

Preventing Premature Convergence via Cooperating Genetic Algorithms

Cyril Fonlupt

Philippe Preux

Denis Robilliard

Laboratoire d'Informatique du Littoral,
BP 719,

62228 Calais Cedex, France,

Phone: +33-3-21-97-00-46 Fax: +33-3-21-19-06-61

fonlupt@lil.univ-littoral.fr

1 Introduction

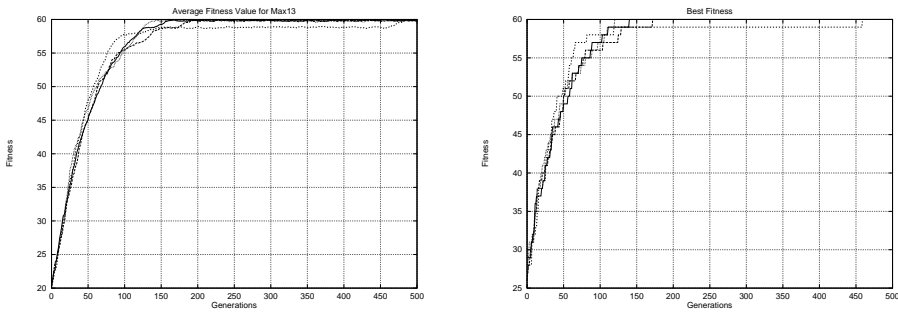
The definition of the hardness of a problem for GA's has been tackled, eventually leading to the notion of deception [Gol89, HG94, Dav87]. It has been known for a while that the hardness of a problem is inherently related to the representation that is used. This fact will be illustrated below by showing that an easy problem (1's counting problem) can become nearly unsolvable after a change of representation. Then, we show how a set of cooperating GA's, each solving the same problem but using different representations, can succeed in solving a difficult problem by exchanging individuals. Remapping the research space with a change of representation is not a new idea [BV90, MW92, KD95]. Remapping may involve an increase or a decrease of the number of local optima by dynamically changing the neighborhood of individuals. In this paper, we show how cooperation can help finding the global optimum. The idea is that when a GA is stuck on a point (because of the representation it uses), it is likely that an other GA, using an other representation provides it with new individuals from which the search can go further. Our point is not to use parallelism to speed up the computation but to use the parallelism as a new way of exploring the research space.

2 The *Max1* problem

1's counting problem (*max1*) is generally held as an easy problem to solve for a GA, though it has also been shown that this problem is not so easy because of the presence of hitch-hikers [MFH91, Gre93, Pre95].

In the *max1* problem, the fitness of an individual scales with the number of alleles which value is 1. Instead of the usual binary representation, we solve this problem in base 3 (and call the problem *max13* in this case). Hereafter, we will denote GA_r a canonical GA using a representation in base r , *i.e.* the set of alleles is $\{0, 1, \dots, r-1\}$. GA_3 applies its operators on genotypes which gene alphabet is ternary, the alleles being thus taken from $\{0, 1, 2\}$. The fitness of an individual is then the number of genes which value is 1. The mutation acts by drawing an allele at random and assigning it to a gene chosen at random. (The number of mutated genes is, as usual, specified by the mutation rate.) The experiments are performed with our **peg** simulator (parallel Enhanced Genesis) [DFP94], which is a parallel version (island model) of the well-known **genesis** simulator [Gre], modified to allow non binary genotypes. Individuals may be exchanged between demes and each deme has its own set of parameters. For the moment, **peg** is run in sequential mode. The population is made of 50 individuals, the length of genotypes is $\lambda = 60$, two-point crossover is applied with probability 0.6, mutation with probability 0.001. Ranking and elitism are used.

As expected, experiments show that *max13* is as easy to solve as the usual *max1*. Figure 1 shows the evolution of fitness along generations during 4 representative runs of the GA. After a fast increase of fitness, GA_3 converges towards an average value close to 60 (the length of genotypes). The optimal genotype has been found 100 times out of 100 runs.



(a) Average fitness in the population during 4 runs on *Max13*

(b) Best fitness of the population during these runs

Figure 1: 4 runs of GA_3 on *Max13*

$max1k$ denotes the 1's counting problem in base k . It is well known that solving $max1k$ will get harder when k grows up ; but we are only interested in showing that the problem $max13$ is not a difficult to tackle for GA_3 .

2.1 Is *max13* really easy?

In this section, we show that $max13$ can be transformed into a very difficult problem, simply by changing the representation. Instead of always using a representation in base 3, we use another base and individuals are temporarily transformed into base 3 for their evaluation (see Fig. 2). Mutation and crossover are applied in the original base. Thus, the algorithm is still optimizing $max13$.

In this framework, the optimal genotype is no longer found; the GA converges towards a genotype rather far from the optimal (see Fig. 3).

Note that it is unsurprising that this change of representation makes a problem very hard for the GA. Suppose that some function f has a set of values which is any permutation of the numbers $1 \dots 3^n$. Then arguably, it is possible to find the recoding such that $g(x) = x$.

Thus the hardness of the problem is related to the representation that is used. In fact, this change of behavior of the GA may be explained from the following way: it has been argued [Wei90] that if the fitness of neighboring individuals is correlated (*ie* the fitness of an individual is correlated with the fitness of its neighbors), a local search algorithm is likely to perform better. On the contrary, if there is no correlation between the fitness of neighboring points, a local search algorithm will behave badly. In our case, by changing the representation, the correlation between neighboring individuals is "broken" [FRP97, Hor97].

An other point worth mentioning is that mutation, in the case of the binary representation, only modifies by 1 the fitness of an individual when optimizing $max1$. Using a non-binary representation, the effect of mutation is not so simple due to Hamming cliffs. For instance, the individual 0000000000102020101 for GA_3 is identical to the genotype 00000001111111111111 for GA_2 , the optimal genotype. However, mutating the rightmost gene in GA_4 results in a genotype of fitness 1.

3 Cooperating GA's

As it has been shown in the last sections, it is quite clear that the representation increases or decreases the difficulty of the problem. Instead of using only one representation, our idea is to have such GAs running concurrently each optimizing the same object, but using its own representation. After a given number of generations, each GA provides other GA's with its best local optima found so far.

In our experiments, each GA_i works on a population of 50 individuals of length $\lambda = 40$. The parameters are the same for each GA_i and like those introduced before. In our model, every 20 generations, each GA_i

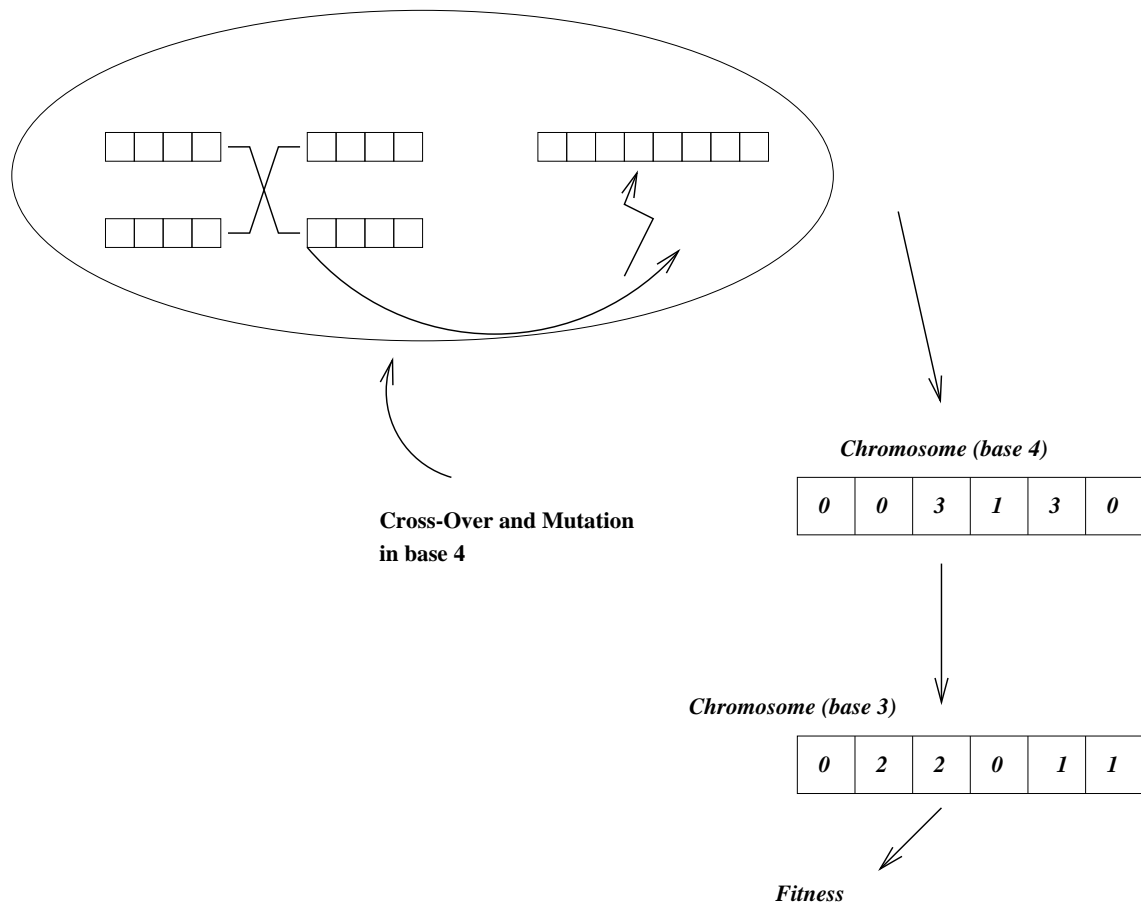
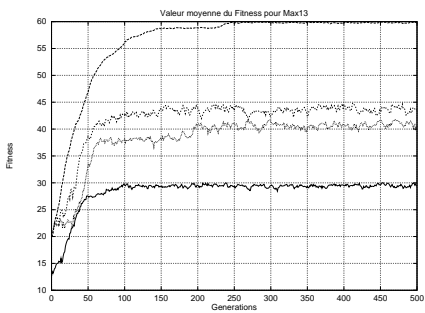
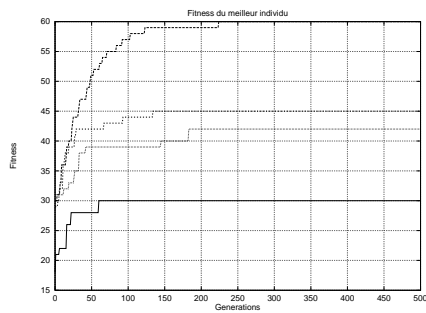


Figure 2: Cross-Over and mutations are made in base 4. When the fitness is computed, the chromosome genotype is turned into base 3. Note that the search is larger in base 4 than in base 3. Non feasible chromosomes are assigned a null fitness in base 3.



(a) Average Fitness for $Max13$, each slope represents a different base. The highest slope uses a representation in base 3



(b) Best Fitness in the population. The highest slope uses a representation in base 3

Figure 3: Runs on GA_3 on $Max13$ for different representations

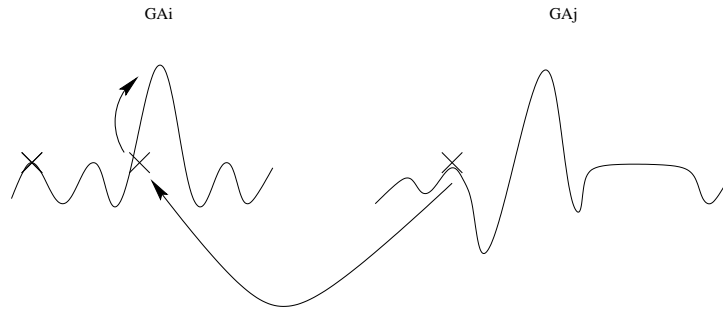


Figure 4: In this case GA_i and GA_j are stuck in a local optimum. GA_j provides GA_i with its best local optimum found so far which is a point leading to the global optimum for GA_i .

broadcasts to every $GA_{j|j \neq i}$ its 10 best individuals. We introduce elitism in our model by automatically inserting the individuals in the target GA if their fitness is better.

The aim goes as follows: a certain point is reached by a certain GA_i where it gets stuck. An other GA_j might provide GA_i with a point which would allow it to search further. Using more than one representation, a local optimum for GA_j may become an interesting new starting point for GA_i (see Fig. 4).

Beginning with *max1*, we modify it so that a global optimum is being difficult to be found when using a binary representation. Instead of the genotype where all alleles are 1, we define the string 11...10000 as the global optimum. The fitness of this genotype is twice the fitness of the genotype which contains only 1's, that is $2 \times \lambda$. GA_2 is not at ease with this problem which necessitates 4 mutations of the "rightmost" genes to occur. The probability for this event to occur can be made as small as desired by varying the length of genotypes.

To sum up the fitness of an individual is:

$$f(x) = \begin{cases} u(x) & \text{where } u \text{ is the unitation function if } x \neq 11\dots10000 \\ 2 \times \lambda & \text{if } x = 11\dots10000 \end{cases}$$

100 experiments **without cooperation in parallel** (*i.e.* each GA_i works in its own representation on the same objective function without exchanging the best optima found) were run with genotypes of length 40. The sub-optimal genotype has been found during all the runs by GA_2 while the optimal genotype has never been found.

However using cooperation, GA_2 is able to find the local optimum (Fitness = 40). Once this chromosome is migrated to other GA's, they succeed to improve this local optimum. The global optimum is then found by GA_3 . Figure 5 explains this cooperation. GA_2 easily reaches the local optimum after a few iterations. This sub-optimum is then migrated to all other GA's. The other GA's improve the local optimum and find the global optimum (fitness=80).

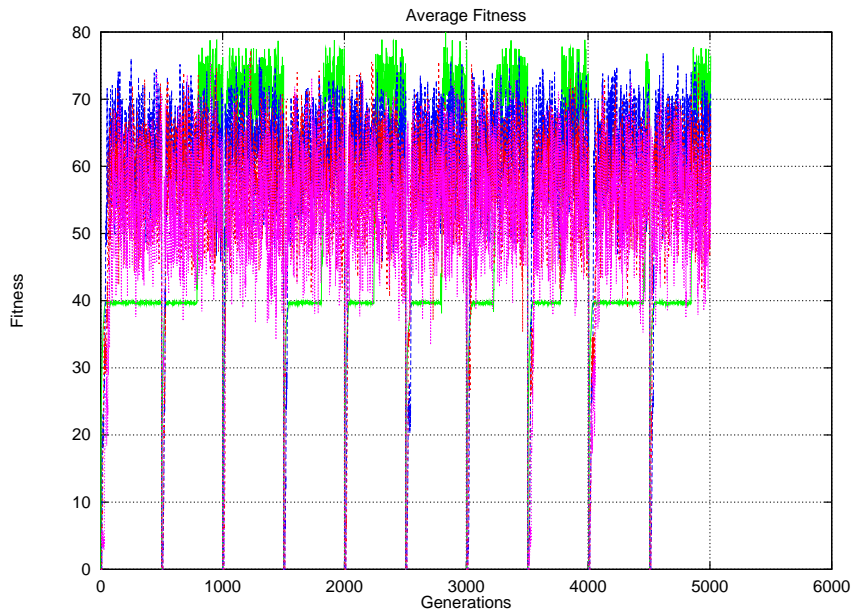
Even if this problem may seem quite artificial, it shows how cooperation between GA's can improve the search and make the neighborhood evolve. Each GA shares its best solution with other GA's and makes it more likely to avoid getting stuck on local optima.

4 Future works and Conclusion

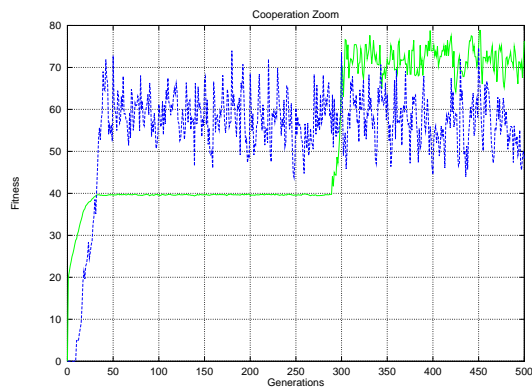
Two points are highlighted in this paper.

Our first point deals with something already known. We have shown on a simple example that the prototypical easy problem for GA's (1's counting problem) can be transformed into a nearly unsolvable problem when changing the representation that is used by the GA.

Our second point is original. We have argued and shown experimentally that cooperating GA's, optimizing the same objective function but working with different representations of data and exchanging individuals some times to times, can be successful at solving to optimality a problem otherwise difficult. Thus, we have shown that, from an experimental point of view, cooperating GA's can solve problems that a sequential GA is



(a) Cooperation between GA's to escape a local optimum. For 10 experiments triggered every 500 generations, 4 GA's are working in their own base (2, 3, 4 and 5). In 9 out of 10 experiments, the global optimum was found. GA₂ (*ie* GA working in base 2) easily reaches the local optimum after a few generations. This sub-optimum is migrates to all other GA's. The other GA's improve this local optimum and find the global optimum. Note that without the help of GA₂, GA_{3,4,5} are unable to find the local optimum and of course the global optimum.



(b) Zoom of Figure (a)
 In order to have a clear Figure, we have focused our attention on a single experiment and we have only plotted the best individuals. In this experiment, it is quite clear that GA₂ easily reaches the local optimum. This genotype then migrates to the other GA which in turn finds the global optimum.

Figure 5: Illustration of cooperation between GA's

very unlikely to solve. They help each other by exchanging (at random) individuals which create new paths of research and permit a GA to follow them. These immigrants are also useful to untrap a GA that is stuck on a point. We think that from a pragmatic point of view, cooperation between GA's can alleviate premature convergence and lead to the discovery of local better optima.

We are now applying the technique introduced in this paper to the traveling salesman problem. Each deme is solving the TSP and the best local solutions are recombined to provide new starting points.

References

- [BV90] David L. Battle and Michael D. Vose. Isomorphisms of genetic algorithms. In Gregory J.E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 242–251. Morgan Kaufmann, 1990.
- [Dav87] Lawrence Davis, editor. *Genetic Algorithms and Simulated Annealing*. Research Notes in Artificial Intelligence. Morgan Kaufmann, San Mateo, CA, USA, 1987. ISBN: 0-934613-44-3.
- [DFP94] David Duivivier, Cyril Fonlupt, and Philippe Preux. PEG: Parallel enhanced genesis. Technical report, Laboratoire d'Informatique du Littoral, Calais, France, 1994.
- [FRP97] Cyril Fonlupt, Denis Robilliard, and Philippe Preux. Comparing 2-opt and city swap operators. submitted, 1997.
- [Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [Gre] John J. Grefenstette. *Genesis v 5.0*.
- [Gre93] John J. Grefenstette. Deception considered harmful. In [Whi93], pages 75–91, 1993.
- [HG94] Jeffrey Horn and David E. Goldberg. Genetic algorithm difficulty and the modality of fitness landscapes. In [WV94], 1994.
- [Hor97] Wim Hordijk. A measure of landscapes. *Evolutionary Computation*, 1997. To appear.
- [KD95] Jason Kingdon and Laura Dekker. The shape of space. Technical Report RN/95/23, University College London, London, England, 1995.
- [MFH91] Melanie Mitchell, Stephanie Forrest, and John H. Holland. The royal road for genetic algorithms: Fitness landscape and GA performance. In [VB92], pages 245–254, 1991.
- [MW92] Keith Mathias and Darrel Whitley. Remapping hyperspace during genetic search: Canonical delta folding. In L. Darrel Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 167–186. Morgan Kaufmann, 1992.
- [Pre95] Philippe Preux. Influence of the size of the research space on the discovery of optima; case studies. (unpublished manuscript), 1995.
- [VB92] Francesco Varela and Paul Bourguine, editors. *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. MIT Press, Cambridge, MA, USA, December 1992.
- [Wei90] E.D. Weinberger. Correlated and uncorrelated landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336, 1990.
- [Whi93] Darrell Whitley, editor. *Proc. of the Workshop on Foundations of Genetic Algorithms*, Vail, CO, USA, 1993. Morgan Kaufmann, San Mateo, CA, USA.
- [WV94] Darrell Whitley and Michael D. Vose, editors. *Proc. of the Workshop on Foundations of Genetic Algorithms*, Estes Park, CO, USA, 1994. Morgan Kaufmann, San Mateo, CA, USA.