Controlling Crossover through Inductive Learning

Michèle Sebag[1] and Marc Schoenauer[2]

[1] LMS, CNRS-URA 317, Ecole Polytechnique, F-91128 Palaiseau
[2] CMAP, CNRS-URA 756, Ecole Polytechnique, F-91128 Palaiseau
{Michele.Sebag,Marc.Schoenauer}@polytechnique.fr
Presented at PPSN III, Jerusalem, October 1994

Abstract. Crossover may achieve the fast combination of performant building blocks ; but as a counterpart, crossover may as well break a newly discovered building block. We propose to use inductive learning to control such disruptive effects of crossover. The idea is to periodically gather some examples of crossovers, labelled as "good" or "bad" crossovers according to their effects on the current population. From these examples, inductive learning builds rules characterizing the crossover quality. This ruleset then enables to control further evolution : crossovers classified "bad" according to the ruleset are refused. Some experimentations on the Royal Road problem are discussed.

1 Introduction

Various heuristics have been designed to face the central dilemma of genetic algorithms, i.e. the Exploitation of promising regions *vs* the Exploration of new regions [1]. Among these heuristics, see for instance fitness scaling, niching, restricted mating [2]. However, these heuristics must be parameterized by errors and trials, given the lack of theoretical results ; furthermore, these heuristics should be adaptively parameterized : e.g. restricted mating is more adapted to the end than to the beginning of evolution.

This paper focuses on the adaptive control of the crossover operator. Crossover is commonly acknowledged the more powerful engine of evolution, as it enables to combine performant building blocks. But crossover may make a promising parent disappear[3] as well as give birth to a promising offspring. The danger of breaking a building block increases with its length, as shown by the Schemata Theorem [3] : the longer the schema, the more likely a crossover point belongs to its region. But whatever the length of a newly discovered schema, crossover is a danger to it.

This paper investigates the use of inductive learning [6] to prevent using disruptive crossovers ; the idea is to characterize crossovers that are "bad" with respect to the current population, to avoid using them in the next generations. Inductive learning must be periodically redone as the crossover quality heavily

[3] Mutation can also destroy it, but with a usually much smaller rate than that of crossover.

depends on the current population.
Section 2 briefly introduces the problem taken as example all along this paper, the Royal Road problem [7] ; on this problem we discuss the kind of knowledge needed in order to control crossover. Section 3 deals with gathering examples of the crossover behavior and evaluating their potentialities for evolving further generations. The proposed frame integrating GA and inductive learning is described in section 4. Last, section 5 presents our first experimental results.

2 Knowledge about Crossover

This section first recalls the Royal Road problem, then discusses the parts respectively devoted to learning and evolving in an integrated scheme.

2.1 Description

The Royal Road problem (RRP) was conceived by M. Mitchell et coll. [7, 8] to study into detail the combined features most adapted to GAs (laying a *royal road*...). Let the problem space be $\Omega = \{0,1\}^P$ and let $H_{i,j}$ denote the schema with i consecutive 1s, beginning at locus j. The first version of the RRP defined on $\Omega = \{0,1\}^{64}$ involves schemas $H_{8,1}, H_{8,9}, H_{8,1+8*k}$, $H_{16,1}, H_{16,17}, H_{16,1+16*l}$, $H_{32,1}, H_{32,33}$ and $H_{64,1}$. The order of a schema is its number of fixed bits ; the fitness of individual x is defined as the sum of the orders of the schemas x belongs to. The authors expected this fitness landscape to be GA-easy because of (a) its building block structure and (b) the reinforcement due to the fact that high-order schemas are composed of low-order schemas.

However, the RRP is definitely *not* GA-easy ; the analysis made by Mitchell and coll. [8] is that the relative failure of GA is due to a "hitchhiking" phenomenon. The idea is that the first individual belonging to a high-level schema, say $H_{32,1}$, will crowd the population ; in the meanwhile, its 32 last bits, "hitchhiked" by the performant first 32 ones may cause any schema concerned with the last 32 bits, such as $H_{8,33}$ or $H_{16,33}$, to disappear. Evolution then goes from scratch regarding the last 32 bits.

2.2 Broad Lines

Let us sketch the behavior required from a "smart" crossover on this problem :

1. No requirement should be put on crossover in the first stages of evolution in order to achieve fast emergence of the low-order building blocks;
2. After a performant building block has emerged i.e. when it comes to have representatives in the population, it should be sequestered according to [8], i.e. crossover should not break it ; e.g. after schema $H_{8,1}$ has emerged, no crossover should break apart the first eight bits in the representatives of this schema.
3. The fact that a crossover is indesirable of course depends on the current population.

Do such requirements make sense in real-world problems ?
They all rely on the assumption that the global fitness function may be decomposed into parts defined on subsets of the search space. These subsets, called *schemas* in a boolean space, can be thought of in a more general frame ; they are called *formae* by Radcliffe [9].

But how are *formae* or *schemas* interrelated ? The answer to this key question should rule the crossover control. To take an example, breaking apart the 5^{th} and 6^{th} bits should be generally forbidden after schema $* * * *\ 1\ 1\ * *$ has emerged — if schemas do not overlap ; but what if schema $* * * *\ 0\ 1\ 1\ *$ remains to be discovered ? If two schemas overlap and have different fixed bit values, avoiding to disrupt the firstly discovered schema may prevent to ever discover the other one. Let us call *conflicting fitness* a fitness landscape involving schemas that have different fixed bit values for some bits, and are nevertheless of high average fitness. Then, conflicting and non-conflicting fitnesses must be handled through different kinds of <induction - GA> coupling.

2.3 Learning or Evolving ?

Let us precise the scope of the intended crossover control :

- A We may demand that at least one offspring of a representative of a given schema, still belongs to the schema. This requirement ensures this schema is not fleetingly discovered. In this approach, learning precisely rules out *how a given* individual must be combined[4].
- B We may demand that a given region is transmitted from any individual to at least one of its offsprings. Patterns existing in this region will thus remain unchanged in the population (except through the selection and mutation effects) ; patterns will evolve in other regions of the search space. What is learned here is something about *how to combine* individuals, whatever the individuals to be crossed.

Option A is clearly more powerful : it applies whatsoever the underlying fitness landscape, be it conflicting or not. In coupling A, induction is devoted to detect the good building blocks discovered so far, while GA provides a smart exploration and combination of these building blocks.

Option B is simpler and more rigid : it forbids some crossover points. Would this control be defined once for all, this option would clearly be inadequate to a conflicting fitness landscape. But what if this control is re-defined periodically ?

On the other hand, our goal in this paper is to study the potentialities of coupling inductive learning and GA. Studying the B coupling on the Royal Road problem[5] may at least tell us if our approach is worth considering any further.

So, though we are well aware of the limitations of our choice, option B will be the only one considered in the sequel of this paper : we tackle the characterization of crossovers that are adapted to a whole population.

---

[4] A still more precise control would concern *who* to combine with a given individual.

[5] The limitations of option B should not be too severe regarding the Royal Road landscape, since it is not conflicting.

3 Learning to Crossover

Since inductive learning needs labelled examples, we must define what representation of crossover we use, and what label, standing for the "desirability" of a crossover, meets our goal of crossover control.

3.1 Representation of a crossover

Let us consider a bit-string representation, and let $\Omega = \{0,1\}^P$ be the search space. A crossover c may then be represented by a *mask* ; let c be given as $(c_1, \ldots c_P)$, $c_i \in \{0,1\}$:

$$\begin{matrix} x_1\ x_2\ ..\ x_P \\ y_1\ y_2\ ..\ y_P \end{matrix} \rightarrow \begin{matrix} x'_1\ x'_2\ ..\ x'_P \\ y'_1\ y'_2\ ..\ y'_P \end{matrix} \text{ with } x'_i = \begin{cases} x_i \text{ if } c_i = 1 \\ y_i \text{ otherwise} \end{cases} \text{ and } y'_i = \begin{cases} y_i \text{ if } c_i = 1 \\ x_i \text{ otherwise} \end{cases}$$

Crossover c may thus be represented by an element of Ω.

In a real-valued representation ($\Omega = R^P$) a widely used cross-over operator involves barycentric recombination of the parents components [5, 9] :

$$x'_i = \alpha_i x_i + (1 - \alpha_i) y_i \ ; \ y'_i = (1 - \alpha_i) x_i + \alpha_i y_i$$

with α_i in [0,1] or [-.5, 1.5]. A crossover can therefore be represented by vector $(\alpha_1, ..\alpha_P)$, that still belongs to (a subset of) Ω.

3.2 Labelling a crossover

Intuitively, a crossover is *good* if it leads to offsprings with better fitness than that of parents ; it is *bad* if offsprings fitness is worse than that of parents ; otherwise, it is *inactive*.

Choosing good crossovers. The more natural idea is to perform only good crossovers. But a crossover considered good has led to discover a promising individual in the previous generations ; and, since the effects of crossover are reversible, if it is applied on the same individual, it will destroy it. So, a crossover observed to be "good" on a given population is far from being good for ever.

Rejecting bad crossovers. The second natural idea is to reject bad crossovers. Notice that good and bad crossovers evolve quite differently along genetic evolution. A crossover, good at a given step of evolution, may be bad in the very next step. Conversely, if a crossover shows bad, it is because it breaks more useful schemas than it leads to discover ; this fact is likely to persist as long as the population does not change too much. It thus makes sense to characterize the disruptive crossovers by learning rules. These rules enable to reject further disruptive crossovers. This way, the crossover control biases the generation of crossover masks (otherwise random).

Rejecting inactive crossovers. A drawback of inductive learning is that, in order to characterize the class of good or bad crossovers, we need examples of these classes. And, as evolution goes on, the number of good or bad crossovers decreases, and most crossovers become inactive.

But actually, inactive crossovers are still worse than disruptive ones (at least after a given stage of evolution) : with disruptive crossovers evolution goes backwards (hopefully to other promising regions) and the search goes on ; with inactive crossovers, evolution stops. So we ended in deciding to characterize inactive crossovers and to refuse them.

More precisely, we propose the following strategy :

- when less than 70% of the observed crossovers are inactive, phase we denote by *classical*, we refuse disruptive crossovers ;
- when more that 70% of the observed crossovers are inactive, phase we denote by *modern*, we refuse inactive crossovers.

The transition between both phases strongly depends on the niching phenomenon [2]. If only one *species* is to be found, then the population will (in case of success) converge toward a single optimum, i.e. to a uniform population. Long before that, most crossovers would be inactive. Retaining solely active crossovers is equivalent to speeding up the evolution. In opposition, if several *species* are to be found, it is usually irrelevant to cross individuals belonging to different species. Learning and rejecting disruptive crossovers then can play the same role than any niching heuristics (e.g. restricted mating).

4 An integrated frame

This section first describes the GA we use, then give the broad lines of the coupling. Last, the interactions between GA and inductive learning are detailed.

4.1 Genetic Algorithm

The genetic algorithm we use is a lab-made software based on the standards [1] : bit-string encoding, roulette wheel selection with fitness scaling, two-points crossover at a rate of 0.6 with both offsprings replacing the parents, and mutation at a rate of 0.05. The evolution stops either after 1000 generations or when the fitness is constant over the population. The fitness scaling (number of offsprings for the best individual) varies in the experiments (see 5.1).

4.2 Broad Lines of the Coupling

We combine GAs and inductive learning in the following scheme (Fig.1) :

1. A first darwinian period involves N generations of a classical GA ;

2. The next step corresponds to gaining experience about evolution. Examples of crossovers are generated and their behaviors are observed with respect to the current population (section 4.3). A set of rules is built from these examples by inductive learning (section 4.4).
3. Then a "civilized" period involves GA controlled according to the available experience. During M generations (M is called *civilization length*[6]), crossovers classified by the ruleset as disruptive or inactive are refused (section 4.5).
4. After these M generations, the population has evolved and acquiring new experience is necessary. So, the process goes back to step (2).

We call *civilization* the phase of rules induction together with the consecutive M generations obeying these rules.

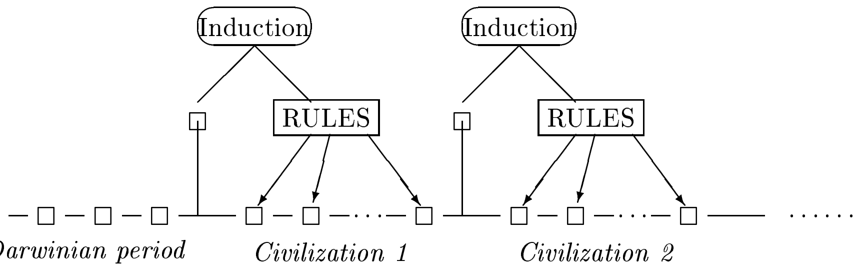


Figure 1 : Combining GAs and Induction (□ stands for a generation)

4.3 Gathering crossover examples

Let L be the size of the genetic population. L crossover examples are built as follows :
• A 2-point crossover mask is randomly generated;
• Two individuals are randomly selected in the population;
• These individuals are crossed according to the crossover mask. The mask is labelled as good, bad or inactive, depending on whether the fitness of the best offspring is better, worse or equal to the fitness of the best parent;
• If this mask belongs to a class that includes less than 70% of the L examples, it is added to the example set.

The last condition may result in an never-ended loop if all crossovers belong to the same class. To settle this problem, if the example set is not completed in $5 \times L$ trials, an empty ruleset is considered all along this civilization (that period is therefore darwinian). The current phase is said *classical* if less than 70% of the examples are inactive, and *modern* otherwise.

4.4 Learning rules

We use a star-like algorithm [6], described into detail in [12, 11]. A rule consists of an hypothesis part and a conclusion part. The hypothesis can be thought of as

[6] In the following, we take $M = N$.

a schema in the crossover space ; the conclusion here belongs to the set { good, bad, inactive }. A rule is said to cover a crossover mask iff this mask belongs to the schema standing for the hypothesis of the rule.

The induction we use is a bottom-up process : from any example (crossover mask) Ex, we try to find out rules covering Ex and not covering any example belonging to a class other than that of Ex. Among the many solutions, only rules covering the maximum number of examples are retained ; furthermore, we require that a rule covers at least 2 examples. This algorithm does not necessarily provide consistent rules. When an unlabelled crossover is covered by rules of different conclusion, the final decision is determined via a majority vote among these rules.

4.5 Using the rules

A 2-point crossover mask is initialized at random. During the civilized periods, this mask is checked as follows :
• if the ruleset is empty, the mask is accepted.
• if the mask is classified as bad, and the current phase is classical, the mask is refused;
• if the mask is classified as inactive, and the current phase is modern, the mask is refused;
• otherwise, the mask is accepted.

Again, this decision process may result in a never-ended loop if all crossovers are rejected. To prevent this, a gradual relaxation of the ruleset is used :
• rules are weighted ; the weight of a rule is set to the number of learning examples it covers ;
• only rules with weight greater than a threshold s are considered. Threshold s is initialized to 1 (i.e. all rules are considered in the beginning).
• if, on L consecutive trials, less than $\frac{L}{20}$ crossovers are retained, threshold s is incremented of 1.
So, if no crossover is accepted, less and less rules are taken into account.

5 Results and Discussion

5.1 Protocol

Any presented result corresponds to the average result obtained on 30 independant runs. The population size is set to 100, 200 and 500 individuals. The fitness scaling (number of offsprings for the best individual) is set to 1.5 and 2. The civilization length varies from 0 (no learning at all) to 10.

5.2 Results on the Royal Road

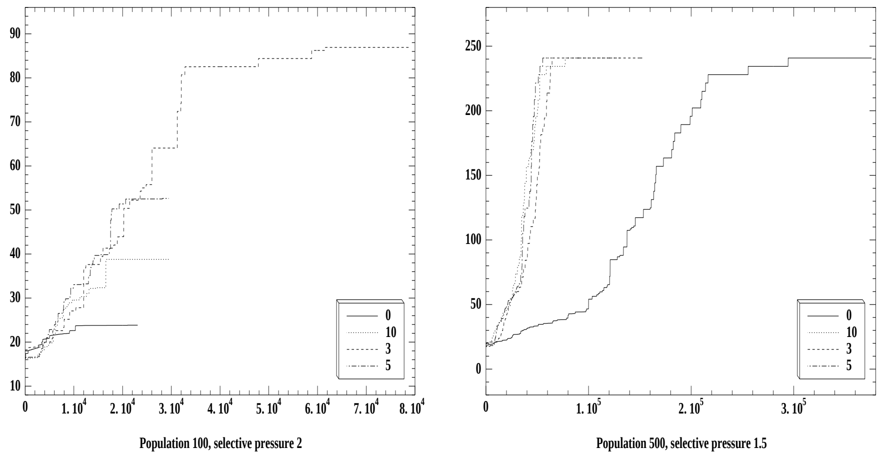
The Royal Road problem is modified as in [8] : if $N(i)$ denotes the number of schemas of order i (with i equals 8, 16, 32 or 64) individual x belongs to, the

fitness of x is the sum, for all orders i of 0 if $N(i) = 0$ and $i + N(i) \times .2$ otherwise (instead of $N(i) \times i$).
Table 1 shows the number of hits of the global optimum over 30 runs. The mean number of function evaluations needed to hit the optimum is given between parentheses for population size 200 and 500 (meaningless for population size 100).

Pop. Size	100		200		500	
Pressure	1.5	2.0	1.5	2	1.5	2
C. Length						
0	0	0	16 (110.200)	14 (112.400)	30 (255.000)	29 (210.000)
3	1	5	28 (42.400)	23 (44.600)	30 (57.700)	30 (39.240)
5	2	2	28 (36.240)	25 (59.230)	24 (62.920)	18 (79.490)
10	1	1	24 (62.920)	18 (79.490)	30 (54.100)	30 (76.100)

Table 1 : Number of hits (mean nb of evaluations needed)

Note that the results obtained without learning are not identical to those of Mitchell et al. [8], since they never found the optimum in 10^6 evaluations, for a population size of 2000. This difference is due to the fitness scaling mechanism : in [8], the number of offsprings of the best individual is set to the ratio between its fitness and the average fitness over the population.



(2.a) (2.b)

Figure 2 : Average best fitness function of the number of evaluations

Another criterion of performance is the evolution speed. Fig. 2 (a) plots the mean best fitness obtained for N nb of evaluations, for a population size 100, and a civilization length of 0, 3, 5 and 10. Fig. 2 (b) gives the results corresponding to a population size of 500. Of course, the number of function evaluations includes the number of experiments needed for learning.

5.3 Discussion

A first remark is that, as expected, civilized evolution (i.e. evolution + learning) discovers the global optimum more often than darwinian evolution (Table 1). Furthermore, when both evolutions do reach the optimum, civilized evolution requires less computational effort (Fig. 2.b). On the other hand, when neither evolution does not reach the optimum, civilized evolution goes higher (Fig. 2.a).

Second, it seems that civilized evolution needs a scaling factor smaller than darwinian evolution, of which we propose the following explanation. Selection ensures the survival of performant schemas through duplication of performant individuals. But in civilized evolution, no disruptive crossover is to fear : performant schemas are ensured to survive. Learning thus replaces to some extent, the individual selection by the crossover selection.

Last, we claim that our approach is more general than the heuristic proposed in [8] to overcome the difficulties of the Royal Road problem. Introns are zones of bits not contributing to the fitness [4] ; Mitchell et coll. modify the representation of the Royal Road by separating order-8 schemas by introns of length 24. The use of introns therefore significantly decreases the chances for a crossover to be disruptive : a crossover point may freely arise in an intron zone without disturbing any low-order schema in the current individuals.

However, this heuristic is not easy to use in real-world problems since it requires strong presumptions as to the localization of schemas. In the meanwhile, a learning approach does not require any such a priori information.

6 Conclusion

This paper investigates the coupling of GA and inductive learning according to the following scheme : a first period, called "darwinian evolution" only involves some generations of classical GA ; next periods, called "civilizations", are two-step processes :

• A learning step gathers examples of crossovers and observes their behavior on the current population. These examples enable inductive learning to build a set of rules characterizing relevant crossovers.

• Then, a civilized evolution step consists of M generations, where M is called "civilization length", that differs of classical GA in that that the crossovers selection is biased according to the current ruleset ; only crossovers relevant according to this ruleset, are retained.

In the end of a civilization, the population is significantly different from what it was in its beginning. Another learning phase thus open to a new civilization.

This approach asks a key question : inductive learning provides the evolution with an explicit memory — compared to the implicit memory consisting of the population itself. The implicit memory is safely (if not fast) optimized by GA and modifications of the fitness landscape can lead to powerful optimization strategies [10]. We may then fear explicit memory to disturb the evolution mechanism. Of course, our experiments only considered the artificial problem of the Royal Road, and many more experiments are needed to validate the scope, if any, of our approach.

However, at a very theoretical level, an argument on behalf of explicit memory could be the following. Implicit memory makes few differences between discovery and re-discovery. But this difference is of great importance for the *acceleration of evolution* : an anthropomorphic analogy of the explicit memory we propose to equip the evolution with, is the invention of writing ...

Further research will consider other problems, especially deceptive ones. We also plan to investigate the influence of the civilization length and of the scaling factor on the hybrid evolution we called "civilized" evolution.

References

1. D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989.
2. D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Proceedings of the* 2^{nd} *International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum Associates, 1987.
3. J. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, 1975.
4. J. R. Levenick. Inserting introns improves genetic algorithm success rate : Taking a cue from biology. In R. K. Belew and L. B. Booker, editors, *Proceedings of the* 4^{th} *International Conference on Genetic Algorithms*, pages 123–127, 1991.
5. Z. Michalewicz. *Genetic Algorithms+Data Structures=Evolution Programs.* Springer Verlag, 1992.
6. R.S. Michalski. A theory and methodology of inductive learning. In R.S Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning : an artificial intelligence approach*, volume 1. Morgan Kaufmann, 1983.
7. M. Mitchell, S. Forrest, and J.H. Holland. The royal road for genetic algorithms : Fitness landscapes and ga performance. In F. J. Valera and P. Bourgine, editors, *Proceedings of the First European Conference on Artificial Life-93*, pages 245–254. MIT Press/Bradford Books, 1993.
8. M. Mitchell and J.H. Holland. When will a genetic algorithm outperform hill-climbing ? In S. Forrest, editor, *Proceedings of the* 5^{th} *International Conference on Genetic Algorithms*, page 647, 1993.
9. N. J. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5:183–20, 1991.
10. M. Schoenauer and S. Xanthakis. Constrained ga optimization. In S. Forrest, editor, *Proceedings of the* 5^{th} *International Conference on Genetic Algorithms*, pages 573–580. Morgan Kaufmann, 1993.
11. M. Sebag. Using constraints to building version spaces. In L. De Raedt and F. Bergadano, editors, *Proceedings of ECML-94, European Conference on Machine Learning.* Springer Verlag, April 1994.
12. M. Sebag and M. Schoenauer. Incremental learning of rules and meta-rules. In Porter B. and Mooney R., editors, *Proceedings of ICML-90, International Conference on Machine Learning.* Morgan Kaufmann, 1990.