

Using Distribution-based Operators in Genetic Algorithms

Rafael Nogueras*

Carlos Cotta*

*Dpto. de Lenguajes y Ciencias de la Computación, Univ. de Málaga
Campus de Teatinos (3.2.49), 29071 - Málaga, Spain
rafaelns@teleline.es

1 Introduction

Genetic Algorithms (GAs) constitute a very efficient search model that has provided excellent results in different domains during the last fifty years. However, new methods offering additional possibilities are emerging. Estimation of distribution Algorithms (EDAs) are one of these methods. In this work, we study the combination of both approaches. To be precise, we consider the use of a Bayesian Network (BNs) to improve the best individuals found by the GA. BNs are a probabilistic model that we utilize to predict the performance of a particular individual in its subsequent mutations and crossovers. This can be used to provide hints on which the most convenient way to mutate one particular individual, or to recombine two different individuals is.

The remainder of the paper is organized as follows. First, an overview of previous work regarding the use of BNs in Evolutionary Computation (EC) methods. The HPA (Hybrid Probabilistic Algorithm) model is described and exemplified in Section 3. Next, an extensive empirical evaluation of the different variants of HPA is presented in Section 4. Finally, Section 5 summarizes our results, and outlines future research.

2 Related Work

EDAs are undoubtedly a prime example of the use of BNs in the EC arena [5, 3, 4]. These methods are mainly based on evolving probability distributions, describing better and better solutions as the search progresses. Unlike GAs, the population of solutions is just a mere temporary device used to estimate a probability distribution. It is actually this probability distribution the central element of the algorithm: initially, it is a flat (or random) distribution over the whole search space; subsequently, the population is obtained by sampling the distribution, and the best individuals are used to infer the probability distribution for the next step. The population is disposed after this, and the process is repeated.

As it can be seen, an EDA has just a few parameters: how many individuals are sampled, how many are retained, and –most importantly– how the probability distribution is modeled

Vienna, Austria, August 22–26, 2005

(some associated parameters appear here). Hence, EDAs are somewhat simpler in this sense than GAs. Indeed, the adequate parametrization of evolutionary algorithms is a problem for itself, as Grefenstette showed in [2].

The particular model used to describe the probability distribution must match the features of the search space, i.e., it must be capable of capturing any underlying relationship between the decision variables that might exist. BN provide a very general model, capable of grasping complex inter-variable relationships (this is done for example in the Bayesian Optimization Algorithm – BOA [6]). In general, the process would be as follows:

1. **let** $\Psi \leftarrow \text{RandomBN}$;
2. **while** $\neg \text{Termination}$ **do**
 - (a) **let** $pop \leftarrow \text{Sample}(\Psi, N)$; /* N is the population size */
 - (b) **let** $pop' \leftarrow \text{SelectBest}(pop, M)$; /* $M < N$ */
 - (c) **let** $\Psi \leftarrow \text{LearnBNFromData}(pop')$;

Notice that while GAs evolve using crossover and mutation operators on the information pool (the population), aiming at producing the optimal solution, EDAs attempt to characterize this optimal solution in a probabilistic sense. Following related ideas, a Probabilistic Model Building Genetic Algorithms (PMBGA) are described in [8]. They try to build a probabilistic model of the population to maintain building blocks in subsequent generations.

In order to perform a deeper analysis, other EDAs that assume different underlying probabilistic models will be considered. To be precise, these are the following:

- Univariate, when no dependency among variables: UMDA, PBIL.
- Bivariate, when pairwise dependencies: MIMIC, COMIT, BMDA
- Multivariate, when multiple dependencies: BOA

3 Hybrid Probabilistic Algorithm (HPA)

As anticipated before, the HPA is a GA in which the reproductive operators are guided by a BN. The generation of this BN will be dealt later on. At this point, let us assume we have this BN, and let us get into the functioning of the operators. First of all, the mutation operator is as follows:

1. Use the BN to estimate the probability that a each gene has its current value. This is done using the marginal probabilities for root variables in the BN, and using the corresponding conditional probabilities for the remaining ones.
2. The gen whose current value is most unlikely is selected to mutate.

According to this description, the underlying idea of this mutation operator is to bring solutions back to the promising region of the search space. Of course, new solutions can be discovered on its way back. As to recombination, the process is a generalization of the previous one. Let s and s' the solutions to be recombined, and let d be the descendant. Then:

1. **let** $\langle i_1, \dots, i_n \rangle$ be the topological order of variables in the BN.
2. **for** $j \in [1..n]$ **do**
 - (a) **if** $s_{i_j} = s'_{i_j}$ **then let** $d \leftarrow s_{i_j}$
 - (b) **else**
 - i. **let** Π be the parents of variable i_j in the BN.
 - ii. **let** $\Pi|_d$ be the values of variables in Π in the partially defined descendant (recall that $\Pi = \emptyset$ for root variables).
 - iii. **let** $d_{i_j} \leftarrow \text{Sample}(\text{BN}, \Pi|_d)$

Here, the idea is to produce a probably promising (according to the information comprised in the BN) combination of the parental alleles. This constitutes a much more computationally amenable approach that e.g. the *provably* promising combination achievable by using an exact technique such as Branch and Bound for recombination [1].

The BN used in the operators can be obtained in different ways. In this work we have opted for by keeping a record of a couple of pilot executions of a plain GA, but other possibilities exists, e.g., using a pilot EDA, generating an elite set using multiple runs of a GRASP procedure, etc. Preliminary tests indicate that the BN exerts a strong control on the evolution of the population, inducing premature convergence to suboptimal solutions. To fix this issue, we have considered a mixed model that uses the Bayesian operators during some phases of the evolution and traditional genetic operators in the remaining ones. The followed criterion is based on the mean fitness of the population: when this value is closed to a particular percentage of the best fitness, classical genetic operators are used.

Therefore, it is shown that the HPA reaches (only in three or four evaluations) the best fitness obtained by the GA and, from there, the HPA follows its evolution, improving the result of the GA, as it can be seen in the Figure 1 (middle) with the curves of HPA'. Figure 1 (right) shows a comparative with the several versions of EDAs described in Section 2.

4 Experimental Results

This section provides some examples of the application of the HPA model for several problems. Unless otherwise noted, experiments have been done with an elitist generational GA/HPA ($popsiz e = 100$, $p_X = .9$, $p_m = 0.01$, $maxevals = 10000$) using ranking selection. All results correspond to series of 50 runs. In all cases, the HPA switched to traditional operators (single-point crossover, and random bit-flip mutation) when the mean fitness of the population is close to the 95 percent of the best fitness. Experiments have been done with a SAT problem, the Rosenbrock function, an alternate bits function (maximizing the number of 10 or 01 substrings

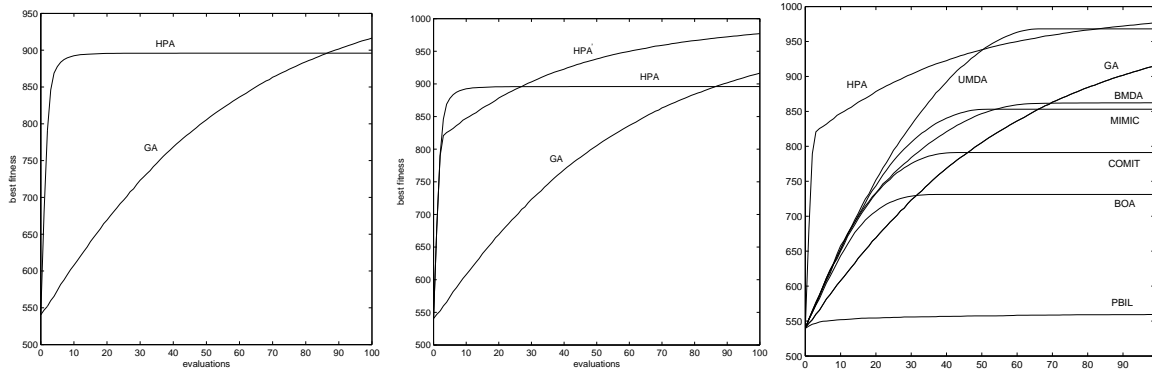


Figure 1: Comparative between the GA and the initial HPA (left), between the GA and the HPA (middle) and between HPA and the EDAs (right) for OneMax problem with 1000 bits.

in a binary sequence), the Massively Multimodal Deceptive Problem (MMDP)¹ and the P-Peaks² function.

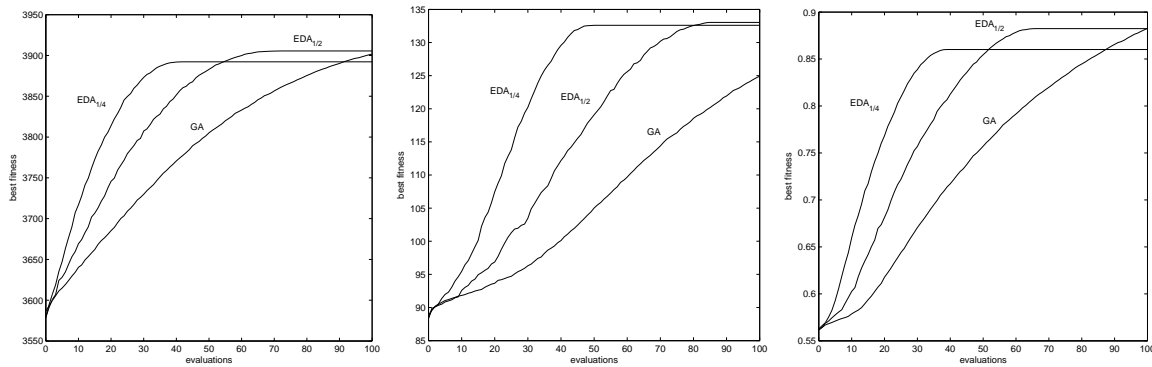


Figure 2: Comparison of GA vs. defined EDA in Section 2 on a SAT problem for 3 variables per clause, 70 variables and 4300 clauses (left), in the MMDP problem with 200 segments (middle), and in P-Peaks function with 1000 bits and 100 peaks (right). In all problems $N = 100$ and $M = 1/2$ of N or $1/4$ of N for the EDAs.

The results are shown in Figs. 2 to 5. As it can be seen, the improvement of the HPAs with respect to the GA is conclusive in all problems studied. This indicates that BNs can lead genetic operators to produce better individuals. Indeed, the HPA always generate better solutions than the GA. Regarding the comparison with the EDA, the HPA always converges faster, and produces similar or better solutions.

¹This problem is defined by the concatenation of k 6-bit segments. The fitness of a string is determined by the sum of each segment fitness, which is calculated on the basis of its unitation: there are two global optima at 0 and 6 ones, and a local optimum at 3 ones; the gradient at 1 and 2 ones leads to the local optimum.

²This function is defined on binary strings of N bits; P points of the search space –the peaks– are selected, and the fitness of a solution is the normalized Hamming distance to the closest one.

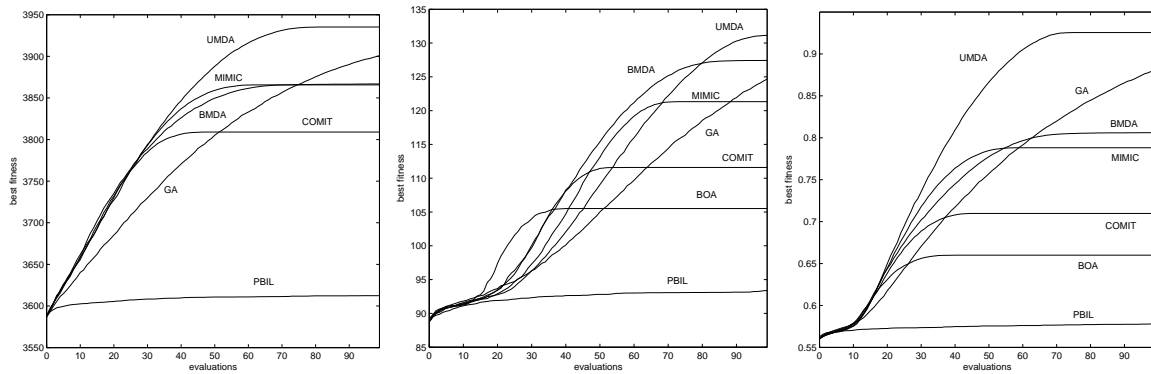


Figure 3: Comparison of GA vs. different EDAs on a SAT problem for 3 variables per clause, 70 variables and 4300 clauses (left), in the MMDP problem with 200 segments (middle), and in P-Peaks function with 1000 bits and 100 peaks (right). In all problems the size of population is 100 individuals and the individuals selected in each iteration are 50.

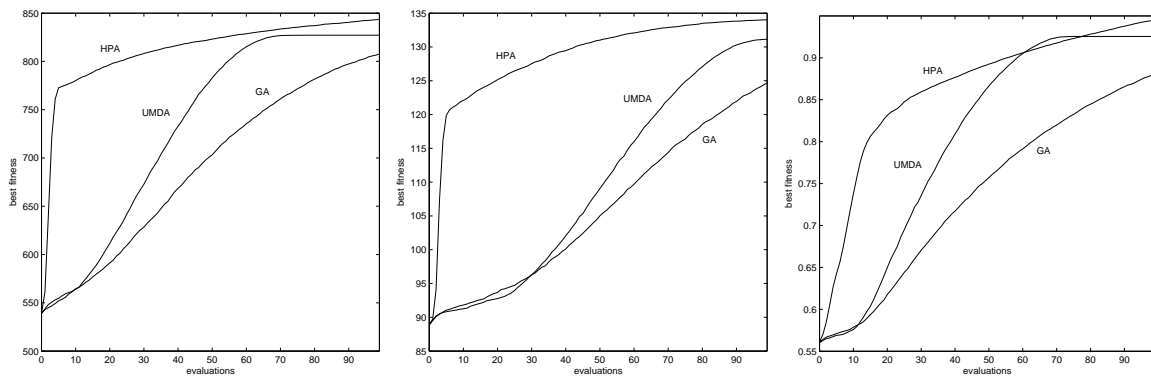


Figure 4: Comparison of UMDA vs. HPA in the alternate bits problem with 1000 bits (left), in the MMDP problem with 200 segments (middle), and in P-Peaks function with 1000 bits and 100 peaks (right).

5 Conclusions

We have presented a hybrid GA that incorporates ideas from EDAs in order to define heuristic reproductive operators. The use of probability distributions within recombination and/or mutation offers a computationally amenable alternative to using more sophisticated, problem-dependent heuristics for performing these tasks. The initial results obtained indicate that this can be a promising approach, since both an EDA and a GA were outperformed by the hybrid.

Future work will try to confirm these results on other problems. Furthermore, there are numerous variants of the algorithm that are worth exploring. These involve alternative mechanisms for obtaining the initial BN, or for governing the usage of the Bayesian operators.

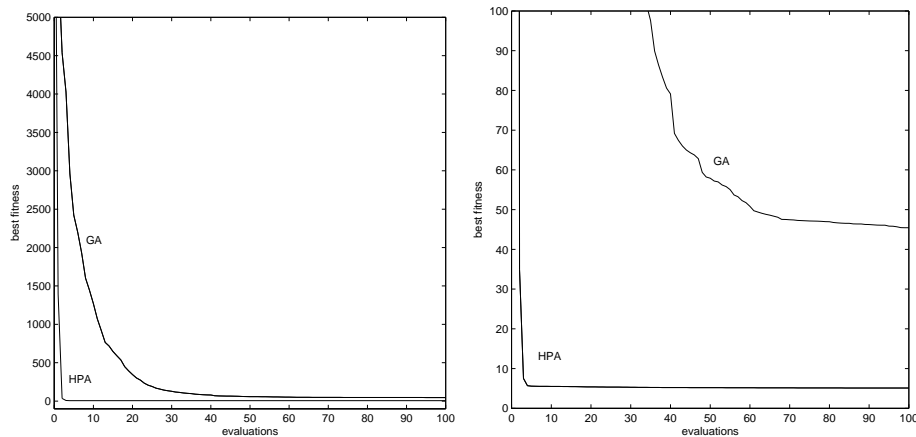


Figure 5: Comparison of GA vs. HPA in the Rosenbrock function with $n = 8$, $length = 128$ bits (left) and the same function with zoom in, where it can be seen that HPA stops very close to the optimum solution (right).

References

- [1] Cotta, C. and Troya, J.M. (2003): “Embedding Branch and bound with Evolutionary Algorithms”. In: *Applied Intelligence* **18**(2): 137–153
- [2] Grefenstette, J.J. (1986): “Optimization of control parameters for genetic algorithms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* **16**: 122–128.
- [3] Larrañaga, P. and Lozano, J.A. (2001): *Estimation of Distribution Algorithms. A new tool for Evolutionary Computation*. Kluwer Academic Publishers.
- [4] Larrañaga, P. and Lozano, J.A. and Mühlenbein, H. (2003): “Estimation of Distribution Algorithms Applied To Combinatorial Optimization Problems”. In: *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial* **19**: 149–168.
- [5] Mühlenbein, H. and Paaß, G. (1996): “From recombination of genes to the estimation of distributions I. Binary parameters”. In: Voigt, H.-M. and Ebeling, W. and Rechenberg, I. and Schwefel, H.-P. (eds.): *Parallel Problem Solving from Nature , LNCS 1141*. Springer: Berlin, 178–187.
- [6] Pelikan, M. and Goldberg, D.E. and Cantú-Paz, E. (1999): “BOA: The Bayesian Optimization Algorithm”. In: Banzhaf, W. and Daida, J. and Eiben, A. E. and Garzon, M.H. and Honavar, V. and Jakiela, M. and Smith, R.E. (eds.): *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*. (Kluwer Academic Publishers, Boston, USA, vol. I) Springer-Verlag, 525–532.
- [7] Pelikan, M. and., Goldberg, D.E. and Lobo, F. (1999): “A Survey of Optimization by Building and Using Probabilistic Models”. Technical Report [99018]. Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.
- [8] Shakya, S.K (2003): “Probabilistic model building Genetic Algorithm (PMBGA): A survey”. Technical Report. School of computing, The Robert Gordon University, Aberdeen, Scotland, UK.