Mimetic Evolution

Mathieu Peyral, Antoine Ducoulombier, Caroline Ravisé, Marc Schoenauer, Michèle Sebag

Mimetic Evolution

Mathieu Peyral[1] Antoine Ducoulombier[2] Caroline Ravisé[1,2]
Marc Schoenauer[1] Michèle Sebag[1,2]

(1): CMAP & LMS
URA CNRS 756 & 317
Ecole Polytechnique
91128 Palaiseau Cedex

(2): Equipe I & A, LRI
URA CNRS 410
Université d'Orsay
91405 Orsay Cedex

{*Prenom.Nom*}*@polytechnique.fr*

Abstract. Biological evolution is good at dealing with environmental changes: Nature ceaselessly repeats its experiments and is not misled by any explicit memory of the past. This contrasts with artificial evolution most often considering a fixed milieu, where re-generating an individual does not bring any further information.
This paper aims at avoiding such uninformative operations, via some explicit memory of the past evolution: the best and the worst individuals previously met by evolution are respectively memorized within two virtual individuals. Evolution may then use these virtual individuals as social models, to be imitated or rejected. In mimetic evolution, standard crossover and mutation are replaced by a single operator, *social mutation*, which moves individuals farther away or closer toward the models. This new scheme involves two main parameters: the social strategy (how to move individuals with respect to the models) and the social pressure (how far the offspring go toward or away from the models).
Experiments on large-sized binary problems are detailed and discussed.

1 Introduction

Biological evolution takes place in a changing environment. Being able to repeat previously unsuccessful experiments is therefore vital. As the result of previous experiments might change, any explicit memory of the past might provide misleading indications. This could explain why all knowledge gathered by evolution is actually contained in the current genetic material and dispatched among the individuals.

Inversely, artificial evolution most often tackles optimization problems and considers *fixed* fitness landscapes, in the sense that the fitness of an individual does not vary along time and does not depend on the other individuals in the population. In this framework, which is the only one considered in the rest of the paper, the evaluation of an individual produces reliable information, and generating this individual again does not provide any further information. Memorizing somehow the past of evolution thus make sense, as it could prevent evolution from some predictable failures. This paper focuses on gathering an explicit collective memory (EC-memory) of evolution, as opposed to both the

implicit memory of evolution contained in the genetic material in the population, and the local parameters of evolution conveyed by the individuals, such as the mutation step size in Evolution Strategies [23]), or the type of crossover applicable to an individual [26].

Many works devoted to the control of evolution ultimately rely on some explicit collective memory of evolution. The memorization process can acquire numerical information; this is the case for the reward-based mechanism proposed by Davis to adjust the operator rates [5], the adjustment of penalty factors in SAT problems [6] or the construction of discrete gradients [11], among others. The memorization process can also acquire symbolic information, represented as rules or beliefs characterizing the disruptive operators (so that they could be avoided) [20], or the promising schemas [22].

Memory-based heuristics can control most steps of evolution: e.g. selection via penalty factors [6], operator rates [5], operator effects [11, 20]... Memory can even be used to "remove genetics from the standard genetic algorithm" [4, 3] as in the *Population Based Incremental Learning* (PBIL) algorithm. PBIL deals with binary individuals (in $\{0,1\}^N$) and it maintains the memory of the most fit individual encountered so far. This memory can be thought of as a *virtual individual*, belonging to $[0,1]^N$. It provides an alternative to the genetic-like transmission of the information between successive populations: any population is generated from scratch by sampling the discrete neighbors of this virtual individual. And the virtual individual is then updated from the best current individual.

Another approach, termed *Evolution by Inhibitions* (EBI), is inversely based on memorizing the worst individuals encountered so far; the memory is also represented by a virtual individual, termed the *Loser* [25]. This memory is used to evolve the current population, by means of a single new operator termed *flee-mutation*. The underlying metaphor is that the offspring aim at being farther away from the loser, than their parents. Incidentally, this evolution scheme is biased against exploring again unfit regions previously explored.

A new evolutionary scheme, restricted to binary search space and combining PBIL and *Evolution by Inhibitions*, is presented in this paper. The memory of evolution is thereafter represented by two virtual individuals, the *Winner* and the *Loser*[1]. These virtual individuals, or *Models*, respectively summarize the best and the worst individuals encountered so far by evolution. An individual can independently imitate, avoid, or ignore each one of the two models; a wide range of, so to speak, social strategies, can thereby be considered. For instance, the Entrepreneur imitates the Winner and ignores the Loser; the Sheep imitates the Winner and rejects the Loser; the Phobic rejects the Loser and ignores the Winner (the dynamics is that of Evolution by inhibitions [25]); the Ignorant ignores both models and serves as reference to check the relevance of the models. This new scheme of evolution, termed *mimetic evolution*, is rather inspired by social than genetic metaphors.

[1] Other metaphors, all likely politically incorrect, could have been used: the leader and the scapegoat, the yang and the yin, the knight and the villain,...

This paper is organized as follows. Section 2 briefly reviews related work dealing with virtual or imaginary individuals. Section 3 describes mimetic evolution and the social mutation operator replacing crossover and mutation. Social mutation is controlled from the user supplied social strategy, which defines the preferred direction of evolution of the individuals. Section 4 discusses the limitations of mimetic evolution, and studies the case where the dynamics of the models and the population go in a deadlock. Section 5 examines how far the offspring must go in the direction of the models; or, metaphorically, which social pressure should be exerted on the individuals. Mimetic mutation is validated on several large-sized binary problems, and the experimental results are detailed in section 6. Last, we conclude and present some perspectives of research.

2 State of the art

With no pretention to exhaustivity, this section examines how imaginary or virtual individuals have been used to support evolution. The central question still is the respective contribution of crossover and mutation to the dynamics of evolution [8, 18, 23]. Though the question concerns any kind of search space, only the binary case will be considered here.

The efficiency of crossover is traditionnally explained by the Building Block hypothesis [12, 9]. But a growing body of evidence suggests that crossover is also efficient because it operates large step mutations. In particular, T. Jones has studied the macro-mutation operator defined as crossing over a parent with a random individual[2]. Macro-mutation obviously does not allow the offspring to combine the building blocks of their two parents; still, macro-mutation happens to outperform standard crossover on benchmark problems, everything else being equal [14].

In retrospect, crossover can be viewed as a biased mutation. The bias depends on the population and controls both the strength and the direction of the mutation. The "mutation rate" of standard crossover, e.g. the Hamming distance between parents and offspring, depends on average on the diversity of the population; and the "mutation direction" of standard crossover (which genes are modified) also depends on the population.

On the other hand, binary mutation primarily aims at preserving the genetic diversity of the population. This can be done as well through crossover with specific individuals, deliberately maintained in the population to prevent the loss of genetic diversity. For instance, the Surrogate GA [7] maintains imaginary individuals such as the complementary of the best current individual, or all-0 and all-1 individuals; crossover alone thus becomes sufficient to ensure the genetic diversity of the population, and mutation is no longer needed. Another possibility is to deliberately introduce genotypic diversity by embedding the search space

[2] Note that this macro-mutation fairly resembles standard crossover during the first generations of evolution, especially for large populations.

Ω into $\{0,1\} \times \Omega$ and identifying the individuals 0ω and $1\bar{\omega}$, as done in Dual Genetic Algorithms [19]. Provided that the number of dual pairs (0ω and $1\bar{\omega}$) is above a given threshold, crossover can similarly replace mutation and ensure genetic diversity.

Evolution can also be supported by virtual individuals, i.e. individuals belonging neither to the population nor to the search space. This is the case in the PBIL algorithm, mentionned in the introduction, where the best individuals in the previous populations are memorized within a vector of $[0,1]^N$. This vector noted $\mathcal{M}$ provides an alternative to crossover and mutation, in that it allows PBIL to generate the current population from scratch: for each individual X and each bit i, value X_i is randomly selected such that $P(X_i = 1) = \mathcal{M}_i$ (where A_i denotes as usual the i-th component of A). $\mathcal{M}$ is initialized to $(0.5, 0.5, ..., 0.5)$ and it is updated from the best individual[3] X_{max} at each generation, by relaxation:

$$\mathcal{M} \leftarrow (1-\alpha)\mathcal{M} + \alpha X_{max}$$

where α in $[0,1]$ is the relaxation factor, which corresponds to the fading of the memory. The main advantage of PBIL is its simplicity: it does not involve any modification of the genetic material. The only information transmitted from one generation to another is related to the best individual; still, it is not necessarily sufficient to reconstruct this best individual. This might hinder evolution in narrow highly fit regions, such as encountered in the Long Path problem [13]. Practically, one sees that even if $\mathcal{M}$ is close to the path, the population constructed from $\mathcal{M}$ poorly samples the path [24].

Evolution by Inhibition involves the opposite memory, that is, the memory of the worst individuals in the previous populations. This memory noted $\mathcal{L}$ (for *Loser*) is also a vector of $[0,1]^N$, constructed by relaxation:

$$\mathcal{L} \leftarrow (1-\alpha)\mathcal{L} + \alpha X_{min}$$

x where X_{min} denotes the average of half the worst offspring, and α is the relaxation factor. In contrast with PBIL which uses $\mathcal{M}$ to generate a new population, $\mathcal{L}$ is actually used to evolve the current population via a specific operator termed *flee-mutation*. Flee-mutation replaces both mutation and crossover; for each individual X, it selects and flips the bits most similar to those of the loser (minimizing $|X_i - \mathcal{L}_i|$). The offspring thus is farther away from the loser, than the parent was. Metaphorically, the goal of this evolutionary scheme is: Be different from the Loser ! And incidentally, this reduces the chance for exploring again low fit regions.

The potential of evolution by inhibitions is demonstrated for appropriate settings of the flee-mutation rate (number of bits mutated): EBI then significantly outperforms PBIL [3, 25], which itself outperforms most standard discrete optimization algorithms (Hill-Climbers with multiple restarts, standard GAs, binary

[3] A more robust variant is to update $\mathcal{M}$ from the *two* best individuals and the worst individual in the population [3].

evolution strategies). But the adjustment of the flee-mutation rate remains an open question.

3 Mimetic evolution

This section focuses on combining evolution by inhibitions and PBIL. Two models memorizing respectively the best and worst individuals met in past generations, are constructed. These models are used to evolve individuals through a single evolution operator.

3.1 Winner-driven evolution of the population

The winner is built by relaxation from the best individuals of the current population, in the same way as in PBIL (section 2). Table 1 illustrates how the winner $\mathcal{W}$ and the loser $\mathcal{L}$ are built from the current population.

	1	2	3	4	5	Fitness	
X	0	0	1	0	0	high	
Y	1	1	1	1	1	high	
Z	0	1	1	0	1	high	
$d\mathcal{W}$	0.33	0.66	1	0.33	0.66		$\mathcal{W} \leftarrow (1-\alpha_w)\mathcal{W} + \alpha_w d\mathcal{W}$
S	0	0	0	1	0	low	
T	1	0	1	1	1	low	
U	1	0	0	1	1	low	
$d\mathcal{L}$	0.66	0	0.33	1	0.66		$\mathcal{L} \leftarrow (1-\alpha_l)\mathcal{L} + \alpha_l d\mathcal{L}$

Table 1. *Individuals and virtual individuals*

Let us first examine how $\mathcal{W}$ can help evolving individual X. Given the most fit individuals of the population (X, Y and Z), some possible causes for being fit are ($bit_2 = 1$), or ($bit_3 = 1$), or ($bit_5 = 1$) (a majority of the most fit individuals has those bits set to this value). Thus, one might want for instance to flip bit_2 and let bit_3 unchanged in X; this amounts to making X more similar to $d\mathcal{W}$, which goes to $\mathcal{W}$ in the limit. Metaphorically, X thus "imitates" the winner $\mathcal{W}$.

Practically, a model-driven mutation termed *social mutation* is implemented as follows. Given the number M of bits to flip (see section 5), one selects these M bits by tournament among the bits. For each one of these M bits, T bits $i_1, \ldots, i_T$ are uniformly selected in $1..N$, and the bit i such that it maximizes $|X_i - \mathcal{W}_i|$, is flipped. This way, the offspring actually reduces its distance to the winner.

This mechanism can be compared to the majority crossover of Syswerda, setting the value of the offspring to the majoritary value of the bit in the population when the two parents differ on that bit [27]. The difference between majority crossover and social mutation is twofold: the majority crossover takes into account *all* individuals in the *current* population; social mutation takes into account the *best* individuals in all *past* populations.

Social mutation is easily refined to also account for the loser. For instance, according to $d\mathcal{W}$, it might be a good idea to mutate bit 5; but $d\mathcal{L}$ suggests that $(bit_5 = 1)$ is not a factor of high fitness. This leads to select the bits to mutate, so that the offspring "imitates" $\mathcal{W}$ and "rejects" $\mathcal{L}$. Practically, one only modifies the tournament criterion: the winner of the tournament is the bit maximizing $|X_i - \mathcal{W}_i|$ - $|X_i - \mathcal{L}_i|$.

Note that the relaxation factors α_w and α_l respectively associated to the winner and the loser could be different, though they are set to the same value (10^{-2}) throughout this paper. In this case, $\mathcal{L}$ changes faster than $\mathcal{W}$.

3.2 Social strategies

However, there is no reason why an individual could only imitate the winner and reject the loser. A straightforward generalization is to define a pair (δ_W, δ_L) in

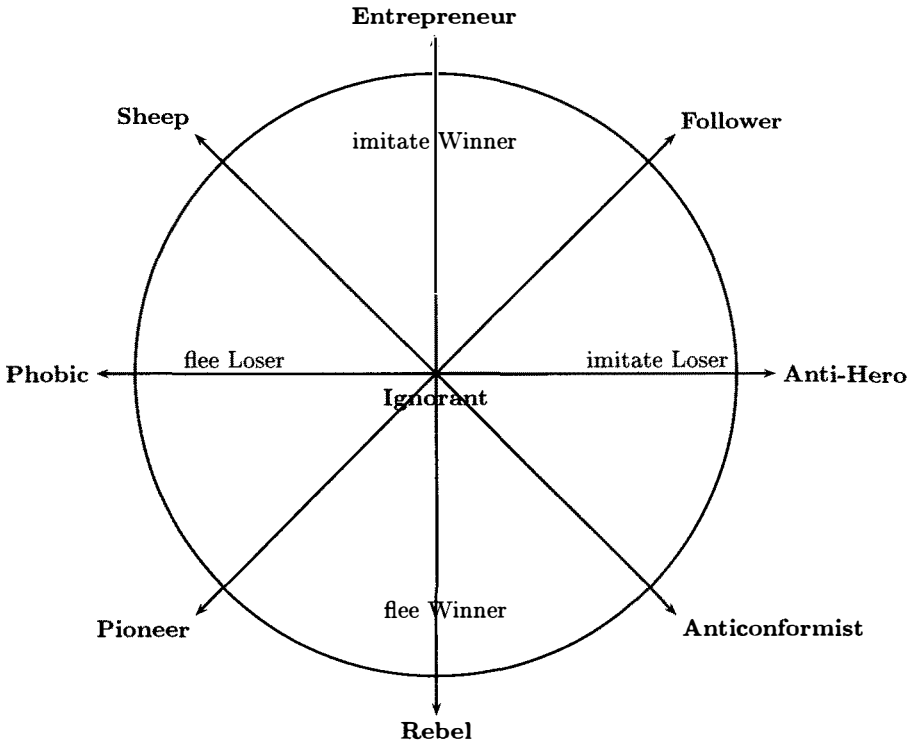

Fig. 1. Social Strategies

$\mathbb{R}^2$, and to select the bits to mutate as those maximizing

$$\delta_W |X_i - \mathcal{W}_i| + \delta_L |X_i - \mathcal{L}_i|$$

One sees that X imitates model $\mathcal{M}$ ($= \mathcal{W}$ or $\mathcal{L}$) if $\delta_M > 0$, rejects $\mathcal{M}$ if $\delta_M < 0$, and ignores $\mathcal{M}$ if $\delta_M = 0$. Social mutation finally gets parameterized by the pair (δ_W, δ_L), termed *social strategy*. Some of these strategies have been given names for the sake of convenience; obviously, other systems of metaphors could have been imagined. We distinguish mainly:

- The *entrepreneur*, that imitates the winner and ignores the loser;
- The *sheep*, that imitates the winner and rejects the loser;
- The *phobic*, that rejects the loser and ignores the winner;
- The *pioneer*, that rejects both the winner and the loser;
- The *rebel*, that rejects the winner and ignores the loser;
- The *ignorant*, that ignores both the loser and the winner.

One notices that social mutation is unchanged if δ_W and δ_L are multiplied by a positive coefficient. With no loss of information, social strategies are thus represented by angles ($\mathbb{R}^2$ being projected onto the unit circle). This angle sets the direction of evolution of the individuals, in the changing system of coordinates given by the winner and the loser. Figure 1 shows the directions corresponding to the main social strategies, with angle 0 corresponding to imitating the loser, and angle $\pi/2$ to imitating the winner.

4 Limitations

As expected, not all social strategies are relevant and some turn out to be misleading. A clear such case is when the winner is close to a local optimum; imitating the winner just drives the population to the local optimum, which does not allow for modifying the winner; a deadlock of mimetic evolution thus occurs.

In a general way, the population and the models depend on each other: at each generation, the best (worst) individuals in the population are used to maintain the models; and these models are used to evolve the population. This dynamics can produce several kinds of deadlock.

If the population does not make any progress, so does the winner; hence any strategy based on solely imitating the winner will fail, that is, fail to re-orient the search. In the meanwhile, the loser will probably fluctuate; either imitating or rejecting the loser could help sidestep this trap.

The loser can also mislead the population. If the loser is close to the optimum, rejecting the loser will deceptively lead the population in bad regions; the loser will then change, providing more reliable beacons and allowing the population to resume its pilgrimage toward the optimum. This may produce an oscillation, satellizing the population around the optimum. Concretely, let us consider a population climbing a hill. As long as the loser is on the same side of the hill as the population, fleeing the loser will make the population duly climb the hill. But

when individuals are equally distributed on both sides of the hill, the loser gets close to the optimum, and now exerts a repelling influence on the population. This perpetuates until the symmetry is broken. Still worse, the loser may be on the other side of the hill than the majority of the population (if the relaxation factor is too large, for instance), and make the population go down the hill...

The distribution of individuals may also cause any strategy to be deceptive. Consider an even distribution of the population between the two schemas, and assume that model $\mathcal{M}$ reflects this distribution (Table 2).

Schema 1	0	1	*	*	*	...	*
Schema 2	1	0	*	*	*	...	*
$\mathcal{M}$	0.5	0.5	*	*	*	...	*

Table 2: Stable distribution of the population

Any individual X most differs from $\mathcal{M}$ by bits 1 and 2 ($|X_i - \mathcal{M}_i| = .5, i = 1, 2$). Imitating $\mathcal{M}$ means reducing the differences between X and M, hence flipping preferably bits 1 and 2 (if the number of bits to mutate is greater than 2 — this will be discussed in the next section). But mutating bits 1 and 2 would perpetuate the distribution of the population between schemas 1 and 2.
Inversely, if the strategy is to reject $\mathcal{M}$, one wants to preserve the differences between X and $\mathcal{M}$, hence bits 1 and 2 will never be modified; and, everything being equal, this would preserve the distribution of the population too.

To sum up, observing a single model may perpetuate the traps visited by the model or the population. This is confirmed by the fact that PBIL enriches the computation of the winner with random perturbations [4], or computes the winner from the two best and the worst individuals [3]. We experimented similar heuristics based on gaussian perturbations of the models.

However, it turns out that guiding evolution according to *two* models is much more robust than with only one model: the influence of each model somehow moderates the influence of the other one, and makes it less harmful. Still, there certainly exists cases where the two combined models deliver deceptive indications. Further research will focus on the deceptivity of social strategies.

5 Social Pressure

Binary mutation is traditionnally parameterized by the mutation rate, usually very low, setting the average number of bits to mutate in the population. And the selection of the mutated bits is done at random.

In opposition, social mutation primarily concentrates on ordering the bits to mutate depending on the individual, via defining its desired direction of evolution (section 3.2). But how far the individual should go in this direction, i.e. the number M of bits to mutate, is still to determine. This parameter, termed *social pressure*, controls the balance of evolution between exploitation and exploration, respectively achieved for low and high values of M. In any case, the social

pressure must correspond to a much higher mutation rate than for standard mutation, as social mutation is meant to replace both mutation and crossover.

A previous work investigated two heuristics for the auto-adjustment of M in the framework of evolution by inhibitions [25]. The first heuristic is inspired from Davis [5], and proceeds by rewarding the values of M leading to fitness increases. The second one is inspired from self-adaptation [23, 1]: the number M is encoded within each individual, and evolution supposedly optimizes M as well as the genotypic information of the individual.
Unfortunately, none of our attempts succeeded in determining relevant global or local values of M; rather, all heuristics rapidly lead to setting $M = 1$. Mimetic evolution thereafter behaves as a standard hill-climber, and soon gets trapped in a local optimum.
In retrospect, reward-based adjustment tends to be risk-adverse, and favor options that bring small frequent improvements, over options that bring rare, though large, improvements; this explains why $M = 1$ is preferred.
On the other hand, the self-adaptation of M fails because the strong causality principle [21] is violated: finding the optimal M amounts to finding an optimal *discrete* value in a very restricted range (say $[2..N/10]$); no wonder that the convergence results of evolution strategies [23, 1] do not apply.

We therefore used fixed schedules to determine M_t at generation t. The simplest possibility is to set M_t to a fixed, user-supplied value M_0. A more sophisticated possibility is to decrease M_t from an initial, user-supplied, value M_0. We used a decreasing hyperbolic schedule borrowed from [2]:

$$M_t = \frac{1}{\frac{1}{M_0} + t \cdot \frac{1 - \frac{1}{M_0}}{T-1}} \tag{1}$$

where T is the maximum number of generations and t denotes the current generation. Social mutation then mutates exactly the integer part of M_t bits in each individual[4].

It now remains to set the initial (or fixed) value of M_t, M_0. We used off-line adjustments loosely inspired from Grefenstette's Virtual GA [10]. More precisely, a (1+10) binary ES is run for a few generations, for each considered value of M_0 (2..5 in the constant schedule, and $\{45, 90, 450\}$ in the decreasing schedule), and one chooses the value of M_0 leading to the best performance.

6 Experimental Validation

This section discusses the experimental results obtained by mimetic evolution on large-sized problems, and focuses on the role of social strategy.

[4] Another possibility, not investigated here, is to mutate on average M_t bits on the population [25].

6.1 Problems

The experiments consider some standard test functions: the OneMax, the Twin-Peaks and the deceptiv) Ugly[5] functions, taken on $\{0,1\}^{900}$.

Other functions are taken from [3], and respectively correspond to the binary coding and the Gray coding of the continuous functions on $[-2,56, 2.56[^{100}$ below:

$$\begin{aligned} y_1 &= x_1 \\ y_i &= x_i + y_{i-1},\ i \geq 2 \end{aligned} \qquad F_1 = \frac{100}{10^{-5} + \Sigma_i |y_i|}$$

$$F_3 = \frac{100}{10^{-5} + \Sigma_i |.024 * (i+1) - x_i|}$$

In the latter case, each continuous interval $[-2,56, 2.56[$ is mapped onto $\{0,1\}^9$; individuals thus belong to $\{0,1\}^{900}$.

The importance of the coding is witnessed, if necessary, by the fact that F_3 only reaches a maximum of 416,63 in its binary version, whereas the continuous optimum is 10^7; this is due to the fact that the continuous optimum ($X_i = .024 * (i+1)$) does not belong to the discrete space considered.

6.2 Experimental setting

The evolution scheme is a (10+50)-ES: 10 parents produce 50 offspring and the 10 best individuals among parents plus offspring are retained in the next population. A run is allowed 200,000 evaluations; all results are averaged on 10 independent runs.

The winner and the loser respectively memorize the best and the worst offspring of the current generation. The The relaxation factors α_w and α_l are both set to .01. The tournament size is set to 20.

Several reference algorithms have been experimented on functions F_1 and F_3: two variants of GAs (GA1 and GA2), two variants of Hill-climbers (HC1 and HC2) [3] and two variants of evolution strategies (ES1 and ES2) [25]. These algorithms served as reference for PBIL and evolution by inhibitions (INH). Another reference algorithm is given by mimetic evolution following an ignorant social strategy (an offspring is generated by randomly mutating 3 bits).

Function	HC1	HC2	AG1	AG2	AES	TES	PBIL	INH	IGNOR
F1 Binary	1.04	1.01	1.96	1.72	2.37	1.87	2.12	**2.99**	**2.98**
F3 Gray	**416.65**	**416.65**	28.35	210.37	380.3	**416.65**	366.77	246.23	385.90

Table 3. Reference results on F_1 and F_3.

More results, and the detailed description of the reference algorithms, are found in [25]. In this paper, we focus on the influence of the social strategy

[5] Defined as 300 concatenations of the elementary deceptive function U defined on $\{0,1\}^3$ as follows: $U(111) = 3$; $U(0XX) = 2$; otherwise $U = 0$.

parameter, and in particular, the social pressure (nb of bits mutated) is fixed to 3.

6.3 The influence of social strategies

A new visualization format is proposed on Figure 2, in order to compare the results obtained for different strategies on a given problem. As social strategies can be represented as angles, results are plotted in polar coordinates: point (ρ, θ) illustrates the results obtained for strategy θ, with ρ being the best average fitness obtained for this strategy. Two curves are plotted by joining all points (ρ, θ): the internal curve gives the results obtained for 50,000 evaluations, and the external one gives the results obtained for 200,000 evaluations. Both curves overlap (e.g. for OneMax and Ugly) when evolution reaches the optimum in about 50,000 evaluations or less.

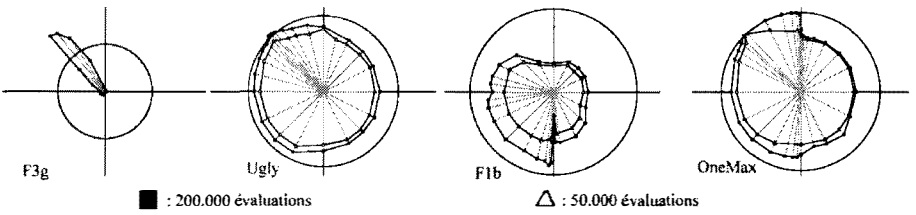

Fig. 2. *Summary results of social strategies*

The graphics obtained for functions TwinPeaks and F_1g, quite similar to respectively those of OneMax and $F_1 g$, are omitted given the space limitations.

Note that the shape of the curve obtained for 50,000 evaluations is very close to that obtained for 200,000 evaluations; in other words, the social strategy most adapted to a problem does not seem to vary along evolution.

In all cases, strategies in the left half space are significantly better than those in the right half space: for all functions considered, the recommended behavior with respect to the loser is the flight. In opposition, the recommended behavior with respect to the winner depends on the function: flight is strongly recommended in the case of F_1, where the Pioneer strategy is the best. Inversely, imitation is strongly recommended in the case of F_3, OneMax and Ugly, where the Sheep strategy is the best. This is particularly true for F_3, where the Sheep is significantly better than other strategies: the respective amounts of imitation of the winner and avoidance of the loser must be carefully adjusted.

7 Conclusion

Our working hypothesis is that the memory of evolution can provide worth information to guide the further steps of evolution.

Instead of memorizing only the best [4] or the worst [25] individuals, mimetic evolution memorizes all striking past individuals, the best and the worst ones. A population can thereby access two models, featured as the Winner and the Loser.

These models define a changing system of coordinates in the search space; an individual can thus be given a direction of evolution, or social strategy, expressed with respect to both models. And mimetic evolution proceeds by moving each individual in this direction. This paper has only considered the case of a single social strategy, fixed for all individuals, and focus on the influence of the strategy on the dynamics of evolution.

Evolution by inhibitions is a special case, driven by the loser model only, of mimetic evolution. As mimetic evolution involves both loser and winner models, the possible strategies get richer. As could be expected, 2-model-driven strategies appear more robust than 1-model-driven ones. Indeed, the more models are observed, the less predictable and deterministic the behavior of the population gets; and the less likely evolution goes in a deadlock.
Experimental results show that for each test function there exists a range of optimal social strategies (close to the Sheep in most cases; and to the Pioneer/Rebel in the others). And on most functions, these optimal strategies significantly outperform the "ignorant" strategy, which serves as reference and ignores both models.

In any case, it appears that memory contains relevant information, as the way we use it fortunately often allows for speeding up evolution. Obviously, other and more clever exploitations of this information remain to be invented. On the other hand, the use of the memory is tightly connected to its content: which knowledge exactly should be acquired during evolution ? Currently the models reflect, or "capitalize", the individuals; another level of memory would capitalize the dynamics of the population as a whole, i.e. the sequence over all generations of the social strategy leading from the best parents to the best offspring.

A further perspective of research is to self-adapt the social strategy of an individual, by enhancing the individual with its proper strategy parameters δ_W and δ_L. This way, evolution could actually benefit from a mixture of different social strategies, dispatched within the population[6].

Another perspective is to extend mimetic evolution to continuous search spaces. Computing the winner and the loser from continuous individuals is straightforward; but the question of how to use them is still more open, than in the binary case.

[6] Incidentally, this would reflect more truly the evolution of societies. But indeed social modelling is far beyond the scope of this work.

A last perspective is to see to what extent the difficulty of the fitness landscape for a standard GA is correlated to the optimal social strategy for this landscape (among which the ignorant strategy). Ideally, the optimal strategy would allow one to compare diverse components of evolution, in the line of the Fitness Distance Correlation criterion [15]. The advantage of such criteria is to provide a priori estimates of the adequacy of e.g., procedures of initialization of the population [16], or evolution operators [17].

References

1. T. Bäck. *Evolutionary Algorithms in theory and practice.* New-York:Oxford University Press, 1995.
2. T. Bäck and M. Schütz. Intelligent mutation rate control in canonical GAs. In Z. W. Ras and M. Michalewicz, editors, *Foundation of Intelligent Systems 9th International Symposium, ISMIS '96*, pages 158–167. Springer Verlag, 1996.
3. S. Baluja. An empirical comparizon of seven iterative and evolutionary function optimization heuristics. Technical Report CMU-CS-95-193, Carnegie Mellon University, 1995.
4. S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithms. In A. Prieditis and S. Russel, editors, *Proceedings of ICML95*, pages 38–46. Morgan Kaufmann, 1995.
5. L. Davis. Adapting operator probabilities in genetic algorithms. In J. D. Schaffer, editor, *Proceedings of the* 3^{rd} *International Conference on Genetic Algorithms*, pages 61–69. Morgan Kaufmann, 1989.
6. A.E. Eiben and Z. Ruttkay. Self-adaptivity for constraint satisfaction: Learning penalty functions. In T. Fukuda, editor, *Proceedings of the Third IEEE International Conference on Evolutionary Computation*, pages 258–261. IEEE Service Center, 1996.
7. I.K. Evans. Enhancing recombination with the complementary surrogate genetic algorithm. In T. Bäck, Z. Michalewicz, and X. Yao, editors, *Proceedings of the Fourth IEEE International Conference on Evolutionary Computation*, pages 97–102. IEEE Press, 1997.
8. D.B. Fogel and L.C. Stayton. On the effectiveness of crossover in simulated evolutionary optimization. *BioSystems*, 32:171–182, 1994.
9. D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989.
10. J. J. Grefenstette. Virtual genetic algorithms: First results. Technical Report AIC-95-013, Navy Center for Applied Research in Artificial Intelligence, 1995.
11. N. Hansen, A. Ostermeier, and A. Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In L. J. Eshelman, editor, *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pages 57–64. Morgan Kaufmann, 1995.
12. J. H. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, 1975.
13. 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.

14. T. Jones. Crossover, macromutation and population-based search. In L. J. Eshelman, editor, *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pages 73–80. Morgan Kaufmann, 1995.
15. T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. J. Eshelman, editor, *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pages 184–192. Morgan Kaufmann, 1995.
16. L. Kallel and M. Schoenauer. Alternative random initialization in genetic algorithms. In Th. Bäeck, editor, *Proceedings of the* 7^{th} *International Conference on Genetic Algorithms*. Morgan Kaufmann, 1997. To appear.
17. L. Kallel and M. Schoenauer. A priori predictions of operator efficiency. In *Artificial Evolution'97*. CMAP - Ecole Polytechnique, October 1997.
18. J. R. Koza. *Genetic Programming: On the Programming of Computers by means of Natural Evolution*. MIT Press, Massachussetts, 1992.
19. P.Collard and J.P. Aurand. Dual ga: An efficient genetic algorithm. In *Proceedings of European Conference on Artificial Intelligence*, pages 487–491. Amsterdam, Wiley and sons, August 1994.
20. C. Ravisé and M. Sebag. An advanced evolution should not repeat its past errors. In L. Saitta, editor, *Proceedings of the* 13^{th} *International Conference on Machine Learning*, pages 400–408, 1996.
21. I. Rechenberg. *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.
22. R.G. Reynolds. An introduction to cultural algorithms. In *Proceedings of the* 3^{rd} *Annual Conference on Evolutionary Programming*, pages 131–139. World Scientific, 1994.
23. H.-P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, New-York, 1981. 1995 – 2^{nd} edition.
24. 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, *Proceedings of the* 4^{th} *Conference on Parallel Problems Solving from Nature*, pages 356–365. Springer-Verlag, LNCS 1141, 1996.
25. M. Sebag, M. Schoenauer, and C. Ravisé. Toward civilized evolution: Developping inhibitions. In Th. Bäeck, editor, *Proceedings of the* 7^{th} *International Conference on Genetic Algorithms*. Morgan Kaufmann, 1997.
26. W. M. Spears. Adapting crossover in a genetic algorithm. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*. Morgan Kaufmann, 1991.
27. G. Syswerda. A study of reproduction in generational and steady state genetic algorithm. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 94–101. Morgan Kaufmann, 1991.