# Extending Population-Based Incremental Learning to Continuous Search Spaces


Michèle Sebag[1,2] and Antoine Ducoulombier[2,1]

LMS, CNRS UMR 7649,
Ecole Polytechnique
91128 Palaiseau Cedex
Michele.Sebag@polytechnique.fr

LRI, CNRS URA 410,
Université d'Orsay
91405 Orsay Cedex
Antoine.Ducoulombier@lri.fr



**Abstract.** An alternative to Darwinian-like artificial evolution is offered by *Population-Based Incremental Learning* (PBIL): this algorithm memorizes the best past individuals and uses this memory as a distribution, to generate the next population from scratch.

This paper extends PBIL from boolean to continuous search spaces. A Gaussian model is used for the distribution of the population. The center of this model is constructed as in boolean PBIL. Several ways of defining and adjusting the variance of the model are investigated.

The approach is validated on several large-sized problems.


## 1 Introduction

Evolutionary algorithms (EAs) [13, 6, 5] are mostly used to find the optima of some fitness function $\mathcal{F}$ defined on a search space $\Omega$.

$$\mathcal{F} : \Omega \rightarrow \mathbb{R}$$

From a machine learning (ML) perspective [9], evolution is similar to *learning by query*: Learning by query starts with a void hypothesis and gradually refines the current hypothesis through asking questions to some oracle.

In ML, the sought hypothesis is the description of the target concept; the system generates examples and asks the oracle (the expert) whether these examples belong to the target concept. In EA, the sought "hypothesis" is the distribution of the optima of $\mathcal{F}$; the system generates individuals and asks the oracle (a routine or the user) what their fitness is. In all cases, the system alternatively generates questions (examples or individuals) depending on its current hypothesis, and refines this hypothesis depending on the oracle's answers.

One core difference between ML and evolution is that ML, in the artificial intelligence vein, manipulates high-level, or *intensional* description of the hypothesis sought. Conversely, evolution deals with a low-level, or *extensional* description of the sought distribution: the distribution of the optima is represented by a collection of individuals (the current population).

The *Population Based Incremental Learning* (PBIL) approach bridges the gap between ML and EAs: it explicitly constructs an intensional description of the optima of $\mathcal{F}$, expressed as a distribution on $\Omega$ [2, 3]. This distribution is alternatively used to generate the current population, and updated from the best individuals of the current population. The advantage of the approach is that, as claimed throughout artificial intelligence [12], the higher level the information, the more explicit and simple the information processing can be. And indeed, PBIL involves much less parameters than even the canonical GAs [6].

PBIL was designed for binary search spaces. It actually constructs a distribution on $\Omega = \{0,1\}^N$ represented as an element of $[0,1]^N$. The basics of this scheme are first briefly recalled in order for this paper to be self contained (section 2). Our goal here is to extend this scheme to a continuous search space $\Omega \subseteq \mathbb{R}^N$. Continuous PBIL, noted $PBIL_C$, evolves a Gaussian distribution on $\Omega$ noted $\mathcal{N}(X, \sigma)$. The center $X$ of the distribution is evolved much like in the binary case; evolving the standard deviation $\sigma$ of this distribution is more critical, and several heuristics to this aim are proposed (section 3). $PBIL_C$ is finally validated and compared to evolution strategies on several large-sized problems (section 4). The paper ends with some perspectives for further research.

## 2 Binary PBIL

### 2.1 Principle

| | | | | |
|---|---|---|---|---|
| Std. Evolution | $\Pi^t$ | $\longrightarrow$ | $\Pi^{t+1}$ | |
| | $\downarrow$ | | $\downarrow$ | |
| | $h^t$ | | $h^{t+1}$ | *unused or marginal information* |
| PBIL | $\Pi^t$ | | $\Pi^{t+1}$ | *marginal information* |
| | $\uparrow$ | $\searrow$ | $\uparrow$ | |
| | $h^t$ | $\longrightarrow$ | $h^{t+1}$ | |

*Figure 1. Comparing the generation steps in standard evolution and PBIL*

Let $\Pi$ denote a population of individuals in $\Omega = \{0,1\}^N$. An element $h$ of $\mathcal{H} = [0,1]^N$ can be associated to $\Pi$, by defining $h_i$ as the fraction of individuals in $\Pi$ having their i-th bit set to 1. Conversely, an element $h$ in $\mathcal{H}$ defines a distribution over $\Omega$: one draws an element $X = (X_1, \ldots, X_N)$ in $\Omega$ by setting $X_i$ to 1 with probability $h_i$.

PBIL relies on the following premises [2]: a) if evolution succeeds, the population $\Pi$ converges toward a single[1] optimum of $\mathcal{F}$; b) the more converged the population $\Pi$, the better it is represented by $h$. Assuming these, PBIL discards all information in the population not contained in $h$: The population is simply considered as a manifestation of $h$. The attention is thus shifted from evolving $\Pi$ by means of mutation and recombination, to evolving $h$ (Fig. 1). To this aim, PBIL uses the information contained in the current population $\Pi^t$: $h$ is evolved,

[1] This claim obviously makes no room for diversity preserving mechanisms [8].

or rather updated, by relaxation from the best individual $X^{max}$ in $\Pi^t$:

$$h^{t+1} = (1-\alpha) \,.\, h^t \;+\; \alpha \,.\, X^{max}, \qquad \alpha \; in \; ]0,1[$$

Distribution $h^t$ can be viewed as the memory of the best individuals generated by evolution. Relaxation factor $\alpha$ corresponds to the fading of the memory: the higher $\alpha$, the faster $h^t$ moves toward the current local optimum.

In contrast to standard evolution, PBIL explicitly explores the space $\mathcal{H}$ of distributions on $\Omega$. And, as noted already, this higher level representation allows for a simpler information processing: besides the population size, PBIL involves a single key parameter, $\alpha$, to be compared to the various parameters controlling mutation and recombination. Further, the exploration is deterministic, in the sense that $h^t$ is deterministically updated from the current population[2].

### 2.2 Discussion

Let us reformulate PBIL as a learning-by-query algorithm, by defining a partial generality order on the set of distributions $\mathcal{H}$. The generality of a distribution $h$ is clearly related to the diversity of the population generated from $h$, and the diversity of the population with regard to bit $i$ is inversely proportional to $|h_i - .5|$. Accordingly, a distribution $h$ is more specific than $h'$, if, for each bit $i$, either $0 \leq h_i \leq h'_i \leq .5$, or $.5 \leq h'_i \leq h_i \leq 1$.

PBIL initializes $h$ to the most general distribution $h^0 = (.5 \ldots, .5)$, and gradually specializes it along generations. Let $X^h$ denote the (boolean) individual most similar to $h^t$; then, $h^t$ is specialized on all bits $i$ such that $X^h_i = X^{max}_i$. The complete convergence of the scheme is avoided as $h^t_i$ never reaches 0 or 1; in theory, PBIL can generate any individual at any time.

In practice, PBIL can suffer from premature convergence. This happens when $h^t$ gets too specific[3], and no new good individual is discovered. PBIL offers two heuristics to resist premature convergence [2]:

• Using the average of the two best individuals in $\Pi^t$, rather than the single best one. This way, $h^t$ is generalized on all bits discriminating these individuals.

• Perturbing $h^t$ with a Gaussian noise: with a given probability (5%), a Gaussian variable with a low standard deviation is added to $h^t_i$. This way, the center of the distribution is durably perturbed, which helps escaping from local minima.

A more fundamental limitation of PBIL comes from the distribution space, which implicitly assumes the linear separability of the problem (genes are considered independent). This distribution space appears too poor to fit complex fitness landscapes, such as the Long Path problem [7]. Previous experiments show that distributions used in PBIL have difficulties to overlap the narrow path [14]. Recent extensions to PBIL have considered richer distribution spaces [4].

---

2 This raises the question of whether PBIL still pertains to the field of "Parallel problem solving from nature": is "nature" allowed to explicitly manipulate distributions? Still, a definition of "nature" is clearly beyond the scope of the paper.

3 Parameter $\alpha$ partly controls the specificity of $h^t$, and plays the same role as selection in GAs: the diversity decreases, everything else being equal, as $\alpha$ goes to 1.

## 3 Continuous PBIL

This section first briefly discusses a previous attempt to extend PBIL to continuous search spaces, then details the proposed method and outlines $PBIL_C$.

### 3.1 Continuous PBIL with dichotomic distributions

To the best of our knowledge, the only extension of PBIL to continuous search spaces has been proposed in [15]. This algorithm explores the search space much like the delta-coding approach [17]. The domain of each gene is divided into two intervals ("low" and "high" values); the current distribution $h$ ($h$ in $[0,1]^N$) is used to determine which interval an individual belongs to:

$$X_i \in [a,b] \qquad Probability(X_i > \frac{a+b}{2}) = h_i$$

$X_i$ is then drawn with uniform probability in the selected interval.
• At each generation, $h$ is updated like in the boolean case, by memorizing whether the best individual takes low or high values for each gene:

$$h_i^{t+1} = (1-\alpha)\ .\ h_i^t \ + \ \alpha\ .\ (X_i^{max} > \frac{a+b}{2})$$

• When $h_i$ gets specific enough ($h_i < .1$ or $h_i > .9$), the population gets concentrated in a single interval (resp. $[a, \frac{a+b}{2}]$ or $[\frac{a+b}{2}, b]$). The search is then focused: the domain of the gene is set to the interval considered and $h_i$ is reinitialized to .5.

In this scheme, evolution gradually focuses on the region most often containing the best individuals. One limitation is that a region which has been discarded at some point is hardly explored ever after, and this violates the ergodicity requirement. Furthermore, the search might be insufficiently focused, given the poor (uniform) distribution used within the selected interval.

### 3.2 Continuous PBIL with Gaussian distributions

Our approach rather explores Gaussian distributions $\mathcal{N}(X,\sigma)$ on the search space $\Omega$, given as products of Gaussian distributions $\mathcal{N}(X_i,\sigma_i)$ on each gene domain. With no loss of generality, $\Omega$ is set to $[0,1]^N$ in the following.

Like *PBIL*, $PBIL_C$ starts with a rather general distribution; then it alternatively uses this distribution to draw the population, and uses the population to update the distribution. The center of the distribution $X^t$ is initialized to the center of the search space $(.5,\ldots,.5)$. At each generation, $X^t$ is updated from a linear combination of the two best and the worst individuals in the current population, inspired from PBIL and Differential Evolution [16]:

$$X^{t+1} = (1-\alpha)\ .\ X^t \ + \ \alpha\ .\ (X^{best,\,1} + X^{best,\,2} - X^{worst})$$

The diversity of the population, controlling the convergence of evolution, depends on the variance $\sigma = (\sigma_1, \ldots \sigma_N)$ of the distribution. Several heuristics have been

investigated to adjust parameters $\sigma_i$.
A• The simplest possibility is to use a constant value. The trade-off between exploration and exploitation is thus settled once for all: the search cannot become too specific and it cannot be speeded up either.
B• A second possibility is to make evolution itself adjust $\sigma$. $PBIL_C$ here proceeds exactly as a self-adaptive $(1, \lambda)$-evolution strategy (ES)[4] where $\lambda$ stands for the size of the population, except that the parent is replaced by the center $X^t$ of the distribution.
C• A third possibility is to adjust $\sigma$ depending on the diversity of the current best offspring; $\sigma^t$ is then set to the variance of the $K$ best current offspring:

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{K}(X_i^j - \bar{X}_i)^2}{K}}$$

where $\bar{X}$ denotes the average of the best $K$ offspring $X^1, \ldots X^K$.
D• Last, $\sigma$ can be learned in the same way as $X$ itself, by memorizing the diversity of the $K$ best offspring:

$$\sigma_i^{t+1} = (1-\alpha)\sigma_i^t + \alpha\sqrt{\frac{\sum_{j=1}^{K}(X_i^j - \bar{X}_i)^2}{K}}$$

### 3.3 Discussion

At first sight, $PBIL_C$ is quite similar to a $(1, \lambda)$-ES, the $\lambda$ offspring being generated from the single parent $(X^t, \sigma^t)$. The difference is twofold.
• In $(1, \lambda)$-ES, the parent is simply replaced by the best offspring, whereas $PBIL_C$ updates $X^t$ by relaxation. Let any offspring $X^k$ be written $X^t + Z^k$, with $Z^k$ being a random vector drawn according to $\mathcal{N}(0, \sigma^t)$. Then it comes:

$$X^{t+1} = (1-\alpha)X^t + \alpha(X^{best,1} + X^{best,2} - X^{worst}) = X^t + \alpha(Z^{best,1} + Z^{best,2} - Z^{worst})$$

The evolution of $X^t$ can be viewed as a particular case of *weighted recombination* as studied by Rudolph [11]; a theoretical analysis shows that weighted recombination with optimal weights should be preferred to the simple replacement of the parents. Interestingly, the heuristic recombination used in $PBIL_C$ is intermediate between two particular cases with good theoretical properties (for $\mathcal{F}(X) = \sum X_i^2$): the half sum of the two best offspring, and the difference of the best and the worst offspring.
$PBIL_C$ uses fixed, hence non-optimal, weights; but note that $\alpha$ intervenes as an additional scaling factor, controlling the variance of $X^t$.
• Independently, the variance of $X^t$ is also controlled from $\sigma^t$. $PBIL_C$ uses global

[4] In self-adaptive ES, besides the $X_i$ an individual $X$ carries the variance $\sigma_i$ of the mutation to be applied on the $X_i$ [13, 1]: Mutation first evolves the $\sigma_i$, then uses the new $\sigma_i$ to perturb the $X_i$. Evolution thus hopefully adjusts the $\sigma_i$ "for free", at the individual level.

mechanisms (options A, B and D) to adjust $\sigma^t$, by opposition to the local adjustment of $\sigma$ achieved by self-adaptive mutation. Actually, the adjustment of $\sigma$ (option D) much resembles the 1/5th rule used to globally adjust $\sigma$ in early evolution strategies [10]. The difference is that the 1/5th rule criterion compares the offspring to the parents, and considers whether a sufficient fraction of offspring is more fit than the parents. In opposition, $PBIL_C$ only examines the diversity of the best fit offspring: it does not need to restrict the exploration, even if the offspring are less fit than the parent, because the center of the explored region moves more slowly than in standard ES.

To sum up, $PBIL_C$ controls the exploration-exploitation tradeoff in a way rather different from that of $(1, \lambda)$-ES. First of all, the single parent does not jump directly to a desirable location (the best offspring, or some weighted combination of the remarkable offspring), but rather makes a very small step toward this desirable location (e.g. $\alpha$ is set to $10^{-2}$ in the experiments). Variance $\sigma$ is adjusted in a similarly cautious way.
It appears that ES takes instant decisions, on the basis of the instant information. On the opposite, $PBIL_C$ maintains a long-term memory, slowly updated from the instant information, and bases its cautious decisions on this long-term memory.

## 4 Validation

This section describes the goal of the experiments and the problems considered. We then report and discuss the results obtained.

### 4.1 Experiment Goals and Problems

Our goal is to study the respective advantages of evolving extensional vs intensional information about the fitness landscape. Practically, $PBIL_C$, evolving an intensional information represented as a distribution, is compared to self-adaptive evolution strategy, evolving an extensional information represented as usual as a population.

| Notation | Definition | | | Domain $\Omega$ |
|---|---|---|---|---|
| $F_1$ | $\frac{100}{10^{-5}+\sum_i \lvert y_i \rvert}$ | with | $y_1 = x_1$<br>$y_i = x_i + y_{i-1}, i \geq 2$ | $[-3,\ 3]^{100}$ |
| $F_2$ | $\frac{100}{10^{-5}+\sum_i \lvert y_i \rvert}$ | with | $y_1 = x_1$<br>$y_i = x_i + sin y_{i-1},\ i \geq 2$ | $[-3,\ 3]^{100}$ |
| $F_3$ | $\frac{100}{10^{-5}+\sum_i \lvert y_i \rvert}$ | with | $y_i = .024 * (i+1) - x_i$ | $[-3,\ 3]^{100}$ |
| $F_6$ | $\sum_i (x_i^2 - A cos(2\pi\, x_i)) + 100A$ | | | $[-5,\ 5]^{100}$ |
| $F_7$ | $\sum_i -x_i sin\sqrt{x_i}$ | | | $[-30,\ 30]^{100}$ |
| $F_8$ | $\sum_i x_i^2 - \prod_i cos(\frac{x_i}{\sqrt{i+1}})$ | | | $[-100,\ 100]^{100}$ |

*Table 1: Fitness functions considered,* $i = 1 \ldots 100$

We deliberately consider large-sized search spaces ($N = 100$) for the following reason. In low or middle-sized spaces, populations or distributions might convey similarly accurate information about the fitness landscape. This is not true in large-sized spaces: any reasonable number of point s can only convey a very poor information about $\mathbb{R}^{100}$. Experimenting $PBIL_C$ in $\mathbb{R}^{100}$ will show how intensional evolution stands the curse of dimensionality.

Functions and search spaces considered are displayed in Table 1. Functions $F_1$ to $F_3$ have been used to evaluated binary PBIL [2]. Besides the size of the search space, $F_1$ and $F_2$ suffer from an additional difficulty, epistasis (the genes are linked via the $y_i$). Functions $F_6$ to $F_8$ have been extensively studied in the literature, for lower-sized search spaces ($N \leq 30$).

### 4.2 Experimental setting

We used two reference algorithms: boolean PBIL working on a discretization of the continuous problem (each continuous variable is coded through 9 binary variables), using either a binary or a Gray coding; and a $(10 + 50)$-ES with self adaptive mutation [1]. In the PBIL case, the size $\lambda$ of the population is set to 50 and the relaxation factor $\alpha$ is set to .01.

$PBIL_C$ involves the same setting as PBIL ($\lambda = 50$ and $\alpha = .01$). Four options regarding the variance $\sigma$ of the distributions have been considered (section 3.2):
A• Constant variance.
B• Self-adapted variance: $PBIL_C$ here behaves like a self-adaptive $(1, \lambda)$-ES, except that the parent is replaced by $X^t$.
C• Instant variance: $\sigma_i$ is set to the variance of the best $K$ offspring in the population. Several values of $K$ were considered: $\lambda/2$, $\lambda/3$, $\lambda/5$.
D• Relaxed variance: $\sigma_i$ is the variance of the best $K$ offspring relaxed over the past generations; the relaxation factor is again set to $\alpha = .01$.

### 4.3 Results

| **Algorithm** | $\sigma$ | **F1** | **F2** | **F3** |
|---|---|---|---|---|
| (10+50)-ES | | 2.91 ±0.45 | 7.56 ±1.52 | 399.07 ±6.97 |
| PBIL + binary coding | | 2.12 | 4.40 | 16.43 |
| PBIL + Gray coding | | 2.62 | 5.61 | 366.77 |
| $PBIL_C$ | A: $\sigma_i = .02$ | 3.56 ±0.36 | 5.87 ±0.42 | 15.02 ±.76 |
| | A: $\sigma_i = .05$ | 3.95 ±0.37 | 8.08 ±0.52 | 28.32 ±1.46 |
| | B: $\sigma$ self-adapt. | 2.41 ±0.22 | 4.49 ±0.50 | 3.04 ±.34 |
| | C: $K = \lambda/2$ | 2.89 ±0.36 | 3.52 ±0.41 | 5.25 ±.59 |
| | D: $K = \lambda/2$ | **4.65** ±0.49 | **10.45** ±0.96 | 685 ±43 |
| | D: $K = \lambda/3$ | **4.40** ±0.41 | **11.18** ±1.36 | **2623** ±204 |
| | D: $K = \lambda/5$ | **4.76** ±0.78 | **10.99** ±1 | **4803** ±4986 |

*Table 2: Best Fitness (averaged on 20 runs) for 200,000 evaluations*
*Best results indicated in bold* *Exact optimum of $F_1, F_2$ and $F_3$ = $10^7$*

Table 2 displays the results obtained on functions $F_1$, $F_2$ and $F_3$. Results obtained by boolean PBIL are taken from [2]; additional results not reported here, show that boolean PBIL significantly outperforms several variants of GAs and Hill-Climbers on these functions. Note that all algorithms end rather far from the actual optimum ($10^7$). Still, $PBIL_C$ significantly outperforms standard ES on these problems — provided that the variance $\sigma$ of the distribution is adequately set. Note also that $PBIL_C$ outperforms PBIL itself, working on a binary or Gray discretization of these continuous problems. This might be due either to the loss of information entailed by discretization, or because PBIL, as already mentioned, explores a too restricted distribution space.

The worst results of $PBIL_C$ are obtained when $\sigma$ is self-adapted or set to the diversity of the current best offspring (options B and C); they are due to a fast decreasing of $\sigma$. And, in retrospect, a vicious circle occurs when $\sigma$ tightly depends on the diversity of the offspring: the less diverse the offspring, the smaller $\sigma$, hence the less diverse the offspring...

Setting $\sigma$ to a constant value (option A; the particular values were chosen after 10,000 evaluations preliminary runs) leads to satisfactory results, even outperforming those of standard ES. Further experiments will show whether this is rather due to the superiority of weighted recombination (replacing a parent by a combination of offspring) over replacement — or to the "long-term memory" effect, as the parent slowly moves toward the weighted combination of the offspring instead of jumping there.

The best option appears to learn the variance $\sigma$ in the same way as the center of the distribution $X^t$ (option D). Further, the fraction $K$ of the offspring considered to update $\sigma$ apparently is not a critical parameter[5].

| **Algorithm** | $\sigma$ | **F6** | **F7** | **F8** |
|---|---|---|---|---|
| (10+50)-ES | | 174 ±29 | **-192.75** ±18.18 | 489 ±115 |
| $PBIL_C$ | B: $\sigma$ self-adapt. | **44.02** ±6.44 | -44.73 ±32 | 71.62 ±14 |
| $PBIL_C$ | D: $K = \lambda/2$ | **45.19** ±4.03 | **-158.47** ±40.87 | 11 $10^{-6}$ $\pm 10^{-6}$ |
| $PBIL_C$ | D: $K = \lambda/3$ | **44.67** ±5.21 | **-167** ±34 | $\mathbf{10^{-6}}$ $\pm 10^{-7}$ |
| $PBIL_C$ | D: $K = \lambda/5$ | **44.43** ±4.52 | **-169** ±27 | $\mathbf{10^{-7}}$ $\pm 10^{-8}$ |

*Table 3: Best Fitness (averaged on 20 runs) for 200,000 evaluations*
*Best results indicated in bold   Exact optimum of $F_6$ and $F_8$ = 0*

These trends are confirmed by preliminary experiments on $F_6$, $F_7$ and $F_8$ (Table 3): $PBIL_C$ significantly outperforms self-adaptive ES on two out of the three problems, the best option for adjusting $\sigma$ being the relaxation from a small fraction of the best offspring.

[5] This holds for all problems except $F_3$, which is the problem with most diversity in the fitness of the offspring. This might be an indication for choosing $K$ adaptively: e.g. retain the offspring whose fitness is greater than a given function of the average fitness and deviation of the fitness in the current population.

## 5 Conclusion

The main originality of PBIL is to reformulate evolution into new, higher-level, terms: rather than specifying all operations needed to transform a population into another population (selection, recombination, mutation, replacement), one only specifies how to evolve or update a distribution given the additional information supplied by the current population. At this level, many core traits of evolution (e.g. diversity, speed of changes) are explicit and can be directly controlled.
Overall, evolution shifts from the stochastic exploration of the search space $\Omega$, to learning a distribution on $\Omega$ by reinforcement from the current population.

This paper extends PBIL from boolean to continuous search spaces, by learning Gaussian distributions $\mathcal{N}(X, \sigma)$. The resulting $PBIL_C$ algorithm can be thought of as a $(1, \lambda)$-ES, with the following differences. ES takes instant decisions, on the basis of the instant information. $PBIL_C$ maintains a long-term memory, takes its decisions on the basis of this long-term memory, and slowly updates the memory from the instant information. Practically, the parent of a $(1, \lambda)$-ES jumps toward the best offspring; in opposition, the center of the distribution in $PBIL_C$ cautiously moves toward a weighted combination of the offspring.
Similarly, self-adaptive ES locally adjusts the variance of mutation by means of instant decisions; in opposition, $PBIL_C$ cautiously updates the variance from the global diversity of the best offspring.

One argument for learning distributions is that it expectedly scales up more easily than evolving populations: a reasonable size population gives little information on large-sized search space. Experimental results on large-sized problems show that $PBIL_C$ actually outperforms standard ES on five out of six problems (with one or two orders of magnitude) and also outperforms the original PBIL working on a discretized version of the continuous problems considered.

Nevertheless, given the size of the search space, $PBIL_C$ ends rather far from the optimum on four out of six problems. Further experiments will consider other problems, and study how $PBIL_C$ behaves in the last stages of exploitation. Another perspective of research is to evolve several distributions rather than a single one. This would relax the main limitation of the PBIL scheme, that is, the fact that it can only discover a single optimum. Indeed, learning simultaneously several distributions is very comparable to evolving several species. The advantage is that comparing an individual to a few distributions might be less expensive and again more transparent, than clustering the population, adjusting the selection or the fitness function to ensure the co-evolution of species.

**Acknowledgments**

Many thanks to Marc Schoenauer, for many valuable comments, and to the second anonymous referee, for very insightful comments and suggestions.

## References


1. T. Bäck. *Evolutionary Algorithms in theory and practice.* New-York:Oxford University Press, 1995.
2. S. Baluja. An empirical comparizon of seven iterative and evolutionary function optimization heuristics. Technical Report CMU-CS-95-193, Carnegie Mellon University, 1995.
3. S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithms. In A. Prieditis and S. Russel, editors, *Proc. of ICML95*, pages 38–46. Morgan Kaufmann, 1995.
4. S. Baluja and S. Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proc. of ICML97*. Morgan Kaufmann, 1997.
5. D. B. Fogel. *Evolutionary Computation. Toward a New Philosophy of Machine Intelligence.* IEEE Press, Piscataway, NJ, 1995.
6. D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989.
7. J. Horn and D.E. Goldberg. Genetic algorithms difficulty and the modality of fitness landscapes. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 243–269. Morgan Kaufmann, 1995.
8. S. W. Mahfoud. A comparison of parallel and sequential niching techniques. In L. J. Eshelman, editor, *Proc. of ICGA'95*, pages 136–143. Morgan Kaufmann, 1995.
9. T.M. Mitchell. *Machine Learning.* McGraw Hill, 1995.
10. I. Rechenberg. *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution.* Fromman-Holzboog Verlag, Stuttgart, 1973.
11. G. Rudolph. *Convergence Properties of Evolutionary Algorithms.* Kovac, Hamburg, 1997.
12. S. Russell and A. Norwig. *Artificial Intelligence, a modern approach.* Prentice Hall, 1995.
13. H.-P. Schwefel. *Numerical Optimization of Computer Models.* John Wiley & Sons, New-York, 1981. 1995 – $2^{nd}$ edition.
14. M. Sebag and M. Schoenauer. Mutation by imitation in boolean evolution strategies. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proc. of PPSN-IV*, pages 356–365. Springer-Verlag, LNCS 1141, 1996.
15. I. Servet, L. Trave-Massuyes, and D. Stern. Telephone network traffic overloading diagnosis and evolutionary computation technique. In *Artificial Evolution'97*, pages 137–144. Springer Verlag, LNCS 1363, 1997.
16. R. Storn and K. Price. Minimizing the real functions of the ICEC'96 contest by differential evolution. In *Proc. of ICEC96*, pages 842–844, 1996.
17. L.D. Whitley, K. Mathias, and P. Fitzhorn. Delta coding: an iterative strategy for GAs. In R. K. Belew and L. B. Booker, editors, *Proc. of ICGA'89*, pages 77–84. Morgan Kaufmann, 1989.


This article was processed using the LaTeX macro package with LLNCS style