Ideal Marriage for fine tuning in GA

Goutam Chakraborty & Basabi Chakraborty
Iwate Prefectural University
Department of Software & Information Science
Iwate Ken, Takizawa Mura, Japan - 020-0193
E-mail: goutam/basabi@soft.iwate-pu.ac.jp

ABSTRACT

While searching for optimum solution using genetic algorithm, it is always critical to properly balance between adequate exploration of the search space during earlier generations, and putting required selective pressure to find the optimum during later generations. This change from more explorative search to close selective search should again depend on the problem and preferably done adaptively. Common approaches to solve this are fitness scaling, ranking of the chromosomes, tournament selection etc. They balance exploration vs. exploitation during the process of genetic selection. Other proposals are to adaptively change the probabilities of crossover and mutation operations as genetic search progresses. Our proposal is to tune this by restricting the choice of partners for crossover over generations. In real life marriages(crossovers) occur between two individuals of similar status in society, only when they are mature in age and usually from neighboring localities. Similar principle is extended in selecting cross-over partners in GA. In the proposed strategy, the probability of cross-over is more when their rank in the whole population is close and their age matured. At early stage restricting crossover to chromosomes of similar rank would lead to bad exploration. The probability function for selecting partners depending on their ranks changes with advancing generations, so that the effect is negligible in the beginning. At later stage the effect is accentuated so as to be able to fine tune good chromosomes to achieve fast convergence and reach optimum values. The scheme is not centralized like elitist approach. The case of effecting crossover only to partners of matured age were separately studied. The effectiveness of this new method is tried on problems of maximizing complex multimodal functions. The results are compared with standard genetic algorithm (SGA) and SGA with "linear fitness scaling". Results show that our strategy is superior in terms of probability of hitting the maximum value as well as speed of finding the maximum.

1. INTRODUCTION

Genetic Algorithm (GA) is a search algorithm based on the mechanics of natural selection [1]. Compared to other approaches, they are superior because, (1) of wide applicability and make few assumptions from the problem domain, (2) and are not biased towards local minimums. At the same time GAs are very efficient to direct the search towards relatively prospective regions of the search space.

The first step in GA is to encode the solution of the problem in binary bit string. This coded solution is often called a chromosome. Next, one has to generate a pool of solutions of the problem to form the initial population. A fitness function has to be defined to measure the goodness of these encoded solutions. Then genetic operators *selection*, *crossover* and *mutation* operate on the population to generate new population (new set of solutions) from the old ones. Good solutions are *selected* with greater probability to next generation, in line with the idea of *survival of the fittest*. *Crossover* operation recombines selected solutions, by swapping part of them, producing divergent solutions to explore the search space. An occasional *mutation* is done on a chromosome by flipping a bit at random position of the encoded chromosome, to facilitate jumping of solutions to new unexplored regions of the search space. As the algorithm continues and newer and newer generations evolve, the quality of solutions improve.

Many strategies for fitness calculation, selection, crossover and mutation are proposed. We here describe the Standard Genetic Algorithm (SGA) in brief. The basic steps of SGA are shown below.

```
Algorithm SGA ( $g, G, \Pi(g), P$ )
begin
    $g = 0$;  /* $g$: generation number */
    Create $P$ members of the initial population $\Pi(0)$;
            /* $P$: population size,*/
            /* $\Pi(g)$: set of members at generation $g$ */
    Evaluate members of $\Pi(0)$;
        /* Calculate fitnesses $f_i$s of all members $m_i$s */
    while ( $g \leq G$ )   /* $G$: maximum generation */
        $g := g + 1$;
        $\Pi''(g) \xleftarrow{selection} \Pi(g-1)$;
        $\Pi'(g) \xleftarrow{crossover} \Pi''(g)$;
        $\Pi(g) \xleftarrow{mutation} \Pi'(g)$;
    endwhile
end
```

Figure 1: Algorithm for Standard Genetic Algorithm

The success of genetic search depends on balancing the two aspects of (1) population diversity i.e. exploring the different regions of the search space, and (2) selective pressure i.e. to get to the exact optimum point. In SGA, the best few members of the initial population could predominate the whole population in a few early cycles due to their much better fitness compared to others and therefore high probability of getting selected. This would result in poor exploration and premature convergence to suboptimal maximum/minimum. On the other hand, at later stage of the search, when high performance regions are identified, fine local tuning is necessary to get quickly to the optimum solution, especially for high precision problems. It is difficult to achieve this quickly by SGA because of disruption of good chromosomes after crossover with bad ones. A number of strategies were proposed [2](chapter 4 and 6) to overcome this problem by setting a balance between diversity (during the beginning) and selection pressure (at the end). We briefly discuss this before introducing our ideas. The existing approaches are:

0-7803-5731-0/99$10.00 ©1999 IEEE

1. To control selection by controlling the differences of fitnesses e.g. fitness scaling, ranking.
2. To control selection by novel selection method e.g. tournament selection.
3. By adaptively changing the crossover and mutation probability, and sometimes positions where mutation should occur.

One of the early proposal was to scale the fitness function [1] (pp. 122-124). To sustain diversity in the beginning, the fitnesses are scaled down so that the influence of high fitness is diminished in the selection stage. At the later stage of search, when most of the chromosomes have similar and high fitnesses, a reverse scaling is done to accentuate the effect of higher fitness and thus facilitating selection of only best chromosomes for faster convergence. Ranking of the chromosomes according to their fitnesses, and not using the exact values of the fitness for selection is another way of scaling of fitnesses throughout all generations to achieve similar goal [4]. There selection pressure could be tuned linearly or non-linearly by a tuning parameter [2] (pp. 60-61). In the approach named non-uniform mutation [6], at later stage of generations mutations are probabilistically performed more towards the tail part of the coded chromosomes. This is to avoid disrupting a good chromosome from its present location by changing bits at the more significant head part. Another class of proposals is to adaptively change the crossover (p_c) and mutation (p_m) probabilities [5] [7]. In [7] the best few chromosomes are disrupted with much less probability than those with weak fitnesses. Thus the weak chromosomes are used for exploration of different regions, and the good ones to find optimum locations.

Our proposed strategy is to make some reservation while effecting crossover between two chromosomes. The approach is quite different from those mentioned above and therefore is not conflicting with them. Here, after randomly selecting two partners for mating, the crossover is actually performed only when the partners are of similar status, or they are enough matured (old). Restricting partners for crossover is done in several other works. In elitism only the top 10-20% performing individuals are permitted to mate. Note that this is contrary to the proposal in [7], but somewhat similar to our approach. This obviously leads to premature convergence and proposals to prevent that are made in [8]. The common 'traits' and 'preferences' found in many organisms for chosing partners is modeled in [9] and experimented through simulations [10] to show how these 'traits' and 'preferences' are adapted in a population. This idea is extended to models like *pygmies and civil servants* in [12] and *selection meeting seduction* in [11], for solving problems with two optimization criterions. In comparison, our chromosomes are not classified in any respect. Their affinity for mating depends on the closeness of their rank and age probabilistically.

We here propose two new crossover strategies, (1) rank based crossover(RBC) and (2) age based crossover (ABC). In the proposed RBC strategy, though two chromosomes are randomly chosen for crossover, the probability that the crossover actually takes place depends on how near they are in the fitness scale. The nearer they are, more is the probability that they will be crossed over. Initially this proximity range covers all the chromosomes (the whole of the fitness scale) so that all randomly chosen pairs are crossed over. This is to allow exploration of different regions of the search space. Slowly when good regions are discovered, we allow crossover between chromosomes of similar rank only. Then good solutions recombine fast to find the optimum location, somewhat like elitism. This allows fast convergence to optimum location. The probability function changes with generations. Different modes of this change were tried in various simulations with RBC.

A chromosome which propagates from one generation to the next, and do not undergo any crossover or mutation gets older. Chromosomes created after crossover or mutation are new borns. A chromosome which does not change is not helping in exploring the search space. In ABC strategy, after two chromosomes are randomly selected for crossover, the actual crossover depends probabilistically on the their ages - the older they are the more is the probability that they will be crossed over. It is to be noted that the RBC, ABC, fitness scaling, adaptive p_c, none of these strategies are conflicting and actually could be simultaneously used to improve the performance of genetic search.

As is obvious, RBC does not improve over SGA for exploration. It improves the speed of convergence and thereby quality of solution due to intense selection pressure at the end. RBC also facilitates tuning this range very easily by tuning the fitness range function. The detail is given in section 2. In section 4 we present experimental results to show how RBC could deliver better results more efficiently, compared to standard genetic algorithm (SGA) as well as genetic algorithm using linear fitness scaling. The details of ABC strategy is given in section 3. Conclusions and future research plans are discussed in section 5.

2. RANK BASED CROSSOVER

Motivation: Our proposed RBC algorithm is same as SGA except for some reservations about effecting crossover on two selected partners. During early generations diversity is important to explore different regions of the search space. As generations progress, the selection pressure is to be increased to find the optimum location quickly. Even at the late stage of evolution, if good solutions (from prospective optimum regions) are disrupted due to crossover with bad solutions, reaching optimum solution would be delayed enormously. In RBC we restrict the crossover, probabilistically, between chromosomes of similar rank to speed up the convergence at the later stage of searching.

Rank Based Crossover

The algorithm of SGA is described in the Introduction. From members of $\Pi(g-1)$ the new population $\Pi''(g)$ is reproduced by selection, where better fitness members are selected with higher probability. For crossover in SGA, pairs of members are randomly selected in turn and crossed over $\lceil (p_c \times P)/2 \rceil$ times, where p_c is the probability of crossover, so that out of P members of the population about $(p_c \times P)$ are get crossed over. The crossed over members, sometimes called children, replace the original chromosomes (also called parents) and form $\Pi'(g)$ population. Mutation is done on $\Pi'(g)$ to get the next generation population $\Pi(g)$.

```
Algorithm RBC ( $\Pi''(g), p_c, P$ )
while ( $no\ of\ crossovers \leq \lceil (p_c \times P)/2 \rceil$ )
    Select randomly a pair $m_i(g)$, $m_j(g)$ from $\Pi''(g)$;
    /* where $\Pi''(g)$ is the population after selection */
    if ( $\varphi(f_i^{nor}(g) - f_j^{nor}(g)) > rand(0,1)$ )
        /* $f_i^{nor}(g), f_j^{nor}(g)$ are normalized */
        /* fitnesses of $m_i(g), m_j(g)$ */
        Perform crossover between $m_i(g)$ and $m_j(g)$;
    endif
endwhile
```

Figure 2: Algorithm for rank based crossover

The basic steps of RBC strategy for crossover is shown above. First the fitness of all members in $\Pi''(g)$ are normalized as follows. If $f_i(g)$ be the fitness of member $m_i(g)$, the normalized fitness denoted by $f_i^{nor}(g)$ is,

$$f_i^{nor}(g) = \frac{f_i(g) - f^{min}(g)}{f^{max}(g) - f^{min}(g)}$$

where, $f^{max}(g)$ and $f^{min}(g)$ are the maximum and the minimum fitness of all the members of $\Pi''(g)$. We need to know only the values for $f^{min}(g)$ and $f^{max}(g)$ for finding this normalized fitness.

As shown in line 01 of Fig. 2, until enough number of crossover is done, two members from $\Pi''(g)$ are selected at random. Let their normalized fitnesses are $f_i^{nor}(g)$ and $f_j^{nor}(g)$. Function $\varphi(f_i^{nor}(g), f_j^{nor}(g))$, which decides the probability of crossover is of the following form,

$$\varphi(f_i^{nor}(g), f_j^{nor}(g)) = exp\left[-\frac{1}{2}\left(\frac{f_i^{nor}(g) - f_j^{nor}(g)}{\sigma(g)}\right)^2\right]$$

The shape of φ is same as the normal function with maximum = 1 at $f_i^{nor}(g) = f_j^{nor}(g)$. φ value decreases as the distance $(f_i^{nor}(g) - f_j^{nor}(g))$ increases. Finally, as written in line 03 of the Algorithm RBC, the crossover is done probabilistically, only when $\varphi > rand(0,1)$. Here $rand(0,1)$ generates a random real number between 0 and 1 with uniform distribution. More close are $f_i^{nor}(g)$ and $f_j^{nor}(g)$, more probable is their crossover.

$\sigma(g)$, a function of the generation number g controls the variance of φ. During earlier generations, when g is low, $\sigma(g) \approx 1$, and thus allowing crossover for any randomly selected pair from $\Pi''(g)$. When g is large and nearing G (the maximum generation), $\sigma(g)$ becomes small. Then φ becomes sharp allowing only pairs with closer fitnesses to cross. Function $\sigma(g)$ is defined as follows:

$$\sigma(g) = 1 - \left[\frac{g}{\lambda G}\right]^{\alpha}$$

where α and λ are the controlling parameters. The way $\sigma(g)$ changes with generations is shown in Fig. 3, with $\lambda = 1.1$ and $\alpha = 2, 1, 0.5$. We used these values in our experiments.

When $\alpha > 1$, $\sigma(g)$ changes slowly in the beginning (when g is low) and rapidly afterwards. Thus exploration is not hampered in the beginning, whereas rapid convergence is forced at the end. On the other hand, when $\alpha < 1$, the selection pressure is high almost from the beginning. For simple search space, this would find the optimum fast, but may miss the global maximum for complex multimodal functions. When $\alpha = 1$, $\sigma(g)$ decreases linearly. Thus the efficiency and success of the algorithm depends somewhat on the complexity of the search space and the choice of α. We will see that $\alpha = 1$, i.e. linear decrement of $\sigma(g)$ is a good choice for all situations. In general we will show by several experiments that RBC strategy works much better compared to SGA as well as SGA with "linear fitness scaling".

3. AGE BASED CROSSOVER

Motivation: In age based crossover (ABC) strategy, the basic algorithm is same as SGA except that after the crossover partners are selected, the actual crossover is done after considerations to their ages. The main motivation is to disrupt chromosomes that are copied from generations to generations. Thus when both the partners are matured (old) the probability of crossover is high. Less is the probability when both the partners are very young. With every chromosome, we keep record of its age. At the time of creation of the initial population the age of all the chromosomes are set to 0. If a chromosome remains from one generation to the next without any change, its age is increased by unity. Once two chromosomes are crossed over, the age of the new born children are set to 0. Also, the age is reset to 0 when a chromosome is mutated. The detail of how the probability of crossover depends on the ages of the two participants is discussed in the next section.

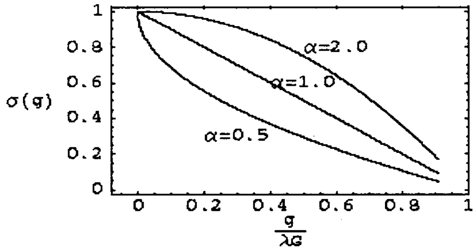

Figure 3: Function $\sigma(g)$ for different α

There are already some proposals to restrict the life time of individuals to one or two generations only, regardless of their fitnesses. This is done at the selection stage - old chromosomes are not allowed to reproduce i.e. go to the next generation. We on the other hand indulge in old individuals, which due to high fitness propagates through generations, to be disrupted by crossover. In addition with fitness scaling, or ranking with low selective pressure, ABC strategy would give good exploration. The detail is explained with psudocode of the ABC algorithm in the next section.

Age Based Crossover

The algorithm of SGA is described in the Introduction. For crossover in SGA, pairs of members are randomly selected in turn and crossed over $\lceil (p_c \times P)/2 \rceil$ times. The crossed over members replace the original chromosomes and form $\Pi'(g)$. Following psudocode presents the working principle of ABC.

```
Algorithm ABC ( $\Pi''(g), p_c, P$ )
01 while ( $no\ of\ crossovers \leq \lceil (p_c \times P)/2 \rceil$ )
02     Select randomly a pair $m_i(g)$, $m_j(g)$ from $\Pi''(g)$;
       /* where $\Pi''(g)$ is the population after selection */
03     if ( ($rand(0,1) \times (\alpha_2/\alpha_P)$) $> 0.5$)
           /* $\alpha_P$ is the av. age of whole population */
           /* $\alpha_2$ is the av. age of $m_i(g)$ and $m_j(g)$ */
04         Perform crossover between $m_i(g)$ and $m_j(g)$;
05     endif
06 endwhile
```

Figure 4: Algorithm for Age Based Crossover

The basic steps of ABC strategy for crossover is shown above. First the average age of all members in $\Pi''(g)$ is found. It is denoted as α_P in Fig. 4. The average age of the two chromosomes, $m_i(g)$ and $m_j(g)$, randomly selected from $\Pi''(g)$ to be crossed over is α_2. $rand(0,1)$ returns a random fraction between 0 and 1 with uniform probability distribution. When the age of $m_i(g)$ and $m_j(g)$ are greater than the average, they crossover with higher probability. Thus on an average old chromosomes are disrupted more often than fresh chromosomes.

4. EXPERIMENTAL SET UP AND RESULTS

The effectiveness of RBC algorithm is demonstrated by solving maximization problem for several univariate and multivariate multimodal functions. The functions are given in Table. 1. These Functions f_1 to f_3 are DeJong's functions and were used by many [1][7].

	Function Form	Domain	Maxvalue
f_1:	$2 + e^{x-10}\cos 10 - x$ if $x \leq 10.0$ $2 + e^{10-x}\cos x - 10$ if $x > 10.0$	[0,20]	3.00
f_2:	$100(x_1^2 - x_2)^2 + (1 - x_1)^2$	[−2.048,2.048]	3905.00
f_3:	$\sum_1^5 integer(x_i)$	[0.0,10.0]	50

Table 1: Functions used in the experiment

Simulation Parameters: We implemented Standard Genetic Algorithm (SGA), SGA with liner fitness scaling, and our proposed GA with RBC. For RBC, we used three different functions for $\sigma(g)$ with $\alpha = 2.0$, $\alpha = 1.0$ and $\alpha = 0.5$. Therefore, in total we compare results of 5 different algorithms. In all runs we set $p_c = 0.6$, $p_m = 0.03$ and $P = 100$. One point crossover method is performed. Because we used the maximum number of generations as the stopping criterion, and because we had used three functions of different complexity, for all functions we have performed our experiments for three different cases with different maximum generation (MAX_GEN) numbers as 100, 1000 and 10000. Independent 50 runs with different initial populations are done for each 3 cases of maximum number of generations. Initial population is same for all the five different algorithms, but are different in the 50 different runs. The following results in the next section are averages over 50 independent runs.

Analysis of results

In this section the results from five different algorithms to find the maxima of the functions f_1 to f_3 are analyzed. We present here only two most significant results.

1. The best fitness value up to a certain generation (average over 50 runs), and
2. The number of times a certain algorithm could hit the 99.9% of the actual maximum of the respective functions.

By (1) we could see how fast the convergence is achieved and by (2) we can judge the probability of hitting the target.

In Fig. 5 to Fig. 7 the functions and the best fitness values found until that generation is plotted against the number of generations. The actual maximum value calculated analytically is also mentioned in the respective captions. For each function the simulation is done with three different maximum generation $G = 100, G = 1000$ and $G = 10000$. In the following figures *star*, *triangle* ($\triangle$), *box*, *pentagon*, and *cross* ($\times$) are used to indicate results obtained using SGA, SGA with fitness scaling, RBC with $\alpha = 1.0$, RBC with $\alpha = 2.0$, and RBC with $\alpha = 0.5$ respectively.

From the results of Fig. 5 to Fig. 7, it is evident that the proposed RBC strategy could reach better quality of results faster compared to SGA as well as SGA with fitness scaling. For simple functions like f_1 and f_2, the maximum is reached soon, in 100 generations using RBC strategy, but not with SGA or even SGA with fitness scaling. Results obtained using RBC strategy are best, and almost similar for different α values. Only in cases of function f_4 and f_5, results with linear decrement of $\sigma(g)$ i.e. with $\alpha = 1.0$, are better compared with $\alpha = 0.5$ or $\alpha = 2.0$. So we may conclude that $\alpha = 1.0$ could be a good choice for every occasion.

Finally in Table. 2 we give the results of how many times an algorithm could reach within 99.9% of the target maximum, out of the 50 random trials (with different initial population). This would give an idea of the probability of success, the hit-ratio, of the different algorithms and compare them. Here we see that the performance of the proposed algorithm is much better compared to SGA and SGA with fitness scaling. We also note that a choice of $\alpha = 2.0$, when there is lots of exploration in the beginning and high selection at the end, gives the best hit-ratio.

5. CONCLUSIONS

New reservation methods while crossing over two chromosomes are being proposed to increase selection pressure and improve exploration of the search space is being proposed. In RBC, crossover partners randomly selected from population are allowed to perform crossover only probabilistically and is controlled by a function. The probability function which controls this permission is very wide in the beginning allowing any two randomly selected members to be crossed. But slowly with progressing generations this function becomes very narrow, allowing members only with close fitness values to be crossed. Simulation results with functions of different complexity show that the best fitness chromosomes are created with higher probabilities using RBC strategy. In ABC, crossover partners randomly selected from population are allowed to perform crossover with greater probability when their average age is more than the population average. This facilitates better and faster exploration of the search space.

It should be emphasized that the proposed RBC and ABC strategies could be used together and in fact in conjunction with other other approaches to achieve similar goals. This is because the existing approaches try to deal with this problem at the selection stage and sometimes by manipulating fitnesses. It would be interesting to perform experiments by using them together.

In case of RBC, depending on the nature of the search space, there would be an optimum choice for α. It is evident that for complex search problems, a higher value of $\alpha > 1.0$ is a better choice for better exploration in the beginning. On the other hand, for simple unimodal problems, $\alpha < 1$ should work faster. The best would be to to adaptively set the value of α with growing generations, by observing the mode of fitness change for the whole population. Also it would be an interesting exercise to combine RBC and ABC by a unique probability function.

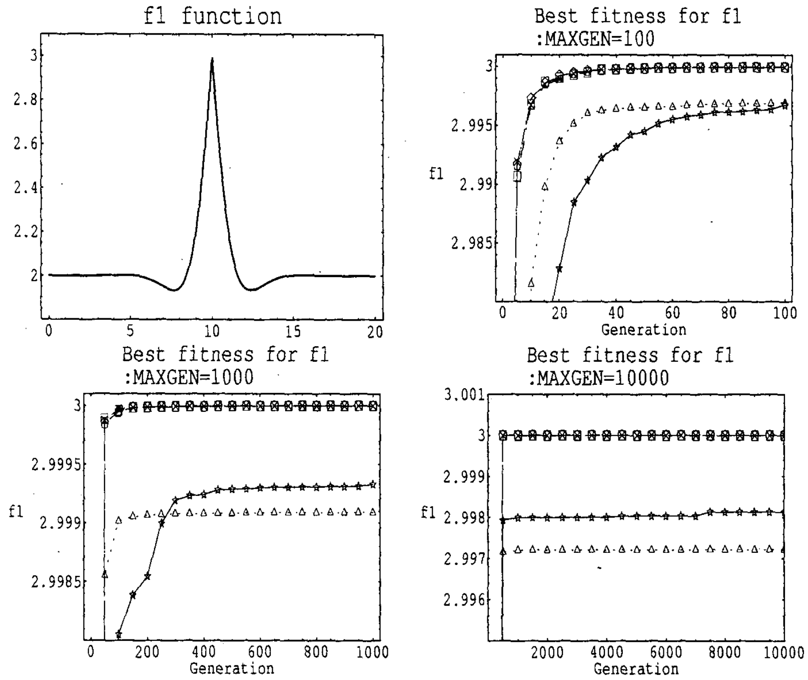

Figure 5: Best fitness until the current generation for f_1, actual maxima is 3.000

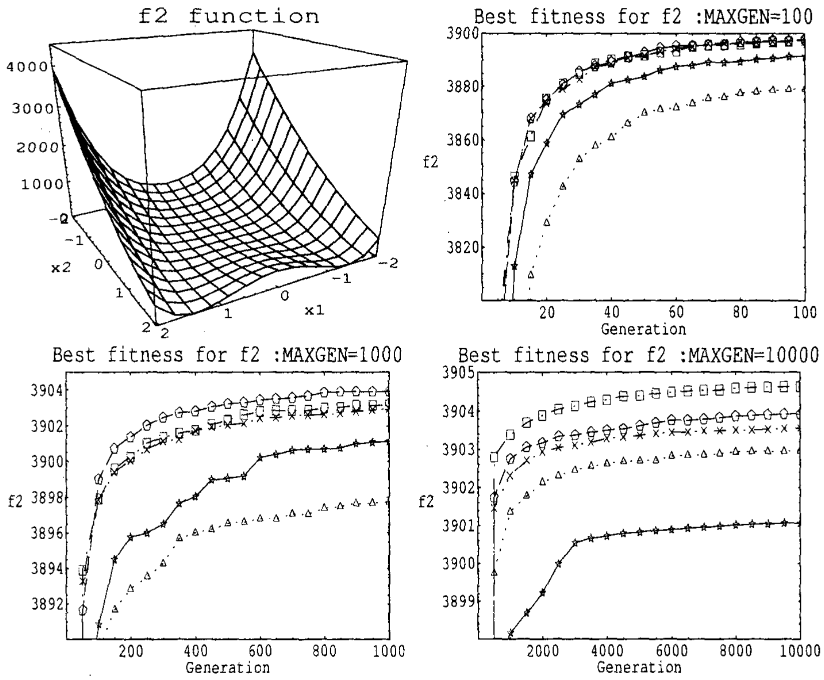

Figure 6: Best fitness until the current generation for f_2, actual maxima is 3905.000

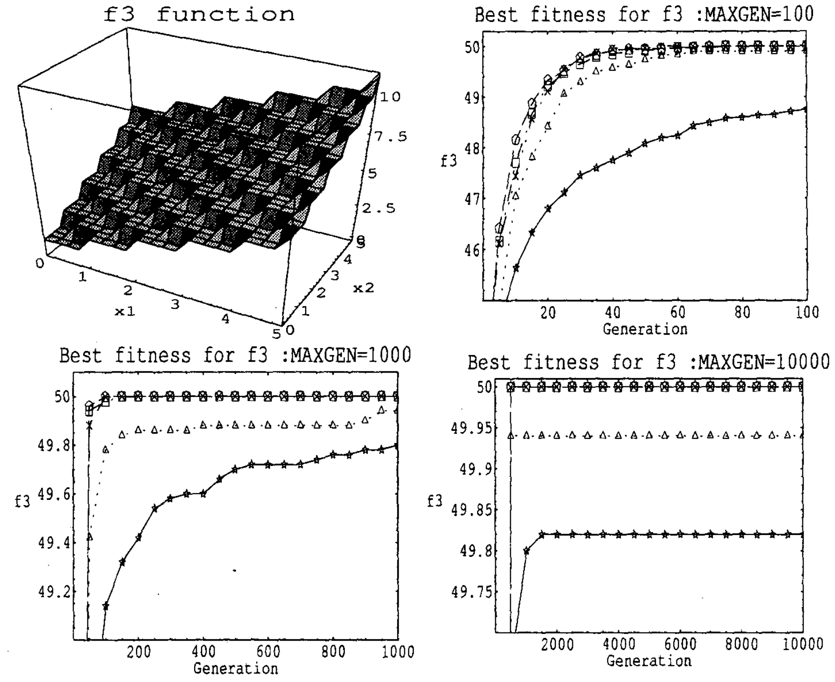

Figure 7: Best fitness until the current generation for f_3, actual maxima is 50.000

	MAX_GEN	SGA	Scale	α=1	α=2	α=.5
f_1	100	37	44	50	50	50
	1000	46	43	50	50	50
	10000	45	44	50	50	50
f_2	100	50	50	50	50	50
	1000	36	24	41	46	39
	10000	34	36	44	50	42
f_3	100	7	45	50	50	50
	1000	42	48	50	50	50
	10000	44	47	50	50	50

Table 2: Hit-ratio for different algorithms

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley,1989.

[2] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*. Springer-Verlag, 3^{rd} edition, 1995.

[3] Melanie Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1996.

[4] Darrell Whitley, "The *GENITOR* Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive trials is Best," in *Proc. of the International Conference on Genetic Algorithms*, pp.116-121, 1989.

[5] J. D. Schaffer and A. Morishima, "An adaptive crossover mechanism for genetic algorithms," in *Proceedings of Second International Conference on Genetic Algorithms*, pp. 36-40, 1987.

[6] T. C. Fogerty, "Varying the probability of mutation in genetic algorithms," in *Proc. of the International Conference on Genetic Algorithms*, pp. 104-109, 1987.

[7] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of crossover and mutation in Genetic algorithms," *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-24, No. 4, pp.656-666, April 1994.

[8] Larry J. Eshelman, "Preventing Premature Convergence in Genetic Algorithm by Preventing Incest", In *Proceedings of the International Conference on Genetic Algorithms*, pp. 115-122, 1991.

[9] M. Kirkpatrick, and M. Ryan, "The Evolution of mating preferences and the paradox of the lek.", *Nature*, vol. 350, pp.33-38, 1991.

[10] R. J. Collins, and D. R. Jefferson, "The evolution of sexual selection and female choice", In F. J. Varela and P. Bourgine, eds., *Toward a Practice of Autonomous Systems: Proceedings of First European Conference on Artificial Life*. MIT Press. 1992.

[11] Edmund Ronald, "When Selection Meets Seduction", In *Proceedings of the International Conference on Genetic Algorithms*, pp. 167-173, 1995.

[12] Conor Ryan, "Pygmies and Civil Servants", In *Advances in Genetic Programming*, Kenneth E. Kinnear Ed. pp. 243-263. MIT Press 1994.