

A Simulation-optimization Approach to Air Warfare Planning

Douglas Popken

Systems View

9139 Roadrunner St.

Highlands Ranch, CO 80129-5758

dpopken@systemsview.com

Louis Cox

Systems View

9139 Roadrunner St.

Highlands Ranch, CO 80129-5758

lcoxdenver@aol.com

How can computer-aided planning systems deal with the complexities, uncertainties, and rapidly shifting information needed to support air warfare operational planning? This paper uses a hierarchical decomposition of decision-making, coupled to a predictive simulation model that estimates the probability distribution of the outcomes of candidate operational plans. The approach will generate, evaluate, and improve Blue plans while assuming that Red intelligently reallocates its forces, using stochastic optimization, to counter Blue's moves. Evaluation of each Blue plan is accomplished via a Stochastic Evaluator that draws multiple samples of potential outcomes and Red force levels for a given Blue force structure and combined target composition. The evaluation metric is the net discounted value from enemy targets hit. Linear programming heuristics and simulation generate Red's adaptively optimized responses, outcomes, and inferred relative marginal force values. The results of this two-player, multi-level simulation-optimization approach for operational planning and decision-making demonstrate that automated plan optimization and embedding of optimization algorithms into an operational planning cycle operating over a multi-period conflict can be made practical for current computers by hierarchical decomposition of planning and on-line decision and optimization problems into computationally practical tasks.

Keywords: Decision making, simulation optimization, complex systems control

1. Introduction

1.1 Related Efforts

For more than forty years, planning and optimization models for air combat have been proposed based on simplified stochastic games and differential games [1]. Karlin [2] discusses the application of dynamic programming, linear programming, and network flow analysis to the problem. Dresher [3] provides a number of interesting and still-relevant formulations on topics such as the "Tactical Air-War Game" and "Defense of Targets of Many Values." The advent of modern methods of simulation optimization and advances in machine learning and optimization heuristics, as well as improved computers, now suggests that such simplified models can be replaced with complex yet more realistic

models that incorporate many of the complexities, details, and uncertainties required to better support real-world conflicts.

Significant research, largely within the DARPA JFACC (Joint Force Air Component Commander) program [4], has investigated planning of military air operations for individual missions. Many of these efforts emphasized an applied optimal control-system based mathematical framework. Mukai, et al. [5] provide a differential game formulation for opposing air and ground units. Ordinary differential equations describe unit movement (on a 2-D grid) and attrition. The controls are the speed, direction, and engagement "intensity" of each unit in each time period. The value of the game is determined by a quadratic payoff function that tends to: 1) minimize distance between units and their designated target locations, and 2) maximize enemy losses for any given level of friendly losses. Cruz, et al. [6] formulate a similar problem where the unit state describes location, the number of "platforms," and the number of weapons

per platform. The controls are movement, salvo size, and choice of target. The objectives again are to maximize enemy losses while minimizing friendly losses. McEneaney and Ito [7] attempt to optimize both aircraft routes and engagement decisions made against hostile missile units and fixed strategic targets. The unit state describes location (for aircraft and mobile missiles) and health. The objective is to destroy strategic targets while each side tries to minimize its own losses and maximize its opponent's. The controls are determined via a two-level hierarchical optimization procedure. First, the aircraft routing through the hostile region is determined; next, the time ordering of aircraft engagements against missile sites is determined.

Each of the JFACC efforts above discretizes the decision space, creating a "curse of dimensionality" (see discussion below) for combinatorial optimization algorithms. Each also assumes perfect knowledge of the true system state at all times. Finally, some involve decisions that may best be left — at least for now — to humans. A more recent paper by McEneaney, et al. [8] refocus their earlier model on unmanned combat air vehicles (UCAVs) and lower levels of a decision hierarchy. A higher level decides strategic planning and resource allocation; a middle level determines aircraft routes, engagement assignments, and support activities; while the lowest level determines targeting and vehicle guidance. The paper also introduces the notion of having only partial information on the system state while recognizing the severe problem size limitations imposed by combinatorial explosion. Cassandras and Li [9] handle complexity in UCAV management by using a "receding horizon" approach. The main idea is to solve for the optimal reward extracted over a given time horizon, and then to periodically roll the time horizon forward, at which point the problem is re-solved. In addition to this "time decomposition" of the problem, further simplification is provided by integrating target assignment, vehicle routing, and real-time trajectory generation into a function of a single control variable: vehicle headings.

Cave and Busch [10] take a control-system approach to formulating the operational planning problem within an Aerospace Operations Center. They provide examples of how the system could work at the mission level for aircraft being routed to targets across a hostile battlefield. The system explicitly accounts for uncertainty, but modeling of Red counter-action tactics is limited. The existence of higher decision making levels is mentioned, but an integration framework is not discussed.

Surdu and Pooch [11] describe an agent-based simulation approach to real-time military planning. In their framework of "operationally-focused" simulations (OpSim) they define Operations Monitors (agents) whose function is to continually compare the internal

simulation models with the real world operations. When significant deviation occurs, agents launch one of several investigative tools used by decision makers to determine corrective Courses of Action (COAs). Agents are modeled as fuzzy rule bases, but the authors suggest that machine-learning frameworks such as neural nets might provide substantial improvements. A prototype was implemented demonstrating control of brigade and battalion-level ground combat operations.

A heuristic, hierarchical approach to wargaming is found in "Linguistic Geometry" [12]. The approach provides a conceptual framework for linking conflicts at variable levels of resolution and different geographic locations. Cause and effect are linked via heuristic, state-dependent trigger rules. COAs are selected from discrete pre-defined sets. Since outcome uncertainties are not quantified, the system is more appropriate for wargaming than analysis.

In contrast to these previous efforts, the work described here focuses on more realistic and comprehensive modeling of uncertainties for target distributions and engagement outcomes using a fully flexible discrete-event simulation model. Blue and Red optimize their plans using adaptive simulation-optimization heuristics that continually update plans (and interleave them with actions) to incorporate the most recent available information. Our approach is intended to produce high-quality solutions for Blue in practical amounts of CPU time while taking into account Red's intelligent responses.

1.2 Incorporating Uncertainty

Our research suggests that all military conflicts involve two major forms of uncertainty: imperfect knowledge of the current state of the battle, referred to by Clausewitz [13] as the "fog of war," due to limited communication, enemy deception, and incomplete sensor coverage; and incomplete predictive capability, consistent with Ancker's "stochastic attrition process" [14], due to equipment breakdowns, human errors, misreading of enemy goals and intentions, weather, and the chance outcomes of specific engagements.

An operational planning tool potentially can account for uncertainty by estimating a set of plausible outcomes and their likelihoods for a plan; by seeking recommended actions that are robust to current uncertainties; or by bounding the set of likely outcomes (e.g., via a "plausible worst case" outcome). However, bounding does not allow the decision maker to make preferred trade-offs between risk and reward and may be hard to define and implement, given the many possible outcome metrics, each with high uncertainties and complex interdependencies. We therefore focus on the first two approaches by generating an entire range of outcomes

and recommending actions based on the distribution of solution values. Determining a range of outcomes with their probabilities typically requires a sampling-based approach combined with repeated coarse-graining to aggregate statistically indistinguishable outcomes into a manageable number appropriate for the available computational resources. For example, particle filtering (PF) with adaptive importance sampling often is practical even for large-scale problems, but requires considerable care to implement efficiently and can lead to rapid accumulation of errors when the underlying dynamics are not correctly specified, even for relatively simple tasks such as deciding whether the situation being monitored has changed significantly [15]. Methods that propagate approximate maximum-likelihood estimates (MLEs) of distributions while using statistical large deviation bounds to approximate distributions around the point estimates also can be useful, but again only when the system is understood well enough to allow MLEs to be defined. This paper will therefore focus on sampling and coarse-graining approaches in which a discrete-event simulation model produces multiple sample realizations of potential outcomes and these are then used to identify approximately optimal contingency plans and current decisions. Rather than maintaining a particle swarm of high-probability hypotheses as in PF, our approach achieves computational tractability with a rolling plan that discards previous future contingency plans and generates new ones after each cycle of action and observation. Recommended actions are robust to future uncertainties, in that they are explicitly based on multiple scenarios modeling future uncertainties and adaptive responses by both Red and Blue. The actions recommended for implementation at any time are the first steps of multi-period plans that are constructed to maximize expected value (or utility) taking into account a wide range of future scenarios.

1.3 The Curse of Dimensionality

The curse of dimensionality occurs whenever one endeavors to evaluate multiple discrete strategies per period over any extended time period. Busch [16] formulated the discrete air warfare planning problem as a partially observable Markov Decision Process (POMDP) and noted that the approach is “intractable for any realistically sized problem.” To illustrate, suppose that for each time period, Red and Blue must select their next COA from a discrete set. Each decision pair then results in an outcome selected from a discrete set. Even if the decision and outcome sets are small, the time element in operational planning makes the total planning space unmanageably large. For example, when Blue and Red forces each have six alternative COAs available in each of 10 periods, and each Blue/Red decision pair in each

period has only two possible outcomes, then there already are $(36 \times 2)^{10} = 3.74 \times 10^{18}$ different decision/outcome paths. In some problem domains it may be possible to avoid the curse of dimensionality by sharply restricting the length of the planning horizon, e.g., to only two periods [6] for selecting movement, salvo size, and choice of target on a geographic grid. Huggins, et al. [17] reduced the number of decision/outcome paths evaluated by 1) assuming a single Red decision for a given Blue decision; 2) manual entry of a small set of plausible Blue decisions; and 3) using a short planning horizon. However, restricting Red to a single decision tends to underestimate the uncertainty of outcomes. Moreover, it often is undesirable to truncate the planning horizon to just a few periods, as a longer time frame is needed for the rewards to fully reflect near-term outcomes and to coordinate with elements outside of the system being controlled, e.g., ordering the correct types and amounts of logistical and support resources.

Our proposed solution to the curse of dimensionality lies not in truncating the planning horizon, but in collapsing the many discrete paths into a continuum of choices. Potential outcomes are then sampled from the continuum. This is the starting premise for our LP/stochastic simulation approach; see also Yost and Washburn [18], who applied a similar computational strategy to attack aircraft hitting targets in stages.

1.4 The Need for Model Abstraction

Models are needed to determine the outcomes(s) for a given state and Blue/Red decision pair. Detailed, entity-level combat models, in which individual movements and engagements are modeled explicitly, are precluded for reasons of run-time performance and complexity. Quick-running abstract models of complex situations are needed instead. Hillestad, et al. [19] described the need for aggregate, low-resolution combat models, providing as rationale: a typical lack of detailed historical data, the difficulty of replicating small unit decisions, a desire to better comprehend and explain phenomena, budget and schedule constraints, and repeatability considerations. Their conclusions are based on related work, including two-sided game theoretic allocations and apportionments of tactical air forces in the context of combat simulations (e.g., “TacSAGE” [20]).

Schlabach, Hayes, and Goldberg [21] implemented a system for generating many courses of action in the military planning domain using genetic algorithms. A coarse, low-fidelity simulation evaluated each member of a generation of potential COAs. Hill and Surdu [22], in describing their APSS (Anticipatory Planning and Support System), suggest that closed-form solutions to Lanchester equations would be an appropriate evaluator to quickly estimate expected losses. In later work with

the APSS, Huggins, et al. [17] went even further in advocating abstract concept level models. They found that plans focused too much on entity level locations and activities were too difficult to use, as there were too many differences between planned and predicted states (i.e., the plan was constantly overcome by events). Compelling arguments for using abstract models also have been made in the context of planning algorithms for intelligent machines [23].

1.5 Contributions from Decision Sciences

The algorithms developed and tested in this project build on the preceding ideas and extend them with several related advances in decision sciences. The main additional concepts incorporated into our integrated hierarchical planning approach are as follows.

- *Simulation-optimization.* Stochastic gradient (hill-climbing) techniques [24] can be used to simultaneously estimate local response surface shapes (i.e., payoff or reward functions for Blue's choices) from samples generated by probabilistic simulation and adjust Blue's decisions to maximize expected value, given the current estimated response surface. These methods have a long tradition, well developed in what is now called simulation-optimization [25,26], and reaching reach back to the Evolutionary Operations (EVOP), Kiefer-Wolfowitz, and Robbins-Munro iterative adjustment procedures for climbing unknown response surfaces. Our approach exploits simulation-optimization techniques to automatically improve Blue's plan, while adding the novel feature of embedding Red's best response strategy (by "optimizing out" Red's choices via a multi-period linear programming approximation).

- *Game theory and linear programming.* From classical game theory [3] we adopted the idea of letting Red make a best response to Blue's plan, thus encouraging Blue to identify a traditional minimax strategy (when one exists in pure strategies). In effect, this assumes that Red will have good intelligence and respond accordingly to Blue's plans. If this perspective turns out to be overly pessimistic, Blue's plan will be re-optimized immediately (in one period) to take advantage of Red's sub-optimal play. We exploited the idea of shadow prices and hierarchical (Dantzig-Wolfe) decomposition from large-scale and decentralized linear programming to coordinate among levels of the decision hierarchy and to help optimize out Red's planning decisions. In a departure from traditional game theory, solutions were required to be pure strategies, i.e., randomized plans were ruled out as unrealistic to implement, despite their potential theoretical advantages. Finally, we applied ideas of mechanism design from modern game theory,

to formulate the conflict as a bi-level hierarchical optimization problem, in which Blue chooses a plan assuming that Red will choose a best response to it. This avoids the need to find (probably unrealistic and computationally daunting) Bayesian Nash equilibria, and reduces the problem to one that can be solved by adaptive optimization methods.

- *Reinforcement learning and adaptive optimization.* In a real air conflict game, neither player necessarily knows the structure or data of the games — e.g., the payoff functions and the transition functions in Markov games [27]. Common-knowledge priors from which to compute theoretical Bayesian Nash equilibria cannot be justified. To model such realistically incomplete knowledge, we assumed that players only can sample from the distribution of outcomes for different pairs of Red and Blue decisions, via simulation. Thus, the players are treated as both learning about the payoff function and adaptively optimizing their decisions as the game evolves. Our models and algorithms therefore combine ideas from adaptive dynamic programming, especially multi-period rollout heuristics [28] for solving Red's multi-period optimization problem, with "learn as you go" ideas from reinforcement learning. Rollout heuristics take a simple decision rule (based on an LP model, in our case) and apply it several steps ahead to forecast returns over a finite look-ahead horizon. Then, they re-optimize the initial (simple) decision using the forecast returns as an approximate value function. Reinforcement learning estimates value functions for state-specific decisions from sample data [18] and adjusts the decision rules to improve expected values. Both sets of ideas play essential roles in the algorithms and heuristics that follow. We apply them directly to the conflict simulation, rather than to a simplified Markov game, to make the solutions as realistic as possible.

2. Hierarchical Planning Framework

In this effort, the planning process is organized as a hierarchy of decisions as follows:

1. The top-most level, "Force/Roles," allocates forces across combat roles (*Counter Air, Air Defense, Target Reduction, AAF Suppression, and Other*) for the remainder of the planning horizon. This is the broadest, longest-term decision level.
2. The second level, "Targets/Missions," assigns forces to specific targets and missions, both offensive and defensive, in a given planning period.
3. The third level, "Routes/Engagements," assigns specific routes and engagements to individual forces on an assigned mission.
4. The bottom level, "Flight Control," addresses real-time piloting, navigation, and fire control.

We define the five combat roles as follows:

- *Counter Air* — engages Air Defense forces assigned to defend opposing targets
- *Air Defense* — engages any opposing forces assigned to friendly targets
- *Target Reduction* — degrades opposing targets (non AAF) to extract value
- *AAF Suppression* — degrades opposing AAF forces/targets
- *Other* — forces not assigned to combat operations in a given planning period



Figure 1. Levels of operational planning for air warfare

3. Level 1 Algorithms

Level 1 allocates Blue’s forces to roles. We use the following roles for purposes of illustration, although the method is general: *Counter Air*, *Air Defense*, *Target Reduction*, *AAF Suppression*, and *Other*. Level 1 allocates a fraction of each type of Blue force to each role for each time period in the planning horizon. An allocation also can be interpreted as a plan or as a high level “Course of Action” (COA).

The following Blue plan improvement algorithm accomplishes this allocation:

Blue Plan Improvement Algorithm

INPUTS: An initial feasible plan for Blue; initial values for the parameters $(\omega, \alpha, \epsilon) =$ (horizon weight, discount factor, force reallocation proportion step size); $D_1 =$ the starting problem domain data (including probability distribution for Red’s forces, targets, etc.).

OUTPUTS: An array PB_{ijt} giving the fraction of Blue’s force of type i resources allocated to role j in period t , for each t in the planning horizon. This is a Level 1 plan for Blue.

1. **Initialize the horizon** weight ($\omega = .50$), the discount factor ($\alpha = .80$), and the force reallocation proportion ($\epsilon = .10$), to values between 0 and 1. Set the planning iteration index $z = 0$. Initialize using an initial feasible Blue plan. This is an array of non-negative allocation fractions, $PB_{ijt}(0)$, for each force i , role j , and time period t ($t=1,2,\dots,T$) subject to $PB_{ijt}(0) \geq 0, \sum_j PB_{ijt}(z) = 1, \forall$

(This initial plan may be generated randomly, via a heuristic, or imported from an external source.)
2. **Given $PB_{ijt}(z)$, simulate the probability distribution of optimized Red responses and outcomes** using ω and α along with the starting problem domain data (forces, targets, etc.), D_1 . This will provide a mean and a standard deviation, $\pi_{ijt}(z)$ and $s_{ijt}(z)$, of relative marginal value (i.e., shadow price) for each Blue force, role, and time period (more details below).
3. **Use $\pi_{ijt}(z)$ and $s_{ijt}(z)$ to modify Blue’s current plan to improve Blue’s expected objective function.** In general, we seek the reallocation of force where the net, statistically significant gain, is greatest.
4. **Adjust Blue’s allocations in the direction indicated**, by adding ϵ to the gaining role allocation and subtracting ϵ from the losing role allocation (if the losing role allocation is currently less than ϵ we use that proportion instead) to obtain $PB_{ijt}(z+1)$. Set $z = z + 1$.
5. **If no further statistically significant improvements are found, STOP, otherwise return to Step 2.**

In Step 1, default values of the input parameters are provided in parenthesis. The horizon weight, ω , and the discount factor, α , work in tandem. The horizon weight places a value on survival of forces beyond the T period planning horizon. Setting $\omega = 0$ would imply that forces have no value beyond T . As a result, the relative marginal resource values for forces would be skewed in favor of the *Target Reduction Roles*. A long-term perspective, $\omega = 1$, implies that survival is given high consideration. Forces are conserved at the expense of more immediate target value extraction. The discount factor, α , on the other hand, is not a control parameter, but an evaluation parameter. It acknowledges that, other things being held equal, shorter conflicts are preferred to longer ones. Target value extraction occurring early is more valued than the same target value extraction occurring later. The force reallocation parameter, ϵ , affects the speed of convergence and the precision of the solution. A larger value provides faster convergence, but we found that too large a value may cause convergence to suboptimality.

Other notes: The initial feasible Blue Plan could be determined from expert opinion or an initializing algorithm. It is necessary to introduce a “tabu list” [29]

of the most recent reallocations to the plan improvement search in Step 4 to prevent a reallocation from being reversed for some number of improving steps, thus preventing cycling. Without loss of generality, the Blue reallocation in Steps 3 and 4 could be performed with additional constraints on feasible allocation combinations. The constraints could represent logistic considerations, coordination with force elements outside of the model, or expert knowledge on the most effective force combinations. A separate top-level linear solver then would be required.

Step 3 of the plan improvement algorithm generates K sample values for each incumbent plan. Each individual “sample” outcome begins with an instance of the possible current Red force states. The current Red force state is drawn from a probability distribution that is assumed to be a result of the latest intelligence estimates. The sampling procedure provides a means to *probe* the space of all possible outcomes as an alternative to exhaustive enumeration. At Level 1, the outcome is generated according the following general algorithm:

Blue Plan Stochastic Evaluator (Level 1)

For $k = 1$ to K :

Set the estimated Red force levels by sampling from their probability distribution based on $D_1(k)$. Set $V_k(z)$ = estimated net Value to Blue of plan $PB_{ijt}(z)$ to 0.

For $t = 1$ to T

- a) Determine Red’s approximate best response in period t by optimizing over the next $T - t + 1$ time periods using $D_t(k)$ and the horizon weight, ω . Select the first period of the solution as Red’s response in period t , obtaining both Red’s role allocation, $PR_{ijt}(z)$, and Blue’s relative marginal resource or shadow price values π_{ijt}^k .
- b) Using $PB_{ijt}(z)$, $PR_{ijt}(z)$, and $D_t(k)$, create Level 2 plans for period t for both Blue and Red.
- c) Use the Level 2 Plans to determine target assignments.
- d) Use the target assignments to allocate forces, and run the stochastic simulator for one period.
- e) Set the Red and Blue force and target states to those found at the end of the one-period simulation and update the problem data: $D_t(k) \rightarrow D_{t+1}(k)$
- f) Set $V_k(z) = V_k(z) + \alpha^{t-1} V_{kt}(z)$ where $V_{kt}(z)$ is the observed net value to Blue extracted from targets during “Phase 2” (see next section) of the period t simulation.

END

END

$$\text{Compute } \bar{\pi}_{ijt}(z) = \frac{\sum_{k=1}^K \pi_{ijt}^k}{K} \text{ and } s_{ijt}^2(z) = \frac{\sum_{k=1}^K (\pi_{ijt}^k - \bar{\pi}_{ijt}(z))^2}{(K-1)},$$

the average and variance over K iterations for the returned by Step a.

The outputs of the algorithm are the $\bar{\pi}_{ijt}(z)$, $s_{ijt}(z)$, and the $V_k(z)$. Note that the discounted Blue net values, $V_k(z)$, are outputs that can be examined to view performance of the Blue Plan Improvement Algorithm or to compare alternative improvement algorithms (as discussed in Section 5.3). The approach used in the Stochastic Evaluator is illustrated in Figure 2 below. Each sample outcome begins with an instance of the possible current states. An optimization is performed to determine the allocation over the remainder of the planning horizon. The optimization captures long-term effects, however, only the first period optimal allocation is retained and fed into the single period simulation, while the rest is discarded. The short-term response is thereby guided by the long-term response. The short-term simulations capture uncertainty in that attrition occurs probabilistically. The end state of the simulation becomes a new state instance, and the process is repeated over the remainder of the planning horizon. The discounted sum of scores achieved over all periods becomes the total score for the original instance. The distribution of total scores is the full outcome for the current allocation. This approach avoids myopic allocations that would achieve high single period scores at the expense of long-term results. At the same time, it recognizes the limitation of multi-period look-aheads by retaining no more than one period of the optimized results and by explicit consideration of the many uncertainties involved. It reallocates forces only if the improvements are seen to be statistically significant over the planning horizon.

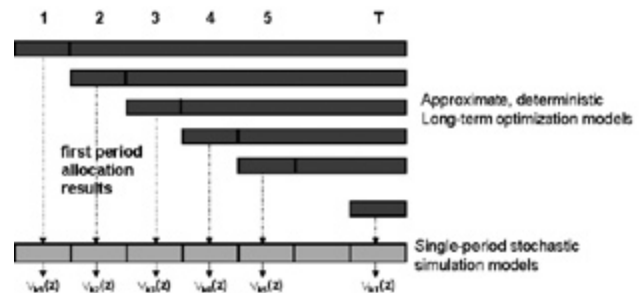


Figure 2. Optimization/simulation scheme to generate T-period outcome

The optimization model used in Step a is formulated as a nonlinear mathematical program (see Appendix) that maximizes Red target value extraction constrained by Blue's current force allocation. Force quantities are relaxed to real values. Because of the "max" constraints on Red attrition, the problem is not solved directly. Instead, a series of linear programming relaxations are solved to provide a good approximate solution.

To generate specific Level 2 Blue and Red target assignments as a function of the Level 1 role allocations, in Step b, a greedy heuristic assigns target-hitting aircraft on each side to the highest-value targets. Supporting forces for the target-hitting aircraft then are sequentially assigned on each side to maximize the probability of mission success. The assignment processes for the two sides proceed via a Blue Phase and a Red Phase. Each phase consists of three main steps in the following sequence:

1. Assign target hitting aircraft to targets.
2. Opposing Air Defense aircraft and AAF forces are assigned to targets in response to 1.
3. Assign Counter Air to targets to escort the aircraft in 1 and defend against the aircraft in 2.

Each assignment step is based on a greedy heuristic that assigns each aircraft to the target where it can produce the highest expected value extraction (target destruction) or value from destruction of enemy aircraft. Aircraft can only be assigned to targets within range of their home bases. Assignment to a given target can occur if a maximum allocation has not been reached, or if all targets already have a maximum allowed allocation. In the case of target-hitting aircraft, the default maximum allowed allocation for the sum of the expected target reduction proportions that will occur is 2.0; this accounts for the stochastic nature of the target strikes and the fact that some aircraft may be destroyed before reaching their targets. In the case of Air Defense or Counter Air, the maximum allocation reflects the expected proportion of opposing aircraft destroyed. The default is 1.0 [30].

The stochastic simulation of Step c is described below.

4. Stochastic Simulator

The attrition and target engagement simulation operates at planning Level 2. Since Level 2 incorporates target and mission planning, the simulation model must distinguish among target locations. The input includes the number of aircraft by side, force type, and role assigned to attack or defend each target. The simulation consists of three main phases.

Phase 1 Pre-Target: *Counter Air, Target Reduction,* and *AAF Suppression* aircraft on each side engage with opposing AAF and other Air Defense for half a time

period. Attrition randomly reduces the force quantities on both sides.

Phase 2 Target: *Surviving Target Reduction* and *AAF Suppression* aircraft on each side engage Targets to extract value (via target reduction) and determine a score. The capability indices at the targets are reduced accordingly and a net value to Blue is computed by subtracting Red's capability weighted value reduction from Blues.

Phase 3 Post-Target: Aerial battle of Phase 1 recommences for the remaining half of the time period with remaining aircraft. The Phase 3 model is identical to the Phase 1 model, but with forces reduced by Phase 1 attrition.

The planning logic assumes that the primary objective of the air campaign is to support the forces (hit Targets) on the ground. If more planes can be allocated to this role, the score will be higher. Clearly, an overly myopic strategy of assigning too many aircraft too soon to *Target Reduction* may not work well. The enemy *Air Defense* may need to be eroded first so that *Target Reduction* forces do not all get destroyed.

4.1 Attrition Logic

The attrition portions of the simulation (Phases 1 and 3) are designed as a multi-weapon type, stochastic Lanchester model. The state of each side (Red/Blue) is provided by a state-vector of weapon-system types (as in [31]) by location. Each unit on each side has a set of parameters designating the rates at which they can destroy opposing force types:

λ_{ij} = the rate (per planning period) at which forces of type i destroy opposing forces of type j when engaged solely against weapons of type j (assume that i and j incorporate the notion of force type, role, and location). These rates are constructed from a composite function that considers relative combat effectiveness, time span of engagement, and interactions between target locations. The composite function takes the form:

$$\lambda(i, j, t; i', j', t') = \lambda(i; i') \times \psi(i, j, t; i', j', t') \times z(i, j; i')$$

where

$\lambda(i, j, t; i', j', t')$ = the rate per mission at which weapons of type i , role j , attacking/defending target location t , destroy opposing weapons of type i' , role j' , assigned to attack/defend at target location t' .

$\lambda(i; i')$ = the rate per hour at which forces of type i destroy opposing weapons of type i' when engaged solely against forces of type i' . (In practice, from external testing/simulations).

$\psi(i, j, t; i', j', t')$ = the allocation of fire from weapons of type i , role j , attacking/defending target location t to opposing weapons of type i' , role j' assigned to attack/defend at target location t' , $\sum_i \sum_j \sum_{t'} \psi(i, j, t; i', j', t') = 1$

$z(i, j; i')$ = the number of hours per mission period spent by force type i engaging enemy weapon systems of force type i' , when assigned to role j .

A full discussion of the composite function is beyond the scope of this paper. Interested readers are referred to [30]. Without loss of generality, we drop the target index, t , in the discussion below.

Since each unit can potentially destroy more than one opposing type, a probabilistic fire allocation scheme based on the fractional allocation method (FAM) of [32] is used. The probability of a unit of weapon type i on side s , selecting an opposing weapon type j on side s' is given as:

$$A_{ij}^s = \frac{C_{ij}^s W_j^{s'}}{\sum_{j=1}^{N^{s'}} C_{ij}^s W_j^{s'}} \quad s, s' \in \{\text{Red, Blue}\}, s \neq s'$$

where:

A_{ij}^s = the allocation of fire from a weapon of type i on side s when that weapon is engaging an enemy of type j ; $i = 1, 2, 3, \dots, N^s$, $j = 1, 2, 3, \dots, N^{s'}$, $s, s' \in \{\text{Red, Blue}\}, s \neq s'$.

Note that $\sum_{j=1}^{N^{s'}} A_{ij}^s = 1$

C_{ij}^s = the corresponding fire allocation weighting coefficient

W_i^s = the number of weapons of type i on side s ; $i = 1, 2, 3, \dots, N^s$, $s \in \{\text{Red, Blue}\}$

N^s = the number of different weapon types on side s ; $s \in \{\text{Red, Blue}\}$

We use a simple allocation where the parameters, C_{ij}^s , are set to 1.0, implying that the allocation is based on relative numbers of opposing forces at a given location. To determine an aggregate destruction rate for each force type we compute:

λ_j^s = the total rate at which weapons of type j on side s are being destroyed by weapons on the opposing side, or

$$\lambda_j^s = \sum_{i=1}^{N^{s'}} A_{ij}^{s'} \lambda_{ij}^{s'} W_i^{s'} \quad s, s' \in \{\text{Red, Blue}\}, s \neq s'$$

The casualty process is modeled as (approximately) a Markov process with rate, λ_j^s . Therefore the time between kills for each weapon type j on side s has an exponential distribution with mean $(1/\lambda_j^s)$, similar to the Bonder-Ferrell approach to determining Lanchester rate coefficients [33]. But since these rates change with the force populations, they must be updated after each kill. This process occurs within the following attrition simulation algorithm:

Set W_i^s = the surviving number of weapons of type i on side s ; $i = 1, 2, 3, \dots, N^s$, $s \in \{\text{Red, Blue}\}$.

Set clock = 0

while clock < T

1. Compute the fire allocation fractions A_{ij}^s
2. Compute the total destruction rate for weapons of type j on side s : $\lambda_j^s = \sum_{i=1}^{N^{s'}} A_{ij}^{s'} \lambda_{ij}^{s'} W_i^{s'}$
3. Determine the next time to destruction, t_j^s , for each weapon type by drawing from an exponential probability distribution $E(\lambda_j^s)$
4. Find $t_{\min} = \min_{j,s}(t_j^s)$
5. If (clock + $t_{\min} < T$)
for s and j such that $t_j^s = \min_{j,s}(t_j^s)$, set $W_j^s = W_j^s - 1$
6. Set clock = clock + t_{\min}

4.2 Target Engagement Simulation

Target engagement occurs during the second or middle phase of the simulation. The surviving Target Reduction and AAF Suppression aircraft on each side hit the target in the target location to which they are assigned. For each target location, the model iteratively considers the targeting aircraft on each side. First, the current time period must be within the window of opportunity defined for the target. The actual damage done by an aircraft is then determined probabilistically. Damage is modeled as an exponential random variable with a mean value equal to the target effectiveness (average capability reduction by target) of the aircraft. After each strike, the target lists are updated and value reduction is accumulated to score the engagement. No further value can be extracted from a target once its capability falls below a desired damage level.

Note that each target has two different values, one according to Blue's value system, the second according to Blue's perception of Red's value system. The target engagement model returns the scores from both value systems. (Higher level algorithms assume that each side seeks to optimize decisions according to its own value system).

5. Application, Results, and Discussion

5.1 Database

The data used to populate the domain model came primarily from two sources. The first is a spreadsheet dataset used by AFRL/IFSB for model testing known as “the Korean Scenario.” It provides data for a subset of US forces — on the order of several hundred individual aircraft — in and around South Korea. The second major source, a well known public Web site (<http://www.globalsecurity.org>), was used to obtain data regarding North Korean forces and bases. The test database contains 13 Blue force types and seven Red (North Korean) force types, including both aircraft and AAF on each side. Blue has 15 base locations, each of which is a target for Red. Red has 201 targets, including bases, SAM sites, ground force concentrations, and various infrastructure targets.

Some domain attributes were approximated with plausible values sufficient for exploring the dynamics of the planning algorithms. The approximations included the combat effectiveness (kills/hr when a unit of force type i is in combat against force type j), target effectiveness (capability degradation per mission by force i against target t) and the target values. Combat effectiveness was evaluated based on relative force strength. Target attack effectiveness was evaluated based on the lbs. of munitions carried by an aircraft versus an estimated lbs. of munitions needed to destroy the target, based on examples from the AFRL/IFSB data set.

5.2 Performance Testing

Runtime requirements of the major algorithms were assessed on a test machine that the MATLAB “bench” function assigned a “relative speed” of 22. This provides a baseline for comparing the performance results to those obtainable on different machines. For a horizon weight, ω , of 1.0 (long term perspective), $K = 10$ samples per plan, a noninformative (flat) prior, and a five-period planning horizon, the optimizer described above was run until no significant value-increasing role allocations could be found. Results are shown in Table 1.

Each plan evaluation requires several minutes for 10 sample iterations, an average of 4.36 minutes each for the first 100 improvements and 2.81 for the last 26. Evaluation times become smaller as the forces are concentrated into fewer roles than the initial flat prior distribution. The times also are somewhat less for smaller ω (not shown), as forces are further concentrated into fewer roles. The time required to fully optimize the Blue Plan with $\omega = 1$ was almost 40 hours. An informed

N (improvement steps)	Total Time (hr)	Average Time/Step (min)	Time/Step for Interval
100	7.26	4.36	4.36
200	13.31	3.99	3.63
300	18.86	3.77	3.33
400	23.98	3.60	3.07
500	28.89	3.47	2.95
600	33.79	3.38	2.94
700	38.63	3.31	2.91
726	39.85	3.29	2.81

Table 1. Level 1 performance testing summary

prior would significantly decrease the optimization time, but we avoided this out of a desire to obtain unbiased results. With lower ω , the relative marginal values (shadow prices) are smaller, since the objective function value of the additional constrained “force resource” is smaller, causing the plan improvement algorithm to quit because of a lack of statistically significant differences in role allocations. For example, at $\omega = 0$, the algorithm quits after about 350 improvement steps. At that point it has operated on allocations in periods 1, 2, and part of 3. It cannot detect differences in allocations beyond period 3. At $\omega = 1$, all forces, roles, and periods were operated on.

5.3 The Effects of Horizon Weight

It is important for plans to reflect varying emphasis on short versus long term considerations. In the short run, there is no value given to survivability beyond the planning horizon. The next test created Level 1 Blue plans using different values for ω . A five-period planning horizon and 10 samples per plan with Blue’s initial plan set to a noninformative (flat) prior was optimized until no significant value-increasing reallocations could be made. Results are as follows:

Horizon Weight	Counter Air	Air Defense	Target Reduction	AAF Supression	Other
0	2.30	4.90	47.47	4.07	6.27
0.5	0.70	19.90	41.67	1.00	1.73
1	1.20	23.00	39.60	0.20	1.00

Table 2. Total allocations by role over the planning horizon

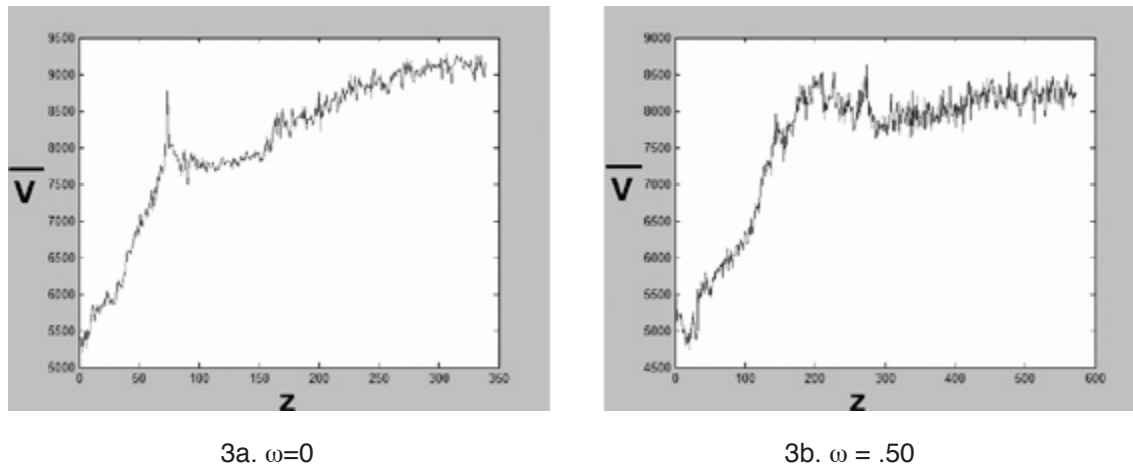
3a. $\omega=0$ 3b. $\omega = .50$ **Figure 3a, 3b.** Mean net blue plan value vs. improving iterations

Table 2 summarizes the resulting plans by summing role allocations over all forces and time periods. As expected, *Target Reduction* is deemphasized with higher ω . The other significant change is in *Air Defense*. At the same time, the other three role allocations show decreases. *Air Defense* increases because of the objectives of the model. Both Red and Blue are trying to maximize target value extraction. By assigning aircraft to *Air Defense*, Blue can both protect targets and reduce Red forces through attrition. This improves longer term survivability since Red then has fewer planes with which to attack Blue forces.

When $\omega = 0$, operating the plan optimization algorithm would again be expected to increase the discounted net Blue plan value. For higher ω , the optimizing algorithm should produce less net Blue value over the planning horizon, since survivability becomes a greater consideration. Figures 3a and 3b show the mean discounted net Blue plan value, V , as a function of improving steps, z , for the cases where $\omega = 0$ and 0.50.

Both cases show a generally steady upward trend in net Blue plan value. With $\omega = 0$, the net time discounted value extracted by the final plan is approximately 65% higher than for the original plan. With $\omega = 0.50$, the value increases by approximately 50%. However, both cases display regions where the net Blue plan value has a locally decreasing trend, most markedly when $\omega = 0$, between 75 and 100 improving iterations. This is not random variation: the phenomenon was observed for different random number seeds. We investigated the reallocations occurring at that point and found that forces were being shifted from *Counter Air* to *Target Reduction*. Our current hypothesis is that the shift results from the linear programming formulation, which assumes that Blue always can extract target value at a fixed average rate per aircraft type. Red, on the other

hand, has an explicit upper bound on the maximum target value extraction. A similar constraint on Blue could render a Blue input allocation infeasible. Adding logic to the algorithm to avoid these allocations would slow performance. So in some situations, Blue *Target Reduction* forces will “overrun” red targets — there are not enough Red targets to hit. This over-allocation is then reflected in the simulation. The planes moved from *Counter Air* to *Target Reduction* by the LP subject all Blue forces to greater attrition. At the same time, the former *Counter Air* forces have no targets remaining to hit. Therefore the overall net target extraction is reduced until additional adjustments are made by the optimizer. Optimizing the plan for a relatively few steps from a given starting point is not guaranteed to increase value when Red targets are near exhaustion.

5.4 Comparisons Between Planned and “Actual” Conflict

The last set of tests exercise the algorithms as they might be employed during actual use, applying them to the same simulated conflict used by the internal planning algorithms. (This highlighted differences due to approximations made by the planner rather than differences due to mismatches between the simulation and reality.) Figure 4 illustrates the evaluation process.

In this situation the planning tool is applied in an adaptive mode. During each cycle, the plan is updated according to the latest Intelligence Assessment. In our testing, the assessment reflects the probability distributions of forces. However, the Intelligence Assessment could be expanded to include updated understandings of weapon effectiveness, i.e., attrition rates, based on the experience of previous periods. Bayesian updating or even Hidden Markov Models could be introduced here [34].



Figure 4. Model use during a planning cycle

The first test demonstrates how the planner adds value to the plans during the operational planning cycles. The stopping condition for the conflict is that Blue can no longer extract net positive value by continuing, (e.g., because all Red targets are destroyed). With $\omega = 0$ and the discount factor set to 0.80, the optimizer was allowed to make N improving iterations, where $N \in \{0, 50, 100\}$. The initial plan is the noninformative (flat) prior, where forces are allocated evenly across feasible roles for five periods. We assumed perfect intelligence estimates for this particular test (relaxed in the next test), so the number of samples per evaluation is $K = 1$. After each cycle, the plan is updated by discarding the already used first period. Then, if there are sufficient periods left in the conflict to allow it, a new period is added to the plan, also set to a noninformative prior. Although it could take many iterations to fully optimize a five-period plan, only the first period of the plan is used at a time as part of rolling adaptive planning; thus, full optimization is not necessary. (In practice, longer plans might be used to facilitate integrating air warfare plans with other military planning, but this is outside the scope of the current model.) Table 3 shows results from five runs made for each value of N .

Clearly, plan optimization adds significant additional value from 0 iterations to 50 iterations. The difference in mean value extracted at 100 versus 50 iterations also is statistically significant but shows diminishing marginal returns from the additional iterations. This suggests a satisficing approach, where optimization is halted prior to the statistically-based stopping condition discussed

earlier. In Figure 3, the solutions follow a generally asymptotic path to the final value, but with a great deal of statistical noise. It may be possible to devise a stopping condition that detects the decreasing marginal return of additional iterations within the noisy mean value outputs. However, the additional iterations also help build the plan for future periods since reallocations tend to operate on earlier periods first, which could be important for coordinating plans with external units.

The second test quantifies the performance of the planner with respect to the accuracy of intelligence. The previous tests assumed that the Intelligence Assessment of Figure 4 was omniscient — it always knew the exact number of Red forces remaining. We now use a poor, systematically biased Intelligence Assessment. It is given an assumed detection probability of 0.90. That is, Blue believes that it can detect 90% of Red's forces, and scales the result accordingly. However, the "true detection probability" actually is 50%. Thus, Blue consistently underestimates Red forces. Both scenarios involve the same number of actual Red forces. Setting $\omega = 0.50$ as before, the number of samples per evaluation, K , will be 10 with poor intelligence and one with perfect intelligence. Table 4 summarizes the results from five runs of each scenario.

Counter-intuitively, runs with poor intelligence extract significantly higher Blue value than those with perfect intelligence. This is because $\omega = 0.50$ implies optimizing a weighted average of Blue value and survivability. Referring to Figures 3a and 3b, the optimized net Blue value is lower with $\omega = 0.50$ versus $\omega = 0$. Poor intelligence thus creates a situation similar to having a lower horizon weight. When intelligence is poor (Red force is underestimated), the optimizer believes that survivability considerations are less necessary, since there are seemingly fewer Red forces to contend with. The resulting plan overemphasizes Target Reduction, with a higher than anticipated attrition rate, and hence excessive extraction of value from current targets at the cost of sparing fewer resources for future needs.

Note that the "Reality/Simulation" component of Figure 4 also could represent the simulation engine within a wargame. Used in this mode, Red and Blue would each have a separate planning tool, but would operate upon a common reality. Each side also would have different Intelligence Assessment objects, representing its own intelligence capabilities. The planning tools would provide recommended allocations to each side during each planning period after determining an assumed (optimal) allocation for the other side. The player would accept or modify the recommendation and enter the final allocation into reality. After running the wargame forward for one period, the current system state would become input to a new Intelligence Assessment and the cycle would be repeated.

N/Cycle	Stopping Point t	Mean Value Extracted	Std. Dev.
0	10-15	7002.8	133.70
50	5-6	8884.7	283.54
100	3	9381.3	346.19

Table 3. Value Extraction during a Conflict

Scenario	Mean Value	Std. Dev.
Omniscient (Perfect Intel)	7572.6	112.81
Poor Intel	8094.5	104.25

Table 4. Discounted Net Blue Values Extracted with Varying Intelligence Quality

6. Conclusions

With several minutes required to evaluate each plan, fully optimizing a plan over a five-period planning horizon required CPU-days on the benchmark machine: a machine 20 times as fast, as seems plausible for the deployment environment, would reduce this time to a few hours, well within the practical needs for on-line decision support of daily mission planning. It is important to note that the test cases began with noninformative prior distributions for the current Blue Plan. If expert users provide more informed prior distributions, the time needed to fine tune them to optimality would be much reduced. Further speed-ups also may be achievable by optimizing the solvers for the nonlinear mathematical programs at the heart of the planning system and/or using statistical approximations to multi-period value functions [18]. Our approach involved successive solution of linear programs, requiring $(T+1)(T+2)/2 - 1$ linear programs for each sample of the Level 1 Stochastic Evaluator, where T is the number of periods in the planning horizon. The advantage of the linear programs is their automatic computation of relative marginal resource (force) values in the form of shadow prices.

The approach described here successfully demonstrates the use of an integrated planning hierarchy as a framework for air warfare planning. It shows that linearization of the decision space can be used to approximate a problem that would otherwise be intractable due to its size and uncertain dynamics. Within the evaluator, linear programming and simulation operate in tandem over an extended planning horizon to generate optimized Red responses, assumed outcomes, and relative marginal force values for Blue. Uncertainty is handled via sampling approaches, and more detailed

plans can be automatically generated that are consistent with top level plans. Lastly, the plan optimization algorithms have been successfully embedded in an operational planning cycle operating over a multi-period conflict, allowing the impacts of poor intelligence and false beliefs about enemy forces to be quantified. This provides a quantitative foundation for assessing and managing Blue’s risks by selecting plans that are robust to uncertainties about intelligence, while perhaps reducing performance for the “best guess” intelligence (in case it proves to be wrong.) Optimally hedging bets (i.e., current decisions) against imperfect intelligence is an important topic for future development that can be explored starting from the approach illustrated in Table 4.

7. Acknowledgements

This research was supported by the Air Force Research Laboratory, Information Directorate, C4ISR Modeling and Simulation Branch (AFRL/IFSB), under contract F30602-03-C-0167. Program Monitor: Gary Plotz.

8. References

- [1] Isaacs, R. 1965. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization.* John Wiley & Sons. New York.
- [2] Karlin, S. 1959. *Mathematical Methods and Theory in Games, Programming, and Economics.* Dover Press, Mineola, NY.
- [3] Dresher, M. 1981. *The Mathematics of Games of Strategy: Theory and Applications.* Dover Press.
- [4] Heise, SA and HS Morse. 2000. The DARPA JFACC program: Modeling and Control of Military Operations, In *Proceedings of the 39th IEEE Conference on Decision and Control*, Vol. 3, 2551-2555, Sydney, Australia, 12-15 Dec.

- [5] Mukai, HA et al. 2000. Game Theoretic Linear-Quadratic Method for Air Mission Control, In Proceedings of the 39th IEEE Conference on Decision and Control, Vol. 3, 2574-2580, Sydney, Australia, 12-15 Dec.
- [6] Cruz, J.B., et al. 2002. Moving Horizon Nash Strategies for a Military Air Operation, IEEE Transactions on Aerospace and Electronic Systems, 38(3), 989-999.
- [7] McEneaney, WM and K Ito. 2000. Stochastic Games and Inverse Lyapunov Methods in Air Operations, In Proceedings of the 39th IEEE Conference on Decision and Control, Vol. 3, 2568-2573, Sydney, Australia, 12-15 Dec.
- [8] McEneaney, WM, BG Fitzpatrick and IG Lauko. 2003. Stochastic Game Approach to Air Operations. Forthcoming in IEEE Trans. Aero. Elec. Systems. Retrieved from <http://math.ucsd.edu/~wmcenean/pubs/jfaccpaper.pdf>. Last accessed Aug 2004.
- [9] Cassandras, CG and W Li. 2003. A Receding Horizon Approach for Dynamic UAV Mission Management, In Enabling Technologies for Simulation Science VII, Alex F Sisti and Dawn A Trevisani, (eds.), Proceedings of SPIE Vol. 5091, 284-293.
- [10] Cave, WC and TE Busch. 2003. Theoretical Underpinnings of Predictive Battlespace Awareness, In Enabling Technologies for Simulation Science VII, Alex F Sisti and Dawn A Trevisani, (eds.), Proceedings of SPIE Vol. 5091, 17-26.
- [11] Surdu, J.R. and U.W. Pooch. 2000. "Simulation Technologies in the Mission Operational Environment." Simulation, the Journal of the Society of Computer Simulation, vol. 74, no. 3, pp. 138-161.
- [12] Stillman, B, V Yakhnis, and O Umanskiy. 2003. LG-ANALYST: Linguistic Geometry for Master Air Attack Planning. In Enabling Technologies for Simulation Science VII, Alex F Sisti and Dawn A Trevisani, (eds.), Proceedings of SPIE Vol. 5091, 414-425.
- [13] Clausewitz, K. von in M. Howard and P. Paret (Eds. And Trans.) On War, Princeton University Press, Princeton, NJ, 1989.
- [14] Ancker, CJ. 1995. A Proposed Foundation for a Theory of Combat, in Bracken, Kress, and Rosenthal (eds.) Warfare Modeling, published by John Wiley & Sons for the Military Operations Research Society, pp. 165-197.
- [15] Namrata V. 2004. Bound on Errors in Particle Filtering with Incorrect Model Assumptions and its Implication for Change Detection, presentation at the IEEE Intl. Conference on Acoustics, Speech & Signal Processing (ICASSP), Montreal, Canada, 17-21 May, 2204. Retrieved from <http://www.cfar.umd.edu/~namrata/nvaswani.pdf>. Last accessed Sep 2004.
- [16] Busch, TE. 2002. Modeling of Air operations for Course-of-Action Determination. In Enabling Technologies for Simulation Science VI, AF Sisti and DA Trevisani (eds.), Proceedings of SPIE, Vol 4716, 35-40.
- [17] Huggins, K, et al. 2002. Concept Level Anticipatory Planning. In Proceedings of the Advanced Simulation Technology Conference, San Diego, CA, 14-18 April 2002, 87-92.
- [18] Yost, KA and AR Washburn. 2000. The LP/POMPD Marriage: Optimization with Imperfect Information, Naval Research Logistics Quarterly, 47, 607-619.
- [19] Hillestad, R, J Owen, and D Blumenthal. 1995. Experiments in Variable Resolution Combat Modeling, in Warfare Modeling, J Bracken, M Kress, and R Rosenthal (eds.). Published by John Wiley & Sons for the Military Operations Research Society, Alexandria, VA.
- [20] Hillestad, R and LR Moore. 1996. The Theater-Level Campaign Model: a Research Prototype for a New Generation of Combat Analysis Models. RAND MR-388, RAND Corporation, Santa Monica, CA.
- [21] Schlabach, JL, CC Hayes, and DE Goldberg. 1999. "FOX-GA: A Genetic Algorithm for Generating and Analyzing Battlefield Courses of Action. Evolutionary Computation, 7(1), 45-68.
- [22] Hill, JMD and J Surdu. 2001. Agent Cooperation and Simulation in Dynamic Plan Building, in Proceedings of the 5th World Multi-Conference on Systems, Cybernetics and Informatics (SCI 2001), Orlando, Florida, 22-25 July, vol. XVI, 45-51.
- [23] Davis, P and J Bigelow. 2001. Meta Models to Aid Planning of Intelligent Machines, In Measuring the Performance and Intelligence of Systems: Proceedings of the 2001 PerMIS Workshop, Mexico City, Sept 4, 2001. Retrieved from http://www.isd.mel.nist.gov/research_areas/research_engineering/Performance_Metrics/PerMIS_2001_Proceedings/Davis_Bigelow.pdf. Last accessed Oct 20, 2004.
- [24] Jacobson, SH and E Yucesan. 2004. Global Optimization Performance Measures for Generalized Hill Climbing Algorithms. Journal of Global Optimization, 29(2), 173-90.
- [25] Dippon J. 2003. Accelerated Randomized Stochastic Optimization. Ann. Statist., 31(4), 1260-1281.
- [26] Swisher, JR, P Hyden, SH Jacobson, and L Schruben. 2000. Simulation Optimization: a Survey of Simulation Optimization Techniques and Procedures. Proceedings of the 32nd Conference on Winter Simulation, Orlando, FL, 119-128.
- [27] Lagoudakis MG and R Parr. 2002. Value Function Approximation in Zero-Sum Markov Games. Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, Alberta, Canada, Aug 1-4.
- [28] Bertsekas DP, JN Tsitsiklis, and C Wu. 1997. Rollout Algorithms for Combinatorial Optimization, Journal of Heuristics, 3(3), 245-262.
- [29] Glover, F and M Laguna. 1998. Tabu Search. Kluwer, Boston.
- [30] Popken, D and LA Cox. 2004. Model Identification and Optimization for Operational Simulations, Technical Report AFRL-IF-RS-TR-2004-237, Air Force Research Laboratory, Information Directorate, Rome, NY.
- [31] Popken, D and LA Cox. 2000. An Investigation of System Identification Techniques for Simulation Model Abstraction, Technical Report AFRL-IF-RS-TR-2000-8, Air Force Research Laboratory, Information Directorate, Rome, NY.
- [32] Anderson, LB and FA Miercort. 1989. Combat: A Computer Program to Investigate Aimed Fire Attrition Equations, Allocation of Fire, and the Calculations of Weapons Scores, IDA Paper P-2248 (DTIC AD-A213660).
- [33] Taylor, JG. 1983. Lanchester Models of Warfare, Volume II, Military Application Section of the Operations Research Society of America, Alexandria, VA.
- [34] Popken, DA and LA Cox. 2003. Model Identification and Optimization for Operational Simulation. In Enabling Technologies for Simulation Science VII, Alex F Sisti and Dawn A Trevisani, (eds.), Proceedings of SPIE Vol. 5091, 294-303.

9. Appendix: Level 1 Mathematical Program

Variables:

ra_{ijt} = # of Red forces of type i allocated to role j at the beginning of t ; $i=1,2,\dots,F_r$; $j = 1-5$

ba_{ijt} = # of Blue forces of type i allocated to role j at the beginning of t ; $i=1,2,\dots,F_b$; $j = 1-5$

r_{ijt} = # of Red forces of type i , surviving period t attrition, that were allocated to role j at the beginning of period t

b_{ijt} = # of Blue forces of type i , surviving period t attrition, that were allocated to role j at the beginning of period t

Assume $j = 3$ represents the *Target Reduction* role. Only forces assigned to this role can directly "extract" value by destroying enemy targets.

$\mathbf{r}(t)$, $\mathbf{ra}(t)$ =row vectors of length F_r*5 of Red force quantities (all i,j) in period t

$\mathbf{b}(t)$, $\mathbf{ba}(t)$ =row vectors of length F_b*5 of Blue force quantities (all i,j) in period t

Input parameters:

- PB_{ijt} = Blue's weights for force i , role j , in period t (these sum to 1.0 across a given (i,t) - this is the current Blue conceptual plan)
- LAMBDAB = rate at which Blue forces (i,j) destroy Red forces (i',j') (2-d matrix)
- LAMBDAR = rate at which Red forces (i',j') destroy Blue forces (i,j) (2-d matrix)
- vb_i = average value reduction (from red's perspective) that Blue force i in the Target Reduction role achieve
- vr_i = average value reduction (from red's perspective) that Red force i in the Target Reduction role achieve
- ω = horizon weight
- sb_i = salvage value factor for Blue force $i = \omega * sb_i^0$
- sr_i = salvage value factor for Red force $i = \omega * sr_i^0$
- $vrTotal$ = the total remaining value of all Red targets (Red perspective)
- $b(0)$ = any initial feasible assignment of current Blue forces to roles
- $r(0)$ = any initial feasible assignment of current Red forces to roles

Optimizing from Red's perspective, the problem is initially formulated as P(T).

P(T):

$$\text{Maximize } \sum_{i=1}^{Fr} \left[\sum_{t=1}^T (vr_i) r_{i3t} + \sum_{j=1}^5 (sr_i vr_i) r_{ijT} \right] - \sum_{i=1}^{Fb} \left[\sum_{t=1}^T (vb_i) b_{i3t} + \sum_{j=1}^5 (sb_i vb_i) b_{ijT} \right]$$

Subject to:

max value obtainable:

$$\sum_{t=1}^T \sum_{i=1}^{Fr} r_{i3t} vr_i \leq vrTotal$$

expected attrition: $t = 1, 2, 3, \dots, T$

$$b(t) = \max(0, ba(t) - ra(t) * LAMBDAR)$$

$$r(t) = \max(0, ra(t) - ba(t) * LAMBDAB)$$

Red reallocation: $t = 1, 2, \dots, T$

$$\sum_{j=1}^5 I_{ij} ra_{ijt} = \sum_{j=1}^5 r_{ij(t-1)} \quad i=1,2,\dots,Fr \quad (I_{ij}=1 \text{ if role } j \text{ is feasible for force } i; 0 \text{ otherwise})$$

Blue reallocation to plan: $t = 1, 2, \dots, T$

$$ba_{ijt} = PB_{ijt} * \sum_{j=1}^5 b_{ij(t-1)} \quad i=1,2,\dots,Fr; j=1,2,3,4,5$$

nonnegativity:

$$ra_{ijt}, r_{ijt}, ba_{ijt}, b_{ijt} \geq 0$$