Bayesian Network Models for Generation of Crisis Management Training Scenarios

Eugene Grois, William H. Hsu, Mikhail Voloshin, David C. Wilkins

Beckman Institute
University of Illinois at Urbana-Champaign
{e-grois | w-hsu | voloshin | dcw}@uiuc.edu

Abstract

We present a noisy-OR Bayesian network model for simulation-based training, and an efficient search-based algorithm for automatic synthesis of plausible training scenarios from constraint specifications. This randomized algorithm for approximate causal inference is shown to outperform other randomized methods, such as those based on perturbation of the maximally plausible scenario. It has the added advantage of being able to generate acceptable scenarios (based on a maximum penalized likelihood criterion) faster than human subject matter experts, and with greater diversity than deterministic inference. We describe a field-tested interactive training system for crisis management and show how our model can be applied offline to produce scenario specifications. We then evaluate the performance of our automatic scenario generator and compare its results to those achieved by human instructors, stochastic simulation, and maximum likelihood inference. Finally, we discuss the applicability of our system and framework to a broader range of modeling problems for computer-assisted instruction.

1 Introduction

Probabilistic networks are used extensively in diagnostic applications where a causal model can be learned from data or elicited from experts [He90, HW95]. In this paper, we first present a formulation of *scenario generation* in computer-assisted instruction as a Bayesian inference problem. Previous approaches to synthesis of training scenarios have predominantly been dependent on the expertise of a human instructor [GD88, GFH94]. The inferential model alleviates this problem by mapping constraints specified by an instructor to a causal explanation in the network model. These constraints are events in the causal model (such as a firemain rupture in training systems for damage control and fire fighting, or a medical complication in training for surgical anesthesiologists) [GD88, De94]. Constraints are literals in a formal specification language for training objectives to be met by a scenario. Explanations, the specification language, and objectives are defined in Section 2.

We then give an efficient randomized algorithm for finding plausible scenarios that meet training objectives, have well-differentiated explanations, are unique, and are diverse. We define a penalized likelihood function (based upon precision and accuracy in meeting training criteria as well as pure likelihood). The output of the algorithm is a set of explanations that maximize this function subject to the specification. These explanations, which we call *proposed scenarios*, determine the fixed parameters for interactive training scenarios (including the initial conditions for dynamic simulation).

Finally, we show that, subject to quantitative metrics for the above criteria, both the acceptance rate of our algorithm is high and the efficiency (number of acceptable scenarios divided by the computational cost of generating all proposed scenarios) is high. We evaluate the system using the penalized likelihood function, relative to naive and perturbation-based stochastic scenario generators. We also study the empirical performance of the system – that is, how its relatively high acceptance rate and asymptotic efficiency allow it to compete effectively with human instructors in designing scenarios. We report on a deployed training system and quantitatively compare the scenarios produced by our system to those applied and tested in the classroom setting.

1.1 Role of Scenario Generation in Crisis Management Training

Immersive Training Systems. It is well known that effective human performance on complex tasks requires deliberate practice under realistic conditions [EKT93]. To provide realistic practice scenarios in an interactive setting, *immersive training systems* are often used [GD88, WFH+96]. In military and medical professions, these systems have evolved in recent years from ad-hoc physical mock-ups to virtual environments with a high degree of automation and standardization [De94, GFH94, WFH+96]. A common aspect of many crisis management tasks is time-critical decision making, requiring a model of problem-solving actions on the environment [HB95, WFH+96]. In this project, we concentrate on specification of simulation parameters by instructors, rather than the later stage of interaction with the trainee.

The effectiveness of computer simulation for training and critiquing has been demonstrated by deployed systems in many high-risk domains, such as surgical anesthesia and ship damage control [HGYS92, WFH+96]. Similar diagnostic models have been designed to support intelligent displays for space shuttle flight control, and for medical monitoring [HB95, HLB+96]. To assess and improve the decision-making skills of future practitioners in time-critical situations, an interactive simulation is typically used to produce a dynamic model of the crisis. A graphical user interface and a visualization or multimedia system (or a physical mock-up with appropriate sensors and actuators) are used to deliver the scenario. Intelligent control of the multimedia components of this system is required throughout the life cycle of a training scenario, from design through deployment to the critiquing and feedback stage of professional education [WFH+96]. Such an ensemble provides an adequate simulation of genuine crisis conditions to evoke realistic stress levels during problem solving [BDS96].

The purpose of scenario generation in this framework, therefore, is to provide collaborative assistance with human instructors to design useful models for dynamic simulation.

Computer-Assisted Instruction (CAI) Issues. We now explain our notion of useful models for dynamic simulation. Much of the research and development effort in computer-assisted instruction (CAI) for crisis management has gone into easing the design burden for simulations of time-critical decision making under stress [BDS96]. This design task can be isolated into the offline *specification* phase (where training objectives are mapped into fixed simulation parameters and initial conditions) and the online *delivery* phase (where interactive simulation occurs). Our research has shown that uncertain reasoning is useful not only in the second phase (where the trainee may interact with intelligent agent models and receive critiquing from an expert system), but in the first (where instructor objectives are used to stochastically generate scenarios) [MW96].

For example, training objectives in medical applications (e.g., complications to be handled by the specialist-in-training) may be achieved by various causes (preexisting conditions, surgical incidents, *etc*.) [GFH94]. In our deployed application (training for shipboard damage control), the objective is to train officers in responding to loss of critical ship functions (maintaining combat readiness and “ship survivability” while minimizing casualties from the damage control process). The specification tool is a graphical probabilistic model of functional dependencies in the ship. Search-based inference is used to determine which causes (damage-initiating events, such as a mine detonation, missile impact, or ignition of a shipboard fire) best explain the *specified* deactivations. For CAI purposes, the entire explanation (not just the “diagnosis”) is useful to instructors, because it aids in plan recognition and the generation of critiques during and after the online simulation phase [De94, BDS96, WFH+96].

1.2 The DCTrain System

In the naval domain, typical damage control scenarios involve explosions, mass conflagrations, flooding, vital system failure, and other types of disaster. A special officer, called the Damage Control Assistant (DCA) is responsible for coordinating all crisis response and recovery efforts. Because real-life major crises on US Naval vessels are extremely rare, DCAs seldom get exposure to serious crisis situations, and as a result find it difficult to obtain the expertise to deal with them if they ever do occur. Training runs on actual ships are extremely expensive, time-consuming, fatiguing, not to mention dangerous – and so are kept to a minimum. As a result, the Navy has identified an acute need for an inexpensive, but effective computerized damage control training system.

In response to this need, the Knowledge-based Systems Group at the University of Illinois has developed the Illinois DCA Training System (DCTrain), a framework of four essentially independent modules that interact with the user to provide an immersive environment for damage control training. Figure 1 shows a diagram of the complete DCTrain system. The Ship Crisis Simulator module comprises a comprehensive numerical/rule-based simulation of ship processes, system functions, and crew actions. The Multimedia/Visualization module implements a powerful interface for user-system interaction. The Critiquing module provides performance feedback to the user. The proposed scenario generation module, utilizing the ScenGen algorithm, is designed to fit into the existing framework. Its task is to design a training scenario in accordance with prescribed training objectives, and guide the simulator through that scenario configuration. As a whole, the system constitutes a comprehensive training tool.

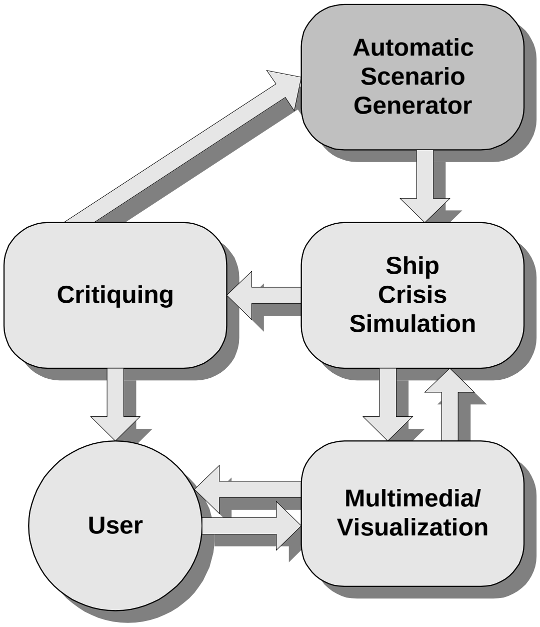


Figure 1. Overview of the Illinois DCA Training System (DCTrain)

Due to its dynamic simulation engine [WFH+96], DCTrain boasts the capability to produce numerous distinct scenarios on demand. A powerful critiquing module, combined with a state-of-the-art multimedia interface provide real-time, and post-mortem context-specific feedback to the user.

In May 1997, DCTrain was tested at the Navy's Surface Warfare Officers' School (*SWOS*) against an existing system called IDCTT [De94]. IDCTT relies on a set of pre-programmed scenario scripts, and as a result offers only one damage control scenario. It also has very limited critiquing capability.

Performance results collected from the May '97 graduating class of DCAs at SWOS clearly indicate the superiority of DCTrain over IDCTT. Exposure to multiple scenarios, and the benefit of expanded critiquing led to significant performance gains in crisis management.

We believe that the addition of an automated scenario generation module to the DCTrain system will improve its training effectiveness yet further. The ability to not only produce scenarios on demand, but to quickly and easily tailor them to specific training requirements, or even to the needs of individual students will provide users with an unprecedented degree of flexibility.

In the following section we give a formal model for scenario representation, and in section 3 we present the ScenGen algorithm for automated scenario generation.

2 A Bayesian Network Model of Training Scenarios

Given that a probabilistic model is needed to map from training objectives to explanations (including placement of both initial and timed events), we seek a diagrammatic representation of the scenario that permits efficient inference. We exploit the property that crisis training scenarios are similar to diagnostic models, and can be *constructed* in a simplified Bayesian network form.

Formal Model: Noisy-OR Bayesian Networks (NOBNs) *Noisy-OR Bayesian networks* (NOBNs) are Bayesian networks which observe the properties of *accountability* and *exception independence*, as defined by Pearl [Pe88]. Accountability is simply a negation-as-failure property for probabilistic models. Specifically, for Bayesian networks with boolean variables, an event is presumed false (i.e., has zero probability) if all its stated causes are. In NOBNs, accountability is achieved using a "leak node" (or "miscellaneous cause") to circumscribe every non-root event [Pe88, RN95]. This is expressed for vertices x with parents $\pi_i(x)$ as:

$$P(x) = 0 \text{ if for all } i, P(\pi_i(x)) = 0 \tag{1}$$

Define an *inhibitor* as a mechanism that mediates the conditional probability of an event given a particular cause (i.e., prevents the effect given the cause). *Exception independence* requires every inhibitor of an event under one cause to be independent of the inhibitors of the event under all other causes. Inhibitors are explicitly represented as events in some graphical models [Pe88, HW95], but it suffices to express them as *noise parameters*, q_i [RN95]:

$$q_i = P(\neg x \mid \pi_i(x) \wedge \forall j \neq i, \neg\pi_j(x)) \tag{2}$$

$$P\left(x \mid \pi'_1(x),\ldots,\pi'_t(x),\neg\pi''_1(x),\ldots,\neg\pi''_f(x)\right)=1-\prod_{i=1}^{f}\left(1-P(x \mid \neg\pi''_i(x)\right) \tag{3}$$

The benefits of NOBNs are as follows:

1. Converting from a general Bayesian network to an NOBN achieves a best-case exponential reduction in the number of interactions. This reduction yields a commensurate increase in inferential efficiency and is due to linearization of conditional probability tables, which require 2^k entries in a general form Bayesian network for a vertex with k parents).
2. NOBNs can be factored for an empirical speedup of 3-4 times [HH97].

The exponential speedup has been realized in real-world diagnostic Bayesian networks – most famously the *CPCS-BN* knowledge base for liver and gall bladder diseases, which is still among the largest Bayesian networks in use [PPMH94, HH97].

2.2 Construction of NOBN Models

The NOBN models used for scenario generation in the ship damage control domain are constructed from diagrams called *damage control plates*, for U.S. Navy ships (e.g., the DDG-51 and DDG-53). Subjective probability assessments were elicited from subject matter experts such as former damage control assistants and instructors at *SWOS*. These instructors were also consulted for compilation of the benchmarking scenarios produced by human experts. Some low-level assessments of probabilistic dependencies (generic events such as water main ruptures, combustion, and electrical deactivation) were contributed by the authors after consultation with domain experts.

Language of Training Objectives. A training objective is a task or skill at which the student is to become proficient (e.g., managing limited water resources for fire fighting in shipboard damage control). In accordance with established training methodology, an instructor selects desired training objectives, and then designs a scenario composed of a series of crisis events requiring application of the desired skills.

An automated scenario generation system can mimic this approach. Training objectives may either be provided by a human instructor, or suggested by an intelligent critiquing module. The system can then choose applicable keyed events from a knowledgebase, and use them to dynamically construct a scenario. Because of the causal nature of event interactions, we prefer to represent such a knowledgebase as a belief network, and due to the efficiency arguments given above, specifically as a NOBN.

In our framework, *constraints* are literals in a monotone disjunctive normal form (DNF) specification language. Each primitive training objective, O_i, denotes one conjunct of constraints c_{ij}. A scenario specification S is the top-level disjunction of all training objectives:

$$S = \vee_{i=1}^{k}\left(O_i\right) = \vee_{i=1}^{k}\left(\wedge_{j=1}^{n_i}\left(c_{ij}\right)\right)$$

This representation allows composite training objectives simply because the disjunction is implicit. When multiple primitive objectives are specified, candidate scenarios can be generated one objective at a time. The results are filtered to obtain acceptable components, then composed by set union.

This achieves three benefits:

- The scenario generation problem is reduced to an offline computation requiring much less interaction with a human designer (i.e., many alternative scenarios can be generated and a criterion applied to reject implausible, unsound, incomplete, and duplicate scenarios).
- Using NOBNs as the probabilistic model for scenario generation reduces empirical complexity. Furthermore, there exists an efficient approximate inference algorithm that produces plausible yet diverse explanations, as is shown in section 3.
- The NOBN model makes stochastic search with a *penalized* likelihood function simple and efficient. For such an approximate inference algorithm, plausibility need not be the sole criterion: precision and accuracy can be assigned quantitative credit as appropriate.

In our implementation, we exploit the first two points using our search algorithm and apply a rejection test as the penalty function. This helps to standardize our evaluation metric, because calibration of a penalty function is somewhat subjective.

3 The ScenGen Algorithm

We have demonstrated how the problem of generating a scenario can be formulated as search-based inference in an NOBN. The deterministic algorithms we discuss in this section produce most probable explanations, which correspond to maximally plausible scenarios. To generate *non-trivially* diverse scenarios (those that are not effectively redundant for training purposes), however, we must relax our maximum likelihood requirement. To meet instructor objectives through inference, we must also *sample from the posterior distribution* of scenarios, given specifications. We now describe our stochastic search-based algorithm.

In order to guide its search through the Bayesian network, our algorithm employs a heuristic which computes the *combined most probable path* (CMPP) from the given set of constraint vertices to a root vertex. For such a set of constraint vertices and a root vertex in a network, the CMPP is composed of the set of *most probable paths* (MPPs), one for each constrain-root pair. This is an admissible heuristic because the CMPP is an overestimate of the actual probability from the constraints to the root. In other words, it is an *underestimate* of the *cost* of any such path . Combining stochastic branch-and-bound with our CMPP heuristic produces a search based-inference algorithm for our NOBN encoding of a scenario design space.

Figure 2 shows a generalized noisy-OR Bayesian network. In this figure, the vertices X and Y are constraints, and vertices R_1 through R_4 are initial events (roots). A parent set consists of one parent for X and one for Y (e.g., {P1(X), P3(Y)}, {P3(X), P3(Y)}). The first part of the search iteration begins by computing the CMPPs for each parent set of the current constraints. Those parent sets resulting in a CMPP probability of 0 are immediately discarded, since they are guaranteed to lead to a dead end. Parent sets resulting in a CMPP probability falling below some predetermined threshold (e.g., one standard deviation from the maximum) are discarded as well, since they will not lead to very likely (realistic) scenarios. The remaining parent sets result in CMPPs with high enough probabilities as to be considered “plausible”.

The second part of each iteration involves stochastically selecting from among the “plausible” parent sets. Which ever parent set is thus chosen becomes the new constraint set.

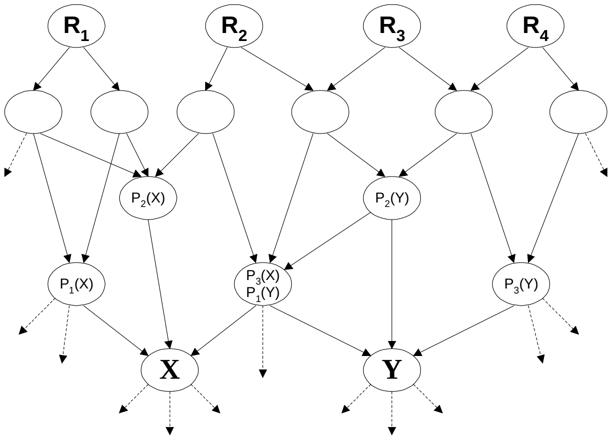


Figure 2. Generalized noisy-Or Bayesian network.

4 Evaluation

In this section we present experimental results from evaluating ScenGen against three competing algorithms.

4.1 Competing Scenario Generation Algorithms

We evaluate the ScenGen algorithm by direct comparison to three alternative scenario generation techniques:

1. Manual design by human subject matter experts (*MDHE*).
2. Naïve random generation (*NRG*).
3. Case-based stochastic perturbation (*CBSP*).

We take scenarios designed by human subject matter experts as our point of reference for scenario quality (i.e., completeness, soundness, and plausibility). Since, in the final analysis, a human expert is the best judge of a given scenario's quality it seems logical to assume that this expert will produce the best possible scenarios. Efficiency, however, will not be very high since the time a human subject matter expert takes to design such a scenario will be orders of magnitude higher than that of even the most expensive computer-based method.

Randomly instantiated scenarios are created by selecting a random initial event (which corresponds to a root node in the NOBN), and then evaluating the NOBN by propagating forward in the graph. This technique is cheap, but in general the quality of most scenarios produced will be poor. Also, efficiency will be low due to the large number of useless scenarios.

Case-based stochastic perturbation is a hybrid of the first two methods. It involves using scenarios designed by human subject matter experts as seeds, and perturbing them so as to obtain variability. This technique has the capability of producing a large number of admissible scenarios by derivation from the human-designed seeds, and can be expected to have reasonably good scenario space coverage due to the stochastic perturbation with respect to the seed. Furthermore, because once the seeds are obtained, the entire process is automated, efficiency should be better than for human experts. However, the downside is that the stochastic nature of the approach is likely to result in at least some unacceptable scenarios, which will lead to a lowering of efficiency.

4.2 Evaluation Criteria

We evaluate the ScenGen algorithm by comparing its performance to that of three other scenario generation techniques. The performance of an algorithm is considered to be the *efficiency* with which it generates *quality* scenarios. There are four metrics that are used to evaluate the quality of a scenario:

1. Empirical Accuracy (*EA*)
2. Empirical Precision (*EP*)
3. Plausibility (*P*)
4. Variability (*V*)

Empirical Accuracy measures the degree to which *all* specified learning objectives are satisfied. The inability of a scenario generation technique to guarantee the satisfaction of all requested learning objectives greatly diminishes its usefulness.

Empirical Precision measures the degree to which *only* specified objectives are satisfied. It has been shown that learning takes place most efficiently when training material is presented in a planned and controlled [GFH94]. For this reason, the generation of "sloppy" training scenarios (i.e., those addressing extraneous learning objectives) is likely to confuse the student, and complicate the learning process.

The Plausibility criterion evaluates the realism of the scenario. The ScenGen algorithm is intended for complex, real-world domains, which generally result in a vast scenario space. Because we define a learning objective simply as a disjunction of event sets, numerous collections of events may satisfy a given learning objective or set of learning objectives. However, not necessarily all of these collections of events are probable or even possible. Thus, we require that a "quality" scenario be *plausible*.

Variability measures how well a given technique can generate *distinct* scenarios. Two scenarios are distinct if they differ in at least one event. Variability is an important criterion because the generation of duplicate scenarios is wasteful of resources, and thus undesirable. We prefer techniques that minimize or even avoid duplication.

The Quality criterion is composed of these four metrics. Formally, the Quality criterion is defines as follows:

Definition Quality criterion. A quality scenario satisfies all specified objectives, and only the specified objectives. It is plausible, and distinct from any other scenario previously generated.

Each time a scenario is generated, it is checked for consistency with the Quality criterion by a series of four rejection tests, corresponding to the four metrics outlined above. Any scenario not passing all four of the rejection tests is deemed unacceptable, and discarded. Those scenarios found to be consistent with the Quality criterion are considered to be "quality" or acceptable scenarios.

Definition Performance. The performance of a scenario generation algorithm is the efficiency with which it generates quality scenarios.

We evaluate the performance of ScenGen by comparing its efficiency in generating quality scenarios with those of competing algorithms. We calculate Efficiency as follows:

Let

n_a	=	number of admissible scenarios
n_t	=	total number of scenarios generated
P_{acc}	=	acceptance probability
C_a	=	average cost of generating one admissible scenario
C_t	=	total cost of generating all scenarios

P_{acc} can be defined as

$$P_{acc} = \frac{n_a}{n_t} \tag{4}$$

We now calculate C_a as follows:

$$C_a = \frac{C_t}{n_t} \times \frac{1}{P_{acc}} \quad (5)$$

$$C_a = \frac{C_t}{n_t} \times \frac{n_t}{n_a} \quad (6)$$

$$C_a = \frac{C_t}{n_a} \quad (7)$$

Since Efficiency is defined to be

$$E = \frac{1}{C_a}, \quad (8)$$

given equation (4), we can express it as

$$E = \frac{n_a}{C_t}. \quad (9)$$

4.3 Methodology

Figure 1 shows the a portion of the NOBN used by ScenGen, RSI, and CBSP[1] to generate scenarios. The full network consists of approximately 150 vertices , and roughly 400 edges. A representative learning objective, which involved dealing with loss of electrical power and chill water was selected for the evaluation.

One trial of the experiment consists of asking one of the algorithms to generate one "quality" scenario. Each algorithm is run for approximately 200 trials. Importantly, as the experiment proceeds, the effects of each trial are accumulated. In accordance with equation (4), the acceptance probability is determined by dividing the number of admissible scenarios found in trial run so far by the total number of scenarios requested. The scenarios are found to be acceptable if and only if they are consistent with the Quality criterion, as described in section 4.2. In each trial, the Efficiency is calculated by dividing the number of admissible scenarios found by the total cost of generating all scenarios, as given by equation (9).

5 Results

Plots 1 and 2 show, in graphical form, the results of comparative evaluation experiments run on ScenGen, RSI, and CBSP. Plot 1 shows the incremental cost (in terms of time) of generating increasing numbers of admissible scenarios. Plot 2 compares the three algorithms on their efficiency of generating acceptable scenarios.

Looking at Plot 1, we see that all three algorithms appear to exhibit an asymptotically increasing incremental cost behavior. This is explained by the fact that the algorithms are operating in a finite domain. The more acceptable scenarios that are found, the more difficult it becomes to find additional acceptable, yet distinct scenarios.

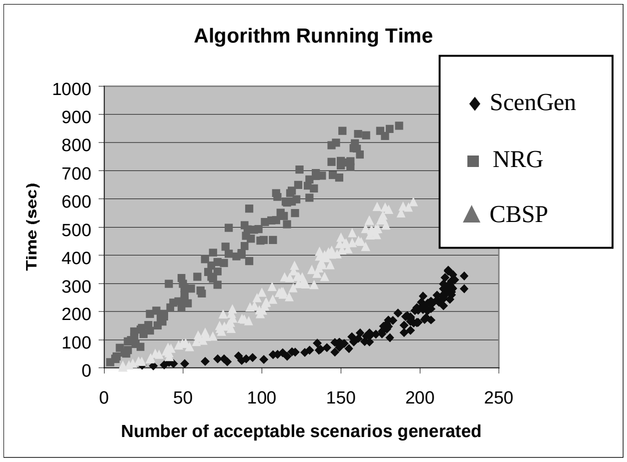
In terms of absolute cost per acceptable scenario, we see that NRG performs by far the worst. This is due to the fact that the vast majority of scenarios it generates are rejected by the Quality criterion, and thus discarded. This results in a large, steadily growing average cost per acceptable scenario. Interestingly, CBSP does slightly better than ScenGen for the first few acceptable scenarios it finds. This is explained by the fact that given a "good" scenario as a seed, it is not difficult to obtain several more "good" scenarios by small, inexpensive random perturbations. However, as additional distinct scenarios are requested, the random perturbations degenerate into a random search, much like NRG. ScenGen, on the other hand, has a very low rejection rate, which outweighs the rather high cost it incurs in generating individual scenarios, since it needs to generate approximately an order of magnitude fewer scenarios total than its competitors.

Unfortunately, the very selection bias that gives ScenGen its power, also significantly limits its accessibility of many portions of the search space. We see that of the three algorithms it is the first to exhaust its space of accessible scenarios. For the given domain model, it is unable to find more than about 230 distinct acceptable scenarios. For NRG and CBSP we do not actually see a vertical asymptote in the plot, but there is certain to be one because there exists only a finite number of event combinations in the model.

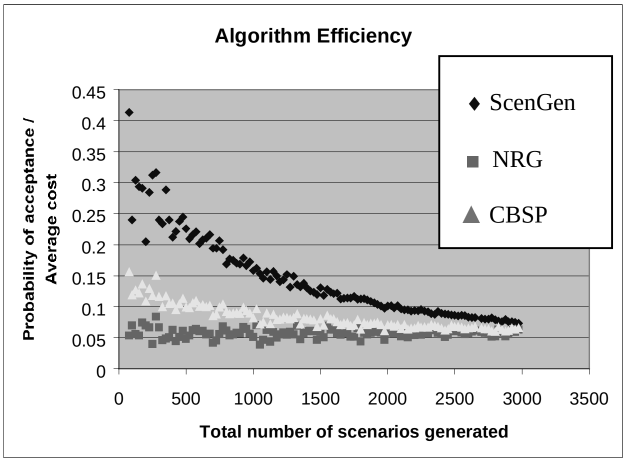
In light of such a comparison, ScenGen's selection bias may appear as a disadvantage. However, when all trade-offs are taken into account, it is seen that this is by no means the case. As can be seen from Plot 2, ScenGen finds those scenarios which are accessible to it far more efficiently that either of the other two algorithms. In addition, the actual location of ScenGen's vertical asymptote (as those of NRG and CBSP) is determined to a large extent by the size and complexity of the domain. Thus, in large and complex domain ScenGen's performance relative to the other two algorithms will actually improve.

We have succeeded, thus, in demonstrating that ScenGen outperforms both CBSP and RSI, its two main competitors. Because it generates only admissible scenarios, and in spite of the high cost associated with the required search heuristics, ScenGen achieves the best efficiency of the three algorithms. We believe that given a

[1] Since only a small number of human expert-designed scenarios were available, MDHE could not be included in any statistical comparison. It was used for qualitative comparison only.



Plot 1. Algorithm Running Time



Plot 2. Algorithm Efficiency

larger search space (with a correspondingly larger number of admissible scenarios), the gap between ScenGen and CBSP will be seen to widen dramatically. ScenGen will not be hampered by a rapidly diminishing number of admissible scenarios, and will be able to maintain a high acceptability rate even when many quality scenarios are requested. This will most likely cause its efficiency to shoot up, while those of the other two algorithms will remain the more or less constant, or perhaps even drop (which will be due to the drawback of the random nature of these algorithms becoming more pronounced in a large search space).

6 Conclusion and Future Work

We have presented a Noisy-OR Bayesian network model for simulation-based training, and ScenGen, an efficient search-based algorithm for automatic synthesis of plausible training scenarios from constraint specification. ScenGen is shown to outperform other randomized methods, such as those based on naïve random instantiation of scenarios, and stochastic perturbation of known "good" scenarios. ScenGen's strength lies in its ability to guarantee the "quality" of each and every scenario it generates through a carefully designed selection bias. While other algorithms are far less costly on a per scenario basis, the multitude of low-accuracy scenarios they produce (or their inability to find a sufficiently large number of "good" scenarios") drastically reduces their respective efficiencies, and results in poor overall performance.

One of our goals for the near future is to integrate ScenGen into DCTrain, the Illinois immersive damage control training environment. DCTrain combines the state-of-the-art in immersive multimedia environments with an accurate numerical/rule-based simulation of ship processes and crew behavior. The system has been extensively field-tested at the Surface Warfare Officer's School (*SWOS*), and found to be an effective and highly useful training tool [BDS96]. At the present time, all scenarios used with DCTrain must be manually constructed by domain experts. The integration of ScenGen would tremendously simplify the problem of scenario generation, thus making the system far more versatile, and greatly increasing its effectiveness.

7 Acknowledgements

The authors would like to thank Pavel Yusim for his significant contributions to the first implementation of the ScenGen algorithm. We gratefully acknowledge the support of ONR grant N00014-95-1-0749.

8 References

[BJM83] L. R. Bahl, F. Jelinek, R. L. Mercer. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179-190, 1983.

[BDS96] M. R. Baumann, M. A. Donovan, and J. A. Sniezek. *Evaluation of the Integrated Damage Control Training Technology (IDCTT).* University of Illinois, Knowledge Based Systems Group, Technical Report UIUC-BI-KBS-96005, 1996.

[De94] P. Derosier. *Integrated Damage Control Training Technology (IDCTT).* Center for Interactive Media in Medicine (CIMM), Bethesda, MD, 1994.

[EKT93] K. A. Ericsson, R. T. Krampe, and C. Tesch-Romer. The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review*, 100(3): 363-407, 1993.

[Fo73] G. D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268-278, 1973.

[GD88] D. M. Gaba and A. DeAnda. A Comprehensive Anesthesia Simulation Environment: Re-creating the Operating Room for Research and Training. *Anesthesia*, 69:387-394, 1988.

[GFH94] D. M. Gaba, K. J. Fish, and S. K. Howard. *Crisis Management In Anesthesiology*. Churchill Livingstone, New York, NY, 1994.

[HLB+96] B. Hayes-Roth, J. E. Larsson, L. Brownston, D. Gaba, and B. Flanagan. *Guardian Project Home Page*, URL: http://www-ksl.stanford.edu/projects/guardian/index.html,1996.

[HW95] D. A. Heckerman and M. P. Wellman. Bayesian Networks. *Communications of the ACM*, 38(3):27-30, March, 1995.

[He90] M. Henrion. Towards Efficient Probabilistic Diagnosis in Multiply Connected Belief Networks. In R. M. Oliver and J. Q. Smith (eds.), *Influence Diagrams, Belief Nets, and Decision Analysis*, p. 385-410. John Wiley and Sons, New York, NY.

[HB95] E. Horvitz and M Barry. Display of Information for Time-Critical Decision Making. In *Proceedings of UAI-95*, URL: http://research.microsoft.com/research/dtg/horvitz/VISTA.HTM, 1995.

[HGYS92] S. K. Howard, D. M. Gaba, G. S. Yang, and F. H. Sarnquist. Anesthesia Crisis Resource Management Training: Teaching Anesthesiologists to Handle Critical Incidents. *Aviation, Space, and Environmental Medicine*, 63:763-770, 1992.

[HH97] K. Huang and M. Henrion. Efficient Search-Based Inference for Noisy-OR Belief Networks: TopEpsilon. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, p. 325-331, 1997.

[Le89] K.-F. Lee. *Automatic Speech Recognition: The Development of the SPHINX System*. Kluwer Academic Publishers, Boston, MA, 1989.

[MW96] O. J. Mengshoel and D. C. Wilkins. Recognition and Critiquing of Erroneous Agent Actions. In *AAAI-96 Workshop on Agent Modeling*, p. 61-68. AAAI Press, Portland, OR, 1996.

[Ne93] R. M. Neal. *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

[Ni80] N. J. Nilsson. *Principles of Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA.

[Pe84] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, Reading, MA, 1984.

[Pe88] J. Pearl. *Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference.* Morgan-Kaufmann, San Mateo, CA, 1988.

[PPMH94] M. Pradhan, G. Provan, G. Middleton, and M. Henrion. Knowledge Engineering for Large Belief Networks. In *Proceedings of UAI-94*, p. 484-490. Morgan-Kaufmann, Seattle, WA, 1994.

[RN95] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[WFH+96] D. C. Wilkins, C. Fagerlin, W. H. Hsu, D. Kruse, and E. T. Lin. *Design of a Damage-Control Simulator*. Technical Report UIUC-BI-KBS-96005, UIUC, Knowledge Based Systems Group, 1996.