure 6-11a.

The Red Forces, Blue Forces, and LAD charts show the quantity of each aircraft type that is *available* on each day of the campaign. The charts do not indicate the number of aircraft sent into battle.

Additional charts on the worksheet are shown in Figure 6-11b. Those charts show number of kills by Blue and Red aircraft. Colors and dimensions are used to display comparisons of kills to show how specific aircraft perform.

Figure 6-11b. Classic SLAACM: Output Charts (continued)

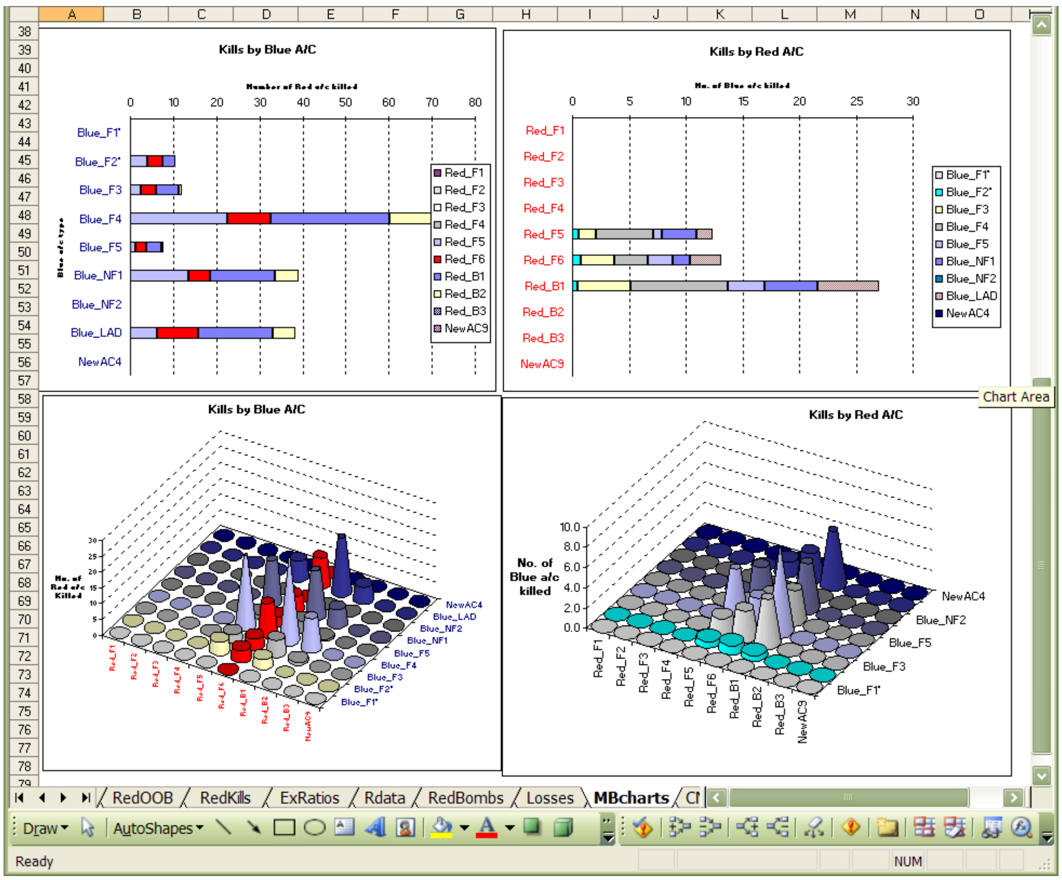
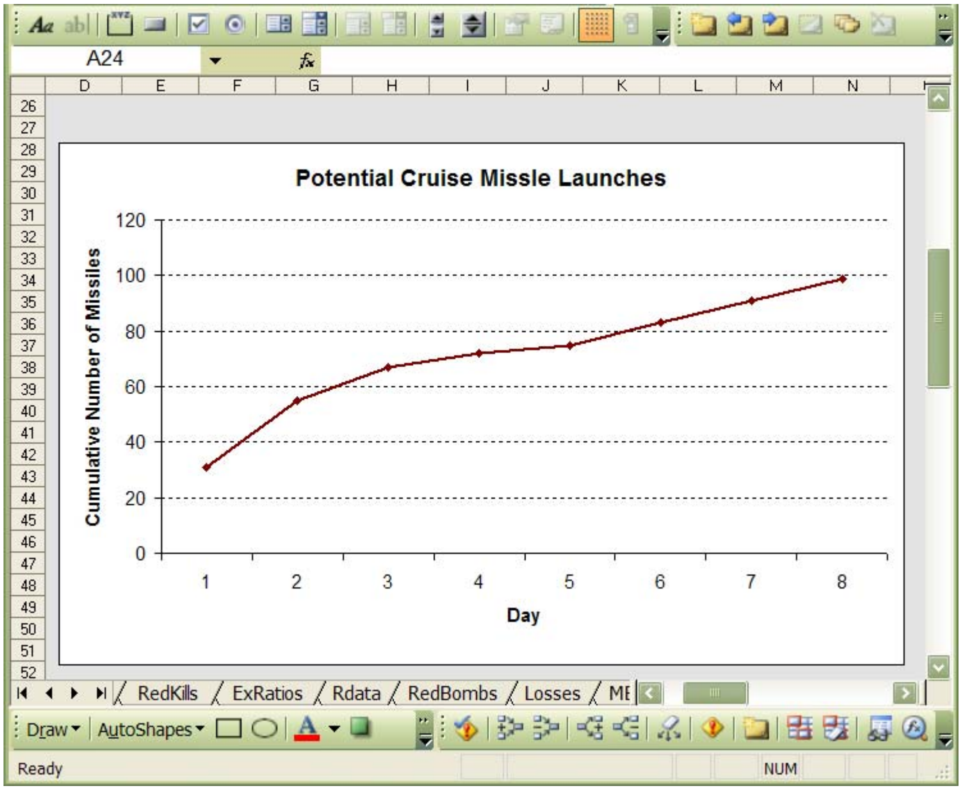


Figure 6-12 shows a graph of the potential number of cruise missile launches. As discussed in the previous section, this result assumes that the Red bombers that got through Blue defenses were armed with cruise missiles in lieu of bombs and were able to launch their cruise missiles.

Figure 6-12. Classic SLAACM: Cruise Missile Launches



Worksheet Tables

Because SLAACM is housed in an Excel workbook, the user can browse the data used to build the charts. All of the charts in Figure 6-11 are built from data in worksheet tables:

- The Red Forces chart uses the data on the "RedOOB" worksheet.
- The Blue Forces and LAD charts use data from the "BlueOOB" worksheet. (As we discussed earlier, the BlueOOB worksheet is both an input and an output. After running SLAACM, available aircraft quantities are loaded into the BlueOOB worksheet for each day of the campaign.)
- The Tons of Bombs chart uses data from the "RedBombs" worksheet.
- The Cruise Missile chart is built from data on the "CM" worksheet, where the chart resides.

It can be helpful to look at aircraft losses to get a better idea of how the campaign progressed. SLAACM has a worksheet named "Losses" that shows the loss data. Figure 6-13 shows the Losses worksheet. Blue and Red losses are shown by aircraft type. In addition, the worksheet shows the standard deviation of the losses for Blue aircraft.

Figure 6-13. Classic SLAACM: Losses

Blue Losses

	1	2	3	4	5	6	7	8	9	10	11	12	Total	Std. Dev.
Blue_F1	0	0	0	1	1	0	0	0					2	1.77
Blue_F2	0	0	0	0	0	0	0	0					0	0.33
Blue_F3	1	0	0	0	0	0	0	0					1	1.77
Blue_F4	1	0	2	1	0	0	0	0					4	2.48
Blue_F5	1	1	1	0	0	0	0	0					3	1.94
Blue_NF1	1	1	1	1	0	0	0	0					4	2.28
Blue_NF2	0	0	0	0	0	0	0	0					0	0.00
Blue_LAD	2	1	1	1	0	0	0	0					5	2.53
NewAC4	0	0	0	0	0	0	0	0					0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
TOTAL	**6**	**3**	**5**	**4**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
cumulative	6	9	14	18	19	19	19	19	19	19	19	19		

Red Losses =IF(RedOOB!D4<>"", RedOOB!C4+RedSupply!C4-RedOOB!D4, "")

Red F1	0	0	0	3	0	0	0	0				
Red F2	36	22	33	34	0	0	0	0				
Red F3	15	10	11	8	0	0	0	0				
Red F5	10	6	10	10	4	0	0	0				
Red F6	14	9	14	14	4	0	0	0				
Red B1	9	5	9	10	0	0	0	0				
Red B2	21	14	19	17	4	0	0	0				
Red B3	6	4	5	7	0	0	0	0				
NewAC9	0	0	0	0	0	0	0	0				
NewACz	0	0	0	0	0	0	0	0				
TOTAL	**111**	**70**	**101**	**103**	**12**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
cumulative	111	181	282	385	397	397	397	397	397	397	397	397

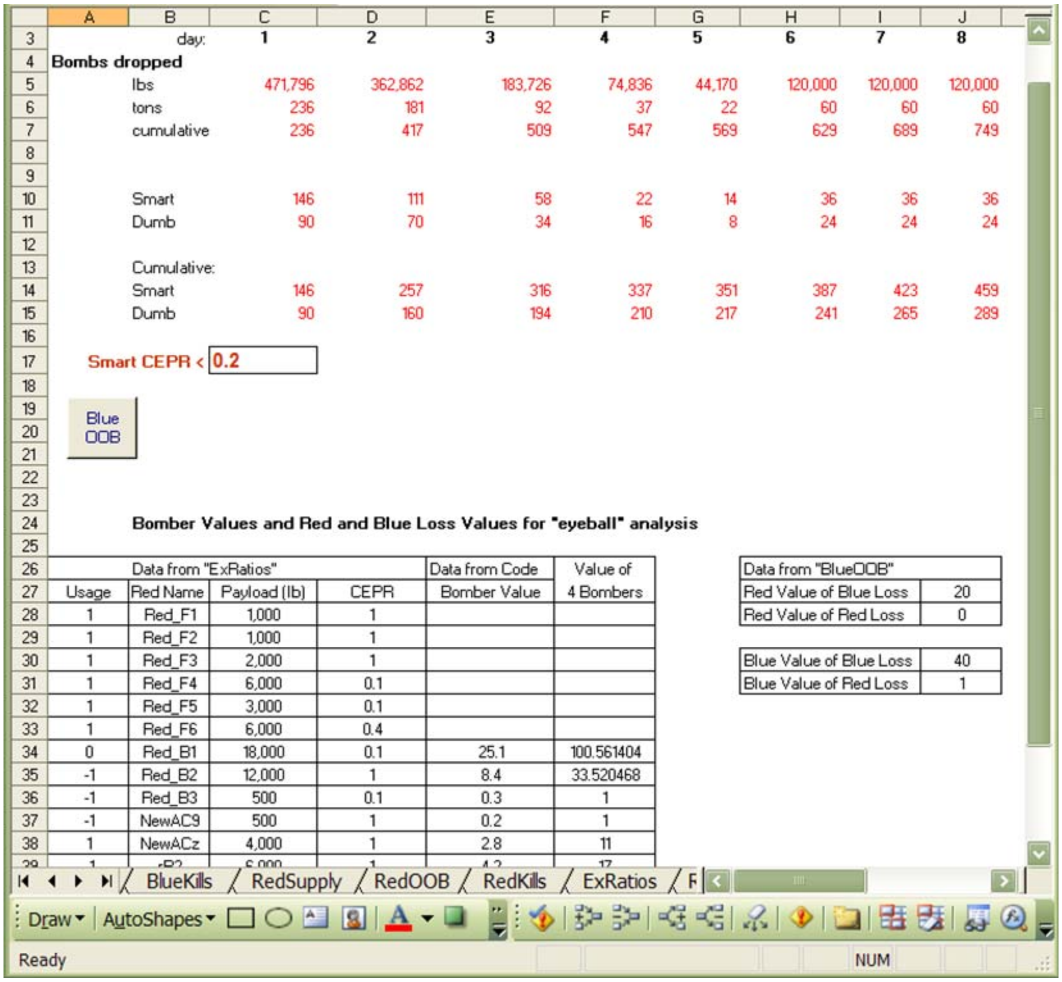
Losses / MBcharts / CM / RedOpt / RedIP / BlueOpt / BluIP

Ready NUM

Figure 6-14 shows the pounds and tons of bombs dropped by Red aircraft each day of the campaign. In addition, the bomb types are broken out by smart and not-smart types. These are the data used to build the “Tons of Bombs Dropped” chart in the lower right corner of Figure 6-11a. Upon inspection of the data in Figure 6-14, we see that the campaign actually ends on Day 5, not Day 4 as it had appeared on the chart; one can see that 2 tons of bombs were dropped on Day 5.

The RedBombs worksheet also contains a table that shows bomber payloads, cepr, and bomber values. The values for bomber effectiveness used in the optimizations are calculated based on payload tonnage and accuracy as in previous versions of SLAACM. Now, however, those calculated values—along with the fighter values—are displayed on the RedBombs worksheet to support “eyeball” analysis by the user. The RedBombs spreadsheet with this display is shown in Figure 6-14.

Figure 6-14. Classic SLAACM: Red Bombs



	day:	1	2	3	4	5	6	7	8
Bombs dropped									
	lbs	471,796	362,862	183,726	74,836	44,170	120,000	120,000	120,000
	tons	236	181	92	37	22	60	60	60
	cumulative	236	417	509	547	569	629	689	749
	Smart	146	111	58	22	14	36	36	36
	Dumb	90	70	34	16	8	24	24	24
	Cumulative:								
	Smart	146	257	316	337	351	387	423	459
	Dumb	90	160	194	210	217	241	265	289

Smart CEPR < 0.2

Blue OOB

Bomber Values and Red and Blue Loss Values for "eyeball" analysis

Data from "ExRatios"				Data from Code	Value of
Usage	Red Name	Payload (lb)	CEPR	Bomber Value	4 Bombers
1	Red_F1	1,000	1		
1	Red_F2	1,000	1		
1	Red_F3	2,000	1		
1	Red_F4	6,000	0.1		
1	Red_F5	3,000	0.1		
1	Red_F6	6,000	0.4		
0	Red_B1	18,000	0.1	25.1	100.561404
-1	Red_B2	12,000	1	8.4	33.520468
-1	Red_B3	500	0.1	0.3	1
-1	NewAC9	500	1	0.2	1
1	NewACz	4,000	1	2.8	11

Data from "BlueOOB"	
Red Value of Blue Loss	20
Red Value of Red Loss	0
Blue Value of Blue Loss	40
Blue Value of Red Loss	1

User Note

The output worksheets in SLAACM are dynamic and are overwritten during each model run. Users who want to save run data should copy and paste the values into an external file.

ANALYSES OF SAMPLE RUNS

In this section, we conduct a comparative analysis of the heuristic and LINGO using two SLAACM runs to illustrate some of the subtle differences that occur under reasonably similar scenarios.

First, we run SLAACM using the heuristic, with the inputs shown in Figures 6-1 through 6-5; we then repeat the run using LINGO. As shown in Table 6-1, the results are identical; the heuristic performs the same as the LINGO optimization. This case is not uncommon, and it gives us confidence that the heuristic can be a good approximation for the optimal solution (the heuristic and the exact integer programming formulations are covered in detail in Chapter 2).

Table 6-1. Classic SLAACM: Results of Initial SLAACM Run

Item	Heuristic	LINGO
Tons dropped		
Smart bombs	1,241	1,241
Dumb bombs	174	174
Aircraft losses		
Blue	19	19
Red	397	397

To illustrate a situation in which the heuristic and LINGO solutions differ, consider the same scenario, except that all Blue aircraft are smart. This change requires checking the boxes on the BlueOOB worksheet (Figure 6-3). The results are shown in Table 6-2. For both optimizers, the tonnage of smart bombs dropped is reduced by almost 50 percent, demonstrating the impact of smart Blues preferentially engaging the highest-value Red packages. Table 6-2 also shows significant differences in the heuristic results and LINGO's exact IP results.

Table 6-2. Classic SLAACM: Results of Setting All Blues to Be Smart

Item	Heuristic	LINGO
Tons dropped		
Smart bombs	640	602
Dumb bombs	289	289
Aircraft losses		
Blue	19	17
Red	402	404

LINGO's optimal solution resulted in 38 fewer tons of smart bombs being dropped, 2 fewer Blue aircraft losses, and 2 additional Red aircraft losses. In short, the IP solution using LINGO resulted in Blue reducing the amount of damage while sustaining fewer losses. This example illustrates what we have seen in other results, as well as what one might expect. The heuristic and the exact solution to the IP problem in LINGO generally produce qualitatively the same results. However, if one is looking for very precise results, the heuristic may fall short of true optimality by several percentage points.

SUMMARY

Classic SLAACM is a fast, flexible, robust model that contains the key input parameters necessary to define the characteristics of a realistic air-to-air campaign. Model results in both tabular and graphical format clearly display the impacts of parameter choices and provide insight into campaign scenarios. The Excel workbook implementation allows results to be easily copied and pasted to other applications for reporting and presentation.

The examples show differences in results using the heuristic optimizer and solving the exact IP formulation with LINGO. In the many practical cases we have analyzed, it has been rare for the heuristic results to differ from the LINGO results by more than 10 percent, but careful analyses may require the use of exact solution methods, especially for actual campaign planning.

Chapter 7
Users Guide to Attack SLAACM

Attack SLAACM is housed in a Microsoft Excel workbook with extensive Visual Basic for Applications code. The current version of Attack SLAACM was designed using object-oriented programming principles. In addition, it has an option to conduct integer programming by linking to an external optimization program called LINGO.[1] Using Excel to house SLAACM has several advantages. Most analysts are familiar with the Excel environment and will be able to navigate it easily. Other than acquiring the SLAACM workbook, users do not need to install additional software unless they want to run integer programming optimizations in LINGO. Finally, users can easily copy and paste output values and charts into their preferred presentation formats for displaying results to others.

In previous SLAACM documentation, friendly forces are identified as Blue and Green, and enemy forces are Red. With the ability of friendly forces to defend or attack, we have changed the nomenclature in this report to "Attack" and "Defense." The previous version has been called "Red SLAACM," "Air Defense SLAACM," and "Classic SLAACM," and the new SLAACM version incorporating attack has been called "Blue SLAACM" and "Attack SLAACM." For consistency, in this chapter we use the terms "Classic SLAACM" and "Attack SLAACM" to differentiate between the two versions. Despite the names, the new version includes both defensive and offensive scenarios and is a superset of the older version.

This chapter describes how to operate Attack SLAACM. It begins with a description of the model inputs. It then describes how to run the model once all of the input parameters are set up. Next, the chapter describes the outputs reporting Attack SLAACM's results. The last section addresses how the outputs are analyzed to determine the effects of the various engagement scenarios.

INPUTS

Attack SLAACM's input parameters are defined in six worksheets:

- BlueSupply, which is used to specify the quantities of Defense aircraft available on each day of the campaign.
- RedSupply, which is used to specify the quantities of Attack aircraft available on each day of the campaign.

[1] For more information on LINGO, see http://www.lindo.com/products/lingo/.

- BlueOOB, which is used to define the order of battle and other pertinent parameters. BlueOOB is the main sheet from which SLAACM runs are conducted. (SLAACM also has a RedOOB worksheet, but it is purely an output worksheet and does not receive any user inputs.)
- ExRatios, which is used to specify engagement parameters such as loss ratios and breakpoints.
- SiteData, which is used to define SEAD vs. SAM and SAM vs. bomber parameters.
- CM, which is used to specify the number of cruise missiles that a Red aircraft could carry.

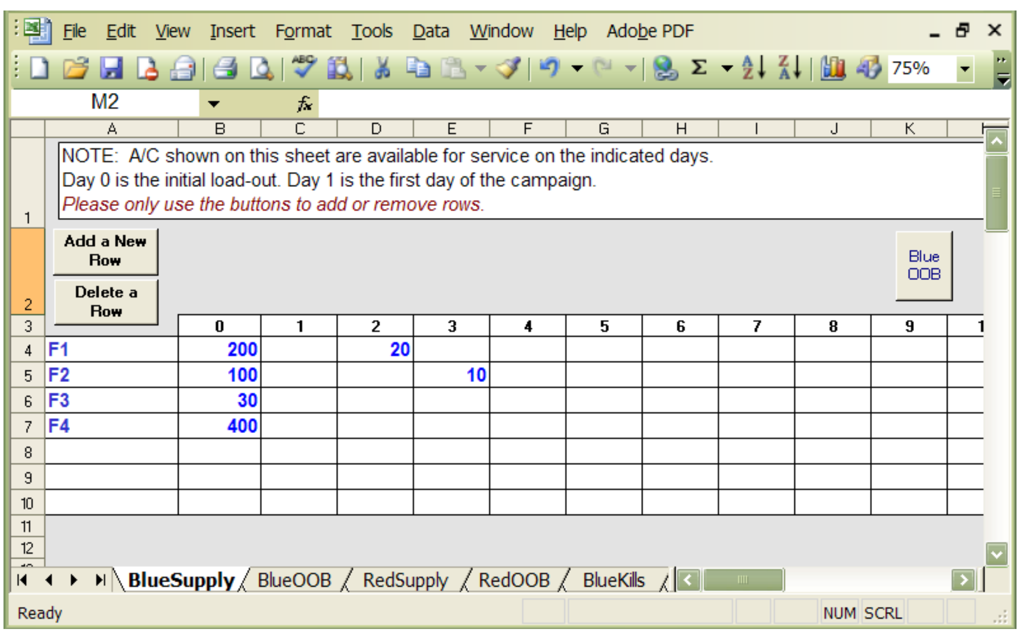
We discuss each sheet below and describe the proper way to complete each one.

BlueSupply and RedSupply Worksheets

For each force, Blue (Defense) and Red (Attack), the user will need to set up the initial quantities of aircraft available and the reinforcements available on each day of the campaign. The user will need to type those quantities into the appropriate worksheets, called BlueSupply and RedSupply.

Figure 7-1 is an example of the BlueSupply worksheet.

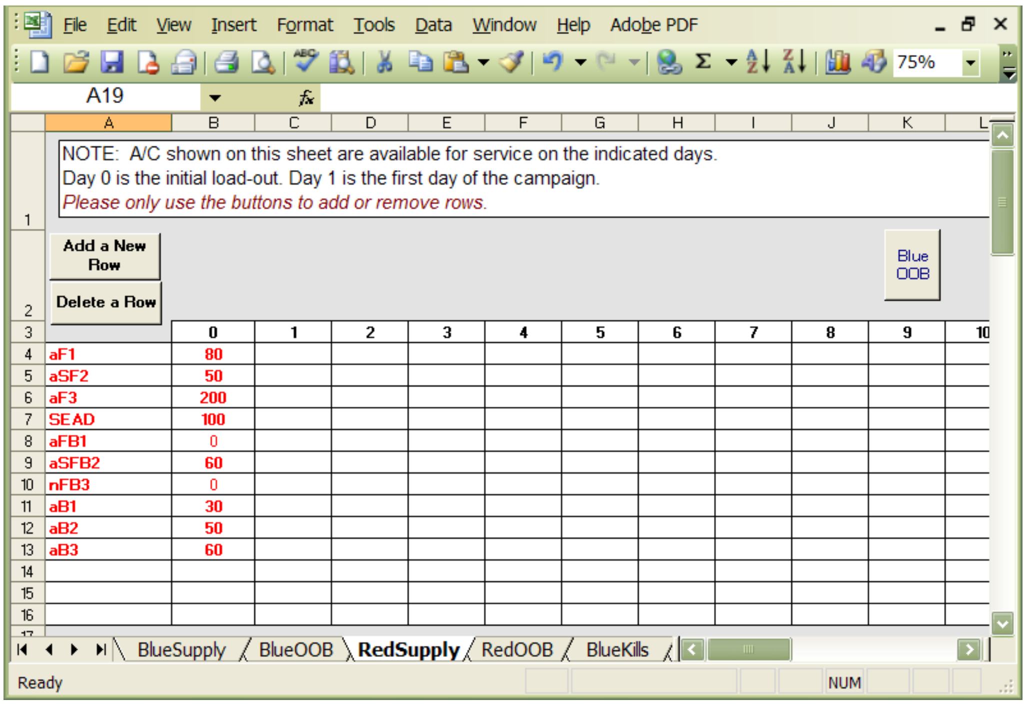
Figure 7-1. Attack SLAACM: BlueSupply Worksheet



As shown in the figure, the worksheet has a row for each Blue (Defense) aircraft type and a column for each day of the campaign. Initial aircraft quantities are entered into the column labeled 0. Reinforcements may be provided for subsequent days of the campaign. In the figure, we see that reinforcements are entered for Day 2 and Day 3.[2]

The RedSupply worksheet is designed in the same way as the BlueSupply sheet. Figure 7-2 is an example. In the figure, we see that only initial aircraft quantities are provided and no reinforcements are specified.

Figure 7-2. Attack SLAACM: RedSupply Worksheet



The BlueSupply and RedSupply worksheets can accommodate up to 20 aircraft types each. If new rows are needed to accommodate additional aircraft types, they can be added using the "Add a New Row" button.

Caution!

Any time a row is added or deleted, it is imperative that the **buttons** on the BlueSupply and RedSupply worksheets are used to add or delete the rows.

Do not simply insert or delete rows and columns.

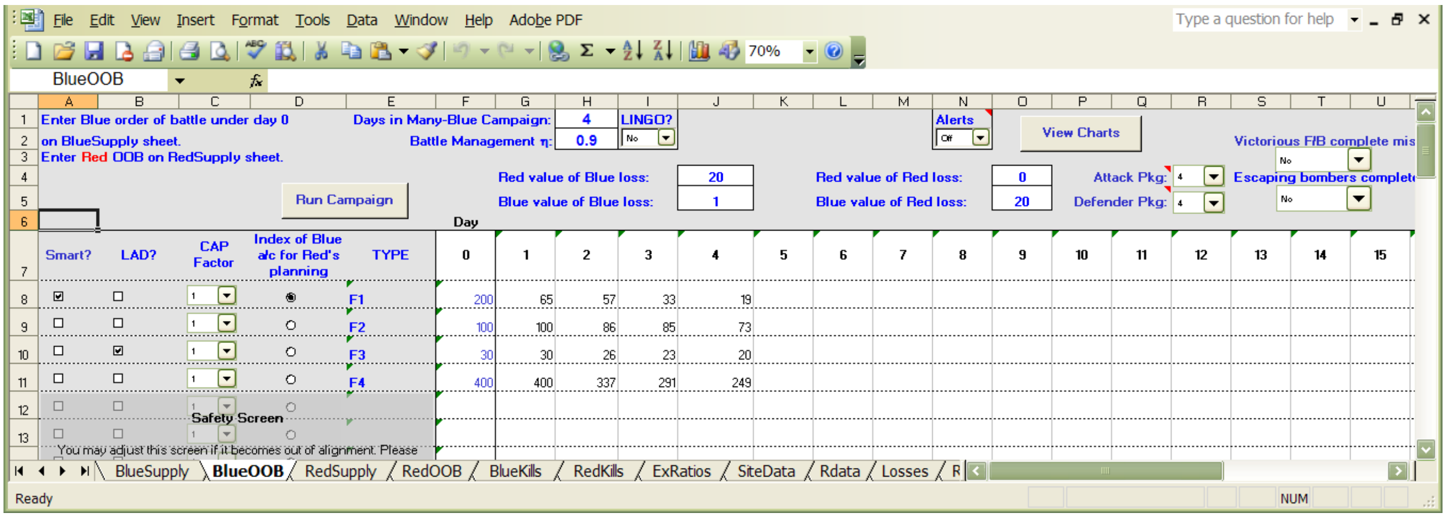
Several of SLAACM's worksheets reference the aircraft types. The "Add a New Row" and "Delete a Row" buttons must be used to keep the worksheets synchronized. Using typical worksheet approaches to inserting and deleting rows will likely result in model errors or misinterpretation of results.

[2] Note that a day is used in the model as a convenient designator to represent an individual attack/defense sortie. In an actual campaign, there may be more than one sortie on a particular calendar day, or, conversely, sorties may be distributed among several calendar days.

BlueOOB Worksheet

The worksheet named BlueOOB (which stands for Blue order of battle) is the main worksheet for input parameters for the Blue (Defense) aircraft. It is also the worksheet used to launch campaign analyses. BlueOOB also serves as an output worksheet, as does the RedOOB worksheet; the output portion of BlueOOB will be discussed in the output section of this chapter. Figure 7-3 shows the BlueOOB worksheet.

Figure 7-3. Attack SLAACM: BlueOOB Worksheet

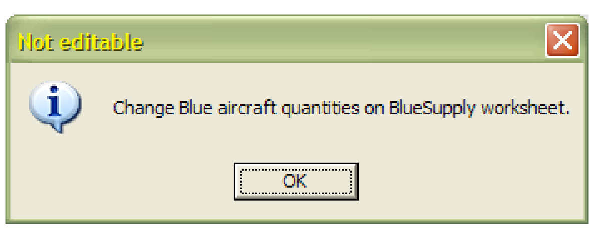


Caution!

Rows or columns should never be deleted from this worksheet!

The aircraft listed in the BlueOOB worksheet are linked to those shown in the BlueSupply worksheet (Figure 7-1). The Day 0 quantities are also linked to those in the BlueSupply worksheet. The user who attempts to click the aircraft names or Day 0 quantities on the BlueOOB sheet will receive the warning shown in Figure 7-4.

Figure 7-4. Warning: Do Not Alter Aircraft of BlueOOB Worksheet



The first four columns on the BlueOOB worksheet define aircraft-specific input parameters for Blue:

- *"Smart?" column.* The user may use this column to identify specific Blue (defense) aircraft as smart. Smart Blue aircraft can determine the composition of Red attack packages before intercepting them and can coordinate their defensive response among themselves to provide optimal defenses. This coordination and optimization is discussed in Chapter 5. Multiple Blue aircraft types may be designated as smart. In Figure 7-3, the first aircraft is selected to be smart.

- *"LAD?" column.* The user may use this column to identify certain Blue (defense) aircraft as local air defenders. SLAACM treats LADs differently from other Blue types when considering Blue dispatch options. In general, a non-LAD Blue has a penalty for its own losses that is significantly greater than a LAD Blue, which presumably is less loss averse due to defending its own territory. Some non-LAD Blues may have a negative payoff function for engaging certain Red packages dispatched. Blue will not dispatch that type. LADs, however, are dispatched in all cases. Multiple Blue aircraft types may be designated as LADs. In Figure 7-3, the third aircraft is designated as such.

- *"CAP Factor" column.* This factor addresses the situations in which Blue (defense) fighters are stationed at remote airfields and must divide their forces to maintain continuous combat air patrol coverage over the battle space. The user may select a CAP factor for any Blue aircraft type, so that only a specified fraction of the inventory of that Blue type is available. CAP factor values are chosen from a drop-down box for each aircraft type. Values range from 1 to 10. A CAP factor of 1 indicates that no CAP limitation is placed on that aircraft type. In Figure 7-3, all aircraft have CAP factors of 1. However, if an aircraft had been assigned a CAP factor of 4, that aircraft would have only n/4 available aircraft on the first day. If there are not enough aircraft to fulfill the combat air patrol at all times, those flights are not dispatched.

- "*Index of Blue a/c for Red's planning*" *column*. Only one aircraft type can be selected for this parameter. In Figure 7-3, the first aircraft type is designated as Red's "planning aircraft." This aircraft type is used by Red to compute its expected payoff function and optimize its dispatch choices. Experience has shown that Red optimization is relatively insensitive to the selection of the Blue type, so normal practice is to select the most numerous Blue type for Red planning.

In addition to the aircraft-specific parameters, several other input parameters must be selected on the BlueOOB worksheet.

At the top of the worksheet, the number of days in the campaign is specified. In Figure 7-3, the number of campaign days is 4. The longer the campaign, the longer it will take SLAACM to run, since each campaign day requires the full breadth of SLAACM computations. Initially, the user may need to start out with a longer campaign and then decrease its length as a steady state in aircraft losses is reached.

Beneath the number of campaign days is a parameter labeled "Battle Management η." This parameter is the battle management efficiency factor. η is a value greater than 0 and less than or equal to 1. Simply stated, η less than 1 indicates the percentage of time that Blue defender will be unable to find a Red attack package. In Figure 7-3, the efficiency factor is shown as 0.9, which means that 10 percent of the not-smart Blue defenders will chase "ghost" attackers. No losses will result, but in those cases, Blue will not be confronting an actual attacker, so its efficiency is reduced. Since smart Blues know Red's attack package compositions in advance, we assume they have perfect management.

Next to the campaign days and the efficiency factor is a drop-down box labeled "LINGO?" If "No" is selected, as shown in Figure 7-3, SLAACM will use its internal heuristic optimization to calculate engagement results. If "Yes" is selected, LINGO will be used to perform integer optimization (a licensed copy of LINGO software is required to be installed on the PC for this feature to work). Both the heuristic and the exact integer programming solution are described in detail in Chapter 5. If LINGO is used, there are two LINGO input worksheets: "RedOpt" and "BlueOpt." However, those sheets are already complete and should not be altered by the user.

Below the parameters just described are four value boxes:

- Red value of Blue loss
- Blue value of Blue loss
- Red value of Red loss
- Blue value of Red loss.

These parameters establish the value of killing an opponent, as well as the penalty a side takes for losing one of its own. These values, which are applied to fighter aircraft only, are used in the optimization objective function. The Red value of Blue loss represents the benefit to Red of destroying a Blue aircraft, while Blue value of Blue loss is the penalty to Blue for losing one of its own aircraft. Values can be between zero and 100.

In the case shown in Figure 7-3, we see that the value of a Blue loss to Red is 20, while the penalty to Blue for losing its own aircraft is only 1. Therefore, Blue's aversion to a loss is one-twentieth the value of Red's benefit for obtaining the Blue loss. Conversely, we see in this example that Red is not penalized for a Red fighter loss (the value is zero) and Blue achieves a benefit for a Red fighter loss equal to Red's benefit for killing a Blue aircraft (the value is 20).

Further to the right are two drop-down boxes for Attack package and Defender package designations. New engagement scenarios have been added to Attack SLAACM to represent potential U.S. attack package options using both normal and low-observable aircraft . Table 7-1 lists the attack package options.

Table 7-1 Attack Package Options

Drop-down value	Attack package
2	Basic hostile 12-ship (4 LE, 4 CE, 4 B) attack package
3	4 escort, 2 SEAD, 2 bomber attack package
5	4 escort, 2 bomber attack package
6	2 escort, 2 SEAD, 1 bomber attack package
7	2 escort, 1 bomber attack package

The defender package selections are enumerated in Table 7-2.

Table 7-2. Defender Package Options

Drop-down value	Defender package
2	2 defenders
4	4 defenders (basic friendly defense package)
8	8 defenders (basic hostile defense package)

On the far right hand side of Figure 7-3 are two yes/now drop-down boxes. The first is labeled "Victorious F/B complete mission?" The second is labeled "Escaping bombers complete mission?" If yes is chosen, then fighter-bombers and bombers, respectively, that escape attack go on to complete their mission and deliver bombs to their target. If no is chosen, then escaping fighter-bombers and bombers do not complete their bombing missions.

The final input selection on BlueOOB is a drop-down box labeled "Alerts." The default setting for this feature is "Off." However, if "On" is selected, message boxes will become visible as the analysis runs.

ExRatios Worksheet

The ExRatios worksheet contains a row for each Red attack aircraft that has been designated on the RedSupply worksheet. The worksheet has several parameter columns, discussed below, and a column for each Blue defender aircraft labeled "F v. Blue_x" where "F" indicates that Red is a fighter and "x" corresponds to the label provided on BlueSupply. Columns are repeated for each Blue aircraft, except they are labeled "B v. Blue_x" where "B" indicates that Red is a bomber and "x" is as just described. The default setting in SLAACM assumes that loss ratios are for engagements between 4 Blues and 8 Reds, with the Reds and Blues fighting to annihilation.

Those assumptions can be changed on this sheet under "Red Start," "Blue Start," "Red Quit," and "Blue Quit." Red Start and Blue Start correspond to the engagement type, e.g., 8 and 4, respectively. Red Quit and Blue Quit denote the breaking points; for example, 0 and 0 means that both sides will fight to annihilation. The user will want to specify values that correspond to the conditions under which the loss ratios were obtained.

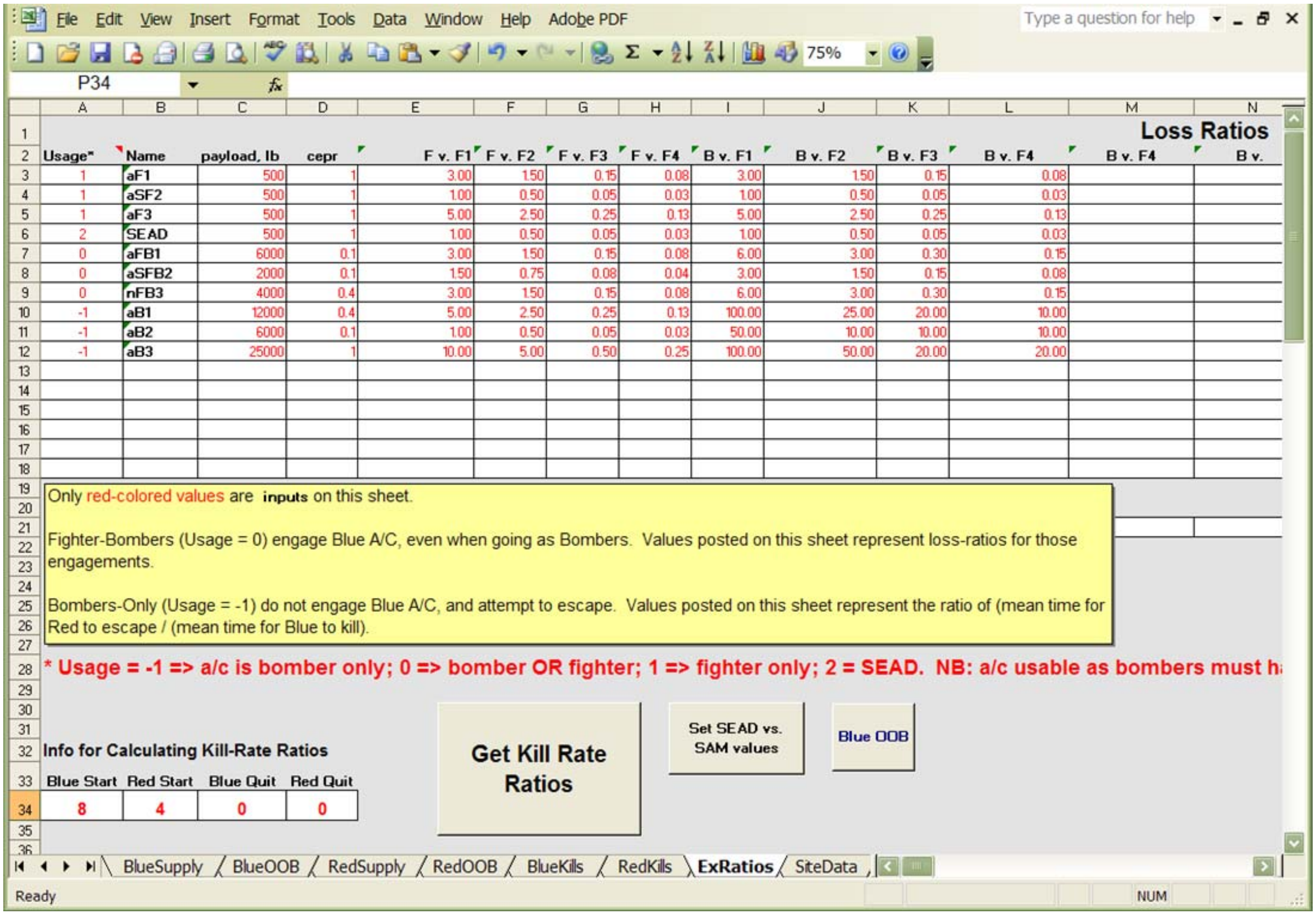
To keep the rows and columns synchronized with the BlueSupply and RedSupply worksheets, it is imperative that no rows or columns on ExRatios be inserted or deleted.

Caution!

As stated earlier, use the add and delete buttons on the BlueSupply and RedSupply worksheets to add or delete rows.

Figure 7-5 is a screenshot of the ExRatios worksheet.

Figure 7-5. Attack SLAACM: ExRatios Worksheet



Usage*	Name	payload, lb	cepr	F v. F1	F v. F2	F v. F3	F v. F4	B v. F1	B v. F2	B v. F3	B v. F4
1	aF1	500	1	3.00	1.50	0.15	0.08	3.00	1.50	0.15	0.08
1	aSF2	500	1	1.00	0.50	0.05	0.03	1.00	0.50	0.05	0.03
1	aF3	500	1	5.00	2.50	0.25	0.13	5.00	2.50	0.25	0.13
2	SEAD	500	1	1.00	0.50	0.05	0.03	1.00	0.50	0.05	0.03
0	aFB1	6000	0.1	3.00	1.50	0.15	0.08	6.00	3.00	0.30	0.15
0	aSFB2	2000	0.1	1.50	0.75	0.08	0.04	3.00	1.50	0.15	0.08
0	nFB3	4000	0.4	3.00	1.50	0.15	0.08	6.00	3.00	0.30	0.15
-1	aB1	12000	0.4	5.00	2.50	0.25	0.13	100.00	25.00	20.00	10.00
-1	aB2	6000	0.1	1.00	0.50	0.05	0.03	50.00	10.00	10.00	10.00
-1	aB3	25000	1	10.00	5.00	0.50	0.25	100.00	50.00	20.00	20.00

The first column on the ExRatios worksheet is labeled “Usage.” For each Red aircraft type, a value of 0, 1, −1, or 2 needs to be entered. A value of 0 indicates a fighter-bomber. A value of 1 indicates a fighter. When bombers are designated by the Usage = −1 code, they are assumed to try to escape rather than engage the Blue fighters. Because the bombers are escaping and not engaging in combat, the values for fighter vs. bomber encounters in the ExRatios spreadsheet no longer represent loss ratios. Rather, they represent the ratio of the mean time for Red to escape to the mean time for Blue to kill. Bombers with the Usage = 0 code are assumed to be fighter bombers and will engage; their encounter values in the ExRatios spreadsheet represent loss ratios. A value of 2 indicates a SEAD aircraft. (A note reminding the user of this difference is on the ExRatio spreadsheet and is shown in Figure 7-5.)

The second column indicates the name of the Red aircraft type; it is linked to the RedSupply worksheet. The third column, “Payload, lb,” is used to enter the number of pounds of bombs that the associated Red aircraft can carry.

The fourth column is labeled “cepr,” which stands for the circular error probable ratio (CEP/CEP$_{ref}$). The cepr is a value greater than 0 and less than or equal to 1, where 1 indicates that the bomber cannot deliver smart bombs (guided munitions) and 0.2 (or less) indicates that it can. The smart bomb threshold of 0.2 can be adjusted if necessary on the worksheet named “RedBombs.” (We do not include the smart bomb threshold on the ExRatios input worksheet because we recommend that the default value of 0.2 be used.)

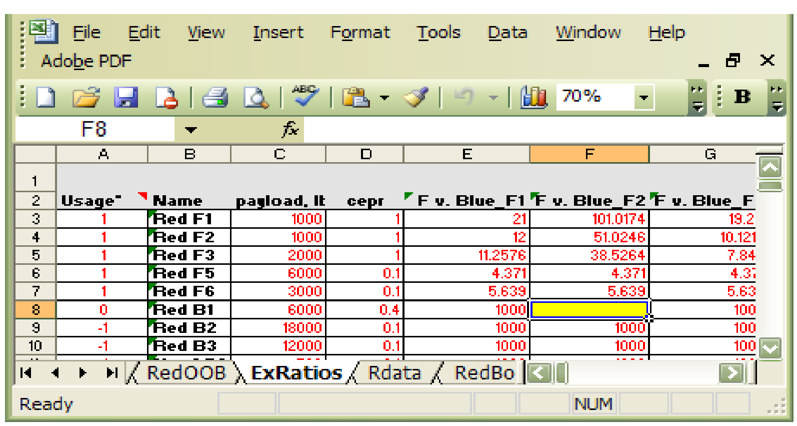
The remaining “F v. Blue_x” and “B v. Blue_x” columns are filled with loss ratio values for Usage values of 1, kill-rate ratios for Usage values of 0, and the ratio of the mean time for Red to escape to the mean time for Blue to kill. Those values are generally determined from experience or simulation, and are beyond the scope of this users guide. It is expected that users will have access to relevant loss ratio data. Once the loss ratios are completed, the large button at the bottom of the screen labeled “Get Kill Rate Ratios” needs to be clicked. That button will run a macro that uses the loss ratios to compute kill-rate ratios. The kill-rate ratios are automatically entered onto the worksheet labeled “Rdata.” As an added safeguard, every time SLAACM is run, the kill-rate ratio macro is run to ensure that the most recent values are entered onto Rdata.

SLAACM will not run if the ExRatios worksheet contains any blank values. If a loss ratio is left blank on ExRatios, the error message shown in Figure 7-6 will appear.

Figure 7-6. Attack SLAACM: Loss Ratio Error

Then the user will be taken to the ExRatios worksheet and any blank values will be highlighted as shown in Figure 7-7.

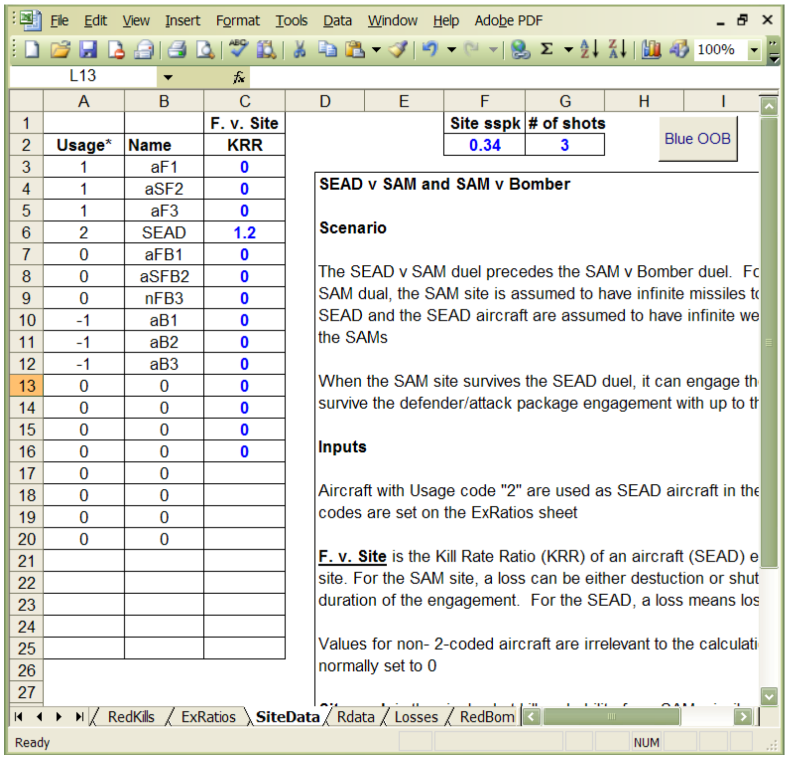
Figure 7-7. Attack SLAACM: Blank Loss Ratio Highlighted



SiteData Worksheet

The SiteData worksheet is particular to the Attack SLAACM model. Using this worksheet, users define SEAD and SAM parameters. Figure 7-8 is a screenshot of the SiteData worksheet.

Figure 7-8. Attack SLAACM: SiteData Worksheet



Aircraft designated as SEAD by using the value of 2 on ExRatios worksheet can have their kill-rate ratios set on the SiteData worksheet. In Figure 7-8, the fourth listing is a SEAD aircraft. On the SiteData worksheet, a KRR of 1.2 has been entered, indicating that this SEAD aircraft has a KRR of 1.2 against a SAM site. For the SAM site, a loss can be either destruction or shutdown for the duration of the engagement. For the SEAD, a loss means loss of the aircraft. For aircraft that are not designated as SEAD, KRR values on this sheet are ignored, so by convention we set them to zero.

In addition, on the SiteData worksheet, the single-shot kill probability for a SAM missile against a bomber can be set in the box labeled “Site sspk.”

Finally, the number of missiles that a SAM site fires at the set of bombers reaching their target is set using the box labeled “# of shots.”

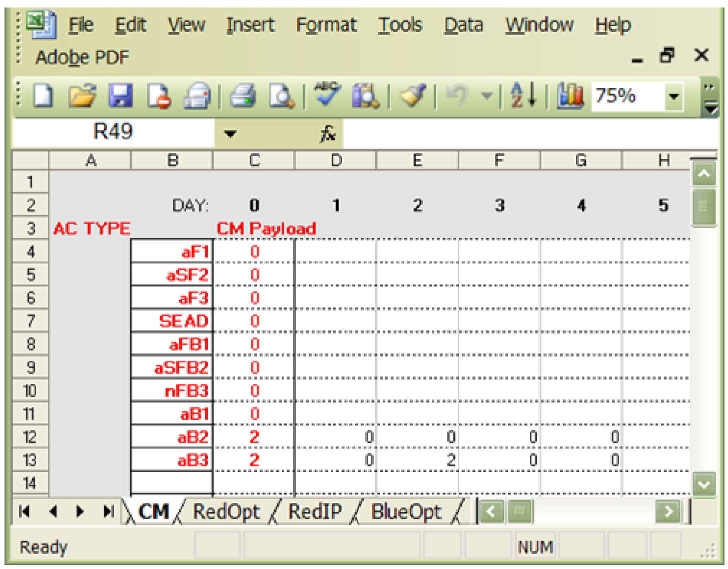
SEAD aircraft engage the SAM site (the SAM has unlimited missiles for this duel). The SAM site does not engage the escort fighters, and the defending fighters do not engage the SEAD aircraft. If the SEAD aircraft are destroyed by the

SAM, the SAM site (with up to three missiles) engages the bombers. In scenarios with no SEAD aircraft, the SAM site always engages the bombers with three missiles. The limitation to three missiles accounts for the limited time interval on which the bombers are on their bomb run, rather than for any limit to the number of missiles at the SAM site.

CM Worksheet

The final user inputs are on the worksheet named “CM,” which stands for cruise missiles. On this worksheet, the user can specify the number of cruise missiles a Red aircraft could carry. Figure 7-9 shows that Column C is used to input the cruise missile capacity of each aircraft.

Figure 7-9. Attack SLAACM: CM Worksheet



No tradeoff between cruise missiles and bomb payload is required because the cruise missile computation is a secondary analysis that is meant to provide “what-if” information to the user. In this computation, the number of Red aircraft of each type that get past Blue defenses each day is multiplied by the number of cruise missiles specified on the CM worksheet. This analysis merely gives an idea of the potential number of cruise missiles that Red could launch if its aircraft were armed to do so.

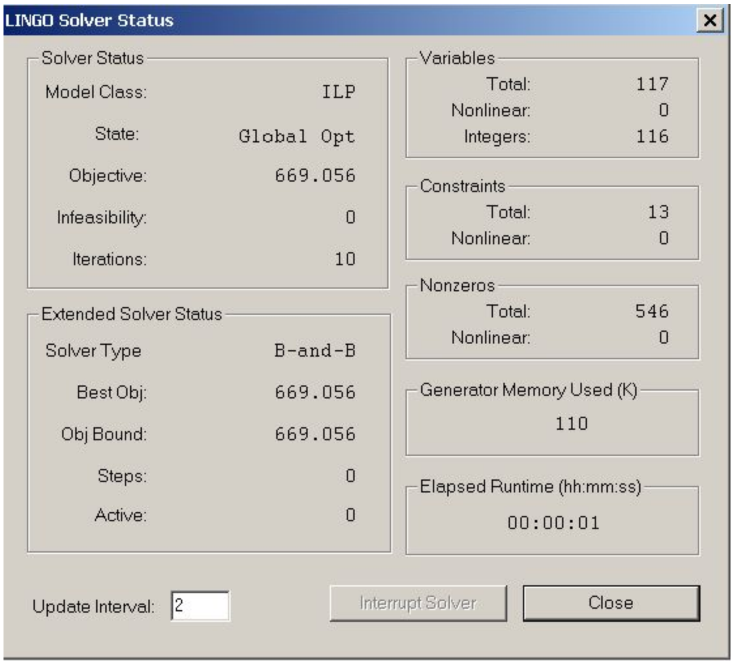
RUNNING SLAACM

Once all of the inputs have been entered, the user needs to return to the BlueOOB worksheet. To launch the analysis, the user clicks the “Run Campaign” button.

If the heuristic is being used, no other action is required on the part of the user.

If LINGO is being used, then a series of message boxes will appear to let the user know what day is being considered and which optimization is taking place. The user simply needs to click “OK” to proceed. The optimization in LINGO usually takes only a couple of seconds, and the screen often clears automatically. If it does not clear by itself, the user may need to click “Close,” provided LINGO has solved for the globally optimal solution (noted by “Global Opt” in the “State”). This case is shown in Figure 7-10.

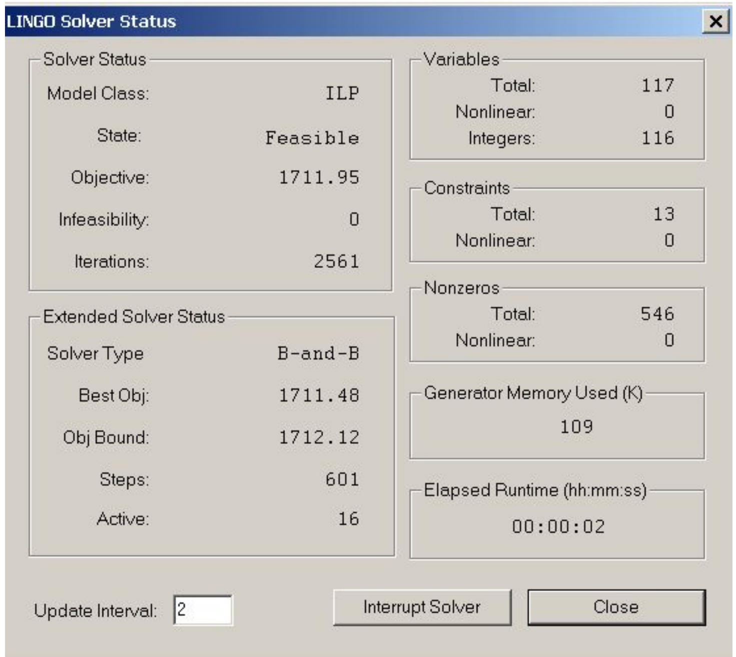
Figure 7-10. Attack SLAACM: LINGO Solution for Red Optimization



If the “State” shows “Feasible” as its status, then the user should wait until the global optimal is reached. If the solver is running for a long time, and the difference between the “Best Obj” and “Obj Bound” is small (e.g., less than 0.5 percent), then the user may click “Interrupt Solver”; the best feasible solution at that point will be entered into SLAACM. This event occurs rarely in SLAACM, since LINGO is usually able to solve these integer programming problems very quickly.

Figure 7-11 shows LINGO before it has reached optimality; it has reached a feasible—but not necessarily optimal—solution. In this example, the difference between the best objective and the objective bound is less than 1/1,712 = 0.05 percent, so one could certainly interrupt the solver at this point.

Figure 7-11. Attack SLAACM: Feasible LINGO Solution



OUTPUTS

Once the analysis has been completed, the user has several output results to review. Attack SLAACM produces daily records of Red and Blue orders of battle, of Red and Blue losses, of their variances due to engagements, of tons of smart and not-smart bombs delivered, and of cruise missiles delivered.

Seven worksheets report Attack SLAACM's output results:

- BlueOOB
- RedOOB
- RedBombs
- Losses
- MBCharts
- CM
- sam.

We discuss the outputs—charts and worksheet tables—below. By reviewing the charts and tables, the user can gain an understanding of specific campaign outcomes.

Charts

The main results are presented graphically. From the BlueOOB worksheet, the user can click the “View Charts” button, or simply proceed to the worksheet tab labeled “MB Charts.” There are eight charts on the MB Charts worksheet. Figure 7-12a shows the “dashboard” of the four main SLAACM results.

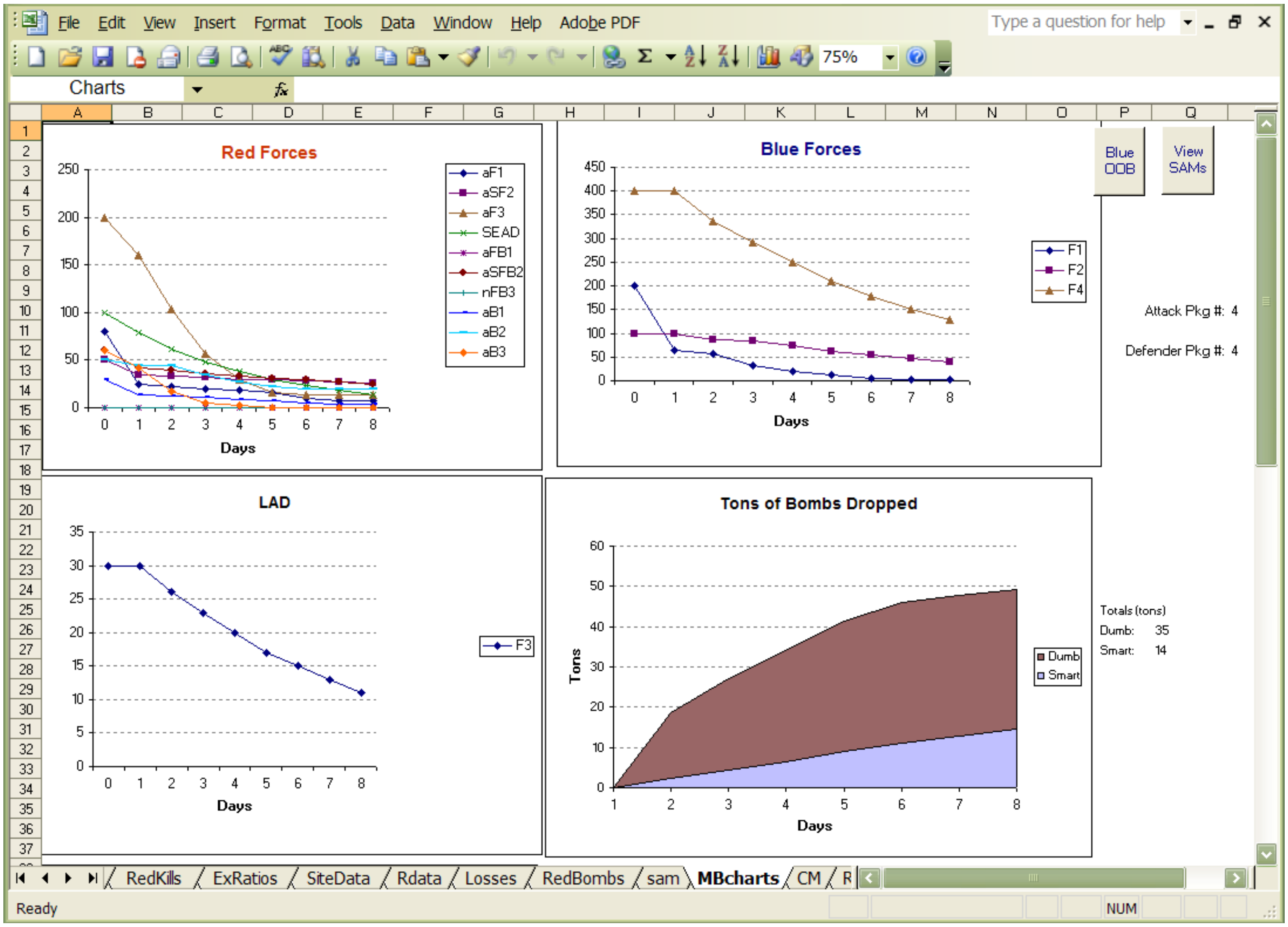
The “Red Forces” chart in the upper left corner of Figure 7-12a shows the quantity of Red Attack aircraft on each day of the campaign. In the example, most Red aircraft are annihilated by Day 5. A quick check of the “Tons of Bombs Dropped” chart in the lower right corner shows that it looks like the bomb levels were decreasing by Day 6, indicating that fewer Red bombers were getting through Blue defenses beyond that day. However, the “Blue Forces” chart shows a steady decline in Blue Defenders, which means that some level of attackers continue to engage with them.

Blue_LADs are broken out separately from the Blue Forces and are shown in the chart in the lower left corner of Figure 7-12a.

The Red Forces, Blue Forces, and LAD charts show the quantity of each aircraft type that is *available* on each day of the campaign. The charts do not indicate the number of aircraft sent into battle.

The attack and defender packages that were selected on BlueOOB are displayed on the right side of Figure 7-12a.

Figure 7-12a. Attack SLAACM: Output Charts



Additional charts on the worksheet are shown in Figure 7-12b. Those charts show number of kills by Blue and Red aircraft. Colors and dimensions are used to display comparisons of kills to show how specific aircraft perform.

Figure 7-12b. Attack SLAACM: Output Charts (continued)

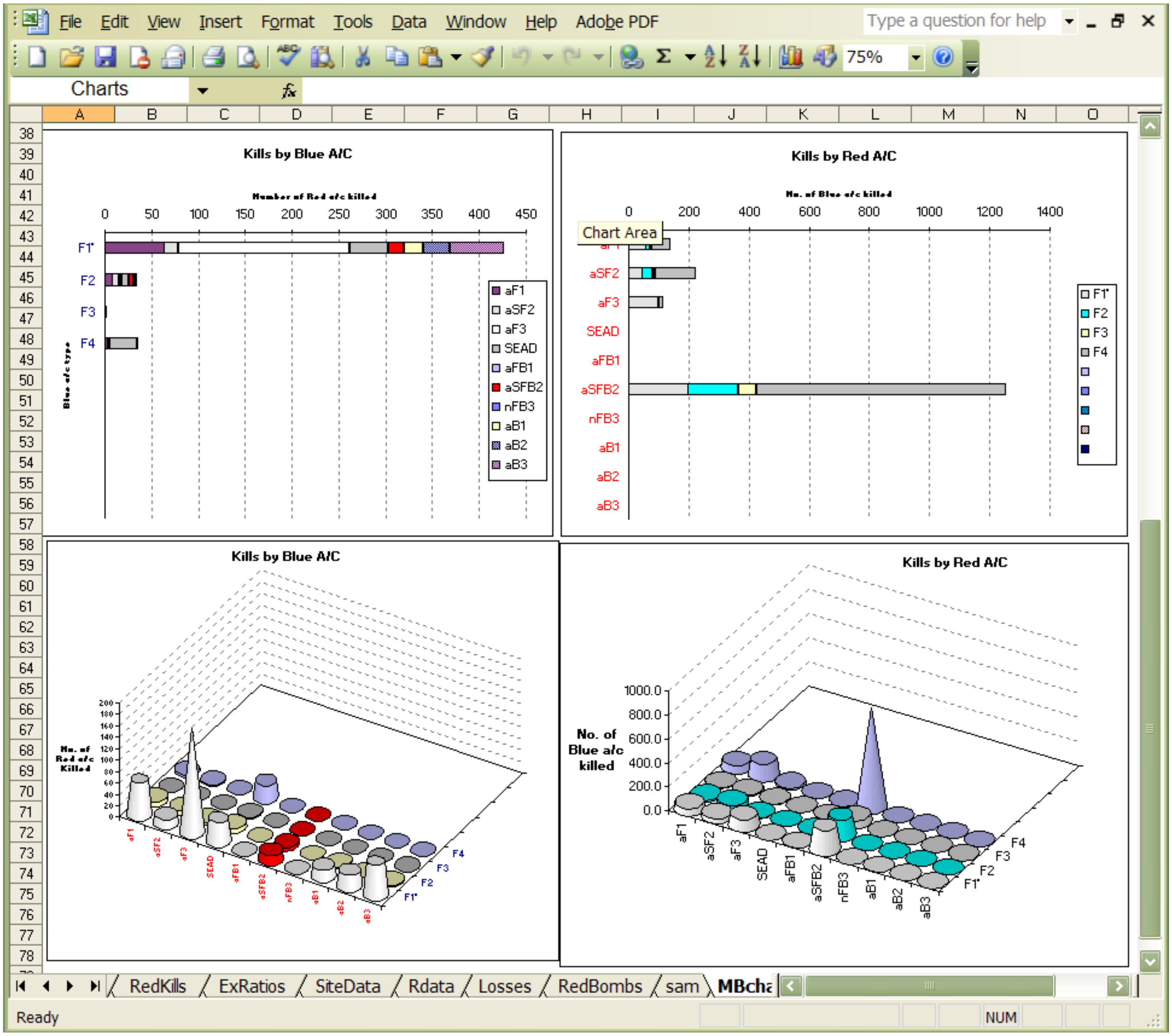
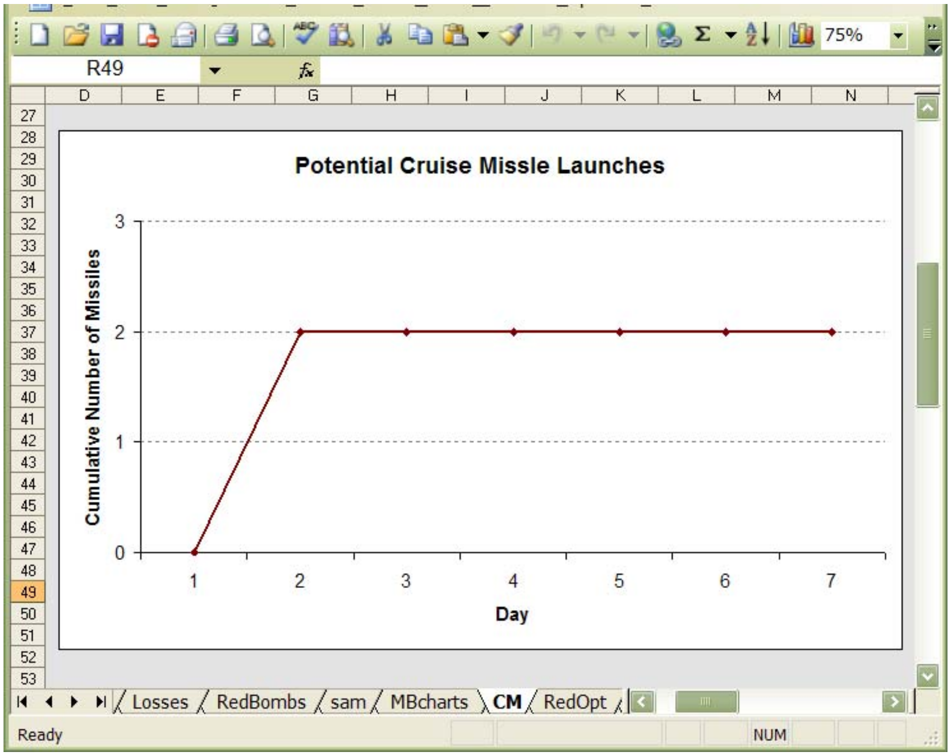


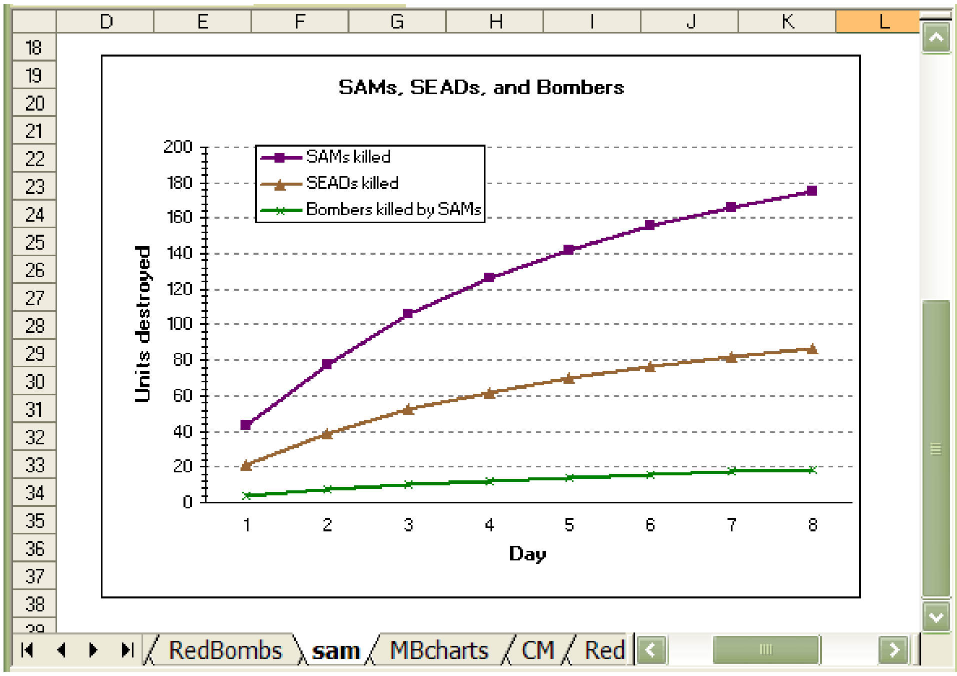
Figure 7-13 shows a graph of the potential number of cruise missile launches. As discussed in the previous section, this result assumes that the Red bombers that got through Blue defenses were armed with cruise missiles in lieu of bombs and were able to launch their cruise missiles.

Figure 7-13. Attack SLAACM: Cruise Missile Launches



The “sam” worksheet shows results of SAM and SEAD activities. Figure 7-14 is a screenshot of this worksheet. The “sam” worksheet shows, by day, the number of SAMs killed, SEADs killed, and bombers killed by SAMs. In the above example, we see that about 175 SAMs, 86 SEADs, and 19 bombers are killed in this scenario.

Figure 7-14. Attack SLAACM: SAM, SEAD, and Bomber Results



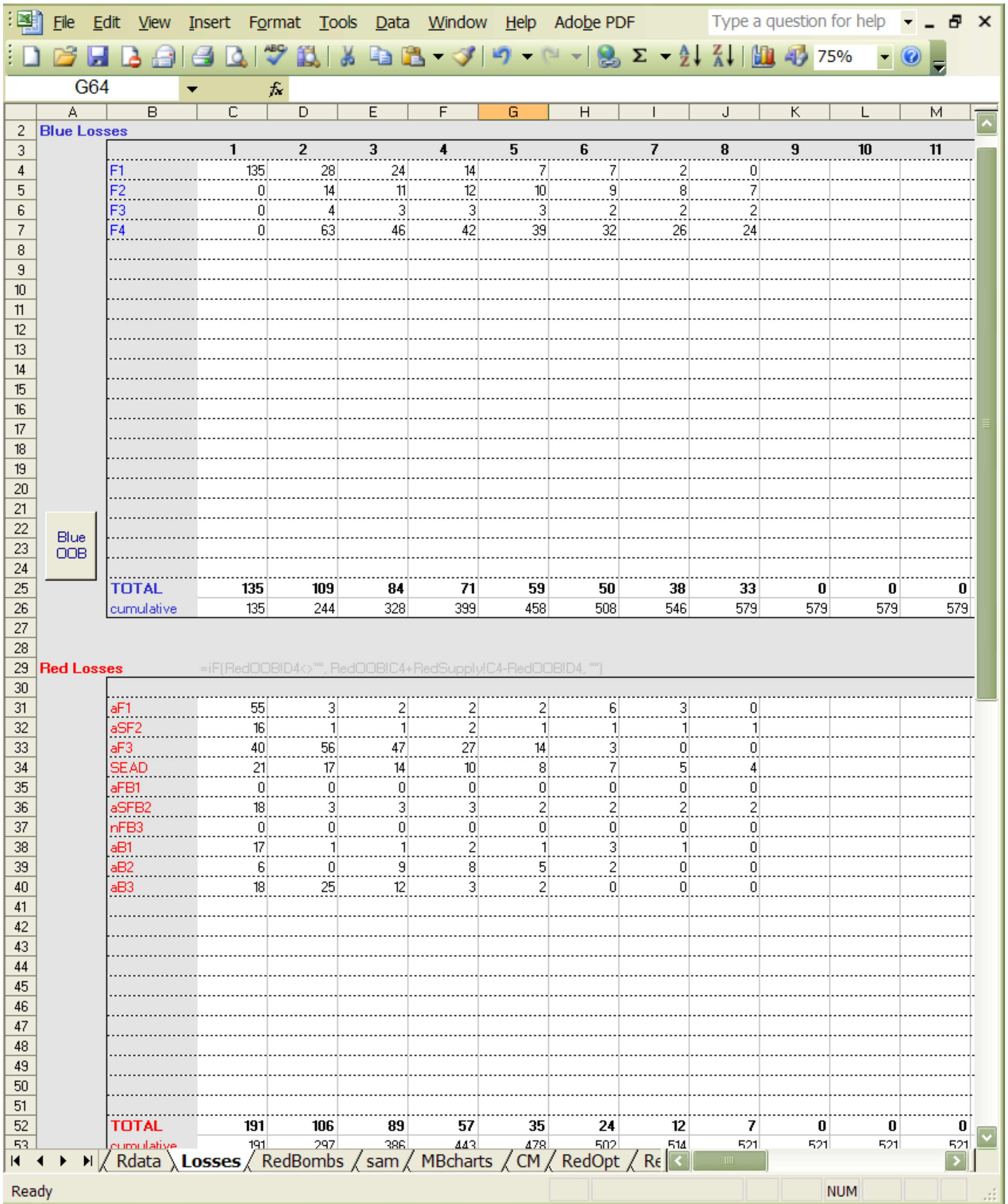
Worksheet Tables

Because SLAACM is housed in an Excel workbook, the user can browse the data used to build the charts. All of the charts in Figure 7-11 are built from data in worksheet tables:

- The Red Forces chart uses the data on the "RedOOB" worksheet.
- The Blue Forces and LAD charts use data from the "BlueOOB" worksheet. (As we discussed earlier, the BlueOOB worksheet is both an input and an output. After running SLAACM, available aircraft quantities are loaded into BlueOOB worksheet for each day of the campaign.)
- The Tons of Bombs chart uses data from the "RedBombs" worksheet.
- The Cruise Missile chart is built from data on the "CM" worksheet, where the chart resides.
- The SAMs, SEADs, and bombers chart is built from data on the "sam" worksheet.

It can be helpful to look at aircraft losses to get a better idea of how the campaign progressed. SLAACM has a worksheet named "Losses" that shows the loss data. Figure 7-15 shows the Losses worksheet. Blue and Red losses are shown by aircraft type. In addition, the worksheet shows the standard deviation of the losses for Blue aircraft.

Figure 7-15. Attack SLAACM: Losses



Blue Losses

	1	2	3	4	5	6	7	8	9	10	11
F1	135	28	24	14	7	7	2	0			
F2	0	14	11	12	10	9	8	7			
F3	0	4	3	3	3	2	2	2			
F4	0	63	46	42	39	32	26	24			
TOTAL	**135**	**109**	**84**	**71**	**59**	**50**	**38**	**33**	**0**	**0**	**0**
cumulative	135	244	328	399	458	508	546	579	579	579	579

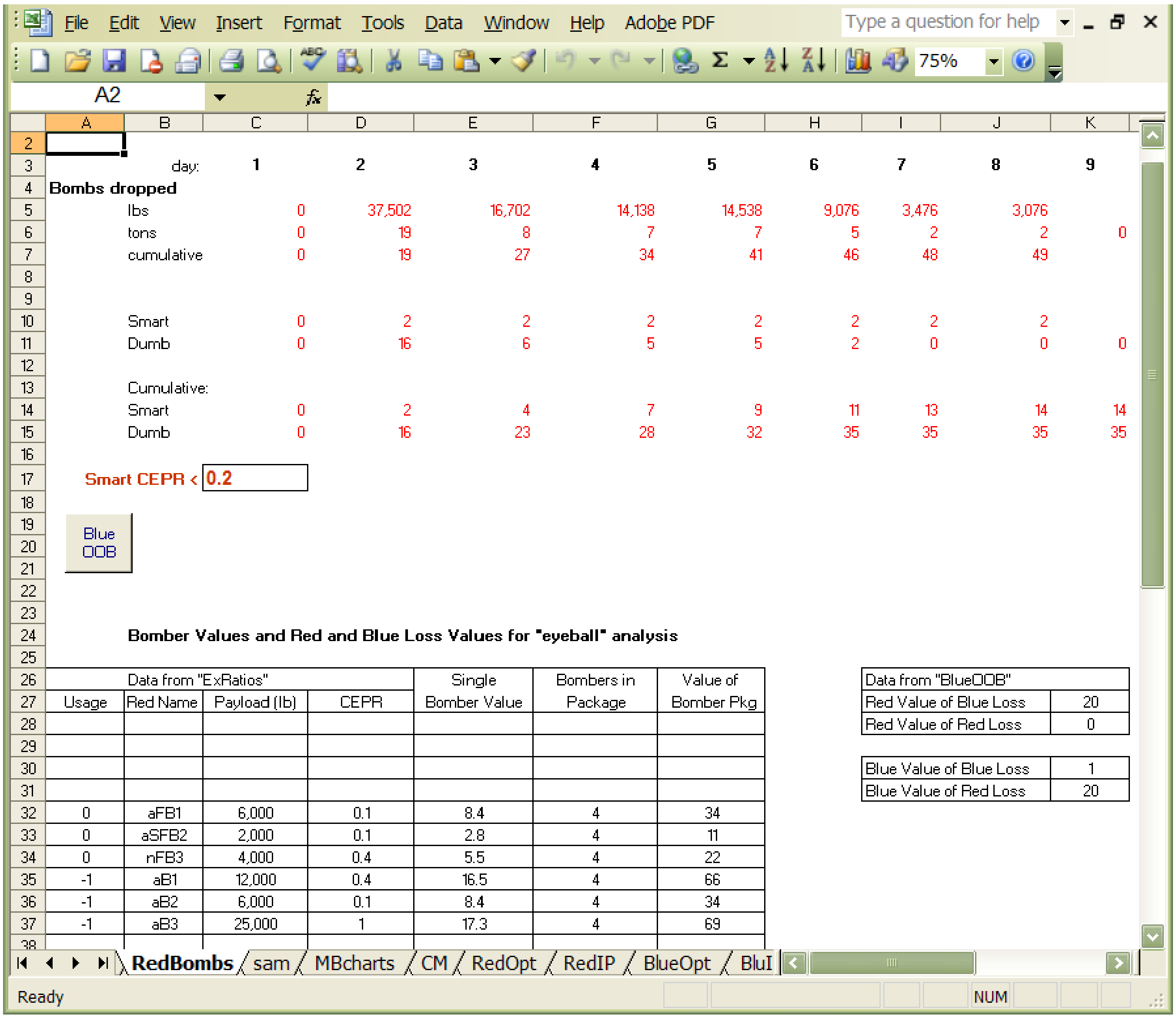
Red Losses =iF(RedOOB!D4<>"", RedOOB!C4+RedSupply!C4-RedOOB!D4, "")

aF1	55	3	2	2	2	6	3	0			
aSF2	16	1	1	2	1	1	1	1			
aF3	40	56	47	27	14	3	0	0			
SEAD	21	17	14	10	8	7	5	4			
aFB1	0	0	0	0	0	0	0	0			
aSFB2	18	3	3	3	2	2	2	2			
nFB3	0	0	0	0	0	0	0	0			
aB1	17	1	1	2	1	3	1	0			
aB2	6	0	9	8	5	2	0	0			
aB3	18	25	12	3	2	0	0	0			
TOTAL	**191**	**106**	**89**	**57**	**35**	**24**	**12**	**7**	**0**	**0**	**0**
cumulative	191	297	386	443	478	502	514	521	521	521	521

Figure 7-16 shows the pounds and tons of bombs dropped by Red aircraft each day of the campaign. In addition, the bomb types are broken out by smart and not-smart types. These are the data used to build the “Tons of Bombs Dropped” chart in the lower right corner of Figure 7-12a.

The RedBombs worksheet also contains a table showing bomber payloads, cepr, and bomber values. The values for bomber effectiveness used in the optimizations are calculated based on payload tonnage and accuracy, as in previous versions of SLAACM. Now, however, those calculated values—along with the fighter values—are displayed on the RedBombs worksheet to support "eyeball" analysis by the user. The RedBombs spreadsheet with this display is shown in Figure 7-16.

Figure 7-16. Attack SLAACM: Red Bombs



	day:	1	2	3	4	5	6	7	8	9
Bombs dropped										
	lbs	0	37,502	16,702	14,138	14,538	9,076	3,476	3,076	
	tons	0	19	8	7	7	5	2	2	0
	cumulative	0	19	27	34	41	46	48	49	
	Smart	0	2	2	2	2	2	2	2	
	Dumb	0	16	6	5	5	2	0	0	0
	Cumulative:									
	Smart	0	2	4	7	9	11	13	14	14
	Dumb	0	16	23	28	32	35	35	35	35

Smart CEPR < 0.2

Blue OOB

Bomber Values and Red and Blue Loss Values for "eyeball" analysis

Data from "ExRatios"				Single	Bombers in	Value of
Usage	Red Name	Payload (lb)	CEPR	Bomber Value	Package	Bomber Pkg
0	aFB1	6,000	0.1	8.4	4	34
0	aSFB2	2,000	0.1	2.8	4	11
0	nFB3	4,000	0.4	5.5	4	22
-1	aB1	12,000	0.4	16.5	4	66
-1	aB2	6,000	0.1	8.4	4	34
-1	aB3	25,000	1	17.3	4	69

Data from "BlueOOB"	
Red Value of Blue Loss	20
Red Value of Red Loss	0
Blue Value of Blue Loss	1
Blue Value of Red Loss	20

User Note

The output worksheets in SLAACM are dynamic and are overwritten during each model run. Users who want to save run data should copy and paste the values into an external file.

SUMMARY

Attack SLAACM is a fast, flexible, robust model that contains the key input parameters necessary to define the characteristics of a realistic air-to-air campaign. Model results in both tabular and graphical format clearly display the impacts of parameter choices and provide insight into campaign scenarios. The Excel workbook implementation allows results to be easily copied and pasted to other applications for reporting and presentation.

Appendix A
Direct Computation of Long-Time Limiting Probabilities of Boundary States

The structure of the evolution equations for probability distributions of the engagements we consider makes it possible to find the long-term limits of the absorbing boundary states directly, in finitely many steps, without formally solving those differential equations. In some interesting cases, the finite steps may be carried out by straightforward iteration. This appendix explains these facts and gives some examples.

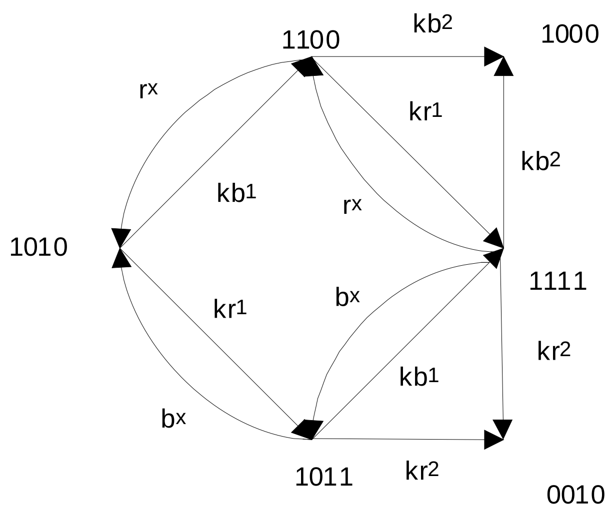
The evolution equations for the state probabilities in our engagement models decompose naturally into a set of equations for the transient states and a set for the absorbing boundary states. Equations 2-5 and 2-6 are examples of this for the basic M vs. N probabilistic engagement model. The advanced engagement models of Appendix D also have this property, except for the low-observable model, which is treated completely by the method given in Appendix D.

If we make the probabilities of the transient states the components of a vector x, and the components of the absorbing boundary states into the components of a vector y, the evolution of the state probabilities can be described by

$$\begin{aligned} \dot{x} &= Ax \,,\ x(0) = x_0 \\ \dot{y} &= Bx \end{aligned} \qquad \text{[Eq. A-1]}$$

To illustrate these ideas for an example that, unlike the one of Chapter 2, has recurrent states, let us consider treating a 1 vs. 1 engagement with a two-phase kill model with break-lock. We use the state description (m, i; n, j) where m denotes the number of Blue aircraft, i the number of Blues tracking opponents, n the number of Red aircraft, and j the number of Reds tracking opponents. Figure A-1 is a diagram of the states in this engagement and their transitions.

Figure A-1. Diagram of 1 vs. 1 Engagement



The evolution equations are

$$
\begin{aligned}
\dot{P}_{1010} &= -(kb_1 + kr_1)P_{1010} + b_x P_{1011} + r_x P_{1110} \\
\dot{P}_{1110} &= -(kb_2 + kr_1 + r_x)P_{1110} + kb_1 P_{1010} + b_x P_{1111} \\
\dot{P}_{1011} &= -(kr_2 + kb_1 + b_x)P_{1011} + kr_1 P_{1010} + r_x P_{1111} \\
\dot{P}_{1111} &= -(kb_2 + kr_2 + r_x + b_x)P_{1111} + kr_1 P_{1110} + kb_1 P_{1011} \\
\dot{P}_{1000} &= kb_2(P_{1110} + P_{1111}) \\
\dot{P}_{0010} &= kr_2(P_{1011} + P_{1111})
\end{aligned}
\qquad \text{[Eq. A-2]}
$$

The first four of these describe the transient states, and the last two describe the absorbing boundary states. Defining x_1 as P_{1010}, x_2 as P_{1110}, x_3 as P_{1011}, and x_4 as P_{1111}, and defining y_1 as P_{1000}, y_2 as P_{0010}, we see that, in this case, the matrices A and B of Equation A-1 are

$$
A = \begin{pmatrix}
-(kb_1 + kr_1) & r_x & b_x & 0 \\
kb_1 & -(kb_2 + kr_1 + r_x) & 0 & b_x \\
kr_1 & 0 & -(kr_2 + kb_1 + b_x) & r_x \\
0 & kr_1 & kb_1 & -(kr_2 + kb_2 + r_x + b_x)
\end{pmatrix}
\qquad \text{[Eq. A-3]}
$$

and

$$
B = \begin{pmatrix}
0 & kb_2 & 0 & kb_2 \\
0 & 0 & kr_2 & kr_2
\end{pmatrix}.
\qquad \text{[Eq. A-4]}
$$

Now, the solutions of systems of ordinary differential equations of the form $\dot{x} = Ax$ for constant coefficient matrices A can always be expressed as finite linear combinations of generalized exponential functions of the time, that is, functions of the form $t^j e^{\lambda t}$, where the j are positive integers and the λ are the eigenvalues of the matrix A.[1] If the transients are, in fact, transient, then the real parts of the λ are all strictly less than 0. It follows that the $\hat{x}_i$, defined by

$$\hat{x}_i \equiv \int_0^\infty x_i(t)dt \qquad \text{[Eq. A-5]}$$

exist. Then, on integrating the differential equations of Equation A-1 from 0 to ∞, using the initial condition on x and y, and remembering that the $x_i(t)$ tend to zero as $t \to \infty$, we find

$$\begin{aligned} -x_0 &= A\hat{x} \\ y_{\lim} &= B\hat{x} \end{aligned} \qquad \text{[Eq. A-6]}$$

where $y_{\lim}$ denotes the limit of y as $t \to \infty$. It follows that

$$y_{\lim} = -BA^{-1}x_0 . \qquad \text{[Eq. A-7]}$$

Equation A-7 expresses the long-time limiting values of the absorbing boundary state probabilities as the result of finite operations, that is, matrix inversion and multiplication.

Let us continue to illustrate these concepts with the 1 vs. 1 example considered above. For a numerical example, we take $kr_1 = 1$, $kr_2 = 2$, $r_x = 1$, $kb_1 = 3$, $kb_2 = 4$, and $b_x = 2$. Then the matrices A and B take the values

$$A = \begin{pmatrix} -4 & 1 & 2 & 0 \\ 3 & -6 & 0 & 2 \\ 1 & 0 & -7 & 1 \\ 0 & 1 & 3 & -9 \end{pmatrix} \qquad \text{[Eq. A-8]}$$

and

$$B = \begin{pmatrix} 0 & 4 & 0 & 4 \\ 0 & 0 & 2 & 2 \end{pmatrix}. \qquad \text{[Eq. A-9]}$$

It is as well to check that the eigenvalues of A are distinct and negative. Direct calculation shows that this is in fact the case: the eigenvalues are approximately

[1] W.E. Boyce and R.C. DiPrima, *Elementary Differential Equations*, Third Edition (New York: Wiley, 1976), Sections 7-6 through 7-9.

−2.44, −6.44, −6.56, and −10.56. Assured by this that the $\hat{x}_i$ exist, we go on to find

$$y_{\lim} = -BA^{-1}x_0 = \begin{pmatrix} 0.827 \\ 0.173 \end{pmatrix}. \quad \text{[Eq. A-10]}$$

That is, the probability that the Blue aircraft defeats the Red one is roughly 83 percent.

The eigenvalues of the plant matrix A have another use: they tell one whether or not it makes sense to look for the long-time limiting values of the absorbing boundary-state probabilities. Those probabilities are of interest if, but only if, an actual engagement can continue long enough for the system to be in boundary states with a probability near one. The time for this to happen is the time for which the transient states' probabilities are all much less than one.

When the eigenvalues of A are distinct (and, of course, have negative real parts) so that the $x_i(t)$ are linear combinations of the functions $\exp(\lambda_i t)$, that time will be no greater than the time at which $\exp(\lambda^* t)$ becomes much less than one, where λ^* is the eigenvalue closest to zero. The time to make that happen is approximately $2/\lambda^*$. If the rate parameters kb_1, … of our example are in inverse minutes, the mean time for the Blue aircraft to acquire a target is 20 seconds, the mean time for them to launch a successful missile is 15 seconds, and the mean time for them to break a Red lock is 30 seconds, while the mean time for the Reds to make lock is 60 seconds, the mean time for them to launch a successful missile is 30 seconds, and the mean time for them to break lock is 60 seconds. The negative eigenvalue of the plant matrix A with smallest magnitude is −2.44 inverse minutes, which implies that the time for the transient phase of the 1 vs. 1 engagement to be over is roughly 50 seconds. Very likely, the combatants will have enough fuel to fight that long, and so the long-time limit is meaningful in our example.

Although we have found numerical evaluation of Equation A-7 quite helpful in generating insight with examples of modest dimension, determining long-time limiting probabilities in this way may not be practical for systems of large dimension. In an important class of engagement models, however, the difficult task of computing $A^{-1} x_0$ can be done with a straightforward (although possibly lengthy) iterative scheme. That class is those engagement models whose diagrams are acyclic, like the example of Chapter 2. For these models, the plant matrix A is lower triangular, and this provides the iterative scheme. In these cases, one has

$$\begin{aligned} &a_{11}\hat{x}_1 = -1 \\ &a_{21}\hat{x}_1 + a_{22}\hat{x}_2 = 0 \\ &\ldots \end{aligned} \quad \text{[Eq. A-11]}$$

so that

$$\begin{aligned} \hat{x}_1 &= -1/a_{11} \\ \hat{x}_2 &= -\hat{x}_1\, a_{21}/a_{22} \\ &\ldots \end{aligned} \qquad \text{[Eq. A-12]}$$

With the $\hat{x}$ determined, the long-time limiting values of the absorbing boundary states follow from Equation A-7.

Determining the set of states actually occupied in a given engagement can be somewhat tedious. For example, finding the long-time limit of a missile-tracking engagement model for 4 vs. 4, when each aircraft has 6 missiles, involves more than 42,000 states. Obviously, it is not practical to evaluate these cases by hand. In these cases, we generally use C++ code to find both the set of states occupied and the solution (Equation A-12).

Appendix B
Analyzing Large State-Space Engagement Models with ASSIST and STEM

Evolution equations like Equations 2-5 and 2-6 of Chapter 2 are simple in principle. They are systems of linear ordinary differential equations with constant coefficients. The solutions of the initial value problems formed by such equations and specifications of the starting probabilities of the various states can be written symbolically—and perhaps treated practically—with matrix exponentiation, and they generally offer no difficulty to numerical solution by difference schemes, but they can be of large dimension.

A basic 4 vs. 4 engagement has 24 states and 32 transitions; adding 6 missiles to the Blue aircraft produces 1,632 states and 9,087 transitions; adding 6 missiles to both Red and Blue produces 108,534 states and 2,416,252 transitions. Also, it is somewhat awkward to determine the long-time limits of the absorbing boundary states by integrating the differential equations numerically. This appendix discusses NASA-developed tools that we have found helpful in dealing with both these issues.

The decision to apply the NASA tools to the problem of fighter combat was serendipitous. One of us was engaged in Markov analysis of fighter combat, and another of us was using the NASA tools for safety-related reliability analysis. The need for a tool to efficiently develop relatively complex fighter combat models surfaced during informal discussions. Using the NASA tools, we can develop and run complex models for Blue versus Red combat, including tracking of Blue missile use, in relatively short order. These results provide insight into the combat problem and support development of the SLAACM algorithms.

NASA has three tools that can be applied to fighter combat. Two are Markov computational analysis programs. The third is a sophisticated utility program that generates the inputs (models) for the analysis programs. The tools were developed by NASA to estimate failure probabilities in highly reliable, reconfigurable avionics and space electronics.

The two analysis programs are Scaled Taylor Exponential Matrix (STEM)[1] and Semi-Markov Unreliability Range Estimator (SURE).[2] The former is a pure Markov analysis tool in which all failure rates are constant with state probabilities

[1] NASA, *The PAWS and STEM Reliability Analysis Programs*, Technical Memorandum 100572, R. Butler and P. Stevenson, March 1988.

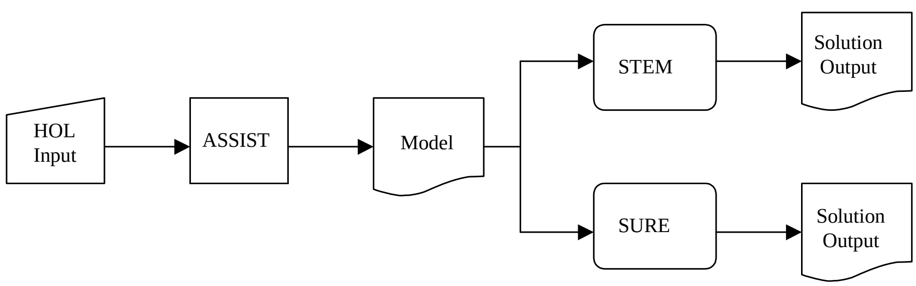
[2] NASA, *SURE Reliability Analysis, Program and Mathematics*, Technical Paper 2764, R. Butler and A. White, 1998.

having generalized exponential form. The latter is a semi-Markov model that allows use of non-exponential reconfiguration probabilities. STEM and SURE use identical input files, but some commands are processed only by SURE. For reasons discussed below, STEM is the tool used for fighter combat calculations. The utility program ASSIST (Abstract Semi-Markov Specification Interface to the SURE Tool)[3] generates the STEM and SURE input files, or "models." ASSIST allows the straightforward generation of extremely complex Markov models.[4]

Reliability analyses are typically conducted for short model times compared to the failure rates of the components studied, such as a 10-hour flight and a failure rate of 0.0001 failure per hour. For air combat models, we are interested in running the engagement to completion so we will run model time units on the order of the reciprocal of the lowest kill rate, e.g., 1,000 units for a kill rate of 0.001. STEM has no problem running such models, but SURE's mathematical algorithms that handle non-Markov recovery rates are not designed to run to completion. Consequently, the remainder of the discussion will focus on ASSIST and STEM.

All three programs were developed for UNIX platforms and have been converted to Windows. STEM is also available on LINUX. The programs are available at no charge from NASA Langley Research Center. Documentation includes an ASSIST users guide, a report on modeling techniques, and reports on SURE and STEM mathematics and performance. Figure B-1 shows the basic relationships of the tools.

Figure B-1. Basic Relationships of the Tools



HOL: Higher Order Language
ASSIST: Abstract Semi-Markov Specification Interface to the SURE Tool – Input model generator
Model: Designation for the input files for the Markov analysis programs, containing variables, states, transitions, and rates
STEM: Scaled Taylor Exponential Matrix – Markov analysis program
SURE: Semi-Markov Unreliability Range Estimator – Markov and Semi-Markov analysis program

[3] NASA Langley Research Center, *ASSIST User Manual*, S. Johnson and D. Boerschlein, September 1993.

[4] NASA, *Techniques for Modeling the Reliability of Fault-Tolerant Systems with the Markov State-Space Approach*, Reference Publication 1348, R. Butler and S. Johnson, September 1995.

FIGHTER COMBAT ANALYSIS

In this section, we use case examples of increasing complexity to describe the capabilities and limitations of the tools for analysis of fighter combat. All the analyses use the common structure of an M vs. N, Blue vs. Red engagement with constant kill rates.

Example 1—Simple 2 vs. 2 Engagement

The 2 vs. 2 engagement demonstrates the basic structure of the models without generating extensive output. The Blue fighter kill rate is 0.1 per unit time, and the Red fighter kill rate is 0.01 per unit time. Listing 1 is the ASSIST code that generates the 2 vs. 2 model.

Listing 1. ASSIST Code for Generating the 2 vs. 2 Model

```
(* ASSIST model for Fighter Combat *)

LIST = 3; (* 1 lists accumulated death state probabilities only *)
(* 2 lists all state probabilities *)
(* 3 lists transitions and all state probabilities *)
ONEDEATH OFF; (* OFF enumerates all absorbing (Death) states *)
(* comment out to consolidate absorbing states *)

(* equipment list *)
nblue = 2; (* blue fighters *)
nred = 2; (* red fighters *)

(* kill probabilities *)
k_rate_blue = 0.1;
k_rate_red = 0.01;

SPACE = (blue: 0.nblue, red: 0.nred);
START = (nblue, nred);

DEATHIF (blue = 0) AND (red>0);
DEATHIF (red = 0) AND (blue>0);

(* Blue kills *)
IF (blue > 0)and (red>0) TRANTO red = red-1 BY blue*k_rate_blue;

(* Red kills *)
IF (red > 0 ) and (blue>0) TRANTO blue = blue-1 BY red*k_rate_red;
```

ASSIST uses a higher order definition language that allows algebraic manipulation of variables and compact description of Markov state transfers.[5] Several features are noteworthy in the listing above. The listing starts with the editing commands LIST and ONEDEATH that control SURE/STEM output. Next, the number of aircraft and their kill rates are input parameters defined as constant types. The SPACE statement defines the range of the Markov state space, i.e., in this case, the first state can vary from 0 to nblue where nblue equals 2. The START statement identifies the initial state populations. The DEATHIF statements define the absorbing states in the model; in this case, the two DEATHIF conditions represent Blue and Red wins. If the ONDEATH OFF command is enabled, STEM and SURE list all

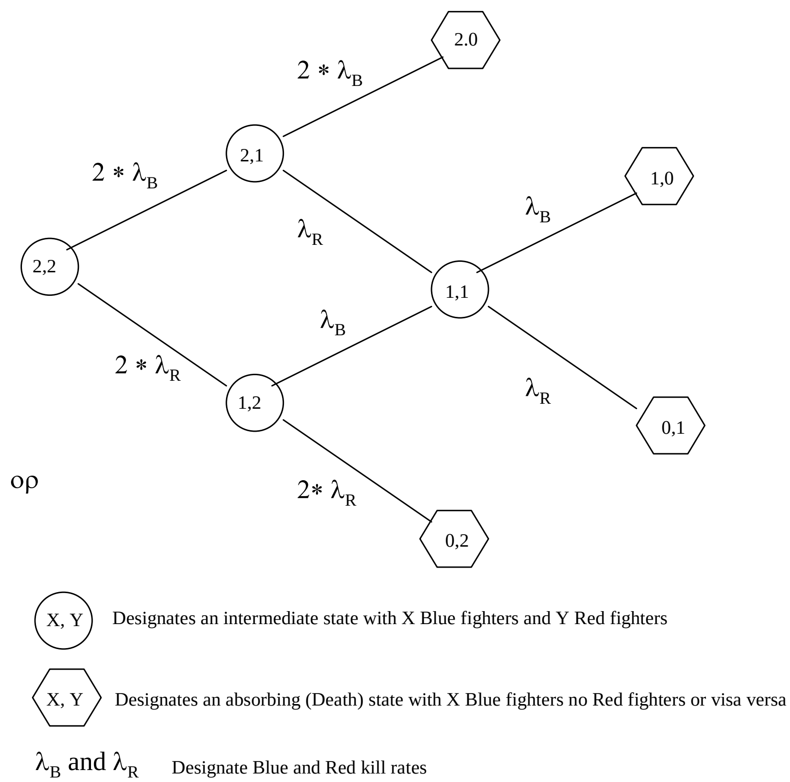
[5] ASSIST inputs are fully defined in the citations of Notes 1–3.

the individual absorbing states; otherwise, they combine the results for absorbing states into the appropriate Death states.[6]

We now come to the powerful TRANTO statements, which define the conditions, nature, and rate of Markov state transfers. In this example, the transfers occur whenever both Blue and Red fighters are available; kills are made one at a time, and the kill rates are proportional to the number of killers. The two TRANTO statements above are adequate to generate the models for any combination of Red and Blue aircraft.

The state diagram for the 2 vs. 2 engagement, Figure B-2, is relatively simple. It shows that there are 8 total states, of which 4 are absorbing (Death) states and 8 are transitions.[7]

Figure B-2. 2 vs. 2 Engagement Diagram



[6] SURE mathematics requires the existence of absorbing states, while STEM can handle models that include recovery from all states.

[7] There is no (0, 0) absorbing state because we chose not to include the case in which the last two opponents simultaneously shoot each other down. Such a state could be modeled with an additional TRANTO statement, but determination of the controlling kill rate would take some careful thought. The pure statistical rate for independent events suggests that it would be equal to K_RATE_BLUE * K_RATE_RED, but simultaneous kills can happen only in certain conditions such as ramming, head-to-head gun attacks, and head-to-head missile attacks. These are only a subset of the configurations that make up the basic Blue and Red kill rates. We assessed the likely rate to be sufficiently low to justify ignoring the (0, 0) state until better data become available.

Running ASSIST with the listing above generates the STEM/SURE input file or "model" shown in Listing 2. Note the 8 states, the 4 absorbing states (0, N or M, 0), and the 8 transitions. The file extension is ".mod."

Listing 2. 2 vs. 2 Engagement Model

```
LIST = 3;
TIME = 100;
NBLUE = 2;
NRED = 2;
MIN_B = 0;
MIN_R = 0;
K_RATE_BLUE = 0.1;
K_RATE_RED = 0.01;


1(* 2,2 *), 2(* 2,1 *) = 2*K_RATE_BLUE;
1(* 2,2 *), 3(* 1,2 *) = 2*K_RATE_RED;
2(* 2,1 *), 4(* 2,0 *) = 2*K_RATE_BLUE;
2(* 2,1 *), 5(* 1,1 *) = 1*K_RATE_RED;
3(* 1,2 *), 5(* 1,1 *) = 1*K_RATE_BLUE;
3(* 1,2 *), 6(* 0,2 *) = 2*K_RATE_RED;
5(* 1,1 *), 7(* 1,0 *) = 1*K_RATE_BLUE;
5(* 1,1 *), 8(* 0,1 *) = 1*K_RATE_RED;

(* NUMBER OF STATES IN MODEL = 8 *)
(* NUMBER OF TRANSITIONS IN MODEL = 8 *)
```

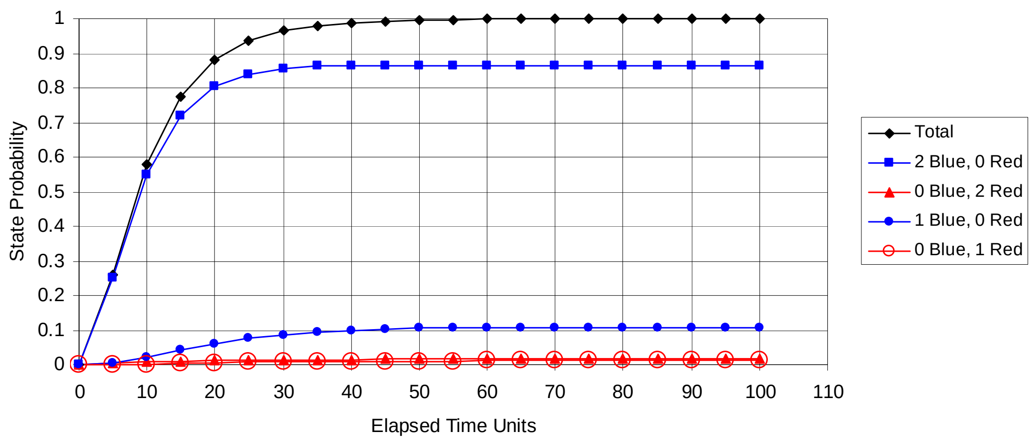
The model file contains the input definitions and transition descriptions needed by SURE and STEM. Transitions are defined by source state number, destination state number, and rate of transfer between the source and destination. The state descriptions, e.g., (*2,2*), are included as comments; "(*" and "*)" are comment delimiters. Note that the absorbing states, such as state 4, (*2,0*), appear only in the destination column. The DEATHIF definitions and ONEDEATH command are not printed. Listing 3 is the STEM output file for the model above, run for 10 time units.

Listing 3. 2 vs. 2 Engagement STEM Output

```
Model = C:\Markov1\fighters\Report2x2.mod
——- RUN #1

D-STATE PROBABILITY ACCURACY
———-——————————-————
4 8.65800856430E-0001
6 1.51513103495E-0002
7 1.08204982318E-0001
8 1.08204982318E-0002

———-——————————-
TOTAL 9.99977647330E-0001 11 DIGITS

STATE PROBABILITY
———-——————————
1 2.78946809287E-0010
2 9.58618467009E-0009
3 1.22878668130E-0006
4 8.65800856430E-0001
5 2.11140182554E-0005
6 1.51513103495E-0002
7 1.08204982318E-0001
8 1.08204982318E-0002
```

Figure B-3 shows the STEM results for increasing units of combat time. As discussed in Chapter 2, we use the reciprocal of the lowest rate as a rule of thumb for setting the model run time. In this case, the lowest rate is K_RATE_RED = 0.01 and its reciprocal is 100. By Time = 70, the total probability of being in an absorbing state is 0.9995, and at Time = 100, it is 0.99998, indicating that our rule of thumb is good. The state probabilities at Time = 100 for the individual absorbing states are (2B, 0R) = 0.8658, (1B, 0R) = 0.1082, (0B, 2R) = 0.0152, and (0B, 1R) = 0.0108.

Figure B-3. 2 vs. 2 Engagement Results versus Engagement Time



Example 2—2 vs. 2 Engagement with Blue Missiles

This example adds the tracking of Blue missiles to the simple 2 vs. 2 engagement. We assume that Blue aircraft each carry 6 missiles, each having a known single shot probability of kill, P_k. It is important to understand in this discussion that Blue and Red kills are still dependent only on the kill rate ratio, and that missile P_k is used only to calculate missile consumption. For simplicity and clarity, we assume, for this example, that Red aircraft have unlimited missiles.

Our basic approach is to expand the state space to (b0, b1, b2, b3, b4, b5, b6, r) to include separate states for Blue aircraft having 0 to 6 missiles. We still need only one state vector for Red aircraft. At the start of an engagement, we have 2 Blue aircraft in the b6 6-missile state and 2 Red aircraft in the r Red state.

We still transition from Red states in only two ways: Red can kill a Blue, or vice versa. Now, however, we can transition from a Blue state in three ways: Blue can kill a Red, Blue can fire and miss, or Red can kill a Blue. To formulate the transfer statements, we need to determine the missile miss rate. We know the Blue kill rate, λ, and we know the missile single shot probability of kill, P_k. We want to estimate the missile usage, including misses. We assume, for now, that the successful missile usage is one missile per kill, i.e., we do not fire salvos. We also assume the kill rate includes both the missile that hit and any that missed.

Now, we define μ to be the missile firing rate, and let n be the number of Blue aircraft engaged. The successful missile rate is then: $n*P_k*\mu$, and the unsuccessful missile rate (miss rate) is $n*(1-P_k)*\mu$.

Based on our assumptions, the successful missile rate must equal the kill rate, i.e.,

$$n*P_k*\mu = n*\lambda \qquad \text{[Eq. B-1]}$$

and, therefore, the missile firing rate is

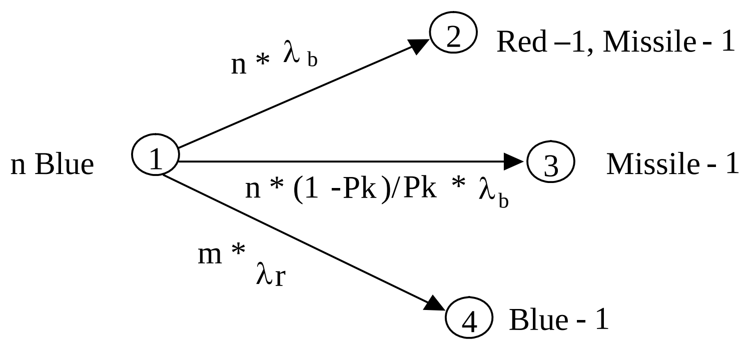
$$n*\mu = n*\frac{\lambda}{P_k}, \text{ and } \mu = \frac{\lambda}{P_k}. \qquad \text{[Eq. B-2]}$$

Substituting for μ, the miss rate, $n*(1-P_k)*\mu$, becomes

$$n*(1-P_k)*\frac{\lambda}{P_k} \text{ or } n*\frac{(1-P_k)}{P_k}*\lambda. \qquad \text{[Eq. B-3]}$$

Figure B-4 is the state transition diagram for the single missile per kill case.

Figure B-B-4. State Diagram for a Single Missile per Kill Engagement



Missile Firing Rate = μ
Missile Kill Rate = $P_k\mu$
Missile Miss Rate = $(1-P_k)\mu$
Engagement Kill Rates = λ_b, λ_r
Aircraft in Engagement = n Blue, m Red

We can generalize the derivation above in a straightforward manner for cases in which multiple missiles are fired per engagement by noting that the failure probability for x failures is $(1 - P_k)^x$ and substituting $[1 - (1-P_k)^x]$ for P_k:

- Define μ to be the missile firing rate.
- Let n be the number of Blue aircraft engaged.

- Let *x* be the number of missiles fired per kill.

The successful missile rate is $n*[1-(1-P_k)^x]*\mu$, and the unsuccessful missile rate (miss rate) is $n*(1-P_k)^x*\mu$.

Based on our assumptions, the successful missile rate must equal the kill rate, i.e.,

$$n*[1-(1-P_k)^x]*\mu = n*\lambda \qquad \text{[Eq. B-4]}$$

and, therefore, the missile firing rate is

$$n*\mu = n*\frac{\lambda}{\left[1-(1-P_k)^x\right]}, \text{ and } \mu = \frac{\lambda}{\left[1-(1-P_k)^x\right]}. \qquad \text{[Eq. B-5]}$$

The miss rate, $n*(1-P_k)^x*\mu$, now becomes

$$n*(1-P_k)^x*\frac{\lambda}{\left[1-(1-P_k)^x\right]} \text{ or } n*\frac{(1-P_k)^x}{\left[1-(1-P_k)^x\right]}*\lambda. \qquad \text{[Eq. B-6]}$$

For *x* = 2 missiles per kill, we have a miss rate of

$$n*\frac{1-2P_k+P_k^2}{2P_k-P_k^2}*\lambda. \qquad \text{[Eq. B-7]}$$

In addition to missile counting, we also add the capability for either Red or Blue to exit the combat when losses reach a preset level by assigning minimum values (min_b, min_r) to the DEATHIF conditions. If min_b = 0 and min_r = 0, the model corresponds to fighting to annihilation; higher values will generate models reflecting disengagement breakpoints. Listing 4 is the ASSIST code for the 2 vs. 2 case with missile tracking.

Listing 4. ASSIST Code for a 2 vs. 2 Engagement with Blue Missile Tracking

```
(* ASSIST code for 2 Blue vs. 2 Red with 6 missiles per Blue aircraft *)

LIST = 3; (* 1 lists accumulated death state probabilities only *)
(* 2 lists all state probabilities *)
(* 3 lists transitions and all state probabilities *)
ONEDEATH OFF; (* OFF enumerates all absorbing (Death) states *)
(* comment out to consolidate absorbing states *)

Time = 100;

(* equipment list *)
nblue = 2; (* blue fighters *)
nred = 2; (* red fighters *)

(* kill rates *)
k_rate_blue = 0.1; (* blue kill rate *)
k_rate_red = 0.01; (* red kill rate *)

pk = 0.85; (* missile kill probability *)
miss_rate = (1-pk)/pk*k_rate_blue; (* missile miss rate *)
```

```
(* minimum number of combatants for state pruning *)
min_b = 0; (* minimum number of blue aircraft *)
min_r = 0; (* minimum number of red aircraft *)

SPACE =
(n0:0.nblue,n1:0.nblue,n2:0.nblue,n3:0.nblue,n4:0.nblue,n5:0.nblue,n6:0.nblue,
red:0.nred);
START = (0,0,0,0,0,0,nblue,nred);

DEATHIF (n0+n1+n2+n3+n4+n5+n6)<=min_b;;
DEATHIF red <= min_r;

(* transition cases: Blue kill then Red kill*)
IF (n6>0) AND (red>0) THEN
TRANTO n6=n6-1, n5=n5+1, red=red-1 BY n6*k_rate_blue;
TRANTO n6=n6-1, n5=n5+1 BY n6*miss_rate;
TRANTO n6=n6-1 BY (n6/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n5>0) AND (red>0) THEN
TRANTO n5=n5-1, n4=n4+1, red=red-1 BY n5*k_rate_blue;
TRANTO n5=n5-1, n4=n4+1 BY n5*miss_rate;
TRANTO n5=n5-1 BY (n5/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n4>0) AND (red>0) THEN
TRANTO n4=n4-1, n3=n3+1, red=red-1 BY n4*k_rate_blue;
TRANTO n4=n4-1, n3=n3+1 BY n4*miss_rate;
TRANTO n4=n4-1 BY (n4/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n3>0) AND (red>0) THEN
TRANTO n3=n3-1, n2=n2+1, red=red-1 BY n3*k_rate_blue;
TRANTO n3=n3-1, n2=n2+1 BY n3*miss_rate;
TRANTO n3=n3-1 BY (n3/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n2>0) AND (red>0) THEN
TRANTO n2=n2-1, n1=n1+1, red=red-1 BY n2*k_rate_blue;
TRANTO n2=n2-1, n1=n1+1 BY n2*miss_rate;
TRANTO n2=n2-1 BY (n2/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n1>0) AND (red>0) THEN
TRANTO n1=n1-1, n0=n0+1, red=red-1 BY n1*k_rate_blue;
TRANTO n1=n1-1, n0=n0+1 BY n1*miss_rate;
TRANTO n1=n1-1 BY (n1/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

(* This line is commented out based on Blues ability to disengage after firing
all his missiles *)
(*IF (red>0) AND (n0>0) TRANTO n0=n0-1 BY
(n0/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;*)
```

Adding the missile Pk, missile miss rate, Blue and Red aircraft minimums, the expanded state space, and the starting conditions is straightforward. The expansion of the transfer statements deserves some discussion.

The basic transition logic is as follows:

- Separate sets of transfers are established for each Blue missile state.
- State transfers occur only when Blue aircraft are in the state and Red aircraft are available.

- Blue kills at a rate proportional to the total number of Blue aircraft. When Blue makes a kill, one aircraft is removed from the Blue state, one aircraft is added to the Blue state having one fewer missiles, and one aircraft is removed from the Red state.

- Blue misses at a rate proportional to the number of aircraft in the Blue state. When Blue misses, one aircraft is removed from the Blue state, one aircraft is added to the Blue state having one fewer missiles, but no aircraft are removed from the Red state.

- Red kills Blue at a rate proportional to the number of Red aircraft and the fraction of Blue aircraft in the Blue state. When Red makes a kill, one aircraft is removed from the Blue state.

- Only Red kills are possible when Blue has no missiles left. This TRANTO statement is commented out in the example listing using "(*" and "*)" operators based on the assumption that Blue can disengage at will. Another alternative would be to give Blues with no missiles a "guns only" kill rate.

Running the ASSIST program above generates a model having 103 states, including 34 absorbing (Death) states, and 301 transitions. Listing 5 shows the first and last few lines of the model file. The state notation in the model shows the state number and the contents of each element in the state. This model has eight elements in each state. The first seven elements are Blue aircraft with 0 to 6 missiles, and the last element is Red aircraft. Thus, State 1 is 1(0,0,0,0,0,0,2,2), indicating 2 Blues with 6 missiles and 2 Reds. The first transition is to 2(0,0,0,0,0,1,1,1), indicating that one Blue has fired a missile and killed one Red. A Blue absorbing state corresponds to 0s in all of the first seven elements, and a Red absorbing state has 0 in the eighth element.

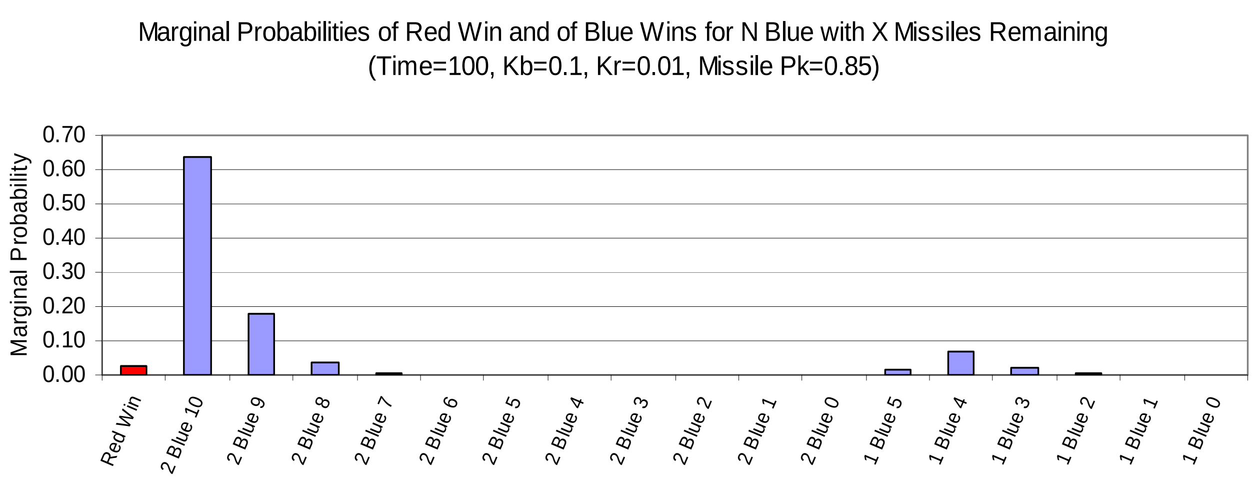
Listing 5. Model File Segments for 2 vs. 2 Engagement with Blue Missile Tracking

```
LIST = 3;
NBLUE = 2;
NRED = 2;
K_RATE_BLUE = 0.1;
K_RATE_RED = 0.01;
PK = 0.85;
MISS_RATE = (1-PK)/PK*K_RATE_BLUE;
MIN_B = 0;
MIN_R = 0;


1(* 0,0,0,0,0,0,2,2 *), 2(* 0,0,0,0,0,1,1,1 *) = 2*K_RATE_BLUE;
1(* 0,0,0,0,0,0,2,2 *), 3(* 0,0,0,0,0,1,1,2 *) = 2*MISS_RATE;
1(* 0,0,0,0,0,0,2,2 *), 4(* 0,0,0,0,0,0,1,2 *) = (2/(0+0+0+0+0+0+2))
*2*K_RATE_RED;
2(* 0,0,0,0,0,1,1,1 *), 5(* 0,0,0,0,0,2,0,0 *) = 1*K_RATE_BLUE;
2(* 0,0,0,0,0,1,1,1 *), 6(* 0,0,0,0,0,2,0,1 *) = 1*MISS_RATE;
2(* 0,0,0,0,0,1,1,1 *), 7(* 0,0,0,0,0,1,0,1 *) = (1/(0+0+0+0+0+1+1))
*1*K_RATE_RED;
2(* 0,0,0,0,0,1,1,1 *), 8(* 0,0,0,0,1,0,1,0 *) = 1*K_RATE_BLUE;
2(* 0,0,0,0,0,1,1,1 *), 9(* 0,0,0,0,1,0,1,1 *) = 1*MISS_RATE;
2(* 0,0,0,0,0,1,1,1 *), 10(* 0,0,0,0,0,0,1,1 *) = (1/(0+0+0+0+0+1+1))
```

```
*1*K_RATE_RED;
3(* 0,0,0,0,0,1,1,2 *), 6(* 0,0,0,0,0,2,0,1 *) = 1*K_RATE_BLUE;
3(* 0,0,0,0,0,1,1,2 *), 11(* 0,0,0,0,0,2,0,2 *) = 1*MISS_RATE;
3(* 0,0,0,0,0,1,1,2 *), 12(* 0,0,0,0,0,1,0,2 *) = (1/(0+0+0+0+0+1+1))
*2*K_RATE_RED;
3(* 0,0,0,0,0,1,1,2 *), 9(* 0,0,0,0,1,0,1,1 *) = 1*K_RATE_BLUE;
3(* 0,0,0,0,0,1,1,2 *), 13(* 0,0,0,0,1,0,1,2 *) = 1*MISS_RATE;
3(* 0,0,0,0,0,1,1,2 *), 4(* 0,0,0,0,0,0,1,2 *) = (1/(0+0+0+0+0+1+1))
                                        .
                                        .
                                        .
97(* 1,0,1,0,0,0,0,2 *), 99(* 1,1,0,0,0,0,0,1 *) = 1*K_RATE_BLUE;
97(* 1,0,1,0,0,0,0,2 *), 100(* 1,1,0,0,0,0,0,2 *) = 1*MISS_RATE;
97(* 1,0,1,0,0,0,0,2 *), 76(* 1,0,0,0,0,0,0,2 *) = (1/(1+0+1+0+0+0+0))
*2*K_RATE_RED;
99(* 1,1,0,0,0,0,0,1 *), 101(* 2,0,0,0,0,0,0,0 *) = 1*K_RATE_BLUE;
99(* 1,1,0,0,0,0,0,1 *), 102(* 2,0,0,0,0,0,0,1 *) = 1*MISS_RATE;
99(* 1,1,0,0,0,0,0,1 *), 72(* 1,0,0,0,0,0,0,1 *) = (1/(1+1+0+0+0+0+0))
*1*K_RATE_RED;
100(* 1,1,0,0,0,0,0,2 *), 102(* 2,0,0,0,0,0,0,1 *) = 1*K_RATE_BLUE;
100(* 1,1,0,0,0,0,0,2 *), 103(* 2,0,0,0,0,0,0,2 *) = 1*MISS_RATE;
100(* 1,1,0,0,0,0,0,2 *), 76(* 1,0,0,0,0,0,0,2 *) = (1/(1+1+0+0+0+0+0))
*2*K_RATE_RED;

(* NUMBER OF STATES IN MODEL = 103 *)
(* NUMBER OF TRANSITIONS IN MODEL = 285 *)
```

At 100 time units, the engagement is 0.99994 complete; the sums of the absorbing missile tracking states agree within four significant figures with the sums of the absorbing states for the simple 2 vs. 2 example. Figure B-5 combines the probabilities for the absorbing states to show the marginal probabilities of Red victory and the marginal probabilities of Blue victory with specific numbers of aircraft and missiles after a 2 vs. 2 engagement of 100 time units.

Figure B-5. 2 vs. 2 Engagement Aircraft and Missile Configuration Probabilities



Example 3—4 vs. 4+4+4 Sequential Engagement

The standard SLAACM campaign scenario has 4 Blue defenders engaging 4 Red long-range escorts (LEs), 4 close escorts (CEs), and 4 bombers sequentially. This sequential engagement can be modeled using ASSIST by imposing on the TRANTO statements the conditions that Blue cannot engage the close escorts or bombers until all the LEs are dead and cannot engage the bombers until all the CEs are also dead. We include individual kill rates for the three pairs of combatants. Listing 6 is the ASSIST code for the sequential engagement. The resulting model has 38 states and 48 transitions when the Blue breakpoint is 2 losses (min_b = 2 in the listing), and has 64 states and 96 transitions when Blue fights to annihilation (min_b = 0). Figure B-6 shows the marginal probabilities for the outcomes for the 2 Blue breakpoint, and Figure B-7 shows the corresponding expected number of surviving aircraft types.

Listing 6. 4 Blue Sequentially Engaging 4 Red LE, 4 Red CE, and 4 Red Bombers

```
(* ASSIST model for Fighter Combat *)

(* This is multistage combat where blue aircraft sequentially engage*)
(* three classes of red aircraft *)
(* This version does not include missiles *)

LIST = 3; (* 1 lists accumulated death state probabilities only *)
(* 2 lists all state probabilities *)
(* 3 lists transitions and all state probabilities *)
ONEDEATH OFF; (* OFF enumerates all absorbing (Death) states *)
(* comment out to consolidate absorbing states *)

(*”Time = 5 to 100 By 5;”*)
(* the time is set here - no matter what the STEM GUI shows *)
Time = 1000;

(* equipment list *)
nblue = 4; (* blue fighters *)
nred1 = 4; (* red fighters - lead escorts *)
nred2 = 4; (* red fighters - close escorts *)
nred3 = 4; (* red bombers or fighter bombers *)

min_b = 2;
min_r1 = 0;
min_r2 = 0;
min_r3 = 0;

(* kill probabilities *)
(* Note: the 1st 12 characters of a variable must be unique *)
blue_red1_k_rate = 0.1; (* 0.1 *)
blue_red2_k_rate = 0.1; (* 0.2 *)
blue_red3_k_rate = 0.1; (* 0.4 *)
k_rate_red1 = 0.02; (* 0.05 *)
k_rate_red2 = 0.01; (* 0.01 *)
k_rate_red3 = 0.005; (* 0.005 *)

SPACE = (blue: 0.nblue, red1: 0.nred1, red2: 0.nred2, red3: 0.nred3);
START = (nblue, nred1, nred2, nred3);

DEATHIF (blue = min_b) AND ((red1>0) OR (red2>0) OR (red3>0));
DEATHIF (red1 = min_r1) AND (red2 = min_r2) AND (red3 = min_r3) AND (blue>0);

(* Phase 1 combat *)
(* Blue kills *)
```

```
IF (blue > min_b) and (red1 > min_r1) TRANTO red1 = red1-1 BY
blue*blue_red1_k_rate;
(* Red kills *)
IF (red1 > min_r1 ) and (blue>min_b) TRANTO blue = blue-1 BY red1*k_rate_red1;

(* Phase 2 combat *)
(* Blue kills *)
IF (blue > min_b) and (red1=min_r1) and (red2 > min_r2) TRANTO red2 = red2-1
BY blue*blue_red2_k_rate;
(* Red kills *)
IF (red2 > min_r2 ) and (red1=min_r1) and (blue>min_b) TRANTO blue = blue-1 BY
red2*k_rate_red2;

(* Phase 3 combat *)
(* Blue kills *)
IF (blue > min_b) and (red3 > min_r3) TRANTO red3 = red3-1 BY
blue*blue_red3_k_rate;
(* Red kills *)
IF (red3 > min_r3 ) and (red1=min_r1) and (red2=min_r2) and (blue>min_b)
TRANTO blue = blue-1 BY red3*k_rate_red3;
```

Figure B-6. Sequential Engagement Marginal Probabilities

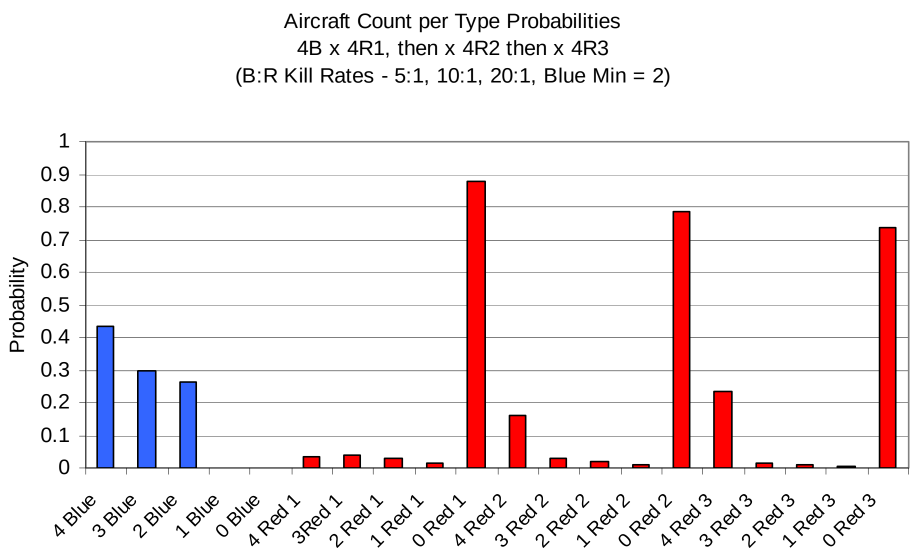
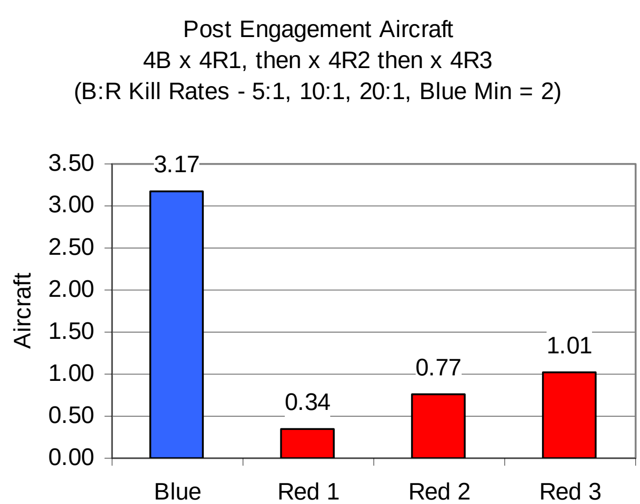


Figure B-7. Sequential Engagement Surviving Aircraft



Example 4—4 vs. 4+4+4 Sequential Engagement with Missiles

The last example is the standard sequential engagement with missiles on the Blue aircraft. Listing 7 is the ASSIST code for this case. This listing generates a model having 3,695 states and 21,729 transitions for the 2 Blue breakpoint case and 3,823 states and 23,454 transitions for the Blue fight to annihilation case. Figure B-8 shows the marginal probabilities of missile counts combined with information on how many aircraft are carrying the missiles for the case in which Blue breaks after two losses. Figure B-9 shows the remaining aircraft count by type. Table B-1 contains the summary results for both the 2 Blue breakpoint and Blue fight to annihilation cases.

Listing 7. 4 Blue vs. 4 Red LE, 4 Red CE, and 4 Bombers Sequentially with Blue Missiles

```
(* ASSIST model for *)
(* This version includes sequential combat with 3 types of red fighters *)
(* and includes Blue missiles *)
(* Fighter Combat using Dave Lee’s construct for blue missile counting*)
(* 7/20/05 This version includes missiles for blues *)
(* 7/20/05 THIS VERSION INCLUDES 6 BLUE MISSILES
(* THIS VERSION CAN LIMIT STATES BASED ON A MINIMUM NUMBER OF COMBATANTS *)
(* The kill rate is for one-on-one and is multiplied for N killers *)
(* Assume one missile per kill even with N killers *)
(* Therefore, the kill rate includes one hit and all misses *)
(* Find miss rate based on ratio of Pmiss:Pk *)

LIST = 3; (* 1 lists accumulated death state probabilities only *)
(* 2 lists all state probabilities *)
(* 3 lists transitions and all state probabilities *)
ONEDEATH OFF; (* OFF enumerates all absorbing (Death) states *)
(* comment out to consolidate absorbing states *)

(*”Time=10 TO+ 100 BY 10;”*)
Time = 1000;

(* equipment list *)
nblue = 4; (* blue fighters *)
nred1 = 4; (* red fighters *)
nred2 = 4;
nred3 = 4;


(* missile equipage *)
b_msls = 6; (* blue missiles *)

(* the first 12 characters in a variable name are significant and must be
unique *)

(* kill rates *)
red1_k_rate_blue = 0.1; (* B:R1 = 5:1 *)
red2_k_rate_blue = 0.1; (* B:R2 = 10:1 *)
red3_k_rate_blue = 0.1; (* B:R3 = 20:1 *)
k_rate_red1 = 0.02;
k_rate_red2 = 0.01;
k_rate_red3 = 0.005;


(* blue missiles miss rates *)
bpk = 0.85; (* missile kill probability *)
red1_bmiss_rate = (1-bpk)/bpk*red1_k_rate_blue; (* missile miss rate against
red1 *)
red2_bmiss_rate = (1-bpk)/bpk*red2_k_rate_blue; (* missile miss rate against
red2 *)
```

```
red3_bmiss_rate = (1-bpk)/bpk*red3_k_rate_blue; (* missile miss rate against
red3 *)

(* minimum number of combatants for state pruning *)
min_b = 2; (* minimum number of blue aircraft *)
min_r1 = 0; (* minimum number of red1 aircraft *)
min_r2 = 0;
min_r3 = 0;

SPACE = (b:array[0.b_msls]of 0.nblue, red1: 0.nred1, red2: 0.nred2, red3:
0.nred3);

(* states 0 to max missiles *)
(* THIS STATEMENT MUST BE CHANGED WHEN THE MISSILE COUNT IS CHANGED! *)
START = (0,0,0,0,0,0,nblue,nred1,nred2,nred3);

DEATHIF (sum(b)<= min_b) OR (b[0]=4);
DEATHIF (red1 = min_r1) AND (red2 = min_r2) AND (red3 = min_r3);

(* state transitions and kills *)
(* advanced escorts *)
FOR I IN [1.b_msls]
IF (b[I]>0) AND (sum(b)>min_b) AND (red1>min_r1) THEN
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1, red1 =red1-1 BY b[I]*red1_k_rate_blue;
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1 BY b[I]*red1_bmiss_rate;
TRANTO b[I]=b[I]-1 BY (b[I]/sum(b))*red1*k_rate_red1;
ENDIF;
ENDFOR;
(* no missile case against advanced escorts *)
IF (b[0]>0) AND (sum(b)>min_b) AND (red1 > min_r1) THEN
(* TRANTO red1=red1-1 BY b[0]*red1_k_rate_blue; *) (* NO BLUE KILLS WITHOUT
MISSILES *)
(* TRANTO b[0]=b[0]-1 BY (b[0]/sum(b))*red1*k_rate_red1; *) (* BLUE ESCAPES IF
NO MISSILES *)
ENDIF;

(* close escorts *)
IF red1 = min_r1 THEN
FOR I IN [1.b_msls]
IF (b[I]>0) AND (sum(b)>min_b) AND (red2 > min_r2) THEN
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1, red2 =red2-1 BY b[I]*red2_k_rate_blue;
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1 BY b[I]*red2_bmiss_rate;
TRANTO b[I]=b[I]-1 BY (b[I]/sum(b))*red2*k_rate_red2;
ENDIF;
ENDFOR;
(* no missile case against close escorts *)
IF (b[0]>0) AND (sum(b)>min_b) AND (red2 > min_r2) THEN
(* TRANTO red2=red2-1 BY b[0]*red2_k_rate_blue; *) (* NO BLUE KILLS WITHOUT
MISSILES *)
(* TRANTO b[0]=b[0]-1 BY (b[0]/sum(b))*red2*k_rate_red2; *) (* BLUE ESCAPES IF
NO MISSILES *)
ENDIF;
ENDIF;

(* bombers *)
IF (red2 = min_r2) AND (red1 = min_r1) THEN
FOR I IN [1.b_msls]
IF (b[I]>0) AND (sum(b)>min_b) AND (red3 > min_r3) THEN
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1, red3 =red3-1 BY b[I]*red3_k_rate_blue;
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1 BY b[I]*red3_bmiss_rate;
TRANTO b[I]=b[I]-1 BY (b[I]/sum(b))*red3*k_rate_red3;
ENDIF;
ENDFOR;
(* no missiles case against bombers *)
IF (b[0]>0) AND (sum(b)>min_b) AND (red3 > min_r3)THEN
(* TRANTO red3=red3-1 BY b[0]*red3_k_rate_blue; *) (* NO BLUE KILLS WITHOUT
MISSILES *)
(* TRANTO b[0]=b[0]-1 BY (b[0]/sum(b))*red3*k_rate_red3; *) (* BLUES ESCAPES
IF NO MISSILES *)
ENDIF;
ENDIF;
```

```
(* FOR NO MISSILE CASES: *)
(* use only red rates if blue has no missiles left and cannot escape *)
(*—- comment out all statements if blue can escape *)
(*—- include blue kill statements to continue combat without missiles *)
(* possibly add rates above and TRANTO statments here for Blue guns *)
```

Figure B-8. Sequential Engagement Missile Marginal Probabilities with Carrier Aircraft Information

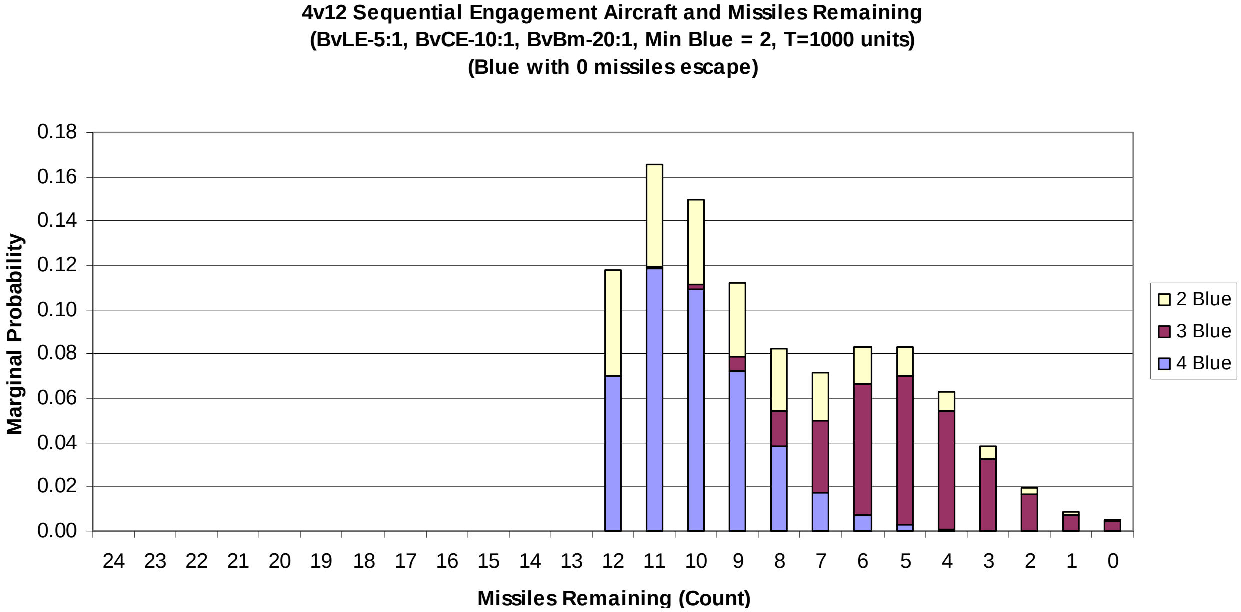


Figure B-9. Sequential Engagement with Missiles, Remaining Aircraft

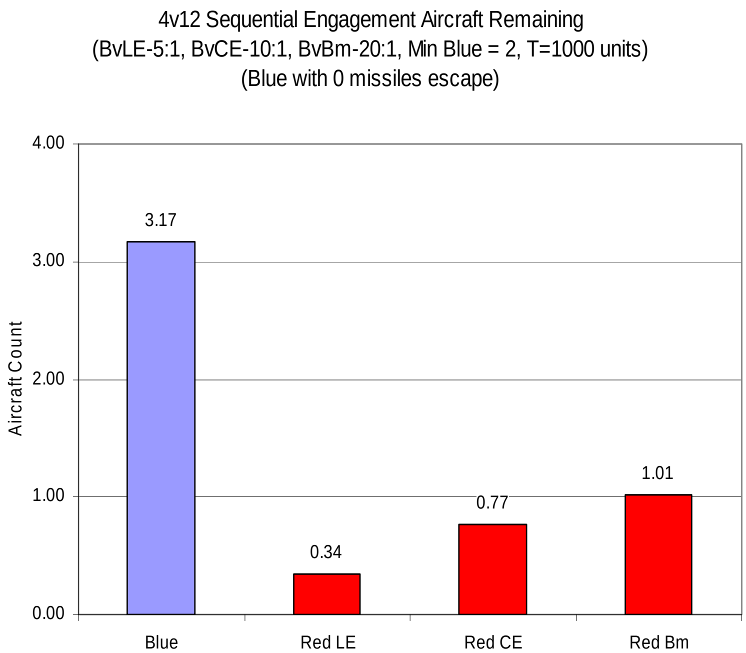


Table B-1. Summary Engagement Results

Scenario	Blue remaining	Red remaining	Missiles used
2 Blue breakpoint	3.2	2.1	16
Blue fight to annihilation	3.0	0.8	18

SUMMARY

In this appendix, we have shown how the NASA-developed Markov tools ASSIST and STEM can be used for analysis of increasingly complex air combat engagements. We have shown through examples how ASSIST is particularly useful for supporting parametric analysis involving large state-space problems. The NASA tools have proved useful for standalone analyses, for prototyping of SLAACM engagements, and for independent confirmation of SLAAM calculations.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
01-2009	Final	9/28/07 – 1/30/09

4. TITLE AND SUBTITLE
Stochastic Lanchester Air-to-Air Campaign Model

Model Description and Users Guides - 2009

5a. CONTRACT NUMBER
W91QUZ-07-F-0076

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Hemm, Robert V.; Author
Lee, David A.; Author
Eckhause, Jeremy M.; Author
Dukovich, John A.; Author

5d. PROJECT NUMBER

5e. TASK NUMBER
PA702

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

LMI
2000 Corporate Ridge
McLean, VA 22102-7805

8. PERFORMING ORGANIZATION REPORT NUMBER
LMI-PA702T1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
OSD/Program Analysis and Evaluation - TACAIR
Mr. Frank Lewis
Pentagon, Room 2C281
Washington, DC 20301-1800

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
A Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The report describes latest version of the Stochastic Lanchester Air-to-Air Campaign Model (SLAACM). The current version adds several attack scenarios to the basic air defense scenarios included in past versions. The attack scenarios include provisions for conventional and low observable aircraft, suppression of enemy air defense (SEAD) aircraft, and surface-to-air missile engagements. Other new features include options for bombers to escape or attempt to complete missions, user-selectable campaign objective function parameters, and enhanced displays of data. The body of the report includes full mathematical description of the model, and users guides for the "classic" air defense SLAACM and the new "attack" SLAACM. Two appendices describe the unique, LMI engagement analysis algorithm and the construction and analysis of large state-space engagements using NASA-developed Markov models.

15. SUBJECT TERMS
Multiple air combat, engagement model, air-to-air combat, campaign model, Lanchester model

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Nancy E. Handy
a. REPORT	b. ABSTRACT	c. THIS PAGE	Unclassified Unlimited	133	19b. TELEPHONE NUMBER *(include area code)*
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED			703-917-7249

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18