



University of
Salford
MANCHESTER

A study of the effectiveness of detailed balance in avoiding convergence in PBIL

Correa, ES and Shapiro, JL

<http://dx.doi.org/10.1007/978-3-540-45240-9>

Title	A study of the effectiveness of detailed balance in avoiding convergence in PBIL
Authors	Correa, ES and Shapiro, JL
Type	Book Section
URL	This version is available at: http://usir.salford.ac.uk/41387/
Published Date	2004

USIR is a digital collection of the research output of the University of Salford. Where copyright permits, full text material held in the repository is made freely available online and can be read, downloaded and copied for non-commercial private study or research purposes. Please check the manuscript for any further copyright restrictions.

For more information, including our policy and submission procedure, please contact the Repository Team at: usir@salford.ac.uk.

A study of the effectiveness of detailed balance in avoiding premature convergence in PBIL

E.S. Correa and J.L. Shapiro

Department of Computer Science

University of Manchester

Oxford Road, Manchester, M13 9PL, United Kingdom

e-mail: {esc, jls}@cs.man.ac.uk

Abstracts: *Estimation of distribution algorithms (EDAs) are a class of evolutionary algorithms that use statistical information to guide further exploration of the search space. A prominent problem with EDAs is that, even on a flat fitness landscape, they do not explore the search space uniformly. This occurs because at each iteration the probabilistic model reinforces the probabilities of generating the best solutions found in the previous population. As a consequence, the search in these algorithms is, in general, strongly dependent on the initial solutions generated by the algorithm. This work investigates the performance of the detailed balance PBIL (DBPBIL) algorithm and extends its application to a non-binary codification. The results obtained by this algorithm show that DBPBIL is much more robust than standard PBIL. More importantly, the implementation of detailed balance to PBIL is shown to be an efficient method of diminishing the convergence pressure imposed by the probabilistic model in EDAs.*

Keywords: EDAs, PBIL, detailed balance, SAT problem, P-median problem.

1 Introduction

Estimation of distribution algorithms (EDAs) are a class of evolutionary algorithms that use statistical information to guide further exploration of the search space. A prominent problem in EDAs is the loss of variability as the search progresses. In essence, it occurs because at each iteration the probabilistic model reinforces the probability of generating the best solutions found in the previous populations. Although this is exactly what it is supposed to do, i.e., to guide the search to promising areas of the search space, this process may accelerate convergence to local optima. In general, the majority of the research effort in EDAs has gone into devising new algorithms based on increasingly complex graphical models. This paper, instead investigates a method to diminish the above-mentioned convergence pressure in a specific EDA. In order to do that a condition called “detailed balance” is applied to the Population Based Incremental Learning (PBIL) algorithm [4]. Detailed balance is a well-known condition in Markov chains. Basically, it says that on a flat fitness landscape the probability of going from a state i to a state j must be the same as the probability of going backwards from state j to state i . In this work, this condition is applied as an attempt to make the probability of picking a certain value to a given variable of the problem in the current population be the same as the probability of not picking this value in the next population. As a result, the algorithm requires more evidence from the fitness function to drive the search to a single point in the search space. As a consequence, not only is the local optimum convergence avoided, but also the variability in the search is maintained for longer.

2 Population Based Incremental Learning

Among the simplest EDAs, PBIL introduced by Baluja [1], combines mechanisms of the generational genetic algorithm with simple competitive learning. Its probabilistic model assumes total independence between variables of the problem being treated. PBIL, like other EDAs, does not maintain a population of solutions. Instead, it uses a probability vector, $p_t(x) = (p_t(x_1), \dots, p_t(x_l))$, to represent characteristics of the good solutions found in previous populations. This vector is then used to sample new solutions. Thus, those characteristics are likely to be transferred to future populations. Considering a binary codification, $p_t(x_i)$, $i = 1, 2, \dots, l$, refers to the probability of obtaining a value of 1 in the i^{th} component of a solution and l is the problem size. Basically, the algorithm works as follows: a population of N solutions is randomly generated. Based on their fitness value, the k ($k < N$) best solutions from the current population are selected. They are denoted by $S_t = x_{1:N}^t, x_{2:N}^t, \dots, x_{k:N}^t$. These solutions are then used to update the probability vector by using the equation: $p_t(x) = p_{t-1}(x)(1 - \alpha) + \alpha \frac{1}{N} \sum_{i=1}^k x_{i:N}^{t-1}$, where $\alpha \in (0, 1]$ is a parameter of the algorithm. A new population is then sampled from $p_t(x)$. Again, S_t is selected, $p_t(x)$ updated, and another population sampled - until a stopping criterion is met.

2.1 Detailed Balance PBIL

Introduced by Shapiro [4], Detailed Balance PBIL (DBPBIL) modifies the dynamics of the standard PBIL (PBIL) in order to make it obey the detailed balance condition. This algorithm imposes reversibility to the transition probabilities and by doing that it significantly diminishes the convergence pressure imposed by the probabilistic model during the search. As a result, the algorithm is less affected by the starting conditions. Finally, and equally important, DBPBIL is much more robust than PBIL as it is experimentally shown later. For a more detailed explanation and results on Detailed Balance PBIL the reader is referred to the paper by Shapiro.

The next section presents one of the two test problems used to assess the performance of the above-mentioned algorithms.

3 Satisfiability problem

The purpose of a satisfiability (SAT) problem is to determine, for a formula of the propositional calculus, if there is an assignment of truth values to its variables for which that formula evaluates to true. A general SAT problem can be defined as follows: given a set of m clauses $\{C_1, C_2, \dots, C_m\}$ on n Boolean variables $v = (v_1, v_2, \dots, v_n)$ with $v_i \in \{0, 1\}$, and a Boolean formula in conjunctive normal form, $f(v) = C_1 \wedge C_2 \wedge \dots \wedge C_m$, determine an assignment of truth values to v so that $f(v)$ evaluates to true. Corresponding to each variable v_i are two literals, v_i and its logical negation $\neg v_i$. In addition, a clause C_i is a set of literals in disjunction in which all variables must be different from each other. In the well-studied MAX-3-SAT problem, every clause C_i has exactly three literals and the goal is to determine an assignment to v that maximises the number of satisfied clauses in $f(v)$. A clause is said to be satisfied when its truth value is true. Last but not least a well-known feature of SAT is phase transition. It is known as the hardest area for determining whether or not a SAT problem is satisfiable. Several experimental studies have shown that this “phase transition” point seems to occur when the ratio between the number of clauses and variables of the problem is approximately 4.25.

3.1 PBIL & DBPBIL on SAT

With the object of assessing the real benefits of the detailed balance condition to PBIL, PBIL and DBPBIL are applied to the random MAX-3-SAT problem. Three sets of 3-SAT problems, with 15, 20 and 30 variables respectively were generated. Each set contains 10 random generated instances, all satisfiable and phase transition. In all the experiments carried out in this work the same sets of problems were used. Every problem in each set was submitted to 100 trials which means that for every size problem 1000 trials were performed for every single value of α . The fitness of a solution is given by the formula: $fitness = \frac{8 \cdot satisfied_clauses - 7 \cdot C}{C}$, where C is the total number of clauses of the problem. Thus, the greatest possible fitness value is 1, which occurs when all clauses are satisfied. The mean of the best fitness obtained by all trials is presented in figure 1.

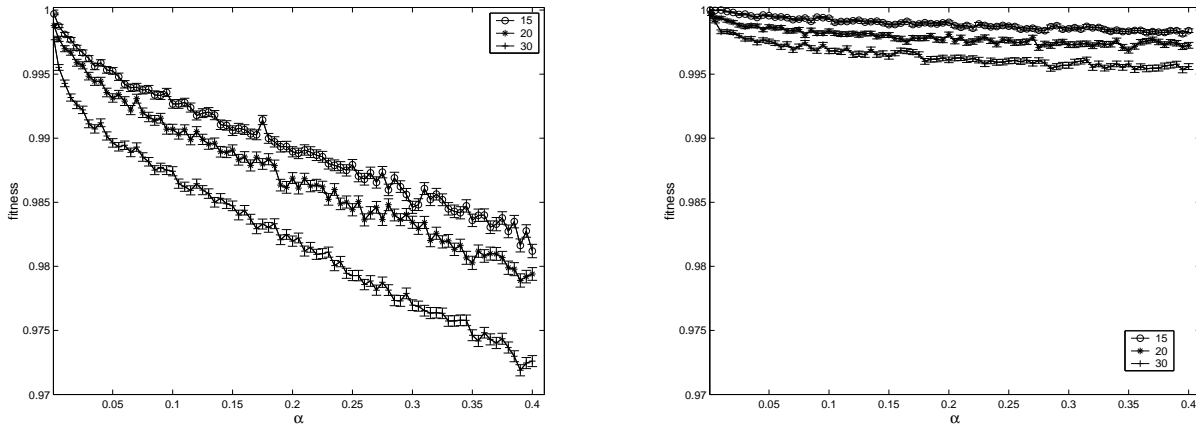


Figure 1: Simulations on PBIL on SAT problem for 15, 20 and 30 variables. Mean of the best fitness found on 1000 trials for 80 different values of α . The population size used is equal to 20 and only the best solution at each generation is selected to update the probabilistic model. **Left:** PBIL. **Right:** DBPBIL.

Figure 1 shows that DBPBIL outperforms PBIL in terms of the quality of the solution found for practically all values of α . These results support the idea that by applying the detailed balance condition to PBIL, the performance of this algorithm is indeed improved. Furthermore, they show that DBPBIL is less sensitive to α ; since the fitness variation to DBPBIL is much smaller than to PBIL.

The stopping criterion used in these experiments was: the algorithm stops when all probabilistic parameters have converged to either 0.05 or 0.95. Figure 2 depicts the mean number of generations for both algorithms until convergence.

On average, DBPBIL seems to take more time to converge. Naturally, this is not surprising, since the algorithm was developed to slow down the convergence pressure imposed by the probabilistic model and generalise the search through the search space rather than being attracted and becoming concentrated on local optima. Furthermore, given the stopping criterion used for both algorithm, DBPBIL will almost always need a greater number of generations than PBIL to stop. In particular, the more the parameters approach 0 or 1, the more DBPBIL rejects movements towards these values. Figure 3 presents the mean of the total number of trials in which the global optimum was found.

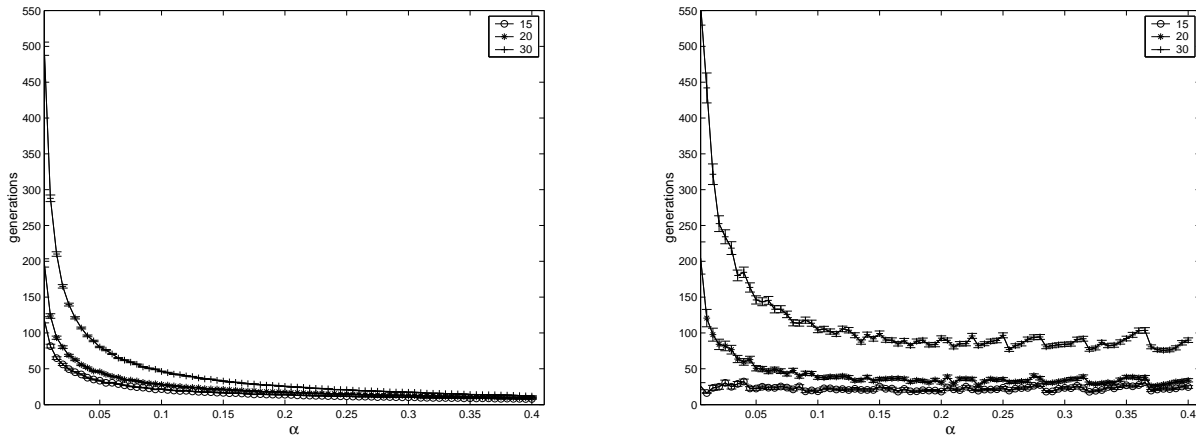


Figure 2: Simulations on PBIL on SAT problem for 15, 20 and 30 variables. Mean of the total number of generations until convergence on 1000 trials. **Left:** PBIL. **Right:** DBPBIL.

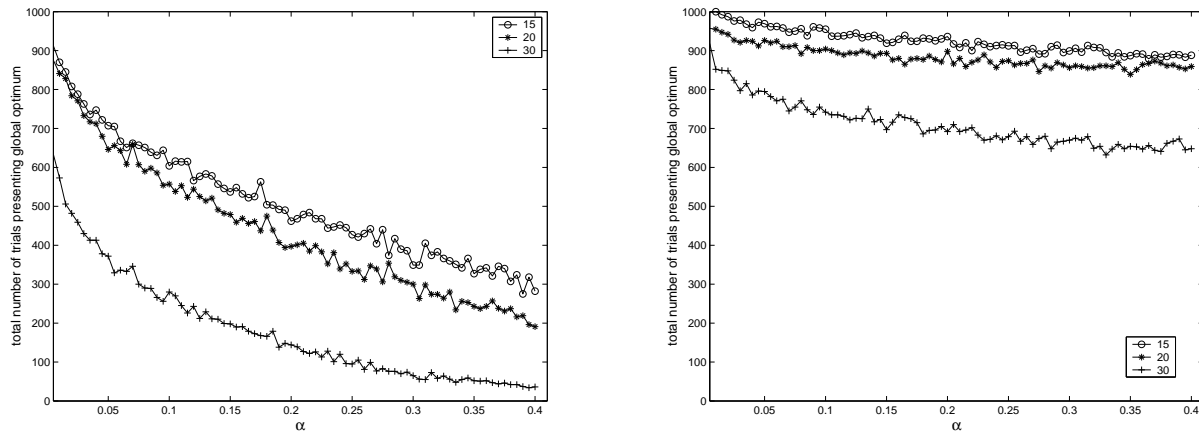


Figure 3: Simulations on PBIL on SAT problem for 15, 20 and 30 variables. Mean of the total number of trials in which the global optimum was found. **Left:** PBIL. **Right:** DBPBIL.

On this criterion DBPBIL worked much better than PBIL. It found the global optimum in more than 60% of the trials for literally all values of α and size problems. By contrast, for the problems with 30 variables, PBIL found the global optimum 60% of the trials only for very small values of α , falling to less than 10% when α is equal to 0.4.

The next section introduces the P-median problem which is the second test problem used to compare the performance of PBIL with DBPBIL. In particular, this problem uses a non-binary codification as presented below.

4 The P-median problem

To put it simple, a median is a facility selected to serve other facilities. The goal of the P-median problem is to select p medians in a set with n facilities ($n > p$). These medians must be selected so that the sum of the weighted distances between every facility not selected as median and its nearest median is minimised. More formally, assuming all vertices of a graph are potential medians, this problem can be defined as follows: let $G = (V, A)$ be an undirected graph where V represents the set of the n vertices belonging to the graph and A represents the edges (each edge represents the distance between a pair of vertices). The goal is to find a subset of vertices $V_p \subseteq V$ (the median set) with cardinality p , such that the sum of the weighted distances between

each remaining vertex in $\{V - Vp\}$ (the demand set) and its nearest vertex in Vp is minimised. For more details on the P-median problem the reader is referred to Correa et al. [2].

4.1 PBIL & DBPBIL on the P-median problem

This application extends DBPBIL to non-binary codification. For this problem a solution is represented in the following way: each solution contains exactly p sites, where p is the desired number of medians, and each site can receive an index (a unique identity number) of a facility selected as median. In this application a solution is represented as a set of facilities indices - there are no duplicated indices and there is no ordering among them. The fitness of a solution is given by the sum of the weighted distances between every facility not selected as median and its nearest median composing the solution in question.

DBPBIL also uses a vector of probabilities with n parameters, $p_t(x) = (p_t(x_1), p_t(x_2), \dots, p_t(x_n))$, where $p_t(x_i)$ represents the proportional probability of the i^{th} facility be selected as median. At the begin, the parameters $p_t(x_i)$ are initialised to 0.5. Every parameter $p_t(x_i)$ is then multiplied by a random number uniformly chosen in the interval $(0, 1)$. After that, the set of the selected medians is defined by ranking the parameters $p_t(x_i)$ in a decreasing order. The first p indices (facilities) corresponding to the first p values $p_t(x_i)$ in the ranking are selected as medians.

To illustrate, suppose that one wants to select 2 medians out of 4 and the multiplied initial probability vector is $p_0(x) = (0.13, 0.01, 0.27, 0.41)$. Ranking $p_0(x)$, $(0.41, 0.27, 0.13, 0.01)$, one would obtain a solution in which the 2 selected medians are the facilities 4 and 3 respectively.

In order to update the probability vector $p_t(x)$, the following equations are used:

$$p_{t+1}(x_i) = \begin{cases} p_t(x_i)(1 - \alpha) + \frac{\alpha}{\text{number_of_medians}}, & \text{if index } i \text{ is present in the best solution,} \\ p_t(x_i)(1 - \alpha), & \text{otherwise.} \end{cases} \quad (1)$$

In both algorithms, PBIL and DBPBIL, only the fittest solution from the current population is used to update the probabilistic model. The equations in 1, however, do not satisfy detailed balance. For this reason they need to be modified. In addition, a rejection method is used in order to make the algorithm obey detailed balance. These modifications can easily be worked out. In spite of that, the mathematics involving them is out of the scope of this paper and therefore the reader is once again referred to [4].

For the P-median problem, 4 instances, with respectively 20, 30, 40 and 50 facilities, were randomly generated on a 100 by 100 square. The weight corresponding to each facility was also randomly determined by a real number between 0 and 100. The same data base was used for all experiments. Figure 4 presents the mean of the best fitness found on 1000 trials.

For the P-median problem the difference between the performance of the algorithms was not so apparent as it was for SAT problems. Nonetheless there is a difference and then again DBPBIL outperformed PBIL. For problems with 20 and 30 facilities, DBPBIL has always converged to a solution near to the best solution ever found by both algorithms. For problems with 40 and 50 facilities, it has not only found better solutions than PBIL for small values of α , but has also been more stable in all simulations.

5 Conclusions

One of the claimed advantages of the EDAs over other evolutionary algorithms is that they involve less parameters to be chosen by the user. Although this generalisation may be true

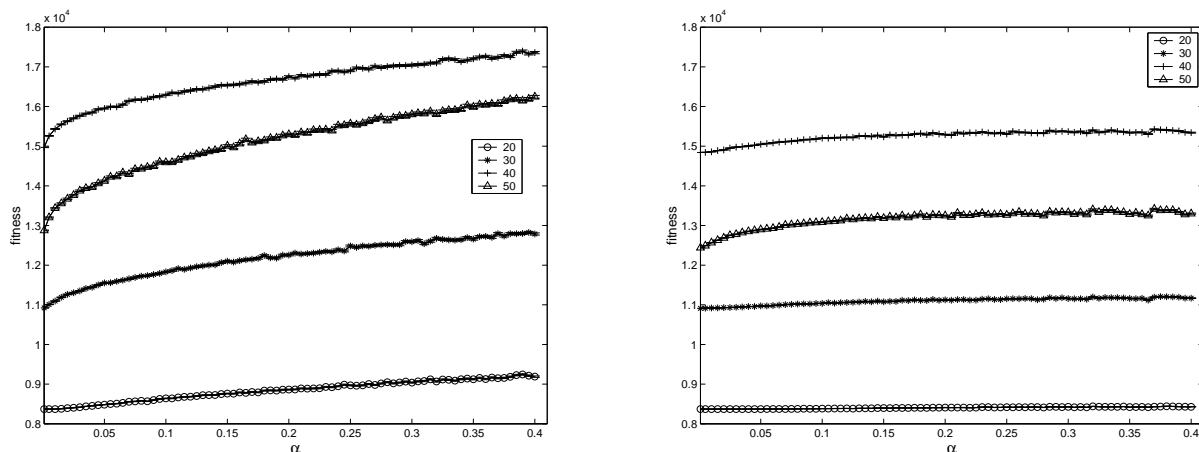


Figure 4: Simulations on PBIL on the P-median problem for 20, 30, 40 and 50 facilities. Mean of the best fitness found on 1000 trials for 80 different values of α . **Left:** PBIL. **Right:** DBPBIL.

it is undeniable that there still exist decision parameters in EDAs. In particular, PBIL has shown to be strongly sensitive to such a parameters. As important as diminishing the number of parameters in an algorithm is making the algorithm robust regarding changes in its parameters. This work investigated the performance of the Detailed Balance PBIL on two different test problems and also extended its application to a non-binary codification. On both test problems DBPBIL has shown to be much more robust than PBIL. The results obtained suggest that the implementation of detailed balance to PBIL is indeed an efficient method for diminishing the convergence pressure imposed by the probabilistic model in PBIL. DBPBIL has not only found the global optimum a greater number of times, but has also found better solutions than PBIL for practically all values of the parameter α on both test problems. To turn to other EDAs, an interesting question would be: does the convergence pressure imposed by the probabilistic model affect the performance of the more complex EDAs models? To conclude, the implementation of detailed balance to a more complex EDA, namely the Mutual-Information-Maximising Input Clustering, De Bonet et al. [3] is ongoing.

References

- [1] Baluja S. Population based incremental learning: A method for integrating genetic search based function optimisation and competitive learning. Carnegie Mellon University, 1994. No. CMU-CS-94-163. Pittsburg, PA.
- [2] Correa E.S., Steiner M.T.A., Freitas A.A., Carnieri C. A Genetic Algorithm for the P-median Problem In: Genetic and Evolutionary Computation Conference - GECCO 2001, 2001, San Francisco, California. Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2001. San Francisco, California: Morgan Kaufmann Publishers, 2001.
- [3] De Bonet J.S., Isbell C.L., Viola P. MIMIC: Finding optima by estimating probability densities. Advances in Neural Information Processing Systems, Cambridge. The MIT Press.
- [4] Shapiro J.L. The sensitivity of PBIL to its learning rate, and how detailed balance can remove it. To appear in: Foundations of Genetic Algorithms VII, 2002.