

©1996-2004 All Rights Reserved. [Online Journal of Bioinformatics](#). You may not store these pages in any form except for your own personal use. All other usage or distribution is illegal under international copyright treaties. Permission to use any of these pages in any other way besides the before mentioned must be gained in writing from the publisher. This article is exclusively copyrighted in its entirety to OJB publications. This article may be copied once but may not be reproduced or re-transmitted without the express permission of the editors.

---

**OJB**  
***Online Journal of Bioinformatics®***

**Volume 5 : 23-31, 2004**

---

***GASP: a new Genetic Algorithm (based on)  
Surviving Probability.***

**A. Carvajal-Rodríguez**

*Departamento de Bioquímica, Genética e Inmunología, Facultad de Ciencias Universidad de Vigo, 36200 VIGO, Spain (acraaj@uvigo.es)*

---

**ABSTRACT**

**A. Carvajal-Rodríguez, *GASP: a new Genetic Algorithm (based on) Surviving Probability. Online Journal of Bioinformatics 5:23-31, 2004.*** A new basic genetic algorithm, called GASP (Genetic Algorithm Surviving Probability) is described. The algorithm differs in some essential properties compared to other genetic algorithms (GA's) and is more accurate than traditional GA's in solving some general problems. In GASP the evolutionary working principle is based in a selection scheme called *absolute selection*. Effect of the *absolute selection* mode is analysed and GASP is compared with the well-known Simple Genetic Algorithm (SGA) via three examples. The third example is a rather novel application of GAs on a biological problem related with in progress research in conservation genetics. Results show that GASP achieves higher accuracy on reaching the optimum in the three example problems and is faster than SGA. Data sets, source code and the biological model used in example 3 are available as supplementary information from <http://webs.uvigo.es/c03/webc03/XENETICA/XB2/antonio/GASP/GASP.htm> It is proposed that GASP-based GAs may represent a powerful new kind of GAs for the exploration of many interesting biological problems.

**KEY WORDS:** Genetic Algorithm, schema theorem, computer simulations in conservation biology problems

---

## INTRODUCTION

A Genetic Algorithm (GA) is a solving problem method that derives its behaviour by analogy to evolution in nature. Briefly, this means that the problem solutions are encoded as a population of individuals. Each individual is a solution and has different "genes" which represent different parameters of the problem. Individuals can undergo mutation and recombination in these "genes". Each individual has a fitness value that depends on how well the individual performs with respect to the problem to solve. Therefore, the offspring number with which one individual contributes to the next generation depends on its fitness value. Through the process of finding the optimal solution, the population will evolve during a number of generations or until a maximum or average fitness is reached. More detailed information about GAs behaviour and implementation can be found in, for instance, Mitchell (1996) or Michalewicz (1996). However, it is important to recall that when GAs are used, a particular codification is needed to represent the parameter set of the problem to be solved. Different possible solutions i.e. distinct parameter combinations are represented in the bit string individuals of the GA. Most work with GAs is focussed on fixed-length character strings although there is a growing field of evolutionary computation where this restriction does not hold (Michalewicz, 1996). Regarding to binary and fixed-length codification, one of the first and simplest GAs (Goldberg, 1989) is known as Simple Genetic Algorithm (SGA). In fact SGA represents the basis for most applications of GAs (Mitchell, 1996). In SGA-based GAs three main basic ideas are present:

- 1) The offspring number of an individual depends directly on the fitness value (the objective function value) of such individual. The higher the fitness the larger the offspring number of the individual.
- 2) Population size is usually held constant and does not depend on the average fitness of the population. Nevertheless, the effect of different population sizes has been studied and used to improve the search with GAs (see chapter 4 of Michalewicz, 1996 and references therein).
- 3) Points one and two are directly related to the mode of selection used in GAs. Thus, the selection operator allows for a probabilistic mechanism to maintain individuals as parents of future progeny despite possessing relatively poorer objective values (soft selection) or, on the contrary, to deterministically retain just the best available individuals to generate future progeny (hard selection). In any case, individuals will be selected at each generation after evaluation of their objective function value with respect to the current population average.

The concepts of hard and soft selection are somewhat different in classical population biology. Basically they refer to selection with respect to population average (soft selection) or to an absolute value (hard selection) that does not need to be present in the population at that time. The latter is related to selection in heterogeneous environments (Hartl and Clark, 1989). The main difference between the biological soft and hard selection concepts is the effect that hard selection has on population size. In the new GA presented in this work, the selection scheme is hard in the sense of population biology. So, to avoid confusions this kind of selection will be called *absolute selection*.

GAs have been applied to fields as diverse as engineering, economics and biology (Mitchell, 1996). Notwithstanding that, there is almost one different GA for each particular problem, but there are some problems that are typically difficult for them. Among such problems, epistasis, i.e. deviation of independence effect of each bit onto the individual fitness value, has been identified as one cause of difficulty for GAs (Naudts and Kallel, 1998; Kumar, 2000). Different strategies have been suggested and used to avoid one fatal epistasis consequence: the search stops at a local population optimum unable to reach the global one (Kumar, 2000 and references therein). One of these strategies is the representational approach, which substitutes the simplest, but epistatic, codification by a more complex non-epistatic representation easier for the GA to solve (Kumar, 2000). However, an inconvenient of this and other approaches is that they imply more complex algorithms and hence specific codification for particular problems.

In this study a new genetic algorithm (Genetic Algorithm Surviving Probability: GASP) which just differs at the three basic points mentioned above, is defined. It will be shown how can GASP solve high epistatic problems that SGA can not. Another advantage of GASP is that it is also a flexible algorithm, via two new parameters,  $R$  (maximum offspring number with which one individual contribute to the next generation) and  $K$  (an *a priori* defined absolute value to compare with each individual fitness), different selection schemes (resembling in some sense soft and hard selection in GA theory) can be implemented with the same algorithm.

## MATERIALS AND METHODS

### Analysis of population dynamics

A schema is a particular pattern of a bit subset, e.g. for a 4 bit string ##10 is an order 2 schema (two defined bits "1" and "0") and length 1 (distance between outermost defined bits) where # represents wildcards ("don't cares"). A well-known theorem in GA theory is the Schema Theorem (Holland, 1975; Goldberg, 1989) which describes the growth of a schema from one generation to the next. The Schema Theorem is often interpreted as implying that short (minimise the recombination effect), low-order (minimise the mutation effect) schemata whose average fitness remain above the population mean will increase their frequencies at each generation.

Regarding GASP, I will give the condition for any particular schema to increase its frequency through generations. Thus, a theorem analogous to the Schema Theorem is obtained for GASP. Furthermore, since population size is variable in GASP, population extinction could occur. The sufficient condition for a population not to become extinct will also be given.

### Comparing GASP and SGA performance

To compare GASP and SGA performances, three examples with different epistatic values (no epistasis, high and intermediate) were studied, using the epistasis variance as total epistasis measure (Davidor, 1991).

#### *Example 1. A simple function.*

Given  $f(x) = x \sin(10\pi x) + 1.0$ , find  $x$  from interval  $[-1,2]$  so that maximise the function.

This problem appears in Michalewicz (1996) as a 22-bit one and is just a simple example to compare SGA and GASP. The epistasis variance of this example is 0.

#### *Example 2. "Needle in a haystack".*

The so-called "needle in a haystack" (NIAH) are problems of finding particular values. Thus, a bit string will have fitness 1 if coincides with one of the searched values and fitness 0 otherwise. Such functions are known as GA-hard NIAH-like functions for which the only efficient algorithm seems to be enumerative search (Kumar, 2000). Consider for instance an 8-bit string and the problem of finding the strings with decimal value 5 or 122. These are two complementary binary strings and this problem is expected to have maximal epistasis variance (Naudts *et al.*, 1997). Specifically, the total measured epistasis variance of this example is 0.99.

#### *Example 3. A biological model.*

Consider a stochastic simulation model for several populations of organisms of a single species in which different genetic and ecological parameters can be set. Allow such populations to be interconnected by migration. Consider that one of this populations undergoes some kind of pollution effect that we can simulate in different ways (model details in Carvajal-Rodríguez *et al.*, 2003). Assume we are interested in investigating parameter combinations for this model to produce extinction of the polluted population. In this case, the aim is to find the genetic and ecological scenario (16 possible combinations) and the type of pollution or human impact effect (5 different kinds plus 10 pair combinations plus uncontaminated case given a total of 16 possible values) in which extinction of the affected population occurs. The total variance epistasis of this 8-bit example with the particular codification used (not shown, available as supplementary

information) is 0.66. With respect to this example, a research on more complex and exhaustive parameter space exploration is in progress.

#### Measures of performance

The main measure of performance was the population average fitness (range 0-1). In all examples the algorithm will stop after a fixed number of generations (150 generations for examples 1 and 2, 10 generations for example 3). Population fitness was recorded in the last generation, or, in the case of examples 2 and 3, if population average fitness reaches a given relative value (1.0 for example 2, 0.95 for example 3) the current generation is recorded and the iteration ends.

In example 1, the measures of performance, were the best value obtained over generations and the generation number in which occurs. In examples 2 and 3 performance measures were the obtained frequency of average fitness that reached the optimum and the average generation number at which it is reached in 50 replicates for a given simulation.

In examples 2 and 3 (with replicates), the criterion of convergence quality (Naudts and Kallel, 1998) was used to classify an algorithm according to its "ability" to solve a given problem. This criterion depends basically on the independence or dependence of the algorithm capability to solve the problem with respect to the choice of the randomly sampled initial population. Accordingly to this criterion, a GA can be recorded as class-1 (the algorithm always achieves the solution in all the replicates and spends almost the same generation number each time), class-2 (the algorithm always achieves the solution in all the replicates but variance in generation number is high), class-3 (the algorithm only sometimes achieves the solution) or class-4 (the algorithm never reaches the solution).

#### Algorithm

The algorithm used in GASP can be described in a few steps:

- 1.- Generate a random population of size  $N$ .
- 2.- Select a random pair of parents from the population.
- 3.- Obtain one son by recombination from each pair of parents. Add one more attempt to the particular counter of one progenitor.
- 4.- Evaluate the fitness of each son. The fitness is obtained dividing the objective function value by  $K$ , which is the maximum, or a reasonable supreme, of the objective function. A son will survive if its fitness value is higher than a number randomly generated from an uniform between 0 and 1. In this way, the fitness value is equivalent to surviving probability.
- 5.- Repeat step 2 until the maximum population size is reached or all parents have spent their fixed number of attempts ( $R$ ).
- 6.- The new population substitutes the old one.
- 7.- Mutate and iterate using this new population.

#### Implementation

The particular implementations of GASP and SGA were in C and the code is freely available upon request. Compilation was made with Linux gcc and the program was run in an Intel Xeon with 2 GHz.

GASP and SGA were run for a set of recombination frequencies  $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$  and mutation rates  $\{10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$  in all cases except for SGA in example 1 in which a previously published result was used (Michalewicz, 1996).

SGA was implemented with *roulette wheel selection* which is a soft selection method (Mitchell, 1996). For GASP, the parameter  $R$  was maintained constant ( $R = 2$ ) in all cases. Parameter  $K$  was the maximum for examples 2 and 3 and a supreme for example 1. The initial population size was 16 for the 8 bit examples and 50 for the 22 bit examples (example 2 was run for 8 and 22 bit string). In every case initial size was considered as the maximum population size. The recombination operator is implemented as *parameterised uniform crossover* (Spears and DeJong, 1991) in which an exchange happens at each bit

position with probability  $p$  belonging to the recombination frequency set given above. This implementation has the advantage of no positional bias (any schemas contained at different positions in the parents can be recombined in the offspring) and it is commonly used in recent GA applications (Mitchell, 1996). SGA and GASP were implemented with parameterised uniform crossover except for SGA in example 1 in which a previously published result was used (Michalewicz, 1996).

## RESULTS

### Schema theorem and population dynamics for GASP

The Schema Theorem (see System and Methods) does not hold for GASP because individual fitness is not compared with current population average. For GASP, the probability for a schema  $\sigma$  to increase its number  $m$  in the population follows

$$E(m(\sigma, t+1)) \geq \sum_{i=0}^{m(\sigma, t)} \sum_{h=0}^R \frac{\phi_{i,h}}{K} P \quad (1)$$

where  $R$  is the maximum offspring number with which one individual contribute to the next generation,  $\phi_{i,h}(\sigma, t)$  is the value of the objective function (absolute fitness) of son  $h$  of the particular individual realisation  $\sigma_i$  of schema  $\sigma$ , and  $K$  is the maximum or a supreme of the objective function. Thus,  $\phi_{i,h}(\sigma, t) / K$  will be the surviving probability (i.e. the fitness) of that son.  $P$  is an inferior bound of the probability of the schema to be not disrupted by recombination and/or mutation.

Equation (1) can also be expressed as

$$E(m(\sigma, t+1)) \geq R \frac{\phi(\sigma, t)}{K} m(\sigma, t) P \quad (2)$$

$$\text{where } \phi(\sigma, t) = \sum_i^{m(\sigma, t)} \frac{\phi_i(\sigma, t)}{m(\sigma, t)} \text{ and } \phi_i(\sigma, t) = \sum_h^R \frac{\phi_{i,h}(\sigma, t)}{R}$$

Thus,  $\phi(\sigma, t)$  is the mean absolute fitness averaged over the  $m$  copies of  $\sigma$  in generation  $t$ . Each individual copy  $\phi_i(\sigma, t)$  is evaluated via the averaged fitness of its expected offspring. As before,  $R$  is the maximum offspring number with which one individual contribute to the next generation and  $K$  is the maximum or a supreme of the objective function. Equation (2) implies that if a schema  $\sigma$  is short and of low order, its frequency  $m$  will increase if

$$\frac{\phi(\sigma, t)}{K} > \frac{1}{R} \quad (3)$$

i.e. if the average relative fitness of the schema is higher than  $1/R$ . Therefore, equation (2) is GASP analogous to the Schema Theorem. Note that if  $R = 1$  only schemata with average value of  $K$  (fitness of 1) can survive in the population. If  $R = 2$  only schemata with average fitness higher than  $1/2$  will increase their frequencies.

Regarding population size, the expected population size at  $t + 1$  for GASP will be the total number of expected surviving offspring, that is:

$$E(N(t+1)) = \sum_i^{N(t)} \sum_h^R \frac{\phi_{i,h}}{K} \quad (4)$$

So, to at least maintain initial population size

$$E(N(t+1)) \geq N(0) \quad (5)$$

must hold and a sufficient condition for the population not to be extinct is obtained just substituting (4) into (5) and changing  $N(0)$  by  $N(t)$ . Condition (5) becomes

$$\frac{\bar{\phi}}{K} \geq \frac{1}{R} \quad (6)$$

where  $\frac{\bar{\phi}}{K}$  is the expected offspring mean fitness of the population *i.e.* the average over all  $N(t)$  parents of the averaged fitness of each parent's expected surviving offspring.

Equation (6) stands for the condition of no population extinction. This condition and equation (4) reflect a selection type in which selection affects population size. I call this selection type *absolute selection*. If the expected population offspring mean fitness at the new population (note this fitness depends indirectly on parental fitness) is lower than  $1/R$ , population size will decrease and eventually the population will become extinct. At this point, GASP will randomly resample new  $N(0)$  individuals and iteration will continue. Such extinction behaviour will help avoiding both the slowness and eventual fall in a local optimum that can occur in SGA.

In the above analysis it has been assumed that population could grow exponentially (whenever  $R > 1$ ). In practice, to avoid an eventual "oversize" in population number a limit to the maximum population size is established. It seems rational to take as this limit the initial population size (as I did) or twice the initial population size, nevertheless, other choices could be considered. The effect of this restriction in the analysis is simply that the  $R$  value (e.g. in the case of  $R = 2$ ) is not in fact 2 but in average lower than 2 and higher than 1. Thus, conditions (3) and (6) become stronger because the value at the right part of the equations is higher. Thus, the intensity of the selection scheme is higher if the population size is limited.

Performance under example 1: a simple function

Find  $x$  from interval  $[-1,2]$  so that maximise  $f(x) = x\sin(10\pi x) + 1.0$ . The published result refers to a run of a SGA with recombination frequency of 0.25 and mutation rate  $10^{-2}$ . The best solution obtained after running 150 generations was recorded at generation 145 and was  $f(x) = 2.850227292$  which corresponds to  $x = 1.850773$  (Michalewicz, 1996 p. 22).

For GASP, I used a  $K$  value of 3 because  $f(x) = 3$  is an obvious supreme of the objective function. Thus, 1 replicate was run during 150 generations for all the 24 combinations of the parameter set. Although all solutions were very similar to those found for SGA, all were faster and most (17 of 24 cases) better. For the 17 better-valued cases the average was  $2.850272649 \pm 0.000003$  and for the 7 worse  $2.756673268 \pm 0.16$ . The average number of generations to reach the best solution was  $79 \pm 33$  in the 17 better-valued cases and  $44 \pm 57$  in the 7 worse ones. The best case was recorded at generation 8 with recombination frequency 0.4 and mutation rate  $10^{-3}$ . The worst case was recorded at generation 1 with recombination frequency 0.2 and mutation rate  $10^{-3}$  (the output for the whole parameter set is available as supplementary information).

Performance under example 2: "needle in a haystack"

In this example, the objective function will have a fitness value of 1 if decimal value of an individual is 5 or 122 and 0 otherwise. The same example was further considered with a 22 bit string. Note that in this case there are only two "good" solutions in a total of 4,194,304 possible.

In Figure 1 the frequency of replicates with population average fitness equal to optimum (optimum for this problem is 1.0) is shown for SGA and GASP for the 8-bit parameter set. For the whole parameter set, the GASP algorithm behaved much better than SGA. All replicates achieved the optimum value with GASP while only a minority (2-12%) of replicates did so with SGA. For example 2 with 22 bits, SGA never finds the solution, but GASP still achieves the solution in 6 cases of the whole parameter combination set. The percentage of replicates achieving optimum average in these 6 cases is 2-4% and all of them belong to parameter combinations with high recombination frequency and/or high mutation rate (not shown, available as supplementary information).

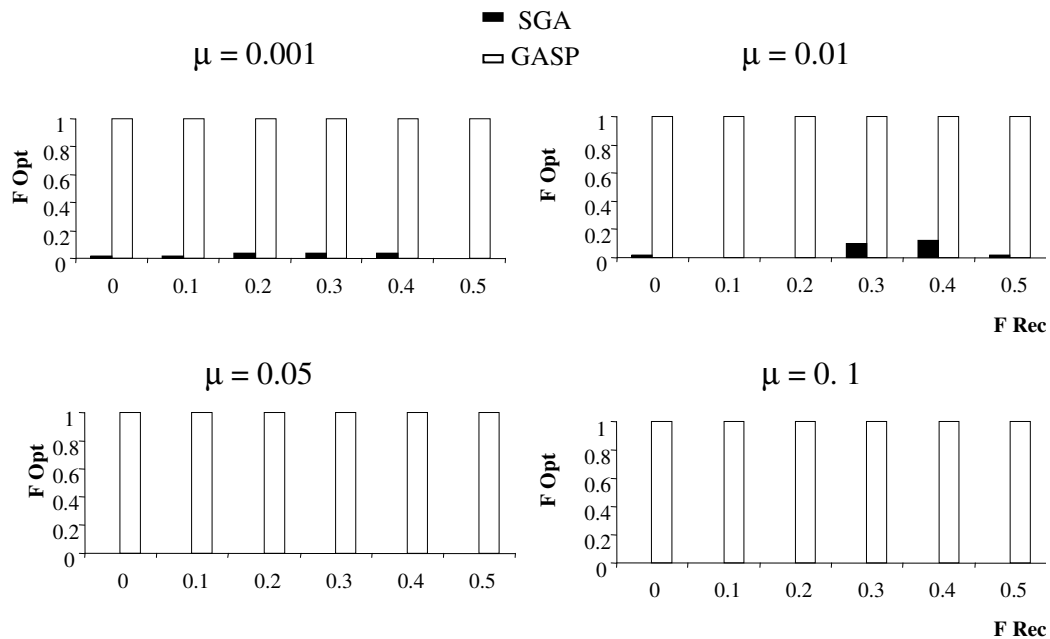


Figure 1. Frequency of replicates in example 2 for which the population average fitness is 1 (optimum) for the different recombination and mutation rate sets. F. Opt: Optima frequencies. F. Rec: Recombination frequency. From the frequencies presented in Figure 1 the algorithms could be classified by means of the quality convergence criterion (see System and Methods). The classification of both SGA and GASP for the whole parameter set is presented in Table 1.

Table 1. Classification of examples 2 and 3 following convergence quality criterion. The asterisk represents the whole range of the parameter set. The cases not shown were of class-4 both for SGA and GASP.

Example	Mutation rate ( $\mu$ )	Recombination frequency	Algorithm	Class
2 (8 bits)	*	*	GASP	1,2
2 (8 bits)	$< 10^{-1}$	*	SGA	3
2 (8 bits)	$10^{-1}$	*	SGA	4
2 (22 bits)	$10^{-2}$	0.0, 0.5	GASP	3
2 (22 bits)	$5 \times 10^{-2}$	0.3, 0.5	GASP	3
2 (22 bits)	$10^{-1}$	0.3, 0.4	GASP	3
2 (22 bits)	*	*	SGA	4
3	*	*	GASP	1
3	$< 10^{-1}$	*	SGA	3
3	$10^{-1}$	*	SGA	4

For SGA with mutation rate of  $10^{-3}$  or  $10^{-2}$  the example can be classified as class-3 because there are strong dependence on the initial choice of population for the algorithm to find the solution. In fact, with a mutation rate of  $10^{-3}$  only in less than 2% of replicates the solution is found for different recombination frequencies. With a mutation rate of  $10^{-2}$  there are two cases (recombination frequencies 0.3 and 0.4) with 10% and 12% of replicates that achieved the optimum. When mutation rate is  $5 \times 10^{-2}$  or  $10^{-1}$  the example is of class-4, that is, the algorithm does not find the solution in any case. For GASP, the situation is very different since the solution is always found independently of the initial population choice. Thus, the example is of class-1 or -2 for all combinations of parameter values assayed. The difference between class-1 and -2 refers to the variance in generation number needed to find solution, small in class-1, large in class-2. Since the average generation number for all replicates and cases was  $5.6 \pm 1$ , it seems that example 2 can be considered of class-1 for GASP. With SGA, the same average was  $147.4 \pm 3$  (note that many cases did not find the solution). Furthermore, example 2 with 22 bits is of class-4 (no solution found) for SGA and of class-3 for GASP (see Table 1). In the latter, the average generation number for all replicates and cases was  $149.5 \pm 1$ .

Performance under example 3: A biological model

In this example, ecological and demographic conditions in which extinction of a biological population takes place are searched (see System and Methods). The objective function for this example is obtained as the maximum biological population size allowed (50) minus the current population size at each generation. Thus, an individual solution with a biological population size of  $N = 50$  has an objective function value of 0 and if  $N = 0$  (extinction) the value is 50. The  $K$  value for the GASP algorithm will be the maximum, i.e.  $K = 50$ . Consequently, an individual with  $N = 0$  will have a fitness of 1 ( $(50 - 0) / 50$ ) and an individual with  $N = 50$  will have a fitness of 0 ( $(50 - 50) / 50$ ).

In Figure 2 the frequency of replicates with average population fitness higher than 0.95 is given for SGA and GASP for the 8-bit parameter set. For the whole parameter set the GASP algorithm behaved much better than SGA. All replicates achieved the optimum value with GASP while only a minority of replicates did so with SGA.



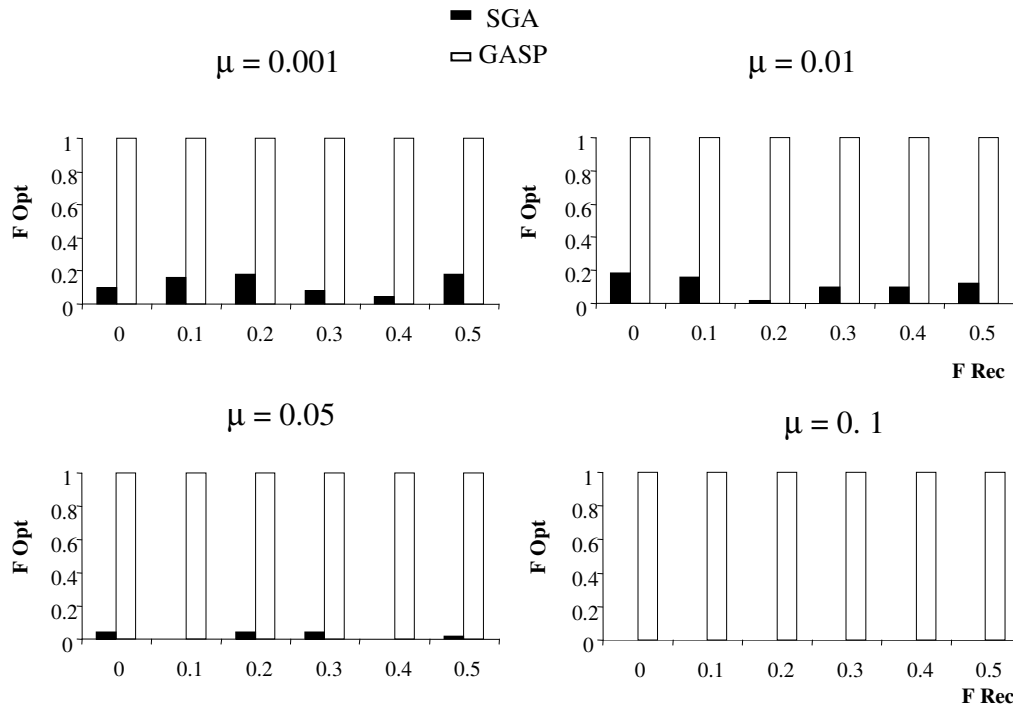


Figure 2. Frequency of replicates in example 3 for which the population average fitness is higher than 0.95 for the different recombination and mutation rate sets. F. Opt: Optima frequencies. F. Rec: Recombination frequency.

From the frequencies presented in Figure 2 we can again classify the algorithms by means of the quality convergence criterion (see System and Methods). The classification of both SGA and GASP for the whole parameter set is presented in Table 1. For SGA we see that if mutation rate is less than  $10^{-1}$ , the example is classified as class-3, that is, there is an initial sample effect although the average optimum is reached in 10-20% of replicates. The same example becomes of class-4, i.e. no solution is found, when mutation rate is as high as  $10^{-1}$ . The number of generations for SGA to reach the solution was 9 or 10 (not shown, available as supplementary information). In regard to GASP once again this is a class-1 example for the whole parameter set, because the optimum was reached for all replicates in 1 or 2 generations (available as supplementary information).

Example 3 has multiple optima solutions and it could be of interest the ability of the algorithm to find more than one optimal solution. To compare how algorithms perform for the number of different optima solutions found, we just run 1 replicate for both algorithms and the whole parameter set. The average number of optima solutions for the whole parameter combination were  $3.46 \pm 1.7$  for GASP and  $1.79 \pm 2.2$  for SGA. The average number of generations spent to find the solutions were  $1.08 \pm 0.28$  for GASP and  $9.92 \pm 0.41$  for SGA. That is, GASP found more solutions than SGA and did it faster.

## DISCUSSION

Since the theory of genetic algorithms was founded by the pioneer work of Holland (1975) there has been multiple variants on its original formulation. These variants concern to the applied selection scheme, population size, the genetic operators, the codification system and so on. Nevertheless, in all of the variants individual selection is referred to current population average fitness and population size is held constant or, at least, is not affected by the selection pressure.

Therefore, between GASP and SGA two fundamental differences are evident:

- 1) An *absolute selection* mode is applied in GASP in which individuals are evaluated with respect to an absolute value rather than to the current population average fitness.
- 2) As a consequence of the former, in GASP the selection pressure affects population size and the population can undergo extinction.

This two points are the consequence of the introduction of two new parameters, namely,  $R$  (maximum offspring number with which one individual contribute to the next generation) and  $K$  (an *a priori* defined absolute value to compare with each individual fitness). If  $R$  is high, selection intensity diminishes its effect and if  $K$  is high (respect to current population values or indeed besides the maximum) selection intensity is increased allowing survival of only the best-fit individuals. However, it is important to note that in no case selection becomes deterministic as in elitism or other classical hard selection schemas of GA theory. Because of this, the *absolute selection scheme*, depending on the  $R$  and  $K$  values, allows to survive individuals despite possessing relatively poorer objective values or, on the contrary, only the best ones will have opportunity to survive and contribute to the next generation. Nevertheless, in the implementation performed here, GASP permits that the search of a solution can occur at two different levels: First, via the selection plus mutation and recombination operators as in other GAs. Second, if in successive generations the fitness in the population can not maintain the population size, extinction could happen and then, resampling a new population will continue the search.

It is also important to note that the variable population size allows GASP to work quickly because implies shorter computation times. Interestingly, a lower population size will occur if the average fitness is low i.e. individuals at population are of less interest. An extreme case of extinction behaviour appears in the so-called needle in a haystack (NIAH) problems (Kumar, 2000), as in example 2 above, in which populations will become successively extinct until the "needle" is found. Here the above mentioned second level of search becomes more important.

All the examples presented demonstrate the efficiency of GASP. In examples 2 and 3 the maximum of the objective function was known. In example 1 a reasonable supreme could be guessed. The interplay between selection intensity (via the  $K$  value) and population size (via  $R$  and maximum population size allowed) allows the finding of optimum in a quick and efficient way.

It is worth mentioning that in a biological context, as in population biology and conservation genetic models, the value to find it is usually known and one wants to maximise (minimise) the population genetic variance or the population size, allele number, etc., until a given value is reached. On the contrary the conditions which drives population to the desired optimum e.g. the migration rate, initial population size, offspring number and so on, are unknown. In this kind of situations, GASP will work faster and better than more classical GAs. Other interesting field for the use of GASP is protein structure prediction. In this context the objective function is a complex one based on the idea of minimising some potential energy function. The problem stands both for finding an adequate function i.e. an adequate potential-energy model, (Mitchell 1996, refs. therein) and an adequate GA. This function has been proposed to be decomposed via a multiobjective approach and then a modified GA was used to solve protein structure (Day *et al.*, 2001). It seems possible, given an energy model, to know the maximum or at least a supreme of such kind of functions and hence to apply a GASP-based algorithm and, in this way, take advantage of its properties.

## REFERENCES

- Carvajal-Rodríguez,A., Rolán-Álvarez,E. and Caballero,A. (2003). A simulation study comparing molecular and quantitative variation for detecting human-induced impacts on genetic diversity, Submitted to Genetical Research.
- Davidor,Y. (1991). Epistasis variance: a viewpoint on GA-hardness. In Rawlins, G. (ed.), Foundations of Genetic Algorithms, Morgan Kauffmann, Berlin, 23-25.

- Day,R.O., Zydallis,J.B., and Lamont,G.B. (2001). Solving the protein structure prediction problem through a multiobjective genetic algorithm. *ICNN*.
- Goldberg,D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York.
- Hartl,D.L. and Clark,A.G. (1989). *Principles of Population Genetics*. Sinauer Associates Inc, Sunderland Mass.
- Holland,J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press; (Second Edition: MIT Press 1982).
- Kumar,V. (2000). Tackling epistasis: a survey of measures and techniques, Technical report at the DCL University of Illinois, IL 61801, 1-12.
- Michalewicz,Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag.
- Mitchell,M. (1996). *An Introduction to Genetic Algorithms*, MIT Press, Massachusetts.
- Naudts,B. and Kallel,L. (1998). Some facts about so called GA-hardness measures, Technical report at the CMAP Ecole Polytechnique, 379, 1-46.
- Naudts,B., Suys,D., and Verschoren,A. (1997). Epistasis as a basic concept in formal landscape analysis, In Back,T. (ed.), *Proceedings of the 7th International Conference on Genetic Algorithms*, Morgan Kaufmann.
- Spears,W.M., and DeJong,K.A. (1991). On the virtues of parameterized uniform crossover, In Belew,R.K. and Booker,L.B. (eds.) *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, La Jolla, CA, 230-236.

#### Acknowledgements

I thank A. Caballero, D. Posada and E. Rolán-Álvarez for their helpful comments on the manuscript. D. Posada also "baptize" GASP. I also thank J.L. Campos and J. J. Pasantes for English corrections. My research is funded by European Union (postdoc contract associated to grant EVK3-CT-2001-00048).