

Final draft, EP98 paper. Final version
of paper is due at conference, March 23.

Genetic Algorithms for Belief Network Inference: The Role of Scaling and Niching

Ole J. Mengshoel and David C. Wilkins
Department of Computer Science and Beckman Institute
University of Illinois, Urbana-Champaign
Urbana, IL 61801
{mengshoe|wilkins}@cs.uiuc.edu

Abstract. Belief networks are graphical models that encode probability distributions; they are used for reasoning and learning under uncertainty. Genetic algorithms are inspired by nature, and are used for adaptation, search, optimization, and learning in complex environments. This paper focuses on the use of a belief network as a genetic algorithm fitness function, and presents theoretical and empirical results related to fitness function sharing and scaling. In particular, we show that niching combined with scaling significantly improves a genetic algorithm's performance for belief network inference.

1 Introduction

Genetic algorithms (GAs) are inspired by nature and are used for adaptation, search, optimization, and learning in complex environments [7]. Belief networks (BNs) are graphical models that are based on probability theory and graph theory, and are used to reason and learn under uncertainty [15]. We consider how to use a GA's function optimization capability as a BN inference algorithm [12] [14], in particular the role of the GA techniques niching and scaling. Exact BN inference is NP-hard [2] [19] and may therefore put a GA's search capabilities to a serious test. GAs have been used to optimize other NP-hard problems [3], however BN inference is a particularly important one, given the prominence of probabilistic approaches in AI.

There is some previous research on using GAs for BN inference. Most closely related to this work is that of Lin et al. [11], Rojas-Guzman and Kramer [16] [17], Gelsema [5], Santos and Shimony [18], and Welch [21]. Most of these efforts focus on using the GA for abductive BN inference; i.e. to compute the most probable explanation given any instantiation of a subset of the BN nodes. This paper considers the role of niching and scaling in a GA used to search for the most probable explanations in a BN. The main contribution of the paper is that it identifies that non-trivial BNs are multi-modal fitness functions where the diversity-preserving technique of niching is of help. The role of scaling in using GAs for BN inference has been explored in earlier research [16] [17]. However, this paper adds to that research by suggesting a new scaling function and also

by finding that scaling *and* niching performs significantly better than either technique alone.

The rest of this paper is structured as follows. Section 2 briefly presents BNs and BN inference tasks. Section 3 reviews niching, in particular the fitness sharing approach to niching. Section 4 discusses theoretical issues in integrating GAs and BNs. Section 5 reports on experiments performed. Section 6 concludes and outlines directions for future research.

2 Belief Networks

A belief network (BN) represents a joint probability distribution in a compact manner, by exploiting a graph structure. This section introduces formal definitions related to BNs, and also defines belief network inference tasks.

Definition 1. Let $\{V_1, \dots, V_n\}$ be random variables, $\{v_1, \dots, v_n\}$ instantiations of those random variables. Here, instantiation v_i goes with random variable V_i . Then $\Pr(\mathbf{v})$ denotes the joint probability distribution over the variables $\{V_1, \dots, V_n\}$:

$$\Pr(\mathbf{v}) = \Pr(v_1, \dots, v_n) = \Pr(V_1 = v_1, \dots, V_n = v_n).$$

The notion of a belief network, also known as a Bayesian network, causal network, or Bayesian belief network, can now be introduced.

Definition 2. A belief network is a tuple $(\mathbf{V}, \mathbf{W}, Pr)$, where (\mathbf{V}, \mathbf{W}) is a directed acyclic graph (DAG) with nodes $\mathbf{V} = \{V_1, \dots, V_n\}$, edges $\mathbf{W} = \{W_1, \dots, W_m\}$. Pr is a set of conditional probability tables. For each node $V_i \in \mathbf{V}$ there is one such table $\Pr(V_i | Pa(V_i))$, which defines a conditional probability distribution over V_i in terms of its parents $Pa(V_i)$.

Consider a belief network over the nodes \mathbf{V} . It can be shown that the joint probability distribution $\Pr(\mathbf{v})$ is as follows, where $pa(V_i) \subset \{v_1, \dots, v_n\}$:

$$\Pr(\mathbf{v}) = \Pr(v_1, \dots, v_n) = \prod_{i=1}^n \Pr(v_i | pa(V_i)), \quad (1)$$

Belief networks are different from most other GA fitness functions in their accommodation of evidence through conditioning. If nodes $\mathbf{E} = \{E_1, \dots, E_e\}$ are instantiated to $\{E_1 = e_1, \dots, E_e = e_e\}$, we can use Bayes' rule to compute the posterior belief over the remaining nodes $\mathbf{X} = \mathbf{V} - \mathbf{E}$:

$$\Pr(\mathbf{x} | \mathbf{e}) = \frac{\Pr(\mathbf{x}, \mathbf{e})}{\Pr(\mathbf{e})} \propto \Pr(\mathbf{x}, \mathbf{e}). \quad (2)$$

$\Pr(\mathbf{e})$ can be computed by marginalization, however this is often not done, since $\Pr(\mathbf{x}, \mathbf{e})$ can in many cases be used directly instead.

All explanations are not equal. In particular, those that are more probable are typically of greater interest, leading to the following definition.

Definition 3. Let all the explanations be ordered according to their posterior probability: $\Pr(\mathbf{x}_1 | \mathbf{e}) \geq \Pr(\mathbf{x}_2 | \mathbf{e}) \geq \Pr(\mathbf{x}_3 | \mathbf{e}) \geq \dots$. The most probable explanation (MPE) is \mathbf{x}_1 . The k most probable explanations (k -MPE) are $\mathbf{x}_1, \dots, \mathbf{x}_k$.

Two typical BN inference tasks are belief updating and belief revision [15]. Belief revision is concerned with computing the MPE or more generally the k -MPE. Exact belief revision has been shown to be \mathcal{NP} -hard [19]. At the same time, inference in many BNs of practical value has proven to be tractable.

The following definitions are needed because of our GAs setting.

Definition 4. A population consists of individuals (or chromosomes). In the current setup, a population consists of a set of explanations $\{\mathbf{x}_1, \dots, \mathbf{x}_s\}$, where s is the population size.

3 Niching in Genetic Algorithms

The simple GA is suitable for function maximization, corresponding to finding the MPE in belief revision. However, the simple GA can have problems with premature convergence, so that a local maximum rather than a global maximum is found. Furthermore, for belief updating, or for computing k -MPE, one is more interested in function covering rather than pure function optimization.

This paper focuses on the technique known as *niching* [6]. Niching is based on the metaphor from nature that different species or subpopulations have different niches. There is no competition between niches, however within a niche there is competition. Within GAs, niching allows different parts of the fitness function to be explored in parallel, with convergence to several rather than just one function maximum.

Goldberg and Richardson asked and answered these two niching questions [6]: Who should share? How much should be shared? The essence of the answers to these questions is to consider the degree of similarity between two individuals in the GA population.

Definition 5. Consider two individuals A_i and A_j . A distance function d_{ij} , which is inverse to the similarity between A_i and A_j , is introduced: $d_{ij} = d(A_i, A_j)$

In addition, a sharing function s is constructed, and the following power law function suggested [6]:

$$s(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{\text{share}}}\right)^\alpha, & \text{if } d < \sigma_{\text{share}} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Here σ_{share} and α are constants. While the distance d between two individuals determines who should share, the sharing function s says how much they should share. The next issue is sharing across the population; this is formalized by the notion of niche count.

Definition 6. Consider individual A_i . The niche count m'_i of A_i is

$$m'_i = \sum_{j=1}^n s(d_{ij}) = \sum_{j=1}^n s(d(A_i, A_j)).$$

Now individual A_i 's shared fitness f'_i can be calculated as $f'_i = f_i/m'_i$, where f_i is the individual's raw fitness as computed by the objective function. The shared fitness f'_i can now be used as a fitness function in the simple **GA**.

Using fitness sharing, Goldberg and Richardson considered two multi-modal functions F_1 and F_2 . F_1 is periodic with five peaks of equal magnitude, while F_2 is periodic with five peaks of decreasing magnitude. On these two functions, the simple **GA** with shared fitness gave niching while the simple **GA** with raw fitness did not. Later, Deb and Goldberg found that sharing outperformed crowding [4]. Both phenotypic and genotypic sharing was considered, as was a mating restriction scheme for improving on-line performance. The sharing function approach has also been applied to deceptive and multimodal functions [8]. For certain deceptive and multi-modal functions, a niching **GA** using sharing found **all** global solutions when the population was sized appropriately and the objective function was scaled to emphasize global solutions [8]. A strength of the sharing function approach is that it gives good control over sharing. A limitation is that $O(n^2)$ steps are used for computing m'_i over the entire population. However, this time complexity can be reduced by using sampling to approximate m'_i .

4 Belief Network Inference using Genetic Algorithms

This section describes our approach to using a **GA** for BN inference. First, we focus on the fitness function to use, then on the encoding of a BN instantiation as a **GA** string as well as on the relationship between belief updating and belief revision when a **GA** is used as an inference engine for BNs.

Which *objective function* should be used? The most natural choice is to use the posterior probability $\Pr(\mathbf{x}, \mathbf{e})$ as defined in Eq. 2, and this is done here.

How should a BN instantiation be *encoded* as a genetic algorithm string? Since a BN is a **DAG**, a topological sort can be used to linearly order the nodes in the BN, and a **GA** string may be organized according to the linear order. We consider BNs where nodes represent discrete random variables, and in particular the special case where all nodes are binary. Let $V_i = v_i$ be the assignment to node number i in such a BN, so $v_i \in \{0, 1\}$. Then the following defines a one-to-one mapping from node V_i to bit b_i in position i in bitstring B :

$$b_i = \begin{cases} 0 & \text{if } V_i = 0 \\ 1 & \text{if } V_i = 1 \end{cases}.$$

This mapping is used for coding and decoding purposes when the BN representation is different from the **GA** representation.

Having presented the objective function and the **GA** encoding, we now consider properties of the objective function. Elsewhere, it has been demonstrated

that BNs can be deceptive [13]. Definition 2, Equation 2, and the fact that the conditional probabilities can be arbitrarily close to zero or one suggest that BN fitness functions that are in the general case multimodal and give fitness values that vary significantly, even between strings that are close in Hamming space. Multimodality is the reason why niching is important; high fitness variance is the reason for scaling.

First, we discuss niching and in particular the sharing approach to niching. Since we consider discrete BNs, genotypic sharing is more appropriate than phenotypic sharing although phenotypic sharing has been found to perform better in some instances [4]. Concerning the niching parameters, the discussion above suggests that $\sigma_{\text{share}} = 1$ is appropriate for belief updating and k -MPE inference. In other words, sharing only takes place between individuals on the same peak. If we let $\sigma_{\text{share}} > 1$, we risk a situation where, say, two close or adjacent individuals (in Hamming-space) both have high fitness, but in the limit one will dominate the other and thus one of two high-probability assignments will get lost. For computing the MPE in belief revision, the optimal σ_{share} -value is less clear, since the role of sharing is here to prevent premature convergence rather than to give covering. We may let $\sigma_{\text{share}} = 1$ also for MPE belief revision, but also consider other sharing values in Section 5.

There is a problem with using the objective function in Eq. 2 directly as the fitness function. In particular, the selection pressure early in a run is too large. In several exploratory runs an individual with relatively high fitness (a super-individual) showed up in the first generation, and because other first-generation individuals typically have very low fitness, the super-individual very quickly takes over the population, even though it is sub-optimal. To counteract this, some form of scaling can be used [7].

The problem of super-individuals in early generations and addressing this by using a scaling function are general issues in GA design—are there aspects specific to BNs? Scaling for BNs was investigated by Rojas-Guzman and Kramer [16] [17]. The functions they used were $c \text{Pr}(\cdot)$, $1/\log^2(\text{Pr}(\cdot))$, and $1/n$, where c is a scaling constant and n is the population size. However, no justification is given for these particular scaling functions, so it is appropriate to develop a theoretical basis for sharing and scaling functions for objective functions that are BNs. Since Eq. 1 is multiplicative, the differences between fitness values might be very large. Taking this into account, we suggest the following root scaling function

$$t_1(x) = x^{1/|\mathbf{V}|^{1/4}}, \quad (4)$$

where $|\mathbf{V}|$ is the number of nodes in the BN. Intuitively, taking the root counteracts the multiplicative (or exponential) effect of Eq. 1. For simplicity, assume that approximately $|\mathbf{V}|/2$ factors in Eq. 1 are relatively 'small' and equal; $x^{1/|\mathbf{V}|^{1/2}}$ then recovers this value. By using $x^{1/|\mathbf{V}|^{1/4}}$ instead, we get a stronger selection pressure. The scaling function t_1 can be contrasted with the inverse logarithmic scaling function t_2 , which was the function suggested by Rojas-Guzman and

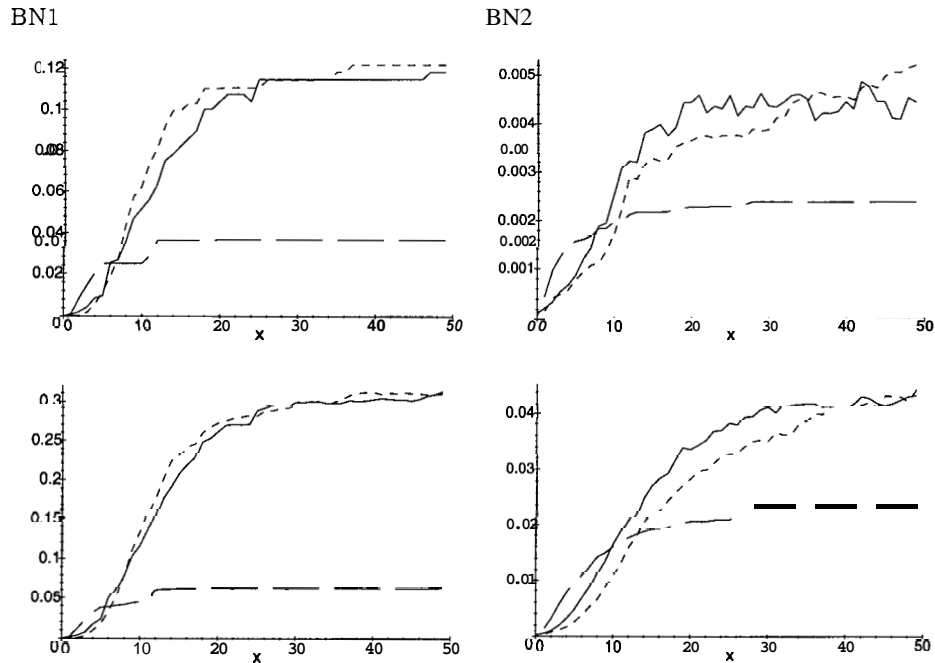


Fig. 1. Objective function evolution for BN1 (left column) and BN2 (right column) for the niching condition with $\sigma_{\text{share}} = 1$. The top row shows best individual, the bottom row shows sum of all unique individuals. The dashed line shows no scaling, the dotted line shows inverse logarithmic scaling, while the solid line shows root scaling.

Kramer that gave the best results in their experiments [16]:

$$t_2(x) = 1/\log^2(x). \quad (5)$$

Is there a difference between belief revision and belief updating when it comes to scaling? We conjecture that root scaling is better for belief updating while inverse logarithmic scaling is better for belief revision. The reason for this is that $t_1(x) \gg t_2(x)$ for small x , meaning that t_1 typically retains more diversity than t_2 , and belief updating is more dependent on diversity than belief revision is. However, it is hard to predict the exact effect of scaling for BNs, and in particular how scaling interacts with sharing. Experimentation is needed, and this is what we turn to next.

5 Experiments in Belief Network Inference using GAs

A simple GA (SGA) variant with niching and scaling as well as a BN interface has been implemented in Common Lisp and integrated with the IDEAL system for BN inference [20]. IDEAL is used for BN representation and objective function

evaluation, not inference, in this study. The only difference between this SGA and the standard SGA [7] is that this one uses stochastic universal sampling [1].

These parameters were used throughout the experiments reported in the following: population size $n = 200$, crossover probability $\Pr(\text{Crossover}) = 0.6$, mutation probability $\Pr(\text{Mutation}) = 0$, sharing parameter $\alpha = 1$, max generation $t_{\max} = 50$, and 30 runs per condition. A mutation rate of zero was used because we wanted to focus on crossover only in this study. Sharing parameters $\sigma_{\text{share}} = 1$, $\sigma_{\text{share}} = 2$ and $\sigma_{\text{share}} = 5$ have been used. We focus on two dependent measures: most probable explanation (or maximum objective function value) and total probability of explanations. The former is most important for belief revision, while the latter is most important for belief updating and for computing the k -MPE.

A BN consisting of 20 binary nodes with maximal in-degree six was created at random. The distribution of this BN, BN3, was then changed in additional BNs to make it more ‘natural’. In BN1, all distributions were 0.9/0.1. In BN2, all distributions were either 0.9/0.1 or 0.6/0.4. Note that only the conditional probability tables were changed; the graph structure is the same for all three BNs. BN1 represents a domain with some uncertainty. BN2 represents a domain where some aspects are slightly uncertain (0.9/0.1 distributions), while other aspects are highly uncertain (0.6/0.4 distributions). BN3 represents a domain where there is a lot of variability in uncertainty. Intuitively, we expect that BN2 represents a ‘typical’ BN. BN1 is on the certain end of the spectrum, while BN3 is on the uncertain end of the spectrum. By using all three BNs for experimentation, the generality of our GA/BN approach is tested.

First, the sharing parameter $\sigma_{\text{share}} = 1$ is tested. In the following, a statistical hypothesis test using two sample averages, unknown variances, and large samples is used. The significance level is 0.05. We focus on BN1 and BN2, since these are the more interesting BNs. Table 1, Table 2, and Figure 1 summarize the experimental results, which are presented in the following. In the tables, mean and standard deviation (st. dev.) refer to the normal (or Gaussian) distribution.

Hypothesis A: *Niching is helpful both for finding the most probable individual and for maximizing total probability mass.* For finding the most probable explanation, consider rows 1–3 versus 4–6 in Table 1. For maximizing total probability mass, consider rows 1–3 versus 4–6 in Table 2. This hypothesis is supported by the data for inverse logarithmic and root scaling, but not for the no scaling condition.

Hypothesis B: *Scaling is helpful both for finding the most probable individual and for maximizing total probability mass.* For the effect of scaling on the most probable individual, consider rows 2–3 versus row 1 as well as rows 5–6 versus row 4, all in Table 1. For the effect of scaling on maximizing total probability mass, consider rows 2–3 versus row 1 as well as rows 5–6 versus row 4 Table 2. The main result here is that with niching, scaling gives a statistically significant effect. When there is no niching, the picture is more complicated. In Table 1, only row 1 versus row 3 for BN1 is significant. In Table 2, all rows except row 1 versus row 2 for BN1 is significant.

Niching	Scaling	BN1		BN2		BN3	
		Mean	St.dev.	Mean	St.dev.	Mean	St. dev
No	No	0.0126	0.0302	0.00126	0.00155	0.000184	0.000130
No	Inv. log.	0.0347	0.0491	0.00221	0.00189	0.000235	0.000124
No	Root	0.0463	0.0544	0.00162	0.00177	0.000245	0.000110
Yes	No	0.0363	0.0481	0.00238	0.00185	0.000363	0.000127
Yes	Inv. log.	0.122	1.23×10^{-17}	0.00521	0.00229	0.000311	0.0000987
Yes	Root	0.118	0.0197	10.004450	0.00211	0.000286	0.0000969

Table 1. Best fit (maximum probability) individual for the three experimental BNs. The no niching condition comprises the three top rows, the niching condition the three bottom rows.

Niching		BN1		BN2		BN3	
		Mean	St.dev.	Mean	St.dev.	Mean	St.dev.
No	No	0.0127	0.0302	0.00135	0.00161	0.000235	0.000200
No	Inv. log.	0.0358	0.0486	0.00320	0.00281	0.00136	0.000670
No	Root	0.0477	0.0545	0.00352	0.00287	0.000940	0.000468
Yes	No	0.0617	0.0713	0.0222	0.0112	0.00613	0.00122
Yes	Inv. log.	0.312	0.0195	0.0444	0.0116	0.00522	0.00113
Yes	Root	0.309	0.0577	0.0435	0.00773	0.00515	0.000716

Table 2. Total probability mass for the three experimental BNs.

Hypothesis C: *Inverse logarithmic scaling is best for finding the most probable individual, while root scaling is better for maximizing total probability mass.* This hypothesis is clearly not supported by the data; if anything the results are the opposite. However, there is no statistical significance at t_{max} . Figure 1 still gives interesting information. For BN1, there is very little difference between the two scaling schemes. For BN2, however, there are some differences. In both cases, root scaling is indeed slightly better than inverse logarithmic scaling up till $t \approx 35$, corresponding to our positive expectations for root scaling. On the negative side, root scaling appears to be less stable than inverse logarithmic scaling.

So far, the sharing value $\sigma_{share} = 1$ has been considered. Even though this value was picked based on fitness function considerations, it is a very low sharing value. Experiments with sharing values $\sigma_{share} = 2$ and $\sigma_{share} = 5$ were also performed, resulting in the following findings. Most significant is the fact that $\sigma_{share} = 5$ does not work well. Comparing $\sigma_{share} = 1$ and $\sigma_{share} = 2$, for the scaled conditions $\sigma_{share} = 1$ outperforms $\sigma_{share} = 2$ except for the inverse logarithmic scaling on BN1. For this case, $\sigma_{share} = 1$ and $\sigma_{share} = 2$ perform approximately the same. These results correspond to what was expected, although it was a little unexpected that $\sigma_{share} = 1$ performed as well as it did compared to $\sigma_{share} = 2$, in particular on the maximal fitness task.

What do the results in this and the previous section mean? Based on these results, we would recommend to use *both* niching and scaling when using a GA for BN inference. In particular, using small sharing values such as $\sigma_{share} = 1$

or $\sigma_{\text{share}} = 2$ is best. As for the issue of root scaling versus inverse logarithmic scaling, our results are less conclusive.

6 Conclusion and Future Work

This work has focused on employing genetic algorithms (GAs) for belief network (BN) inference. In particular, we have shown how niching and scaling significantly improves the GAs performance under proportionate selection. Scaling only gave a statistically significant result when niching was also used. To get full effect from scaling, one also needs to use a niching scheme. This applies to finding the best fit individual as well as maximizing total fitness.

Although GAs are particularly suited for computing the most probable explanations in a BN with discrete nodes, other forms of evolutionary computation could be suited for closely related inference and learning tasks. In graphical models with continuous random variables, evolution strategies should prove useful. Evolutionary programming techniques could also be exploited; notice the similarity between a transition table in a finite state machine and a conditional probability table. Genetic programming could be applied after a generalization of BNs to stochastic programs rather than the deterministic programs that genetic programming usually works with.

There are several other areas for current and future research. First, additional theoretical research on combining niching and scaling in the context of using GAs for BN inference would be fruitful. A second area of interest is the performance of our approach compared to other approaches to BN inference, both traditional ones and approaches based on other GAs or other evolutionary algorithms. A third area of interest is approximate objective function evaluation. A BN can be approximated using the techniques of abstraction and aggregation, and the GA could (initially) use this approximate BN rather than the full-blown BN for objective function evaluation [14].

Acknowledgments: This work was supported in part by ONR Grant N00014-95-1-0749, ARL Grant DAAL01-96-2-0003, and NRL Grant N00014-97-C-2061. Thanks also go to David E. Goldberg for comments on and discussions related to previous versions of this paper.

References

1. J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In Grefenstette [9], pages 14–21.
2. G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.
3. Kenneth A. De Jong and William M. Spears. Using genetic algorithms to solve NP-complete problems. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, CA, 1989. Morgan Kaufman.

4. K. Deb and D. E. Goldberg. An investigation of niche and species-formation in genetic function optimization. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, CA, **1989**. Morgan Kaufman.
5. E. S. Gelsema. Abductive reasoning in Bayesian belief networks using a genetic algorithm. *Pattern Recognition Letters*, 16:865-871, **1995**.
6. D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In Grefenstette [9], pages **41-49**.
7. David E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley, Reading, MA, **1989**.
8. David E. Goldberg, Kalyanmoy Deb, and Jeffrey Horn. Massive multimodality, deception, and genetic algorithms. Technical Report IlliGAL Report No **92005**, University of Illinois, Urbana, **1992**.
9. J. J. Grefenstette, editor. *Proceedings of the Second International Conference on Genetic Algorithms*, Hillsdale, New Jersey, **28-31 July 1987**. Lawrence Erlbaum Associates.
10. John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, **1975**.
11. R. Lin, A. Galper, and R. Shachter. Abductive inference using probabilistic networks: Randomized search techniques. Technical Report **KSL-90-73**, Knowledge Systems Laboratory, Stanford, November **1990**.
12. Ole J. Mengshoel. Belief network inference in dynamic environments. In *Proc. of AAAI-97*, page **813**, Providence, RI, **1997**.
13. Ole J. Mengshoel, David E. Goldberg, and David C. Wilkins. Deceptive and other functions of unitation as bayesian networks. **1998**. Submitted paper.
14. Ole J. Mengshoel and David C. Wilkins. Abstraction for belief revision: Using a genetic algorithm to compute the most probable explanation. In *Proc. AAAI Spring Symposium Series*, Stanford University, CA, March **1998**. To appear.
15. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California, **1988**.
16. C. Rojas-Guzman and M. A. Kramer. GALGO: A Genetic ALGORITHM decision support tool for complex uncertain systems modeled with Bayesian belief networks. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, pages **368-375**, Washington, D.C., **1993**.
17. C. Rojas-Guzman and M. A. Kramer. An evolutionary computing approach to probabilistic reasoning on Bayesian networks. *Evolutionary Computation*, 4(1):57-85, **1996**.
18. Jr. Santos, Eugene, Solomon Eyal Shimony, and Edward Williams. Sample-and-accumulate algorithms for belief updating in Bayes networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages **477-484**, Portland, Oregon, **1996**.
19. E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68:399-410, **1994**.
20. S. Srinivas and J. Breese. IDEAL: A software package for analysis of influence diagrams. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages **212-219**, July **1990**.
21. Robert L. Welch. Real time estimation of Bayesian networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages **533-544**, Portland, Oregon, **1996**.