Generating Hypotheses on Fighting the Global War on Terrorism

Associate Professor Thomas W. Lucas
Naval Postgraduate School

Professor Susan M. Sanchez
Naval Postgraduate School

Lieutenant Colonel Thomas M. Cioppa
U.S. Army Training and Doctrine Command Analysis Center

Lieutenant Arif I. Ipekci
Turkish Army

INTRODUCTION

As the calendar turned to 1992, the conditions that had dominated military planning for almost half a century came to an end with the dissolution of the Soviet Union. In anticipation of a world that required less military manpower, the United States cut its active duty forces by about a third in the decade that followed [1]. This was often referred to as the "peace dividend." Unfortunately, while the nature of the threat changed, the world arguably became a more dangerous place. In fact, Beckett [2] states that: "Between 1990 and 1996 there were at least 98 [significant] conflicts inflicting 5.5 million deaths, but only seven of these were waged between recognized states."

The challenges facing our armed forces are fundamentally different than they were just over a decade ago. Consequently, our forces are undertaking an ambitious effort to fundamentally change. As Secretary of Defense Rumsfeld [3] states: "What's taking place in the conflict [Afghanistan], in the global war on terrorism, and the distinctively new threats we're facing, [provides] the impetus to transformation." Similarly, we need to transform an analysis infrastructure built to analyze a well studied and stable situation. Specifically, we need agile tools and analysis methods that allow us to quickly gain and effectively communicate insights into dynamic, asymmetric situations. In particular, to better assist senior decision-makers in structuring, equipping, and employing military forces to face the new threat we need to understand more about the intangible human elements of combat (leadership, morale, unit cohesiveness, etc.) in medium and low intensity conflicts involving adaptive adversaries. Towards that end, the Marine Warfighting Laboratory's Project Albert seeks to exploit the advances in computing power and new technologies to "provide quantitative answers…to important questions facing military decision-makers," Brandstein [4].

Under Project Albert's guidance, many diverse organizations have built a series of relatively simple models, along with data farming and visualization environments in which they can be explored. These models, by design, are fast-running, flexible, and easy to use. They strive to include only that detail which is absolutely necessary to capture the "essence" of the problem. Furthermore, these models are typically used in an exploratory manner. That is, the models assist us in reasoning about extremely complex systems and processes by helping generate hypotheses or assessing the consequences of assumptions.

Most of Project Albert's models are agent-based simulations. While Project Albert's simulations are small when compared to traditional Department of Defense (DoD) models, they still contain scores of variables an analyst may desire to explore. To do so efficiently requires *experimental designs* (the specification of the input variables) that allow us to efficiently sample from the vast number of potentially interesting input combinations. Furthermore, techniques that help us uncover relationships in the output data are also vital for effective exploration. In this paper, we highlight two methods that we have found particularly helpful in a series of explorations on a variety of models and scenarios—see Lucas et al. [5] and Sanchez et al. [6]. They are applied in a study of guerrilla combat involving a skirmish that author Ipekci experienced. The first method, dealing with generating inputs, is Latin hypercube designs (see [7]). The second method is classification and regression trees (CART)—which are good at uncovering relationships in large data sets (see [8]).

The outline of this paper is as follows. The next section describes the guerilla infiltration scenario we investigate. This is followed by sections that describe the model (MANA) that we use and the experimental design (a specially constructed Latin hypercube). The subsequent section summarizes the results with CART and multiple additive regression tree (MART) models. A concluding section discusses the main findings.

THE SKIRMISH

One of the most prominent examples of guerrilla forces fighting against a conventional force is the recent 15-year conflict between Turkey and the Kurdistan Workers Party (PKK). The Marxist-Leninist PKK was formed in 1974, with the goal of establishing a Kurdish state in southeastern Turkey. The PKK is one of 34 organizations on the U.S. State Department's list of designated foreign terrorist organizations [9]. In Turkey's conflict with the PKK, approximately 100,000 Turkish soldiers fought continuously against a PKK force of about 10,000. The conflict has claimed more than 30,000 lives. While the majority of casualties have been civilians, approximately 4,000 Turkish soldiers have been killed. Author Ipekci served in this conflict as a tank platoon commander in the southeast of Turkey. His experiences motivated this case study.

In September 1999, Lieutenant Ipekci was ordered to move his platoon in order to secure a hilltop along Turkey's border with Iraq. The platoon was composed of two tanks, two armored combat vehicles, and 11 infantrymen. Their mission was to take up a position on the hilltop to protect the area against terrorist activities and interdict forces seeking to enter Turkey. Given the hilltop's strategic value they knew that they were a desirable target for PKK forces.

A few weeks later, before dawn one morning, 11 terrorists equipped with light infantry weapons attacked. The attack began when a two-man reconnaissance team initiated heavy fire upon the platoon in an attempt to distract their attention. The remaining nine attackers split into two squads (of size four and five) and attempted to infiltrate the platoon's position from two directions shortly after the firing commenced. The skirmish lasted for almost half an hour. Just before daybreak, after losing four combatants, while inflicting only minimal losses on the platoon (two injured soldiers and minor equipment damage), the attackers withdrew.

This type of engagement increasingly occurs around the globe in situations where lightly armed guerrillas use speed and surprise to battle superior conventional forces. As such, we use it as a vehicle to examine how things might have changed if Ipekci's platoon was deployed differently, there were more attackers, they were more capable (i.e., had better weapons, combat effectiveness, unit cohesion, or aggression), and other questions. In addition, any real engagement is a single realization of an event that might have substantial variability associated with it. That is, in an otherwise identical scenario, events may unfold quite differently due simply to random chance. Therefore, when determining the lessons learned from historical battles it is important to understand the range of possible outcomes that could have occurred (or could happen in similar situations).

SIMULATING THE ENGAGEMENT IN MANA

The skirmish described in the previous section was replicated in the agent-based simulation Map Aware Non-uniform Automata (MANA), see Figure 1. As an agent-based simulation, the agents (software objects representing infantry soldiers, platoon commanders, tanks, etc.) make decisions autonomously about where to move, whom to shoot at, etc. These agents are aware of, and interact with, their local environment through relatively simple internal decision rules. The rules determine an agent's "personalities," such as their desires to move toward or away from a destination, alive and injured friendly agents, and enemy agents. These traits are often used to model so-called intangibles, such as aggressiveness. Additionally, variables can be defined that affect group behavior—such as the difference in forces required for an agent in a unit to want to advance towards an enemy agent. An agent's physical characteristics include their abilities to sense, communicate, and engage other agents. See Lauren and Stephen [10] for a detailed description of MANA.

From among the available simulations, MANA was selected for the following reasons. MANA's user interface allows one to easily construct and visually assess new scenarios. Individual battles of this size take only a few seconds to simulate on a PC. This, combined with the fact that MANA is resident at super computing centers (specifically, the Maui High Performance Computing Center (MHPCC) and the Mitre Corporation in Woodbridge, Virginia), enables us to generate hundreds of thousands of simulated battles. Critical functional capabilities that MANA affords include the ability to influence agent-movement with way-points, event-driven personality changes (e.g., an agent's desire to move towards the enemy can be programmed to change if he is shot at), and an internal situational map that allows agents to have a memory of enemy contacts.

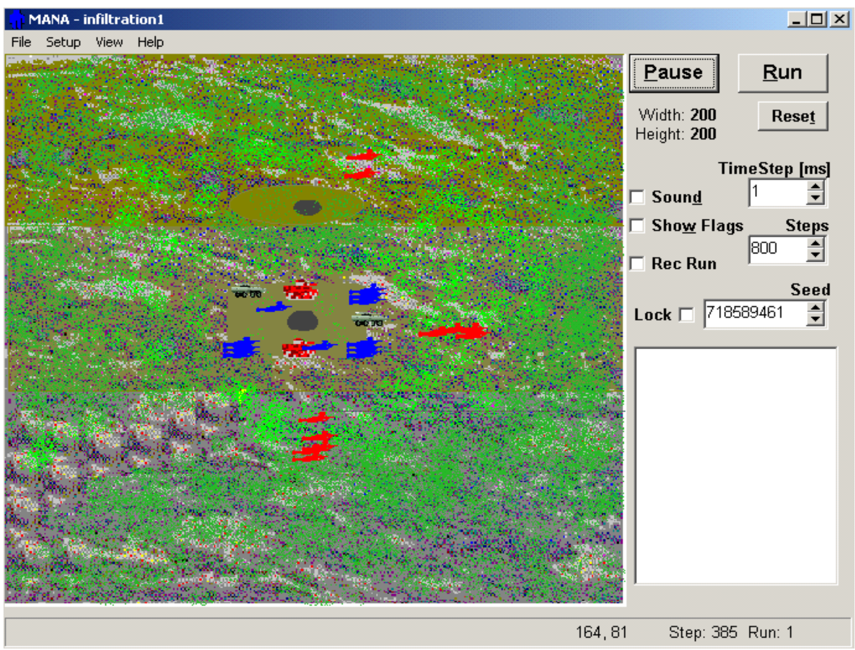

Figure 1: MANA Infiltration Scenario

It is important to emphasize that MANA models physical events (e.g., detections and engagements) with low resolution. Furthermore, as with most combat simulations, the differences between MANA's outputs and the real world have not been quantified. Thus, we do not want to put too much credibility in specific output values. Rather, we are using our exploration to glean insights whose veracity needs to be externally tested—perhaps with real battle data or warfighting experiments. Moreover, we see these results as one part of the *operational synthesis* process. That is, the process of combining the information obtained from a family of diverse analytical tools to provide the most compelling analyses—see Brandstein [4].

DESIGNING THE COMPUTATIONAL EXERIMENTS

This section describes the variables that are selected for exploration to address the issues discussed above. Manual trial and error on hundreds of MANA input variables indicated that we should explore more than a score of them. Of course, if we wish to be able to measure interactions (e.g., synergistic effects) among variables they need to be varied simultaneously. With so many variables to explore, a gridded design is infeasible. Thus, we chose to use a specially constructed Latin hypercube. Since we have found these designs particularly useful in high-dimensional explorations we detail their construction.

MANA Variables Selected for Exploration

Our exploration varies a total of 24 factors. Two are simple excursions from the baseline scenario. They are another Blue force (the defending tank platoon) disposition (to see if this affects results) and a second Red force (the attacking terrorists) attack plan (the new one utilizes three infiltration teams).

We explore the remaining 22 variables in all three scenarios (the base line and two excursions). Generally, they fall into the following classes. Nine of the input variables define the Red force's physical abilities (number, stealth, lethality, and mobility). Three parameters control individual Red agents' propensities to stay with their comrades (alive and injured) and move towards Blue agents—the latter represents their aggressiveness. Two additional Red force variables control which Blue targets (infantry or vehicles) the Red agents prefer to shoot at. Two more variables constrain the Red agents' group behavior. Specifically, they limit the size of Red groups and the difference in manpower (between Red and Blue) required for a Red agent to advance towards the Blue force. Finally, we vary a total of six parameters that define the Blue force's capabilities. These six relate to the Blue agents' stealth, sensing ability, and lethality. For a specific description of the variables and their levels see Ipekci [11].

Our measures of effectiveness are the proportion of Blue agents killed and the proportion of Red agents killed. Clearly, we want to minimize the former while maximizing the latter.

Latin Hypercube Designs

The designs we use are variants of the basic Latin hypercube design. We chose this family of designs because they are easy to construct in a broad range of situations and they generate results that provide flexibility in fitting models where there is considerable *a priori* uncertainty about the forms of the response surfaces—as in our example. Specifically, Latin hypercubes allow us to screen a large number of variables for significance, while simultaneously providing us with the ability to fit complex models (including non-parametric) on the most important variables.

In Latin hypercube sampling, the input variables are treated as random variables with known distribution functions. For each of k input variables, labeled X_i, for $i = 1, 2,\ldots, k$, "all portions of [X_i's] distribution [are] represented by input values" by dividing its range into "n strata of equal marginal probability $1/n$, and [sampling] once from [within] each strata" (McKay et al. [7]). For each X_i, the n sampled input values are assigned at random to the n cases—with all $n!$ possible permutations being equally likely. This determines the column in the design matrix for X_i and is done independently for each of the k input variables. A great strength of basic Latin hypercubes is that they are easy to generate for all k and n.

We illustrate the construction of a basic Latin hypercube when there are five input variables (i.e., $k = 5$) that we wish to explore over the region $[-1, 1]^5$—and must do so with only $n = 11$ samples. In our explorations, we sample each variable uniformly (i.e., their input distribution is a discrete uniform). Thus, all of the five variables will take each of the values (-1, -.8, -.6,…,1) exactly once in the 11-run design. The first input combination is selected by independently sampling once (with all input values being equally likely) from each variable—see the second row (corresponding to Run 1) in Table 1. The second input combination is obtained by independently sampling from the 10 remaining input values for each variable. This creates Run 2. We continue this, until Run 11, where we use the value that is left over for each input variable. The result is that each variable is uniformly sampled over its range. In this illustration we have scaled all of the input variables so that their domain is [-1, 1]. The same process applies to any rectangular region if the experimenter wants to uniformly sample each factor. Moreover, one need not limit themselves to uniform distributions. If there is a need to sample some regions more heavily than others, non-uniform distributions should be used.

Run	***X_1***	***X_2***	***X_3***	***X_4***	***X_5***
1	0.2	-0.8	-0.4	-0.2	-0.8
2	0	0.6	0.2	0	-0.6
3	-0.8	1	-0.8	0.6	1
4	-1	-1	0.4	-0.8	0.2
5	0.4	-0.4	0	0.2	0.8
6	0.6	0	0.8	-1	0.4
7	-0.4	0.4	-0.2	-0.6	-1
8	-0.6	-0.6	-1	0.8	-0.4
9	0.8	-0.2	1	0.4	0.6
10	1	0.8	-0.6	1	0
11	-0.2	0.2	0.6	-0.4	-0.2

Table 1: Example Latin Hypercube Design Matrix

Figure 2 shows the two-dimensional projections of all 10 pairs of input variables in our example. We see that by using uniform distributions the input points are scattered throughout the region to be explored. If we used a traditional two-level, full-factorial (also known as a gridded) design all of these points would be in the corners of the panels. A three-level, full-factorial design adds a point to the centers of the panels, as well as one in the middle of each of the panels' four boundaries. This design, with three levels for each factor, requires 3^5 (i.e., 243) runs. Clearly, Latin hypercubes give us much better space-filling than traditional gridded designs.

This Latin hypercube is just one of many possible designs that could have been generated—depending on the random sampling. In this example, one concern is correlations between the input variables. In fact, the correlation between X_3 and X_4 is −.6. Correlations between input variables can reduce the effectiveness of many analytic procedures—such as regression and CART. When the number of input combinations (n) is sufficiently large with respect to the number of factors (k), there will likely be relatively small correlations between columns in the design matrix, see [5] and [12]. When this is not the case, special Latin hypercubes (even ones that are orthogonal—i.e., with zero correlations between columns of the design matrix) may exist.

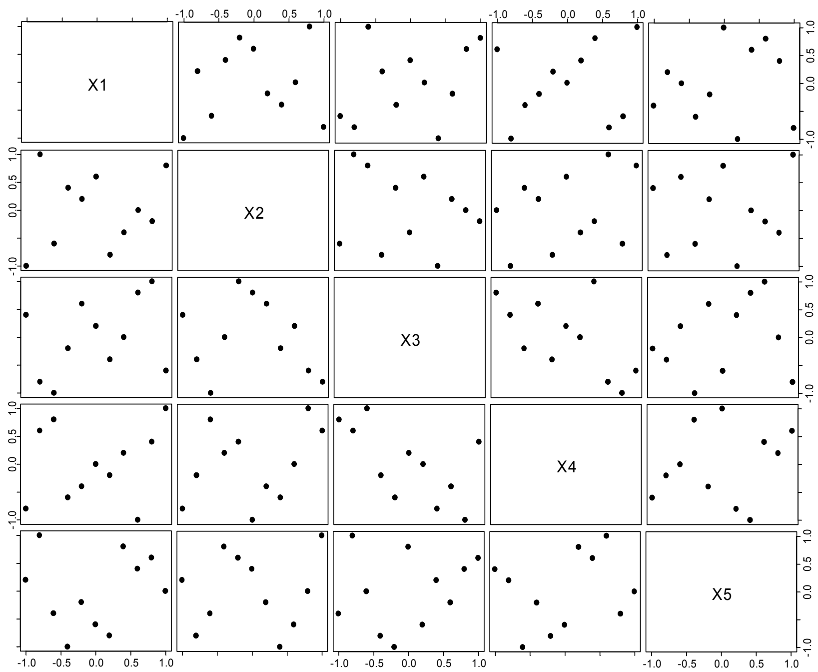

Figure 2: Scatter Plot of All Pairs of Input Variables From Table 1

In our analyses we use the recently developed nearly orthogonal Latin hypercubes of Cioppa [13]. These designs are nearly orthogonal—with a maximum pairwise correlation between columns of the design matrix of less than .03. Furthermore, they also tend to have much better space-filling properties than a randomly generated basic Latin hypercube. For each of the three scenarios, we simulate 513 input combinations of the 22 variables. For each of these, 100 replications are made. Thus, for each scenario 51,300 engagements are simulated.

EXPLORING THE DATA

Making sense of the outputs from hundreds of thousands of simulated battles is quite a challenge. For analysts, this is a good problem to have—much better than having too little data. One advantage of the nearly orthogonal Latin hypercubes we use is that they provide tremendous analytic flexibility. In fact, Ipekci [11] analyzed the data graphically using the software packages S-Plus, Clementine, Ggobi, and Netica. Analytic methods applied to the data range from the simple sign test to a variety of advanced statistical techniques, including cluster analysis, neural networks, regression trees, linear regression, and Bayesian networks. In this section, we summarize our findings on the baseline scenario that we obtain with regression trees. We focus on regression trees because we have found them particularly valuable in finding structure in large simulation output data sets and believe that tree models, as a whole, are underutilized by military operations research analysts. Furthermore, the results are readily interpretable.

Red Killed

This subsection looks at the proportion of Red killed as a function of the 22 factors discussed in the baseline scenario using tree models. Regression trees are hierarchical tree models that show structure in the data by sequentially partitioning it into homogeneous subsets through a series of simple bifurcation rules. One reason trees are becoming popular in data exploration is that they automatically generate models without the necessity of the user specifying the basic form of the relationships between the predictors and the response—for example, a linear, nonlinear, or additive model. In addition, trees do not require distributional assumptions—thus, transformations are not needed. In fact, the results are invariant to monotone re-expressions of the predictors. Moreover, interactions between variables are naturally obtained as the tree is built. Furthermore, trees are robust to outlying data.

The best way to explain how to construct and interpret a regression tree is by example. We now do so with the regression tree fit to the proportion of Red killed data—see Figure 3. Initially, all of the observations start in a single group or "node." A measure of the heterogeneity of the node's responses (called *impurity*) is made. If the all of the responses are the same the impurity is zero. Our impurity measure (using S-plus [14]) is the sum of the squared residuals. We want to partition the data into a set of homogeneous nodes one split at a time. This is done by considering every possible split of the form "$X_i < a$" (where X_i, for $i = 1,2,\ldots,22$, are the independent variables and a is a real number). From among all of these, the split that gives the smallest sum of the impurities of the two child nodes is made. In this case, the data are split into two sets, one containing the observations such that "**Red.Stealth** ≤ 123.5" and the other containing the remaining observations, i.e., "**Red.Stealth** > 123.5".

As we go down the tree, each of the two child nodes are then candidates for splitting—until a stopping condition is met. Specifically, a given node will split if it contains enough observations (as determined by the user) and the split improves upon the tree's purity by a specified amount. For the "**Red.Stealth** > 123.5" node, the data is partitioned into sets by the "**Red.Movement** ≤ 28" and "**Red.Movement** > 28" rules. These are terminal nodes—i.e., they are not split. The model then estimates, for example, that **if {**"**Red.Stealth** > 123.5" and "**Red.Movement** > 28"**} then** the proportion of Red killed is .10450.

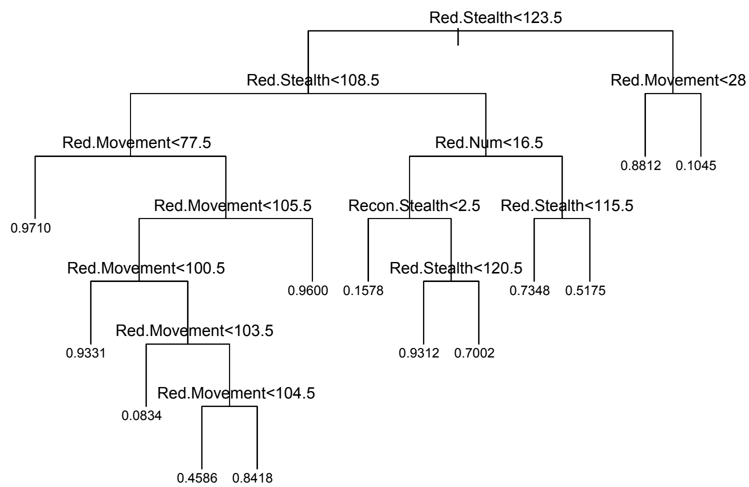

Figure 3: Regression Tree for the Proportion of Red Agents Killed

Our fitted tree model partitions the data into 13 sets—i.e., terminal nodes. Only four of the 22 variables appear in the tree. They are:

Red.Stealth: This parameter affects the probability that an agent can be seen—with higher values meaning the (Red infiltration) agent is less likely to be detected.

Red.Movement: This parameter affects the speed of (Red) agents—with greater numbers meaning faster agents.

Red.Num: This is the total number of Red agents in the infiltration teams.

Recon.Stealth: This is the stealth of the two Red reconnaissance agents.

The tree fit is quite good, with a *residual mean deviance* (which is the sum of the squared differences between the data and what the model would predict divided by the number of samples) of .006. Thus, knowledge of just these four variables is enough to accurately estimate the proportion of Red killed. Also, we obtain quite good discrimination in the response—as the mean losses in the 13 terminal nodes range from below .1 to above .9. Note: Several of the nodes split on **Red.Movement** near the value of 100. It turns out that this is a function of a discontinuity in MANA's movement algorithm—see Wolf [15]. Another note of interest is that none of the variables associated with the Blue force appears in the tree.

What is the importance of the 18 variables that do not appear in the tree? To answer this we use multiple additive regression trees (MART). MART models are designed to predict. They consist of a series regression trees—hence it is difficult to interpret them. However, Hastie et al. [16] provide a heuristic that quantifies the relative importance of the 22 predictor variables depending on how often they appear in the trees and how much they reduce the impurity. Figure 4 displays the relative importance values for Red killed on a scale of zero (of no importance) to 100 (the most important). We see that **Red.Stealth** and **Red.Movement** are the two most important predictors, followed by **Red.Num** and **Recon.Stealth**. Not until the sixth most important predictor do we get a factor associated with the Blue force—in this case the single shot probability of kill for Blue infantrymen. Another interesting point is that the Red personality parameters do not have much influence on the proportion of Red killed in this scenario.

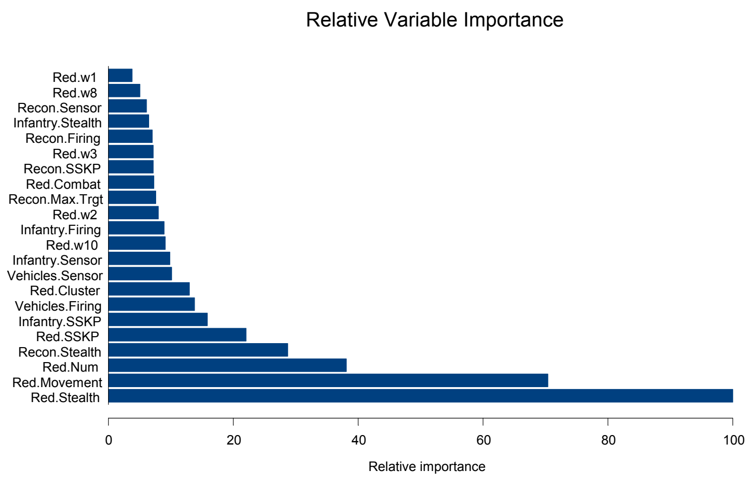

Figure 4: MART's Relative Importance of the Variables for the Proportion of Red Agents Killed

Blue Killed

Figures 5 and 6 display the regression tree and MART's relative importance values for the 22 predictors on the proportion of Blue killed. Here, seven predictors appear in the regression tree—which also has 13 terminal nodes. Once again, all of the tree's predictors are associated with the Red force. Three of the variables, **Red.Stealth**, **Red.Num**, and **Recon.Stealth** were in the proportion of Red killed tree. The four new variables are:

Recon.Firing: This factor controls the range at which Red reconnaissance agents can effectively engage Blue agents.

Recon.Sensor: This parameter defines the range that Red reconnaissance agents are able to detect Blue agents.

Red.SSKP: This variable affects the lethality of the Red infiltration agents.

Red.w1: This parameter determines the Red infiltration agent's propensity to move towards friendly agents (i.e., mass with other infiltration agents) when in contact with Blue agents.

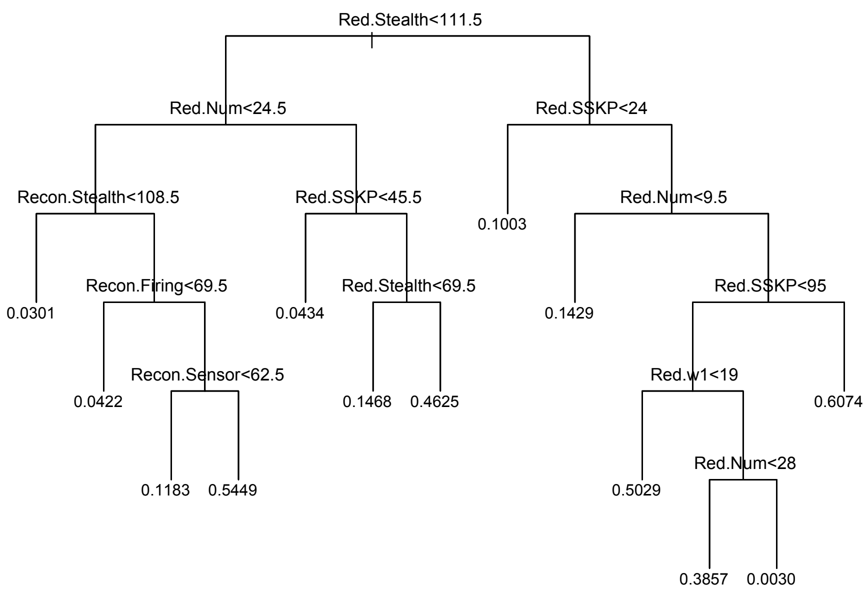

Figure 5: Regression Tree for the Proportion of Blue Agents Killed

Once again, the most important variable (i.e., first split variable) is **Red.Stealth**. The mean residual deviance is a little higher in this tree, with a value of .016, and the mean proportions of Blue killed range from .03012 to .60740. In this scenario, the Red force is particularly lethal when they are stealthy, there are a large number of them, the reconnaissance team has capable sensors, and their single shot probability of kill is high.

We see from MART's relative importance rankings (see Figure 6) that stealth, single shot probability of kill, and the number of Red forces are the most important variables. Once again, it is striking that the most important predictors (the top 13 in this case) are all associated with the Red force.

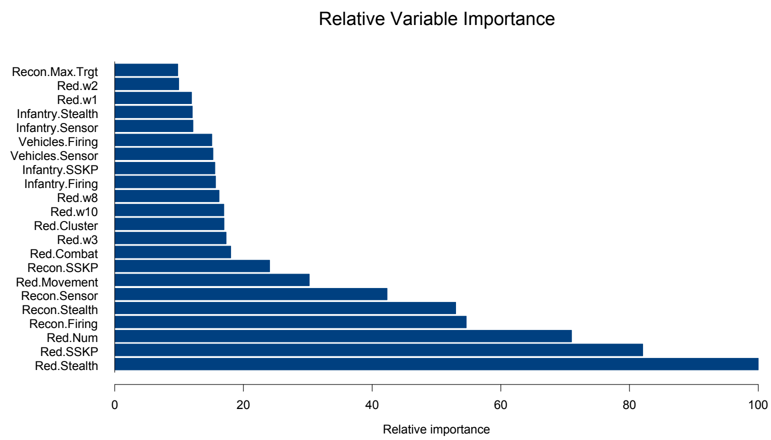

Figure 6: MART's Relative Importance of the Variables for the Proportion of Blue Agents Killed

CONCLUSIONS

Terrorist organizations almost always face conventional forces with vastly superior firepower. Hence, when engaging conventional forces terrorists usually resort to guerrilla tactics. In this exploration we use special Latin hypercubes to see how a large number of variables affect Blue and Red losses in a MANA scenario based on a guerrilla attack on conventional forces. Regression trees help us make sense from the output of over 150,000 simulated battles. They reveal that both Blue and Red losses depend almost solely on factors associated with the Red force—in particular the Red force's stealth and mobility.

Strategically, our findings suggest the importance of taking actions to inhibit terrorists' abilities to mass, train, and acquire weapons and sensors. The results also imply that improvements in the ability to detect terrorists may offer Blue more in both survivability and lethality than enhanced firepower. This might be accomplished through technical means (better sensors) or different force mixes (perhaps more reconnaissance elements). Of course, as with all force-on-force combat simulation generated hypotheses, their veracity, if possible, should be tested with other models, warfighting experiments, and/or examining real data.

REFERENCES

[1] United States Census Bureau, "Table No. 500: Department of Defense manpower: 1950 to 2000," *2001 Statistical Abstract of the United States*, Washington, D.C. Web page: http://www.census.gov/prod/2002pubs/01statab/stat-ab01.html. [Last accessed 13 June 2003]

[2] Beckett, I., *Encyclopedia of Guerrilla Warfare*, ABC-CLIO Inc., 1999.

[3] Rumsfeld, D., Department of Defense News Briefing, 20 August 2002. Web page: http://www.defenselink.mil/news/Aug2002/t08202002_t0820sd.html. [Last accessed 13 June 2003]

[4] Brandstein, A., "Operational Synthesis: Applying Science to Military Science," *PHALANX*, Vol. 32, No. 4, December 1999.

[5] Lucas, T.W., S.M. Sanchez, L. Brown, and W. Vinyard, "Better Designs for High-dimensional Explorations of Distillations," in *Maneuver Warfare Science 2002*, G. Horne and S. Johnson editors, USMC Project Albert, Quantico, VA, 17-45, 2002.

[6] Sanchez, S.M. and T.W. Lucas, "Exploring the World of Agent-based Simulations: Simple Models, Complex Analyses," *Proceedings of the Winter Simulation Conference*, J. Snowdon, J. Charnes, C-H Chen, and E. Yucesan editors, Institute of Electrical and Electronic Engineers, Piscataway, NJ, 2002.

[7] McKay, M.D., R.J. Beckman, and W.J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, **21**, 239-245, 1979.

[8] Venables, W.N. and B.D. Ripley, *Modern Applied Statistics with S-plus*, third edition, Springer-Verlag, New York, 1999.

[9] Office of Counterterrorism, U. S. Department of State, *Designated Foreign Terrorist Organizations*, 09 August 2002. Web page: http://usinfo.state.gov/topical/pol/terror/designated.htm [Last accessed 13 June 2003]

[10] Lauren, M. and R. Stephen, *Map Aware Non-Uniform Automata Version 1.0 Users Manual*, New Zealand's Defence Technology Agency, Devonport Naval Base, Auckland, New Zealand, 2001.

[11] Ipekci, A., "How Agent-based Models Can Be Utilized To Explore and Exploit Non-linearity and Intangibles Inherent in Guerrilla Warfare," Masters thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA, 2002.

[12] Owen, A., "Controlling Correlations in Latin Hypercube Samples," *Journal of the American Statistical Association*, **89**, 1517-1522, 1994.

[13] Cioppa, T.M., "Efficient Nearly Orthogonal and Space-filling Experimental Designs for High-dimensional Complex Models," Ph.D. Dissertation, Operations Research Department, Naval Postgraduate School, Monterey, CA, 2002.

[14] Insightful Corporation, *S-plus 6 for Windows Guide to Statistics, Vol. 1*, Seattle, WA, 2001.

[15] Wolf, E., forthcoming Master's thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA, 2003.

[16] Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer-Verlag, New York, 2001.