

Title:

Robust Action Strategies to Induce Desired Effects

Student paper submission

Submit to:

2002 CCRTS --- Modeling and Simulation

Authors:

Haiying Tu:

Graduate student
Dept. of Electrical and Computer Engineering
University of Connecticut
Storrs, CT 06269-2157
Tel./Fax: (860) 486-2210/5585
E-mail: haiying.tu@uconn.edu

Yuri Levchuk:

Aptima, Inc
Woburn, MA
Tel: (781) 935-3966 × 236
E-mail: levchuk@aptima.com

Krishna R. Pattipati:

Professor, IEEE Fellow
Dept. of Electrical and Computer Engineering
University of Connecticut
Storrs, CT 06269-2157
Tel./Fax: (860) 486-2890/5585
E-mail: krishna@engr.uconn.edu

Robust Action Strategies to Induce Desired Effects

**Haiying Tu
Yuri Levchuk
Krishna R. Pattipati**

Dept. of Electrical and Computer Engineering
University of Connecticut
Storrs, CT 06269-2157
Tel./Fax: (860) 486-2210/5585
E-mail: krishna@engr.uconn.edu

Abstract

A new methodology is given in this paper to obtain a near-optimal strategy (i.e., specification of courses of action over time), which is also robust to environmental perturbations (unexpected events and/or parameter uncertainties), to achieve the desired effects. A dynamic Bayesian network (DBN)-based stochastic mission model is employed to represent the dynamic and uncertain nature of the environment. Genetic algorithms are applied to search for a near-optimal strategy with DBN serving as a fitness evaluator. The probability of achieving the desired effects (namely, the probability of success) at a specified terminal time is a random variable due to uncertainties in the environment. Consequently, we focus on signal-to-noise ratio (SNR), a measure of mean and variance of the probability of success, to gauge the goodness of a strategy. The resulting strategy will not only have a relatively high probability of inducing the desired effects, but also be robust to environmental uncertainties.

Keywords: Effects-based operations, optimization, organizational design, robustness, signal-to-noise ratio, Taguchi method, dynamic Bayesian networks, genetic algorithms, confidence region, hypothesis testing

1. Introduction

Robustness is a key issue in stochastic planning problems under uncertainty. This paper describes the application of dynamic Bayesian networks, along with evolutionary optimization through genetic algorithms, to derive robust strategies that induce the desired effects in a mission environment. The methodology discussed here is applicable to both military organizations and commercial enterprises.

An organization's ability to choose an efficient and effective strategy for its mission execution is critical to its superior performance. Given the dynamic nature of a modern military environment, an effective C2 strategy is to create the desired effects at the right place and at the right time. Actions constitute the means by which an organization attempts to shape the future. However, environmental conditions also affect the feasibility of organization's actions, making some strategies more likely to succeed than others. The uncertainty about the dynamics of potential interactions between organization's actions and its environment could result from two different sources: (i) the inability to predict some of the indirect cross-influence effects of organization's actions [Leblebici81], and (ii) the stochastic nature of the dynamic environment faced by the organization [Emery65]. Consequently, the extent of a potential organization's control over the effects it desires to achieve is limited and, in some cases, indirect. The corresponding models must capture and quantify the influence of organization's actions, various stochastic events, and direct or latent effects.

In most cases, there are a large number of cause-effect relationships within an environment, many of which are not observable by the organization. Probabilistic models, such as dynamic Bayesian networks, are natural candidates for representing uncertainties in a dynamic environment. A robust strategy seeks to maximize the probability of successfully achieving the desired effects, while minimizing its variability.

This paper introduces a framework for devising a robust organizational strategy to induce desired effects in a dynamic and uncertain mission environment. A normative model of the stochastic environment, based on a dynamic Bayesian network (DBN), to infer indirect influences and to track the time propagation of effects in complex systems is developed. For a specified set of mission goals (i.e., desired effects) and organizational constraints, intermediate organizational objectives are derived, and a near-optimal organizational strategy is obtained via genetic algorithms, where the DBN serves as a fitness evaluator for candidate strategies. The results of this paper will form a foundation for current research on dynamic adaptation of organizational strategies.

The remainder of the paper is organized as follows. In section 2, we will formalize the problem as a graph model, which we call an effects-based mission model. This model represents the concepts from effects-based operations (EBO) (see [McCrabb01], [Davis01]) in the form of a Bayesian network. Section 3 describes our approach for this problem, which combines DBN with genetic algorithms to compute robust action strategies. Signal to noise ratio (SNR), computed from Monte Carlo runs, is used as a criterion of robustness. In section 4, two conceptual examples, one commercial and the other military, are solved to demonstrate the feasibility of the methodology. Finally, we conclude with a summary of current research and future research directions.

2. Model and Formulation for Strategy Optimization

A stochastic planning problem in an uncertain environment can be defined as follows: given an initial environment state, determine optimal action sequences that will bring the environment to a specified destination (goal) state at a specified time with a relatively high probability. The destination, in our case, is the set of desired effects.

The process to solve this problem is to:

- (i) Represent the joint dynamics of the organization and its environment;
- (ii) Optimally select appropriate courses of action;
- (iii) Assess the probability of successfully achieving the desired effects and the corresponding risks.

As illustrated in Fig.1, a dynamically evolving *effects-based mission model* $G_k = G(t_k) = (V, E, P_k)$, which can be viewed as a Bayesian network at time t_k , combines knowledge about the organization and its environment. G_k is a directed acyclic graph consisting of a set of nodes V and a set of directed edges E with a fixed structure. Every node is considered as a random variable and can assume Boolean values --- either true ('1') or false ('0'). For each node $v_i \in V$, we define a probability mass function (pmf) $P_k(v_i) = P\{v_i(t_k)\}$ to characterize the environment uncertainty at time t_k .

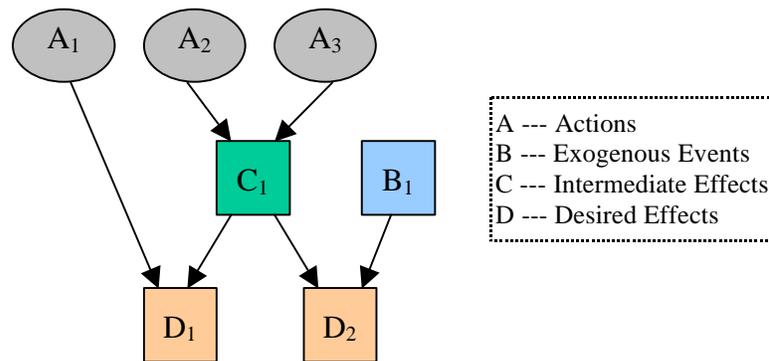


Fig.1: A Simple Effects-Based Mission Model

The dynamic evolution of the effects-based mission model unfolds through a finite horizon timeline, which is discretized into T time slices (from t_1 to t_T). Time slices are used to represent a snapshot of the evolving temporal process [Kanazawa95]. This evolution can be depicted as a DBN as shown in Fig.2 (t_0 is the time before the first time slice). The nodes in this network have causal-temporal relationships with each other. The solid arcs are “synchronic” to portray the causal relationship in a single time slice, and the dashed edges are “diachronic” to show the temporal evolution of the model between neighboring time slices [Boutilier98]. With these assumptions, the DBN is Markovian.

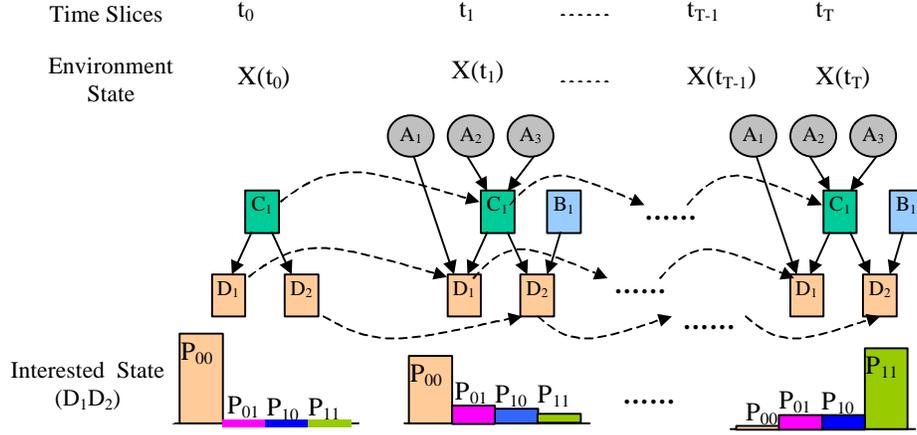


Fig. 2: Time-evolution of Effects-Based Mission Model as a DBN

Based on Fig.2, the key elements of this model are as follows:

- (i) *Critical objects* (including *centers of gravity*, or COG [McCrabb01]) that constitute the environment of interest, with $X(t_k) = \{P_k(v_i) | v_i \in V\}$ to portray the overall state of the environment at time t_k ;
- (ii) *Objectives* (to be framed in terms of *desired effects*) $D = \{D_n | 1 \leq n \leq N_D\}$ specified by the desired outcomes ('1' or '0') and the corresponding terminal time t_{D_n} for each effect: $D_n(t_{D_n}) = 1$ or $D_n(t_{D_n}) = 0$. Here $N_D = |D|$ is the total number of effects we are interested in;
- (iii) *Critical/important exogenous events*, regarded as *noise factors*, whose occurrence is beyond the control of the organization, but will affect the environmental dynamics: $B = \{B_j | 1 \leq j \leq N_B\}$. $N_B = |B|$ is the total number of exogenous events in the environment. In many cases, one has partial knowledge of the statistical properties (e.g., means and variances, probability distributions) for these events. For instance, if event B_1 in Fig.1 occurs with a probability that is uniform between $[0.2, 0.6]$ at time t_k , then $P_k\{B_1 = 1\} = p_1, P_k\{B_1 = 0\} = 1 - p_1$, where $p_1 \sim U[0.2, 0.6]$. The prior pmfs in the model are application specific and are normally elicited from domain experts. We may also consider the enemy actions (or competitor actions in business applications) as exogenous events. Note that some events may have inhibiting effects in that they reduce the probability of achieving certain desired effects;
- (iv) *Feasible actions*, regarded as *control factors*, which can be employed by an organization to influence the state of the environment: $A = \{A_q | 1 \leq q \leq N_A\}$, where $N_A = |A|$ is the total number of feasible actions. Each action will take a value of "true" or "false" at each time slice once the decision maker determines a strategy. That is, $P_k\{A_q = 1\} = 1$ if action A_q is activated at time slice t_k ; otherwise, $P_k\{A_q = 1\} = 0$. If there are no constraints on actions, potential choices for each action consist of 2^T strings of binary digits ranging from '0(t_1) \wedge 0(t_T)' to '1(t_1) \wedge 1(t_T)'. In real applications, however, the available potential actions may be very limited and much less than 2^T . Without loss of generality, we assume that $(r_q + 1) \ll 2^T$ feasible choices for action A_q from a domain $\Omega_{A_q} = \{a_q^0, a_q^1, a_q^2, \wedge a_q^{r_q}\}$ are available. Each element a_q^i ($0 \leq i \leq r_q$) in this set maps to a string ' $a_q^i(t_1) \wedge a_q^i(t_T)$ ' with $a_q^i(t_k) \in \{0, 1\}$ ($0 \leq k \leq T$). Let f_{a_q} be the cost of

selecting string a_q for action A_q . A strategy under a given initial environment state $X(t_0)$ is a set of strings for all the actions: $S = \{(a_1, a_2, \dots, a_{N_A}) \mid a_q \in \Omega_{A_q}, 1 \leq q \leq N_A\}$. Thus, the space of feasible action strategies is $\Omega_s = \Omega_{A_1} \times \Omega_{A_2} \times \dots \times \Omega_{A_{N_A}}$ and the cost of the strategy is $F_S = \sum_{q=1}^{N_A} f_{a_q}$.

- (v) *Intermediate effects* are defined to differentiate those effects that are not desired end effects per se, but are useful in connecting the actions and events to the desired effects. These effects are termed direct effects in Effects-Based Operations [McCrabb01]. All the intermediate effects form a set $C = \{C_m \mid 1 \leq m \leq N_C\}$ with $N_C = |C|$. Fig. 2 shows that only desired effects and intermediate effects are connected by diachronic edges;
- (vi) Direct influence *dependencies* between all the objects of the system and their mechanisms are specified by conditional probability tables (CPTs) in Bayesian networks parlance. We assume that actions and events in our model are root nodes and the desired effects are conditionally independent. Consequently, directed synchronic edges $E = \{<v_i, v_j >\}$ exist only from an action to an intermediate effect or a desired effect ($v_i \in A, v_j \in C \cup D$), from an event to an intermediate effect or a desired effect ($v_i \in B, v_j \in C \cup D$), and from an intermediated effect to a desired effect or another intermediate effect ($v_i \in C, v_j \in C \cup D$). Diachronic edges are directed from immediately prior time slice to the current one for each intermediate and desired effect. Evidently, the number of CPTs needed is $(N_C + N_D)$.
- (vii) The total budget available for the organization is constrained by F_{budget} .

It can be seen that four types of nodes are defined in our effects-based mission model such that $V = A \cup B \cup C \cup D$. The total number of nodes in the mission model is $N = N_A + N_B + N_C + N_D$. Define the timeline starting from initial time t_0 to the terminal time t_T for achieving all the desired effects, where $t_T = \max(t_{D_n})$, ($1 \leq n \leq N_D$). Without loss of generality, the initial environment state $X(t_0)$ is assumed known and deterministic; in other words, all the effects (desired or intermediate) are observed as being “true” or “false”. Given that the total number of nodes is N , the number of possible states is 2^N . The states span from all “false” to all “true” with each state having a probability, say P_M , with the constraint that $\sum_{M=1}^{2^N} P_M = 1$.

Conceptually, the problem is to achieve the desired effects $\{D\}$ with a very high probability at specified times. Thus, we only need to focus on the marginal probabilities of all the D 's. For example, in Fig. 2, if $D_1(t_{D_1})=1$ and $D_2(t_{D_2})=1$ are desired, the objective of our problem is to make the initially $P_{11}(t_0)=0$ to be a “statistically significant” $P_{11}(t_T)$, where P_{ij} is the joint probability of desired effect D_1 in state i and desired effect D_2 in state j . Evidently, $P_{00}(t_k) + P_{01}(t_k) + P_{10}(t_k) + P_{11}(t_k) = 1$ holds for all t_k .

The mathematical formulation of the strategy optimization problem is as follows:

$$\begin{aligned}
& \max_S (P\{D(t_k) | X(t_0), S\}) \\
& = \max_S (P\{D_1(t_{D_1})D_2(t_{D_2}) \wedge \dots \wedge D_{N_D}(t_{D_{N_D}}) | X(t_0), S\}) \\
& = \max_S \left(\prod_{n=1}^{N_D} P\{D_n(t_{D_n}) | X(t_0), S\} \right) \quad (\text{assumption (iii)})
\end{aligned} \tag{1}$$

subject to:

$$F_S = \sum_{q=1}^{N_A} f_{a_q} \leq F_{budget} \tag{2}$$

3. Solution Approach

3.1 Overview of the Solution Approach

As shown in Fig.3, our approach to solve the strategy optimization problem combines concepts from robust design, dynamic Bayesian networks and heuristic optimization algorithms. DBNs, which adopt probability evaluation algorithms such as the junction tree for stochastic inference [Jordan99], are used to model the dynamics of the environment and to calculate the probability of desired effects at specified times. Monte Carlo runs are made to account for uncertainty in system parameters in the inner loop of DBN. That is, disturbances are introduced by randomly choosing network parameters (prior pmfs of events and conditional probabilities). In each Monte Carlo run, DBN will evaluate the joint probability of achieving the desired effects. The results of Monte Carlo runs provide a histogram, and we approximate it as a Gaussian density (based on the Central Limit Theorem) with sample mean and sample variance. Using the sample mean and variance and following robust design techniques of Taguchi [Phadke89], a signal-to-noise ratio (SNR) is computed; this criterion maximizes the probability of achieving the desired effects while minimizing its variability. A genetic algorithm is employed in the outer loop to optimize the action strategies.

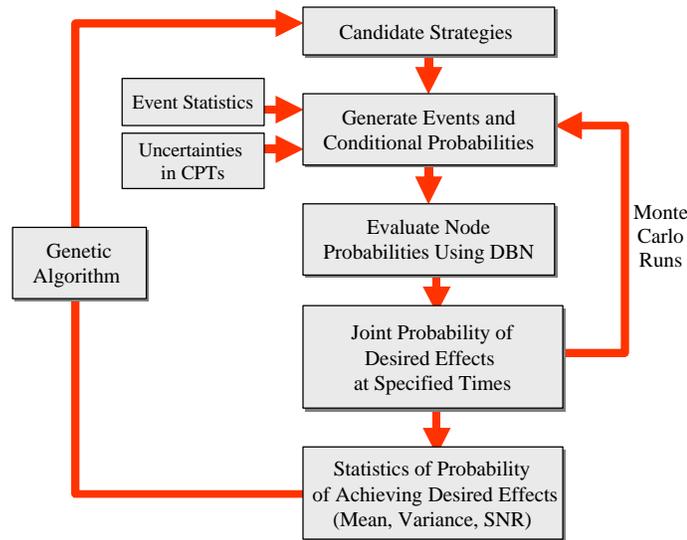


Fig. 3: Approach Overview

Conceptually, the probability of achieving the desired effects is a function of actions A , exogenous events B and time t_k , that is, $P(D) = f(A, B, t_k)$. In iterations of the genetic algorithm, since we choose candidate strategies, thereby fixing the values of A , the probability will be a function of events B and time t_k , that is, $P(D|A) = g(B, t_k)$. Then, in each Monte Carlo run of DBN inference, for the given sequences of actions A , we estimate the occurrence probabilities of exogenous events B . Consequently, from a single Monte Carlo run, we have $P(D|A, B) = h(t_k)$. We can see that Monte Carlo runs inside the DBN inference makes it possible to measure the robustness of a strategy in an uncertain environment in terms of the signal-to-noise ratio.

3.2 Probability Propagation through DBN

Bayesian networks (BN), also known as probabilistic networks, causal networks or belief networks, are formalism for representing uncertainty in a way that is consistent with the axioms of probability theory [Pearl88]. As a Graphical model with strong mathematical background, it has grown enormously over the last two decades. Indeed, there is now a fairly large set of theoretical concepts and results [Jordan99], as well as software tools for model construction, learning and analysis, such as Microsoft's MSBNX [Microsoft], Netica [Netica] and Matlab Toolbox [Murphy].

Given a set of nodes $V = \{v_1, v_2, \dots, v_n\}$, a Bayesian network computes the joint probability of variables in the network via:

$$P(v_1, v_2, \dots, v_n) = \prod_{i=1}^N P(v_i | \mathbf{p}(v_i)) \quad (3)$$

where $\mathbf{p}(v_i)$ is the possible instantiation of the parent nodes of v_i . This equation is derived based on the chain rule of probability and conditional independence [Heckerman 95]. To be precise, given the state of a node's parents, all the ancestors are conditionally independent of the node. Here, we use "parents" to depict the directly fan-in nodes, and "ancestors" to represent the parents' parents, and so on.

A major drawback of the standard theory of Bayesian networks is that there is no natural mechanism for representing time [Aliferis96]. Dynamic Bayesian networks (DBN) are normally used for representing Bayesian networks that also take into account temporal information. As we see from Fig.2, DBN is a compact, factored representation of a Markov process [D'Ambrosio99]. Since the state of the environment is still static during one time slice, DBN can be decomposed as a sequence of static Bayesian networks with certain connections [Barrientos98].

Based on Markov hypothesis, the probability of state at time slice t_k in a DBN, given all the evidence (in our case, actions and events) up to that time is given by [Russell95]:

$$P(X(t_k) | \{X(t_i)\}_{i=0}^{k-1}; \{A(t_i), B(t_i)\}_{i=1}^k) = P(X(t_k) | X(t_{k-1}); A(t_k); B(t_k)) \quad (4)$$

In our model, intermediate and desired effects between adjacent time slices have temporal links; other nodes are supposed to be temporally independent [Tawfik00]. The temporal independence implies that the probability mass functions of node v_i at time t_k and that of v_j at time t_q , which are not temporally connected, are independent, that is, $P\{v_i(t_k) | v_j(t_q)\} = P\{v_i(t_k)\} \triangleq P_k(v_i)$ for $t_k \neq t_q$.

Fig. 4 shows the augmented Bayesian network which is applied for probability propagation in the effects-based mission model. It is logically extended from the initial static Bayesian Network by introducing dummy nodes for all the intermediate and desired effects. Dummy nodes are defined as: $V_i^0 = \{v_i^0 | v_i \in C \cup D\}$ with $P_{k+1}\{v_i^0\} = P_k\{v_i\}$. The corresponding CPTs are listed in Tables I-III. This data will be used in a later example.

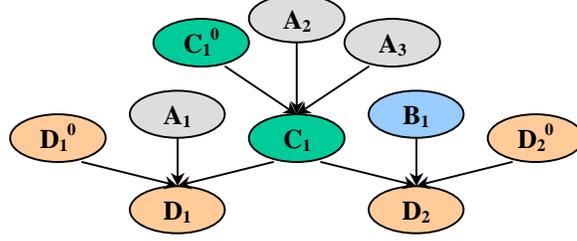


Fig. 4: Augmented Bayesian Network

Table I: CPT for C_1

Parent Nodes			C_1	
A_2	A_3	C_1^0	Yes	No
Yes	Yes	Yes	1.0	0.0
		No	0.4	0.6
	No	Yes	1.0	0.0
		No	0.0	1.0
No	Yes	Yes	1.0	0.0
		No	0.1	0.9
	No	Yes	1.0	0.0
		No	0.0	1.0

Table II: CPT for D_1

Parent Nodes			D_1		
A_1	C_1	D_1^0	Yes	No	
Yes	Yes	Yes	1.0	0.0	
		No	0.4	0.6	
		No	1.0	0.0	
	No	Yes	Yes	1.0	0.0
			No	0.05	0.95
		No	Yes	1.0	0.0
No	Yes	Yes	1.0	0.0	
		No	0.35	0.65	
	No	Yes	1.0	0.0	
		No	0.0	1.0	

Table III: CPT for D_2

Parent Nodes			D_2		
B_1	C_1	D_2^0	Yes	No	
Yes	Yes	Yes	1.0	0.0	
		No	0.1	0.9	
		No	1.0	0.0	
	No	Yes	Yes	1.0	0.0
			No	0.0	1.0
		No	Yes	1.0	0.0
No	Yes	Yes	1.0	0.0	
		No	0.5	0.5	
	No	Yes	1.0	0.0	
		No	0.0	1.0	

In the DBN of Fig. 2, the probability will propagate vertically from causal nodes to effect nodes, and propagate horizontally from one time slice to the next as follows:

- (i) Set the initial pmfs of nodes: $P_0\{v_i\} = P_1\{v_i^0\}$, $v_i \in C \cup D$ based on known $X(t_0)$;
- (ii) Let $k = I$;
- (iii) Select an action strategy: $S = \{(a_1, a_2, \dots, a_{N_A}) \mid a_q \in \Omega_{A_q}, 1 \leq q \leq N_A\}$, where if $a_q(t_k) = 1$, set $P_k\{A_q = 1\} = 1$; else $P_k\{A_q = 1\} = 0$;
- (iv) Randomly select probability mass functions of exogenous events $P_k\{B_j\}, 1 \leq j \leq N_B$;
- (v) Calculate probability mass functions of the intermediate and desired effects using Bayesian model averaging [Madigan96]:

$$P_k\{v_i\} = \sum_{\mathbf{p}(v_i), v_i^0} P\{v_i \mid \mathbf{p}(v_i), v_i^0\} \cdot P_k\{\mathbf{p}(v_i)\} P_k\{v_i^0\}, v_i \in C \cup D;$$

- (vi) Propagate the current probability mass functions to the next time slice:

$$P_{k+1}\{v_i^0\} = P_k\{v_i\}, v_i \in C \cup D;$$

- (vii) Let $k = k+1$. If $k \leq T$, go back to step (iii); otherwise, stop.

3.3 Action Strategy Optimization with GA

3.3.1 Algorithm Overview

Genetic algorithms (GAs) are general-purpose global optimization techniques based on the principles of evolution observed in nature. They combine selection, crossover and mutation operators with the goal of finding the best solution to a problem. GA creates an initial population, evaluates the fitness of each individual in this population, and searches for a near-optimal solution from generation to generation until a specified termination criterion is met. These algorithms have been widely used in areas where exhaustive search maybe infeasible because of a large search space and where domain knowledge is difficult or impossible to obtain.

The use of genetic algorithms requires the specification of six fundamental elements: chromosome representation, selection function, the genetic operators making up the reproduction function, the creation of initial population, termination criteria and the evaluation function [Houck95].

In this section, we use GA to navigate the solution space to obtain a near-optimal action strategy. A typical GA may have a genetic cycle as follows [Stender94]:

- (i) Initialize the population randomly or with potentially good solutions;
- (ii) Evaluate the fitness for each individual in the population;
- (iii) Select parents for alteration;
- (iv) Create offspring by crossover and mutation operators;
- (v) Reorganize the population by deleting old ones and creating new ones while keeping the total size fixed;
- (vi) Go to (iii) until termination criteria are met.

Our implementation of GA for strategy optimization is illustrated in Fig.5. Important steps and fundamental issues will be explained in more detail in the following subsections.

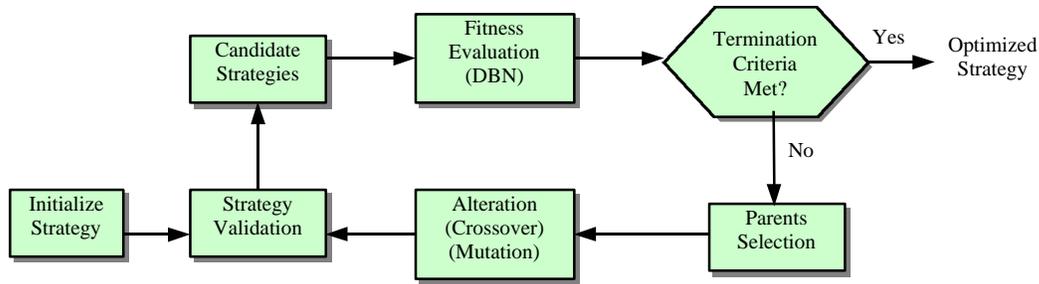


Fig. 5: GA Cycle for Strategy Optimization

3.3.2 Chromosome Representation

For any GA, a chromosome representation is necessary to describe each individual in the solution population. The population in our problem corresponds to candidate strategies to induce the desired effects. We use integer-valued GA in our problem. In section 2, the feasible actions are given by $A = \{A_q \mid 1 \leq q \leq N_A\}$ with $A_q \in \{a_q^0, a_q^1, \dots, a_q^{r_q}\}$. Thus, the chromosome can be represented as a string of integer genes $\mathbf{w} = (\mathbf{w}_1 \mathbf{w}_2 \wedge \mathbf{w}_q)$, where $0 \leq \mathbf{w}_q \leq r_q$. The lower bound “0” corresponds to the null action “do not perform A_q ” in the entire timeline. If $\mathbf{w}_q = 1$, a_q^1 is picked for A_q , if $\mathbf{w}_q = 2$, a_q^2 is picked for A_q , and so on. In other words, the gene is coded to represent the assignment of an action, and the whole chromosome is a code representing an action strategy.

3.3.3 Initial Population and Pre-filtrating

Population initialization is the first step in GA. The most popular method is to randomly initialize the population. However, since GAs can iteratively improve existing solutions, the beginning population can be seeded with potentially good solutions [Houck95], especially for cases where partial knowledge about the solution is known. In our problem, we generate the initial strategy randomly. Thus, for any individual $\mathbf{w} = (\mathbf{w}_1 \mathbf{w}_2 \wedge \mathbf{w}_q)$ in the initial population, \mathbf{w}_q ($1 \leq q \leq N_A$) is randomly selected from $\{0, 1, \dots, r_q\}$. The size of the population can be selected to conform to available computational resources (time and memory) and to accommodate the size of the solution space.

In planning, other important issues such as the cost of a strategy and the available resources need to be considered. A randomly created individual $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2 \wedge \mathbf{w}_q)$ is pre-filtered to satisfy the constraints of cost and resource budgets. For example, verifying for each individual if $\sum_{q=1}^{N_q} f_{a_q} \leq F_{budget}$ is satisfied enables us to check the cost constraint for feasibility.

3.3.4 Evaluation Function

DBN performs the inner loop inference to compute the evaluation function for GA. The evaluation function will map the population candidate into a partially ordered set [Houck95], which will be input to the next step, i.e., population selection.

DBN is used to obtain the probability of achieving the desired effects at certain time slices for a given strategy $P\{D_1(t_{D_1})D_2(t_{D_2}) \wedge D_{N_D}(t_{D_N}) | X(t_0), S\}$. In a noisy environment, this probability is a random variable because of the uncertainty in the statistical description of exogenous events B . In the DBN loop, we generate a histogram of this probability via M Monte Carlo runs, the sample mean and variance are computed via:

$$\mathbf{m} = \frac{1}{M} \sum_{i=1}^M P_i\{D_1(t_{D_1})D_2(t_{D_2}) \wedge D_{N_D}(t_{D_N}) | X(t_0), S\} \quad (5a)$$

$$\mathbf{s}^2 = \frac{1}{M-1} \sum_{i=1}^M (P_i\{D_1(t_{D_1})D_2(t_{D_2}) \wedge D_{N_D}(t_{D_N}) | X(t_0), S\} - \mathbf{m})^2 \quad (5b)$$

Signal-to-noise ratio (SNR) provides a measure of goodness or fitness of a strategy. SNR is computed via [Phadke89]:

$$SNR = -10 \log_{10} \left[\frac{1}{\mathbf{m}^2} \left(1 + 3 \frac{\mathbf{s}^2}{\mathbf{m}^2} \right) \right] \quad (6)$$

This SNR corresponds to larger-the-better type robust design problem. The term $\frac{1}{\mathbf{m}^2} \left(1 + 3 \frac{\mathbf{s}^2}{\mathbf{m}^2} \right)$ is an approximation of mean square reciprocal quality characteristic, which implies maximization of \mathbf{m} , while minimizing \mathbf{s}^2 . The optimized evaluation function, SNR, corresponds to a strategy that has high probability of success, and that is also robust to changes in the environment (unforeseen events, uncertainty in parameters, etc.).

3.3.5 Selection Function

The fitness evaluation provides a partially ordered set of candidate strategies from the best to the worst. If the termination criteria are not met, successive generations are produced from individuals selected from a partially ordered set. There are several schemes for the selection process: roulette wheel selection and its extensions, scaling techniques, tournament, elitist models, and ranking methods [Houck95].

Holland's roulette wheel [Holland75] is the first and maybe the most popular selection scheme imitating the natural selection. However, traditional roulette wheel limits the evaluation function in a way that it must map the solutions to a fully ordered set of values on \mathfrak{R}^+ . Since SNR is negative in our case, we use the normalized geometric ranking method [Joines94] as follows.

When population is $\{S_i | 1 \leq i \leq N_p\}$, the probability of selecting S_i is defined as:

$$P(\text{select } S_i) = \frac{q(1-q)^{r-1}}{1-(1-q)^{N_p}} \quad (7)$$

where q is a specified probability of selecting the best individual, r is the rank of the individual with the best individual ranked at '1'. The best individual will have a better chance of being selected for reproducing an offspring for the next generation.

3.3.6 Genetic Operators

Mutation and crossover are basic operators to create new population based on individuals in the current generation. Crossover takes two individuals and produces two new individuals, while mutation alters one individual to produce a single new solution [Houck95]. Since our chromosome is a string of integers, we employ the following genetic operators to generate individuals for the new strategy:

$$\text{Uniform mutation: } w'_q = \begin{cases} U(0, r_q) & \text{if the } q^{\text{th}} \text{ gene of the chromosome is selected for mutation} \\ w_q & \text{otherwise} \end{cases} \quad (8)$$

Integer-valued simple crossover generates a random number l from $U(1, N_A)$, and creates two new strategies S'_i and S'_j through interchange of genes as follows:

$$w'_i = \begin{cases} w_i & \text{if } (i < l) \\ w_j & \text{else} \end{cases} \quad w'_j = \begin{cases} w_j & \text{if } (i < l) \\ w_i & \text{else} \end{cases} \quad (9)$$

3.3.7 Termination Criteria

GA runs from one generation to the next, evaluating, selecting and reproducing until a predefined termination criterion is met. Typically, there are three kinds of stopping criteria:

- (i) Define a maximum number of generations and stop at a predefined generation.
- (ii) Stop when the population converges. That is, all the individuals in the population have approximately same fitness function.
- (iii) Stop when there is no distinct improvement in the fittest solution over a specified number of generations.

The fittest strategy at the terminal generation corresponds to the optimized strategy.

4. Illustrative Examples and Results

4.1 Business Scenario

4.1.1 Example Description

Fig.1 is a simplified partial model of a marketing problem faced by a hypothetical company. Suppose the company wants to advertise and promote sales via traditional media marketing, as well as Internet marketing (online promotions). The relevant actions are:

A_1 ---Use Sunday newspaper to advertise products and deliver discount coupons to potential customers;

A_2 ---Promote the URL (Uniform Resource Locator) of the company in the Sunday newspaper. This action is also known as integrated marketing that promotes online clients through traditional media;

A_3 ---Advertise on Company's Website and make coupons available for download and print out.

The marketing and sales divisions of the company must choose the best strategy to achieve a direct goal of redeeming the coupons, as well as having the majority of customers return to company's URL. We define the desired effect D_1 as "Promotional coupons are redeemed" and D_2 as "Customers revisit URL". However, aside from the actions taken by the company, other events such as specific customers' preferences may also significantly affect the desired effects. Define "Customers dislike the promoted products" as exogenous event B_1 . The redeemed coupons may either be the coupons published in

newspapers or coupons downloaded and printed from the website. An intermediate effect E_I is used to depict “Coupons are downloaded from website”. In conclusion, we have three potential actions A_1 , A_2 and A_3 ; two desired effects D_1 and D_2 , an intermediate effect C_1 that transfers the influences from A_2 and A_3 to D_1 ; one exogenous event B_1 , which has certain influence on D_2 .

4.1.2 Experiment Results

Suppose the initial effects (desired or intermediate) are all zeros and we desire $D_1(7)=1$ and $D_2(7)=1$. Devise $\Omega_{A_1} = \{a_1^0, a_1^1\}$, $\Omega_{A_2} = \{a_2^0, a_2^1\}$ and $\Omega_{A_3} = \{a_3^0, a_3^1, a_3^2, a_3^3, a_3^4, a_3^5, a_3^6, a_3^7\}$, where $a_i^0 (i=1,2,3)$ implies no action, a_1^1 corresponds to the action to advertise coupons in the Sunday newspaper, and a_2^1 is the action to advertise URL in the Sunday newspaper. Note that these two advertising actions are valid for the entire week. For the third action, a_3^7 corresponds to online coupons being available at the same time as the URL promotion; a_3^6 has one day delay, a_3^5 has two days of delay, and so on. All the actions are listed in Table IV. Event B_1 (customers dislike the promoted products) is supposed to occur with a probability that is uniform between $[0.2,0.6]$.

Table IV: Potential Actions for the Marketing Problem

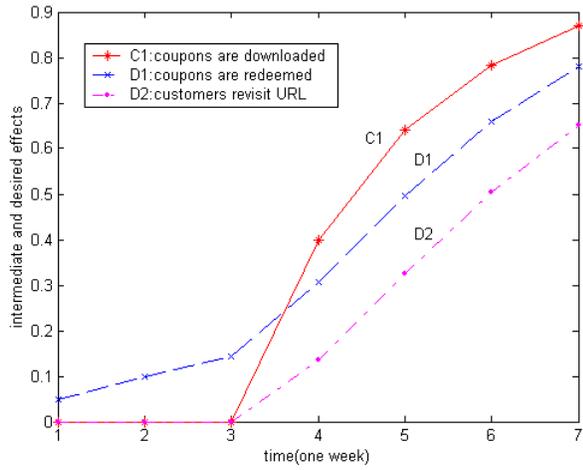
Action	A ₁		A ₂		A ₃							
	a_1^0	a_1^1	a_2^0	a_2^1	a_3^0	a_3^1	a_3^2	a_3^3	a_3^4	a_3^5	a_3^6	a_3^7
t ₁	0	1	0	1	0	0	0	0	0	0	0	1
t ₂	0	1	0	1	0	0	0	0	0	0	1	1
t ₃	0	1	0	1	0	0	0	0	0	1	1	1
t ₄	0	1	0	1	0	0	0	0	1	1	1	1
t ₅	0	1	0	1	0	0	0	1	1	1	1	1
t ₆	0	1	0	1	0	0	1	1	1	1	1	1
t ₇	0	1	0	1	0	1	1	1	1	1	1	1

Consider three strategies: $S_1 = (a_1^1, a_2^1, a_3^4)$, $S_2 = (a_1^0, a_2^1, a_3^7)$, $S_3 = (a_1^1, a_2^1, a_3^7)$. Figs. 6 (a-c) show results of single runs of the DBN with a fixed prior probability for event B_1 : $P(B_1=1)=0.4$. Evidently, the probability of desired effects D_1 and D_2 , as well as the intermediate effect E_I , are functions of time. Since the occurrence of the exogenous event B_1 is random, we generated 100 Monte Carlo runs for these three strategies and computed the joint probability of desired effects $P\{D_1(7)=1, D_2(7)=1|S\}$. Fig. 6(d) shows that the variance of the joint probability may also change with time and clearly the strategy S_3 is the best.

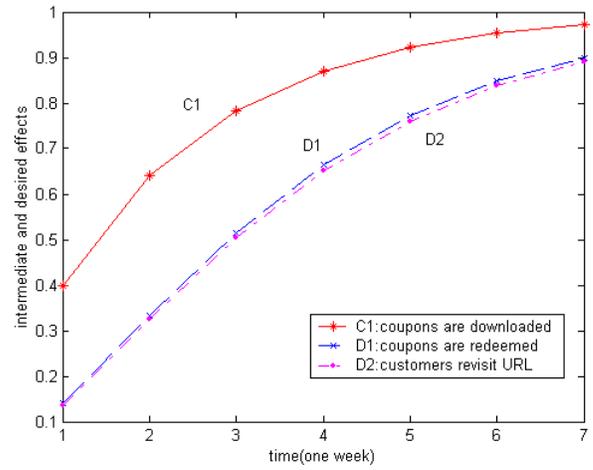
Indeed S_3 is the optimal strategy. The results in Fig. 6(e) were obtained from GA of 20 generations, with each generation having a population of size 10. The sample mean and variance of the joint probability of desired effects for each individual are obtained from 1000 Monte Carlo runs. SNR, as defined earlier, serves as the fitness measurement. Under randomly generated initial populations in Fig. 6(f), Fig. 6(e) shows that the best strategy is $S_{opt} = (a_1^1, a_2^1, a_3^7) = S_3$ and the GA converges in less than 10 generations.

The following conclusions can be made from the results in Fig.6:

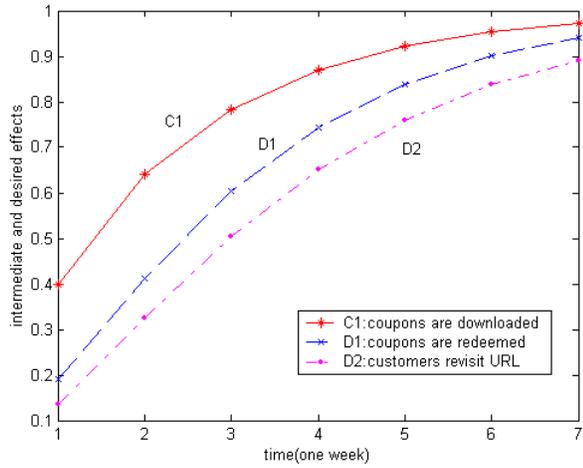
- (i) Since S_3 is substantially better than S_1 , the time to put coupons on website (A_3) cannot lag too much after the advertisement in the newspaper (A_2).
- (ii) Since S_2 and S_3 have similar performance, the benefit of a traditional marketing is very limited. Consequently, action A_1 maybe removed from the action set.



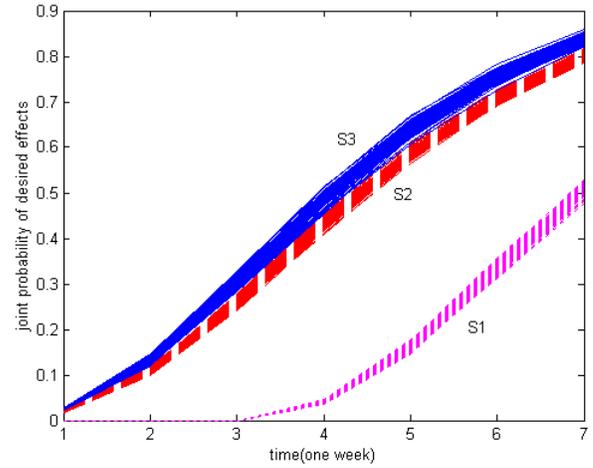
(a) Single Run of S_1



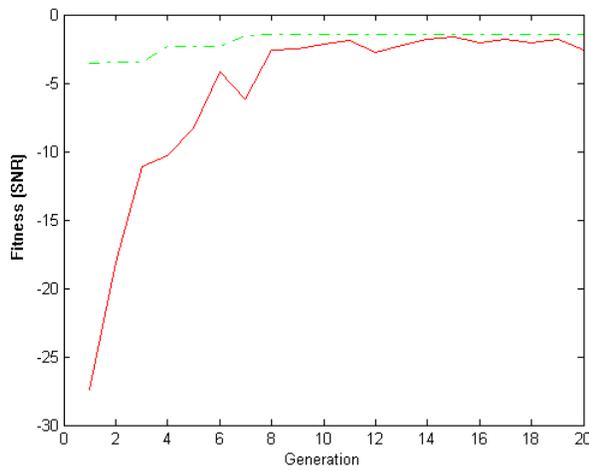
(b) Single Run of S_2



(c) Single Run of S_3



(d) 100 Monte Carlo Runs of S_1 , S_2 and S_3



(e) Strategy Optimization through GA

A1	A2	A3	SNR(DB)
1	1	1	-22.05
0	1	2	-20.64
1	1	5	-3.52
0	1	2	-20.73
1	0	1	-35.12
1	0	1	-34.91
0	1	3	-12.89
0	1	6	-3.55
1	0	0	-100.00
0	1	2	-20.83

(f) Initial Population

Fig.6 Simulation Results

4.1.3 Statistical Analysis

Fig.7 shows a histogram, obtained from 1000 Monte Carlo runs, of the probability of achieving desired effects given strategy S_3 . It can be seen that the histogram is nearly Gaussian which is consistent with the Central Limit Theorem (CLT). In Fig. 8, we plot $P_{00}, P_{01}, P_{10}, P_{11}$ which represent the sample means of $P(D_1(7)=0, D_2(7)=0 | S_{opt}), P(D_1(7)=0, D_2(7)=1 | S_{opt}), P(D_1(7)=1, D_2(7)=0 | S_{opt})$ and $P(D_1(7)=1, D_2(7)=1 | S_{opt})$, respectively. We can see that P_{11} is significantly higher than others. Based on the Gaussian approximation, the following statistical analysis can be performed on the obtained results:

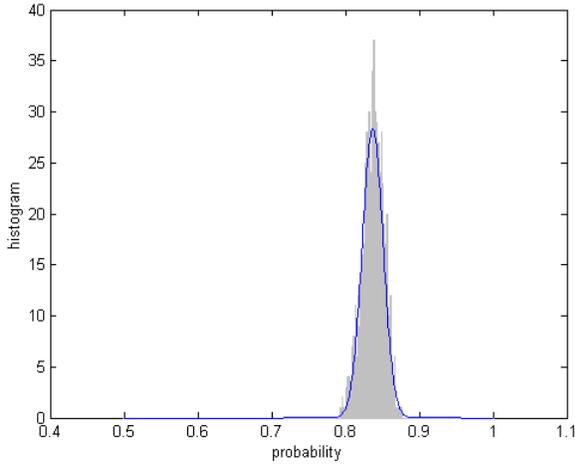


Fig.7 Histogram of 1000 Monte Carlo Runs

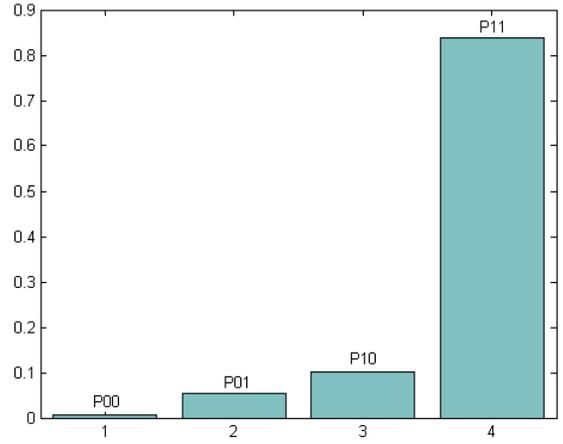


Fig.8 Pdf of Sample Mean

(i) Two-sided Confidence Region

If the sample size is sufficiently large, the two-sided confidence region for the probability of reaching desired effects can be calculated from the sample mean m and sample variance s as:

$$(m - s Z_{\alpha/2}, m + s Z_{\alpha/2}) \quad (10)$$

Here $Z_{\alpha/2}$ denotes the $(1-\alpha)\%$ two-sided probability region for a $N(0,1)$ random variable. With the sample mean $m=0.837$ and standard deviation $s=0.0141$, the 95% confidence region is $[0.8095, 0.8646]$. Thus, given the prior pmfs for the exogenous events and conditional probability tables of the Bayesian network, we can be quite confident that the probability of achieving the desired effects is in the range $[0.8095, 0.8646]$, as illustrated in Fig. 9. A narrower confidence region means better control of the environment.

Fig. 10 shows the propagation of the mean probability of achieving the desired effect D_2 and 95% confidence intervals under strategies $S_2 = (a_1^0, a_2^1, a_3^7)$ and $S_4 = (a_1^1, a_2^1, a_3^6)$. We can conclude from Fig. 10 that different strategies may have very different trajectories and that the confidence regions may also change with time.

In some cases, the confidence regions may overlap with each other for two strategies. In this case, we cannot simply declare one strategy to be superior to another one. Cost of the strategy can be included as a secondary criterion, that is, a strategy with less cost will be preferable to one with a higher cost, even though both may be within the cost budget.

(ii) One-sided Confidence Region

Two-sided confidence region depicts the precision of the predicted probability. Since our purpose is to maximize the probability of achieving the desired effects, another parameter of interest is a lower bound on \mathbf{m} . This results in one-sided probability region as $(\underline{\mathbf{m}}, 1)$, where $\underline{\mathbf{m}} = \mathbf{m} - \mathbf{s} Z_a$. For the above Monte Carlo runs, $\underline{\mathbf{m}} = 0.8139$ for a 95% confidence level. The lower bound tells us that the probability of achieving the desired effects will be no less than 0.8139 with a 95% confidence.

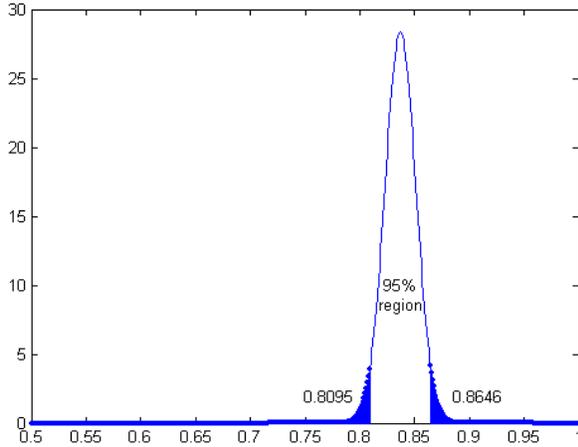


Fig.9 95% Confidence Region

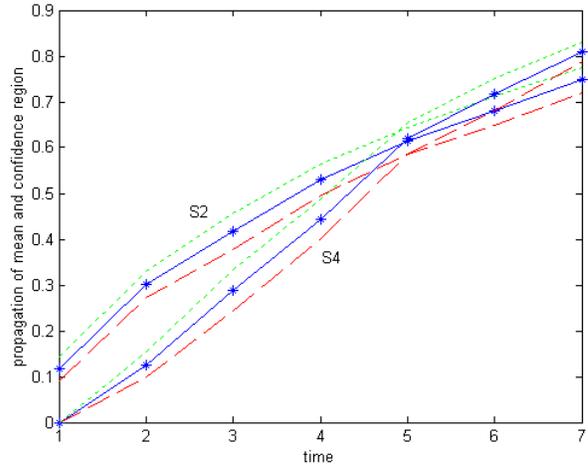


Fig.10 Trajectory of Desired Effect $D_2(t_k) = 1$

(iii) Hypothesis Testing

Suppose the probability of achieving the desired effects is required to be at least \mathbf{m}_0 . Then, the question is: can we accept the results from Monte Carlo Runs? The following binary hypothesis-testing formulation answers this question:

$$\begin{cases} H_0 : \mathbf{m} \leq \mathbf{m}_0 \\ H_1 : \mathbf{m} > \mathbf{m}_0 \end{cases} \quad (11)$$

For a specified tail probability \mathbf{a} , if $\frac{\mathbf{m} - \mathbf{m}_0}{\mathbf{s}}$ exceeds a threshold Z_a , we will reject H_0 and accept that the true value will be higher than \mathbf{m}_0 . Thus, the strategy is acceptable. Otherwise, accept H_0 , that is, the best strategy does not meet our expectation. If the model is credible, the latter result implies that the desired effects are beyond the capability of available actions.

4.2 Military Scenario

4.2.1 Example Description

Friendly forces are assigned to capture a seaport. There is a suitable landing beach with a road leading to the seaport. An approximate concentration of the hostile forces is known from intelligent sources. In addition, friendly intelligence reports that the enemy is using tanks to prevent the infantry advancement along the roads. The mission objective is to capture the seaport, while minimizing the friendly losses due to attrition. Drawing upon intelligence-generated knowledge, the commander identifies the following tactical and operational centers of gravity (COG) that may need to be attacked or defended, as well as other objects of interest whose state will affect the dynamics of the battlespace and the mission outcome: hostile (enemy) air; hostile patrol-boats; hostile tanks; neutral air; neutral patrol-boats; neutral tanks; landing beach and the seaport. This fictitious scenario is shown in Fig. 11. Suppose that the initial environment state at t_0 corresponds to no friendly losses and non-capture of the

seaport, and that the timeline to execute the mission is divided into five time slices (t_1 through t_5). We will measure the mission performance (in term of the joint probability of achieving desired effects) at time t_5 .

The hostile forces are modeled as exogenous events, where: B_1 --- hostile patrol-boats; B_2 --- hostile air; B_3 --- hostile tanks. Each event has an approximate probability, based on intelligence information on the strength of the hostile forces. However, the enemy's decision as to the time at which the enemy uses its forces is unpredictable.

In the same vein, the feasible actions of friendly forces are: A_1 --- neutralize hostile patrol-boats; A_2 --- neutralize hostile air; A_3 --- neutralize hostile tanks; A_4 --- advance to seaport. The feasible actions, along with potential times of their application, are listed in Table V. It is also specified that action A_4 cannot be taken earlier than t_4 due to certain constraints. Since hostile tanks can only be encountered when friendly forces advance to the seaport, the possible times to take action A_3 is t_3 or t_4 .

Desired effects are defined as: D_1 --- capture the seaport; D_2 --- keep friendly losses to a minimum.

The following intermediate effects are designed to connect actions or events to the desired effects: C_1 --- threat from hostile patrol-boats; C_2 --- threat from hostile air; C_3 --- threat from hostile tanks; C_4 --- friendly losses in landing on the beach.

The nodes are interconnected as a Bayesian network. Fig. 11 is the corresponding augmented network by introducing dummy nodes for intermediate and desired effects. The CPTs are listed in Tables VI through XI.

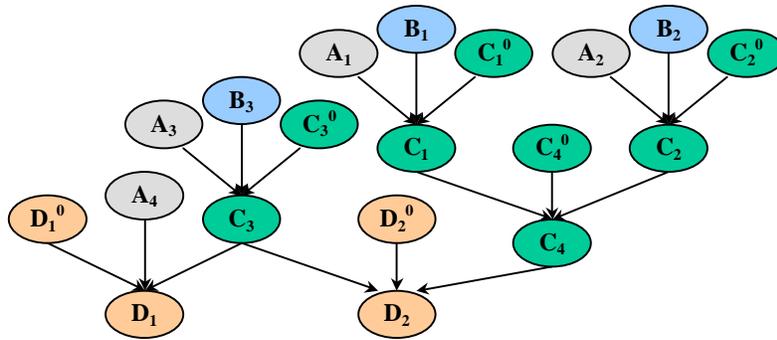


Fig.11 Fictitious Military Scenario

Table V: Potential Actions

Action	A ₁					A ₂					A ₃			A ₄				
	a_1^0	a_1^1	a_1^2	a_1^3	a_1^4	a_1^5	a_2^0	a_2^1	a_2^2	a_2^3	a_2^4	a_2^5	a_3^0	a_3^1	a_3^2	a_4^0	a_4^1	a_4^2
t_1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
t_2	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
t_3	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
t_4	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
t_5	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	1

		No	0.7	0.3
	No	Yes	0.0	1.0
	No	No	0.0	1.0

		No	0.0	1.0
--	--	----	-----	-----

Table VI: CPT for C_1

Parent Nodes			C_1	
C_1^0	B_1	A_1	Yes	No
Yes	Yes	Yes	0.7	0.3
		No	1.0	0.0
	No	Yes	0.7	0.3
		No	1.0	0.0
No	Yes	Yes	0.1	0.9

Table IX: CPT for C_4

Parent Nodes			C_4	
C_4^0	C_2	C_1	Yes	No
Yes	Yes	Yes	1.0	0.0
		No	0.8	0.2
	No	Yes	0.7	0.3
		No	0.2	0.8
No	Yes	Yes	0.9	0.1
		No	0.6	0.4
	No	Yes	0.5	0.5
		No	0.0	1.0

Table VII: CPT for C_2

Parent Nodes			C_2	
C_2^0	B_2	A_2	Yes	No
Yes	Yes	Yes	0.6	0.4
		No	1.0	0.0
	No	Yes	0.6	0.4
		No	1.0	0.0
No	Yes	Yes	0.1	0.9
		No	0.9	0.1
	No	Yes	0.0	1.0
		No	1.0	0.0

Table X: CPT for D_1

Parent Nodes			D_1	
D_1^0	C_3	A_4	Yes	No
Yes	Yes	Yes	0.7	0.3
		No	0.0	1.0
	No	Yes	1.0	0.0
		No	0.0	1.0
No	Yes	Yes	0.5	0.5
		No	0.0	1.0
	No	Yes	0.9	0.1
		No	0.0	1.0

Table VIII: CPT for C_3

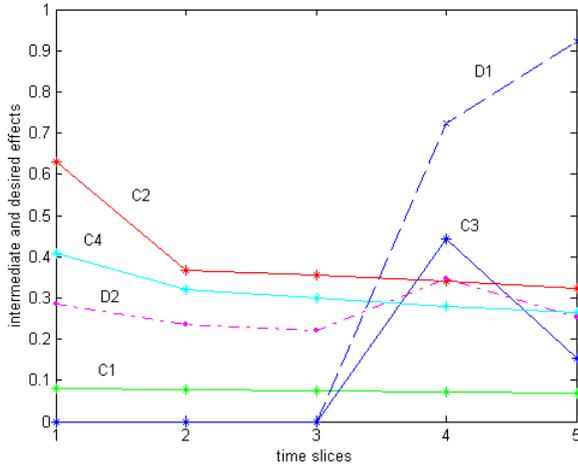
Parent Nodes			C_3	
C_3^0	B_3	A_3	Yes	No
Yes	Yes	Yes	0.3	0.7
		No	1.0	0.0
	No	Yes	0.3	0.7
		No	1.0	0.0
No	Yes	Yes	0.1	0.9
		No	0.5	0.5
	No	Yes	0.0	1.0
		No	0.0	1.0

Table XI: CPT for D_2

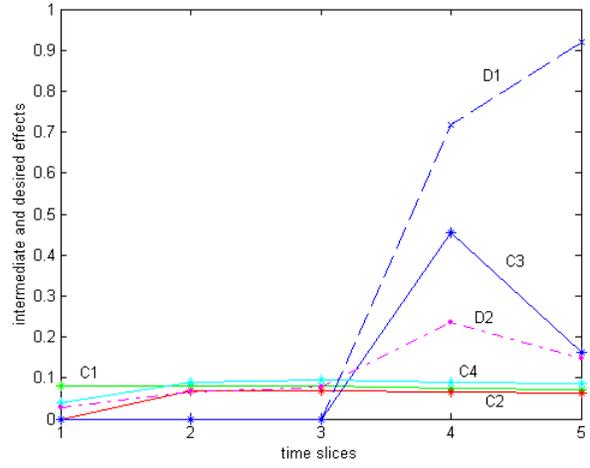
Parent Nodes			D_2	
D_2^0	C_4	C_3	Yes	No
Yes	Yes	Yes	1.0	0.0
		No	0.5	0.5
	No	Yes	0.3	0.7
		No	0.2	0.8
No	Yes	Yes	0.9	0.1
		No	0.7	0.3
	No	Yes	0.2	0.8
		No	0.0	1.0

4.2.2 Simulation Results

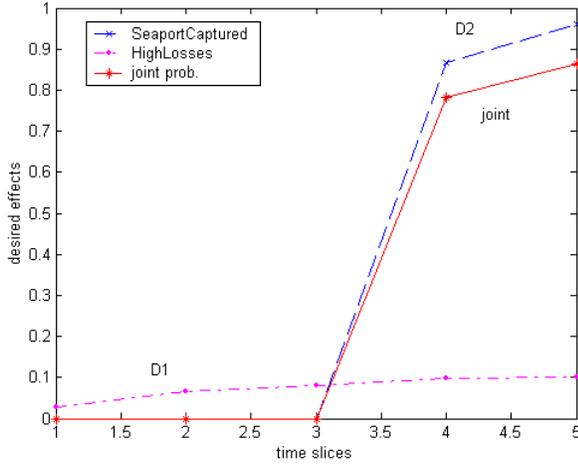
Since the events may happen at arbitrary times, the problem is changed from one of searching for an optimal strategy to that of finding a set of decision rules, that is, given a possible combination of events, which strategy will maximize the probability of achieving the desired effects. Consider two cases: (i) Friendly forces encounter both threats from hostile air and hostile patrol-boats at time t_1 . Whenever friendly forces advance to the seaport, the hostile tanks will defend immediately; (ii) Friendly forces encounter hostile air at time t_1 and encounter hostile patrol-boats at time t_2 ; the hostile tanks will act as in case (i). The results from these two cases under strategy $S_1 = (a_1^1, a_2^2, a_3^2, a_4^1)$ are illustrated in Fig.12 (a) and Fig.12 (b), respectively. In this scenario, we assumed that the intelligence sources are reliable, and the action probabilities of hostile forces are: $P\{B_1 = 1\} = 0.8$, $P\{B_2 = 1\} = 0.7$, $P\{B_3 = 1\} = 0.8$.



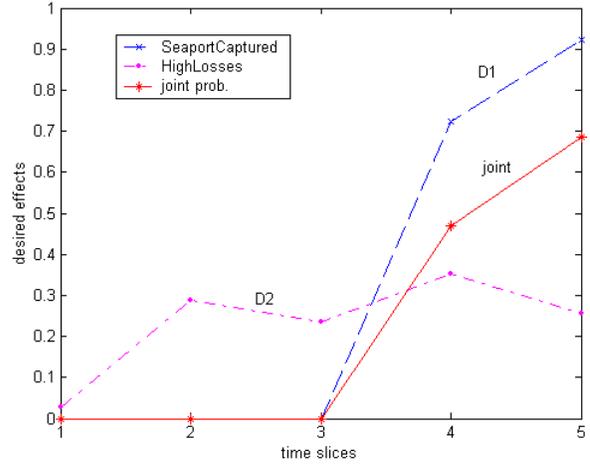
(a) S_1 for case (i)



(b) S_1 for case (ii)



(c) S_2 for case (ii)



(d) S_3 for case (ii)

Fig.12 Simulation Results

Since hostile air and hostile patrol-boats are separately encountered in case (ii), the landing beach will be under a moderate threat. On the other hand, in case (i), the combination of two events may put the friendly forces in the landing beach under severe threat due to the infeasibility of simultaneously dealing with both threats. Thus, the friendly losses will be higher in case (i).

Now, we focus on case (ii) to see which action strategy will be better. Comparing S_1 with $S_2 = (a_1^1, a_2^2, a_3^1, a_4^1)$ and $S_3 = (a_1^1, a_2^3, a_3^2, a_4^1)$, we can see from Figs.12 (b-d) that S_2 is the best among these three strategies because all the hostile forces are immediately neutralized. As a consequence, the friendly losses due to attrition are low. The solid lines in Figs. 12 (c-d) depict the joint probability of achieving both of the desired effects: $P\{D_1(t_k)=1, D_2(t_k)=0\}$. Fig.13 is the result from genetic algorithm, where we use $P\{D_1(5)=1, D_2(5)=0\}$ as a fitness measurement. Indeed, S_2 is the optimal solution from GA.

Additionally, we consider a scenario where the data from intelligence sources is noisy. We model this by assuming that the concentrations of the hostile forces are random. We suppose $P\{B_1(t_1)=1\} = P_1, P\{B_2(t_2)=1\} = P_2, P\{B_3(t_4)=1\} = P_3$, where P_l is uniformly distributed between $[0.6, 1]$,

P_2 is uniformly distributed between $[0.5, 0.9]$ and P_3 uniformly distributed between $[0.7, 0.9]$. Results of $P\{D_1(5) = 1, D_2(5) = 0 \mid S_2\}$ from 1000 Monte Carlo runs are shown in the histograms of Fig. 14, with the Gaussian distribution superimposed. The sample mean and standard deviation are 0.8641 and 0.0089, respectively. The two-sided 95% confidence region of this strategy is $(0.8467, 0.8816)$.

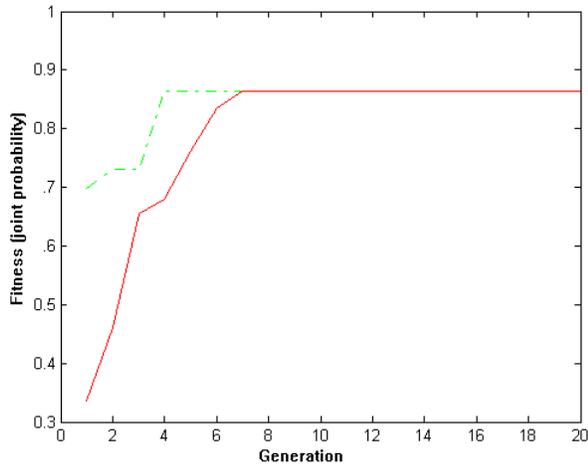


Fig.13 Strategy Optimization through GA

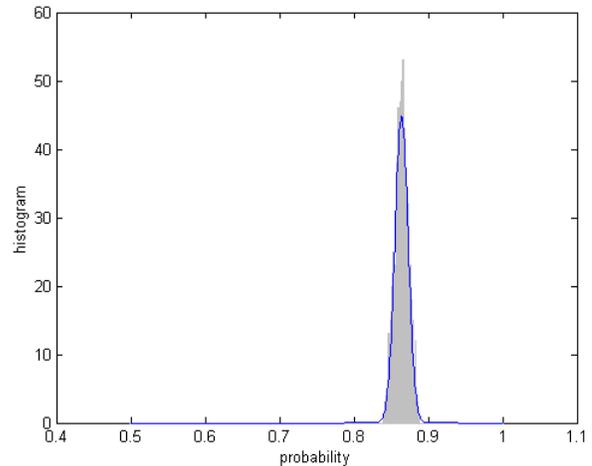


Fig.14 1000 Monte Carlo Runs for S_2

5. Conclusions and Future Work

This paper introduced a general methodology, based on an integration of dynamic Bayesian networks and the genetic algorithms, to optimize strategies for offline decision support. DBN is used for evaluating the probability of achieving desired effects for a given strategy, while GA is applied to search for the optimum solution in a relatively large solution space. Since uncertainty is unavoidable in military as well as business applications, the desired effects indeed are random processes. As a consequence, Monte Carlo runs and probabilistic analysis are employed to determine an action strategy that trades off goodness and robustness. The main contributions of this paper are: the use of DBN to compute time-dependent probability propagation for desired effects; use of GA to optimize action strategies; introduction of signal-to noise ratio (SNR) as a measure of robustness of a strategy in an uncertain environment.

The methodology can be extended to more realistic scenarios. In our examples, we assumed that CPTs are known and time-invariant. When CPTs are elicited from many experts, they may not always be consistent with each other. In this case, we randomize CPTs in Monte Carlo runs. If CPTs are time-varying, the only change needed is to update the CPTs with time.

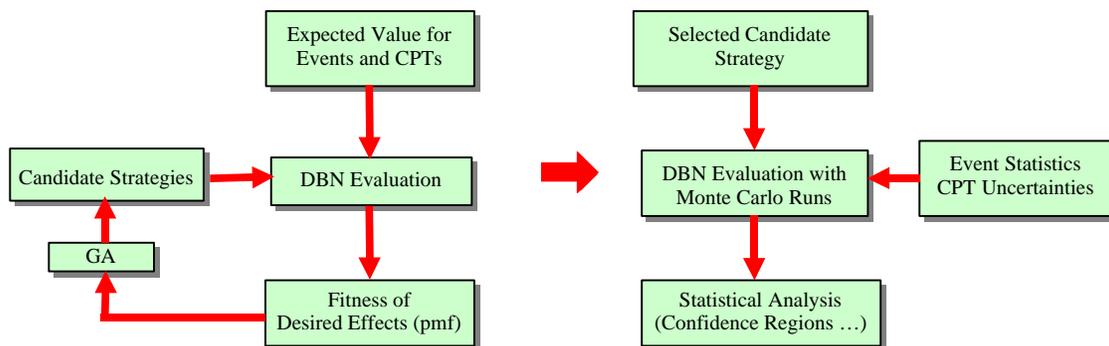


Fig.15: Alternative Optimization Approach

Both GAs and DBN are computationally expensive. Consequently, this method can be applied offline in the planning phase. If computational time is of concern, an alternative optimization approach is shown in Fig. 15. Suppose the expected values of the uncertain prior probability of events and CPTs are known. Then, we will avoid the Monte Carlo runs in the GA loops. This may be advisable because most of the strategies in the solution space tend to be inferior. Once a strategy is selected, further Monte Carlo analysis may be conducted on the optimized strategy only.

In some applications, different desired effects may have different priorities. Using the Gaussian approximation, suppose the probabilities of achieving desired effects D_1, D_2, \dots, D_{N_D} are conditionally independent random variables (nodes without direct connections in a Bayesian network). Define mean vector $\mathbf{m} = (\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{N_D})$ as the expected values for the probabilities of achieving desired effects and let the corresponding covariance matrix $R = \text{diag}[\mathbf{s}_1^2, \mathbf{s}_2^2, \dots, \mathbf{s}_{N_D}^2]$. In this case, we can use a weighted SNR to measure the goodness of a strategy as in the following equation:

$$SNR = -10 \log_{10} \left\{ \sum_{i=1}^{N_D} \frac{\mathbf{v}_i}{\mathbf{m}_i^2} \left(1 + 3 \frac{\mathbf{s}_i^2}{\mathbf{m}_i^2} \right) \right\}, \quad \sum_{i=1}^{N_D} \mathbf{v}_i = 1 \quad (12)$$

In addition to its application in offline strategy planning, the methodology introduced in this paper may be used for strategy execution phase. The strategy obtained by the offline planning is open-loop in that it is an action sequence based on the current forecast of future events [Bertsekas95]. However, in the strategy execution phase, the strategy can be made open-loop feedback optimal based on observed events and intermediated effects. The process works as follows:

- (i) Prune the nodes which have no relevance for future effects given current observations;
- (ii) Adjust the model parameters to conform with the current environment;
- (iii) Optimize the strategy using the methodology of the paper.

Acknowledgements

This work was supported by the Office of Naval Research under contract # N00014-00-1-0101. The authors would like to thank Kevin Murphy from UC Berkeley and Michael G. Kay et al from the North Carolina State University for providing Matlab-based Bayesian network toolbox and Genetic Algorithms, respectively.

Reference:

- [Aliferis96] Aliferis, C.F. and Cooper G.F. *A Structurally and Temporally Extended Bayesian Belief Network Model: Eefinitions, Properties and Modeling Techniques*. Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence. UAI-96. Portland, Oregon: Morgan Kaufmann.
- [Barrientos98] Barrientos, M.A. and Vargas J.E.. *A Framework for the Analysis of Dynamic Processes Based on Bayesian Networks and Case-based Reasoning*. Expert Systems with Applications. 15, 1998.
- [Bertsekas95] Bertsekas, D.P.. *Dynamic Programming and Optimal Control*. Athena Scientific 1995.
- [Boutilier98] Boutilier, C., Dean, T. and Hanks S.. *Decision Theoretic Planning: Structural Assumptions and Computational Leverage*. Sept. 15, 1998.
- [Davis01] Davis, P.K.. *Effects-based Operations*.
<http://www.rand.org/contact/personal/pdavis/davis.online.html>
- [D'Ambrosio99] D'Ambrosio, B.. *Inference in Bayesian Networks*. AI Magazine. Vol.20, No.2, 1999.

- [Emery65] Emery, F.E. and Trist, E.L.. *The Causal Texture of Organizational Environments*. Human Relations. 18, 1965.
- [Heckerman95] Heckerman, D.. *Causal Independence for Probability Assessment and Inference Using Bayesian Networks*. IEEE Transactions on Systems, Man & Cybernetics, Part A (Systems & Humans), Vol. 26, No. 6.
- [Holland75] Holland J.. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor. 1975.
- [Houck95] Houck, C., Joines J., et al. *A Genetic Algorithm for Function Optimization: A Matlab Implementation*. NCSU-IE TR 95-09, 1995.
- [Joines94] Joines J. and Houck C. *On the Use of Non-Stationary Penalty Functions to Solve Constrained Optimization Problems with Genetic Algorithms*. In 1994 IEEE International Symposium Evolutionary Computation. Orlando.
- [Jordan99] Jordan, M.I.. *Learning in Graphical Models*. MIT Press 1999.
- [Kanazawa95] Kanazawa, K., Koller, D. and Russell S.. *Stochastic Simulation Algorithms for Dynamic Probabilistic Networks*. Proc. of the 11th Annual Conference on Uncertainty and Artificial Intelligence. 1995.
- [Leblebici81] Leblebici H. and Salancik G.R.: *Effects of Environmental Uncertainty on Information and Decision Processes in Banks*. Administrative Science Quarterly. 26, 1981.
- [Madigan96] Madigan, D., Raftery A.E., et al. *Bayesian Model Averaging*. Proc. AAAI Workshop on Integrating Multiple Learned Models. Portland. OR. 1996.
- [McCrabb01] McCrabb, M.. *Uncertainty, Expeditionary Air Force and Effects-Based Operations: Concept of Operations for Effects-based Operations*.
- [Microsoft] <http://research.microsoft.com/adapt/MSBNx/>
- [Murphy] Murphy, K. <http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html>
- [Nettica] <http://www.norsys.com>
- [Pearl88] Pearl, J.. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA, 1988.
- [Phadke89] Phadke, M.S.. *Quality Engineering Using Robust Design*. Prentice Hall 1989.
- [Russell95] Russell, S. and Norvig, P.. *Artificial Intelligence, A Modern Approach*. Prentice Hall. 1995.
- [Stender94] Stender, J., Hillerbrand, E. and Kingdon, J.. *Genetic Algorithms in Optimization, Simulation and Modeling*. IOS Press. 1994.
- [Tawfik00] Tawfik, A.Y. and Neufeld E.. *Temporal Reasoning and Bayesian Networks*. Computational Intelligence. Vol. 16, No. 3, August 2000.