Racial Harmony and Function Optimization in Genetic Algorithms - The Races Genetic Algorithm

Conor Ryan

Abstract

This paper introduces a modified genetic algorithm, *RGA*, which, through the use of races, maintains a very diverse population. RGA is shown to be particularly good at solving mulitmodal functions and outperforms current methods, both in discovering solutions and maintaining them.

1 INTRODUCTION

Many papers investigating genetic algorithms concern themselves with the location and optimization of a single global maximum (Beasley, D. et. al. 93). The solution landscape of these problems usually contains a single, dominant peak which the population converges on.

But, many problems of interest contain several peaks in the fitness landscape. Sometimes, only one of these peaks will be of interest, but often (Spears 94) (Deb 89a) (Beasley, D. et. al. 93) it is desirable to find all peaks of the landscape. Unlike problems in which the peak of interest changes with time, (Ryan 94c) (Hillis 92) (Siegel 94), it is important to maintain solutions at all peaks in the landscape, ensuring that one peak does not dominate and ensuring that all peaks are actually discovered by a population. Traditional GAs find locating all peaks in the landscape a difficult problem(Goldberg 87) (Goldberg et. al. 92) and several solutions which are reviewed below have been suggested.

This paper presents a modified version of the Pygmy Algorithm (Ryan 94a) which results in a new method for the maintenance of several solutions to a multimodal function in a single population - the Races Genetic Algorithm(RGA). The tests on the RGA algorithm are those orginally used by Deb and since adopted by Beasley and Spears as the standard test bank for assessing the proficiency of algorithms in multimodal function optimization.

Like Beasley, the measurements of performance used is the number of peaks located in the fitness landscape that exceed a certain *threshold*. Beasley described such peaks as *maxima of interest*. Much of the earlier work on multimodal functions (Deb 89a) (Deb 89b) (Goldberg 87) assumed (a) that the number of maxima of interest in a function is either known or can be estimated, and (b), that these maxima are spread reasonably evenly throughout the search space, however, the RGA algorithm does *not* require any such assumptions and can be applied with little or no knowledge of the type of maxima of interest present in the search space.

Below is a short review of previous work in this area.

Crowding

Much previous work divides the search space into various environmental *niches* occupied by similar individuals. Ideally, a search space could be divided into as many niches as there are maxima of interest in it, and each niche could concentrate on finding its own maxima of interest. One method of encouraging the creation of niches is that of *crowding*, (DeJong 75) Crowding takes the analogy from nature that the fiercest competition for survival between individuals is most likely to take place between those individuals occupying the same environmental niches as it is individuals such as these who are most likely to be competing for similar resources. Thus, when a new individual is entered into a population, it replaces the individual it most closely resembles taken from a small sample of the population. This sample is referred to as the *crowding factor* and is normally set to about two or three.

Although crowding does encourage niches, it has been shown (Mahfoud 93) that it can only encourage a population to exist on one or two peaks in the landscape.

Fitness sharing

The niche analogy is carried a step further by use of the *sharing* mechanism (Goldberg 87) which looks upon fitness as a single resource which has to be shared by individuals occupying the same environmental niche. An individual's fitness is derated by an amount directly related to the number of similar individuals in the population:

$$sh(d) = 1 - \left(\frac{d}{\sigma_{share}}\right)^{\sigma} \; (d < \sigma_{share})$$

While sharing has been shown to outperform crowding in multimodal function optimization (Goldberg 87), it is computationally expensive, as individuals have to be compared to every other individual in the population when calculating the sharing function shown above. It has been suggested by Mahfoud that it might be possible to estimate the sharing function by population sampling which would reduce the computation involved.

Labels

The previous two methods each employ some distance metric to decide what degree of sharing is present. Spears (Spears 94) presented a method involving the use of labels on individuals to measure similarity between them. Specifically, every individual has a label, and only mates with an individual who also has the same label, which promotes the likelihood of individuals searching the same area mating with each other, as individuals searching the same area tend to have the same labels.

Spears did not claim that this method would ensure discovering all peaks, rather he suggested that the use of labels could replace distance metrics in crowding and sharing schemes, thus increasing their efficiency. It is possible that individuals with several different labels could end up searching the same peak, but would not mate with each other because of the different labels. Unlike the sharing and crowding methods, however, using labels does not require the extra assumptions described in Section 1.

Spears' use of labels is discussed further in section 3.1.

Parallel sub-populations

Another technique, reported on by Beasley (Beasley, D. et. al. 93), is to divide a population into various sub-populations and evolve them in parallel. There is, of course, no guarantee that each sub-population will investigate a different maximum of interest, nor is there any reason why each sub-population shouldn't investigate the *same* maxima, a situation likely when all maxima are not of the same height.

Iteration

Running sub-populations in parallel, particularly when there is no communication between them, is tantamount to running a population several times in the hope that each, or at least some, of the runs find different peaks. Again, there is no reason why each run subsequent to the first should find a different peak, and, particularly in functions with uneven peaks, it would be expected that all runs would tend to find the same peak.

The problem of all runs searching the same area was addressed by Beasley (Beasley, D. et. al. 93) who used *derating functions*. Derating functions are used to modify the fitness function from run to run so that areas already searched will not be searched again, so knowledge attained in one run can be used in subsequent runs. The basic algorithm used by (Beasley, D. et. al. 93) is as follows:

1. Initialise: equate *modified fitness function* with raw fitness function.

2. Run the GA using the modified fitness function and keep a record of the best individual found in the run.

3. Update the modified fitness function to give a depression in the region near the best individual, producing a new modified fitness function.

4. If the raw fitness of the best individual exceeds the solution threshold, display this as a solution.

5. If all solutions have not been found, return to step 2.

Of course, the difficulty arises in knowing when all solutions have been found. Beasley (Beasley, D. et. al. 93), however, produced some impressive results using the same test bed of functions as this paper. These results are discussed further in Section 6.

Beasley measured the "region near the best individual" using a distance metric based on the decoded parameter space, which, as individuals decode to a real number, was trivial to measure.

2 RACES IN GENETIC ALGORITHMS - A BRIEF HISTORY

The use of races in genetic algorithms was originally used in multi-objective function optimization (Ryan 94a) (Ryan 94b). Multi-objective problems are problems in which two or more criteria must be satisfied in order for a solution to be considered satisfactory. Usually (Kinnear 93) (Koza 92) there is some weighting given to each criteria in the fitness function, which is typically taken to be the sum of each of the criteria. Such is the nature of genetic algorithms that in the case where one objective is less difficult than the other, individuals tend to concentrate on the easier one, ignoring the more difficult, and possibly more important, objective.

Pygmies and Civil Servants

This problem was addressed in (Ryan 94a) by applying a GA to the problem of evolving minimal network sorters, which must be capable of correctly sorting numbers, but also of doing so in as few exchanges as possible. Rather than try to balance shortness and efficiency in the fitness function, *two* fitness functions were used. One fitness function was biased toward efficient sorters, individuals that excelled at this function being described as Civil Servants, while the other fitness function was biased toward *short* sorters, described as Pygmies. Two lists of individuals were maintained, and, when creating a new individual, one parent was drawn from each list. The Civil Servants put pressure on new individuals to be efficient, while the presence of the Pygmies exerted pressure on individuals to be short. It was found that the two lists improved on the success rate of a standard GA by up to 100%, and by about 10-20% on the success rate of a GA with carefully selected weights.

Gender or Race?

The Pygmies and Civil Servants of the previous section are analogous to different genders, and, to all intents and purposes are considered morphologically different. This difference was due to their performance at each fitness function, and was not any inherent difference in their bit strings or functionality. The Pygmy Algorithm was extended to look upon each group as races, which could either inbreed or outbreed, and again, this improved the performance further.

Inbreeding allowed each group to concentrate on optimizing its own particular area, while outbreeding permitted the spreading of useful gene combinations throughout the entire population. Too much inbreeding, however, could prove disasterous as not only does inbreeding concentrate on optimizing its current area, it also reduces the likliehood of the GA searching other, perhaps better, areas.

The mechanism used for deciding when to inbreed and when to outbreed was a *meta-GA*, which is simply another GA running over the original GA. Each individual had a genetically inherited *Racial Preference Factor* (RPF), which is a measure of that individual's inclination to outbreed, and was examined when that individual was chosen for mating. Like any inherited material, an individual's RPF could be crossed over and the RPF genes evolved in parallel with the population, allowing automatic tuning of the GA parameters during a run.

The use of races in a GA can easily be modified to fit any multi-objective problem simply by including a fitness function for each objective, weighted toward that objective. In general, for n objectives, each fitness function is of the form:

$$FitnessFunction_x = \sum_{i=1}^{n} Weight_{xi}.Obj_i$$

Where $Weight_{xi}$ is the weighting given to the *unmodified* score in objective i by fitness function x. Typically, $Weight_{xx}$ serves to promote the objective associated with $FitnessFunction_x$, while the other weights usually either serve to normalise the score from each objective, or are set to 0, which would increase the emphasis placed on the main objective.

While it is possible to vary the weights, so each objective has a different weight in its own fitness function, it is difficult to decide exactly what each weight should be. For this reason, all the weights are usually left equal, with perhaps the weight associated with the primary objective being slightly greater.

At the end of a run, several solutions are produced, each one excelling at least one objective, ideally in most if not all. It is then simply a matter of selecting which one suits best.

Elitism and the preservation useful gene combinations

RGA, and the GAs used in its development, all use some degree of elitism. In the Pygmy Algorithm, both in the gender and racial model, the top performers for each gender or race were always maintained from one generation to another. So if too much inbreeding was taking place, and poorly performing individuals were appearing, then better performing individuals, that may have appeared much earlier in the run, are still maintained in the parent population.

3 RACES IN MULTIMODAL FUNCTIONS

Optimizing multimodal functions requites a somewhat different approach from multi-objective problems in that the most important goal is the discovery of a number of solutions, whereas multi-objective problems are more concerned with the balancing of a number of goals. However, if one could have fitness functions that would evaluate an individual's suitability for each peak, with one fitness function designed for each peak, then a population would be strongly encouraged to be spread evenly across the search space. Each fitness function would allow convergence on its respective peak. Unfortunately, such a situation is unlikely in the real world, as one simply doesn't know where the peaks are located in the search space, and indeed, how many peaks are present.

If one cannot use information about the peaks themselves, then the only other information available concerns the individuals. In function optimization, individuals are typically real numbers, so calculating the difference between them is a trivial matter. Indeed, it was by a method such as this that Goldberg decided if individuals were close enough that they would be sharing fitness.

By using fitness functions that reward individuals simply because they belong to a particular race, a population can be forced to search many points of the search space at a time.

Combining each of these fitness functions with the multimodal function to be optimized, the problem now effectively becomes a multiobjective one. As well as having to find a peak in the search space, an individual is also expected to belong to a race. Far from complicating the search however, this improves it vastly, as the races promote a wide variety of individuals that can concentrate on different parts of the search. The fitness function for each race is of the form:

$$FitnessFunction_x = Distance(x) + Weight.Function(individual)$$

Where $Distance(x)$ is a measure of how close an individual is to the racial perfect, $Weight$ is a weight given to an individual's performance in the function to be optimized, and $Function(individual)$ is the individual's score in the function.

The racial perfect is decided by dividing the search space into a number of equal parts, one for each race. The racial perfect is then an individual that matches the division for a particular race. The distance of other individuals from the racial perfect is calculated in the same way as Beasley's Sequential Niche Technique (Beasley, D. et. al. 93), which is simply the absolute difference between the decoded individual and the racial perfect.

If, for example, the search space was from 0..1, and it was divided into five races, the racial perfects as outlined in table 1 would appear. The distance of three decoded individuals is also shown to illustrate how simple it is to calculate how suitable an individual is to a race.

Table 1. Scoring individuals for their suitability to five different races. The closer the value of an individual to the racial perfect, the higher the score is.

Race	Racial Perfect	0.33	0.755	0.55
1	0	0.33	0.755	0.55
2	0.25	0.08	0.505	0.3
3	0.5	0.17	0.255	0.05
4	0.75	0.42	0.005	0.2
5	1	0.67	0.245	0.45

So the individual that decodes to 0.33 is most likely to join the second race having a racial perfect of 0.25. But, that same individual is also more likely to join the first or the third race than the individual that decodes to 0.755 by virtue of the fact that it has a smaller distance.

The weight given to the performance of the individual at function optimization is the same for each race. Like most GA parameters, there is no absolute rule for the setting of the weight, so a certain amount of trial and error is involved in the choosing of a weight that ensures the population does make some effort at optimizing the function.

Tracing the ancestry

The question arises now of how to decide to which race an individual belongs. Spears(Spears 94) explained his use of tags with the description of how one decides whether or not they are Portuguese. A person does not decide they are Portuguese simply because they resemble their relations, but because their ancestors were identified as being Portuguese, and they simply "culturally inherited" the label.

The RGA uses this reasoning to a certain degree, but does not adhere to it as strictly as Spears did. Again, using the example of a person trying to decide whether or not they are Portuguese, we can explain how it works. In the event of the person's two parents being Portuguese, there is little doubt that that person will also be Portuguese. Indeed, in Spears' model, mating was restricted to individuals with common tags, so this would always be the case. But, if a Portuguese man was to mate with, say, an Irish woman, what then would the offspring be, Irish or Portuguese? The simplest answer is a bit of both, although depending on the upbringing, the offspring may tend to consider themselves one or the other. Because RGA does not insist on inbreeding (for reasons which will be made clear in the following section) the situation of inter-racial mating is only to be expected.

To overcome the question of deciding which race an individual belongs to, let us return to question of nationality the half-Portuguese, half-Irish offspring who is trying to discover his true nationality. Let us assume that he travels to Portugal and every other Portuguese he meets accepts him as being Portuguese, if this is the case, then it is fair to say that he is Portuguese. He might also decide to visit Ireland, and every Irish person he meets accepts him as Irish. What does this mean? Is he Irish or Portuguese? The only answer can be is that he is both.

On the other hand, the tale might not end with such acceptance, and both Irish and Portuguese decide that he is not one of them, so he is not considered to be a member of either race.

This analogy could be taken much further. His Irish mother could have been the offspring of an Irishman and Englishwoman, so perhaps he might belong to the English race also. The moral of this story is that when races interbreed, there is no way to know which race the offspring will belong to. RGA overcomes this uncertainty by having an individual examine *each* race, and see if the other members of the race accept it as being a member. Because it is crucial to maintain the "cultural identity" of each race, a *Steady State*(Syswerda 89) approach is taken, whereby only one individual is created at a time, evaluated, and tested to see if it can get into the parent population.

The term "nationality" could also be used here, but "nationality" and "race" are considered to be the same in this context. The term "race" is used because it has more genetic connotations, and in RGA it is an individual's genes rather than geography that determines their race.

The implementation, then, of RGA is as follows:

1. Create and test individual.
2. Test to see which, if any, races the new individual can join.
3. If initial population(*Generation 0*) has been created go to step 4, otherwise, repeat steps 1 through 3.
4. Select race at random.
5. Select individual probabilistically from race.
6. Examine if individual wishes to inbreed or outbreed.
7. Select second individual from appropriate race probabilistically.
8. Create and test new individual.
9. Test to see which, if any, races the new individual can join.
10. If finishing criteria haven't been fulfilled, go to step 4.

Ideally, as many races would be created as there are peaks in the landscape, and each race would evolve towards one peak. But one cannot assume that enough information is available to do this, so the safest thing to do is to create at least twice as many races as one *thinks* there are peaks. "Thinks" is emphasised because the exact number of peaks does not matter to RGA, and creating extra races will not adversely affect performance.

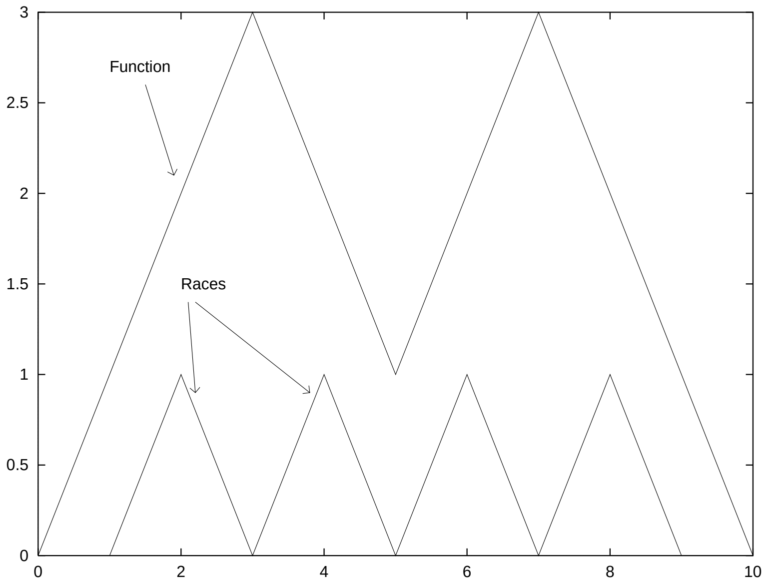

Figure 1. A case where four races are evolving toward two peaks. Neither peak coincides with a race, but both appear between two.

The benefits of inter-racial mating

Practically every other application of GAs to the problem of solving multi-modal functions has involved the use of restricted mating in some sense, usually restricting individuals to mating with individuals on the same peak in the landscape. But individuals occupying the same peak in the function landscape may be spread across several nearby races, as illustrated by figure 2, so this does not suit RGA.

The cost of inter-racial mating

All algorithms considered in this paper are reported upon in terms of their quality at locating and maintaining solutions, not in terms of computational cost. Using races is akin to steady state genetic algorithms (Syswerda 89) except that instead of having one sorted list of N individuals, one maintains M sorted lists of $\frac{N}{M}$ individuals. In this respect, one is attaining a large increase in performance, as outlined below in Section 6, at a cost marginally greater than that of using a steady state genetic algorithm.

In figure 1 there are many more races than peaks in the function, so several races will evolve towards each peak. Inter-racial mating is useful because otherwise each race would search in isolation for the peak, and may not contain sufficient information to converge on it. But, as will be demonstrated in section 6, as the difficulty of a problem increases, inter-racial mating becomes more important.

4 TEST FUNCTIONS

As stated in section 1, the tests introduced by Deb will be used to test RGA. Both Deb (Deb 89a) and Spears(Spears 94) considered how long their algorithms could maintain a stable population spread across each peak as a measure of success, while Beasley measured the probability that his algorithm would actually find all of the peaks. Due to the steady state nature of RGA, once all peaks are found, maintaining stable sub-populations on each is a trivial task, and peaks will never be lost. For this reason, the same measures used by Beasley will be employed. That is, *success rate*, which is the probability that all peaks of a particular function will be found, and *Root Mean Square error* (RMS). RMS is a measure of the accuracy of solutions found. The value for RMS is calculated by finding the distance between each solution found and the exact solution. This distance is then squared and the RMS value is the square root of the mean of these values.

All of the test functions were described by Deb to test his sharing function. In functions F1 to F4 a 30-bit binary chromosome was used to represent x in the range 0 to 1. In function F5 there are two parameters, x and y, which are represented by two 15-bit numbers in the range $-6 \leq x, y \leq +6$.

F1 - equal peaks

The first function has five equally spaced peaks of equal height:

$$F1(x) = sin^6(5\pi x)$$

The peaks are at x values of 0.1, 0.3, 0.5, 0.7 and 0.9.

F2 - decreasing peaks

Deb's second function has peaks at the same locations as F1, but with heights decreasing exponentially:

$$F2(x) = exp\left(-2log(2) * \left(\frac{x-0.1}{0.8}\right)^2\right) * sin^6(5\pi x)$$

In this case, the peak heights vary from 1.0 to .25.

F3 - uneven peaks

The third function described by Deb has peaks of equal values, but at uneven intervals:

$$F3(x) = sin^6(5\pi(x^{\frac{3}{4}} - 0.05)$$

The peaks are located at, roughly, values of x of 0.08, 0.246, 0.45, 0.681 and 0.934.

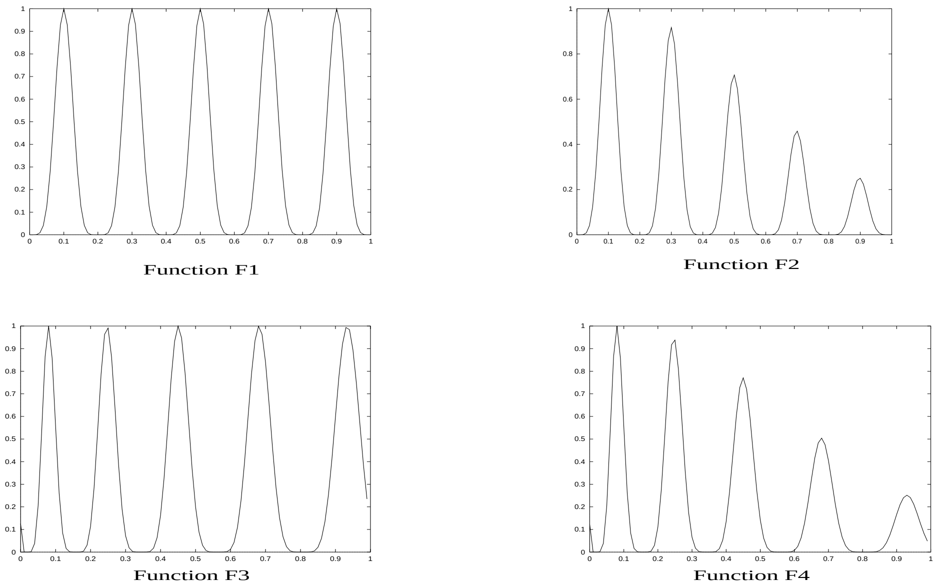

Figure 2. Functions F1 - F4 which provide a variety of solution landscapes

F4 - uneven, decreasing maxima

Deb's fourth function has peaks at the same locations as F3, but with the values of those in F2.

$$F4(x) = exp\left(-2log(2) * \left(\frac{x - 0.08}{0.854}\right)^2\right) * sin^6(5\pi x)$$

F5 - Himmelblau's function

This two-variable function was modified by Deb to be a maximization problem:

$$F5(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$$

There are four peaks, all around 200 in height, at approximately (3.58, -1.86), (3.0, 2.0), (-2.815, 3.125) and (-3.78,-3.28).

5 IMPLEMENTATION ISSUES

Functions F1-F4 were run with races located as shown in Figure 4. The locations of the races were decided simply by dividing the search space (0..1) into ten different divisions.

In no case did the location of any race coincide with the exact location of any peak. In the cases where all five peaks were located, they were distributed

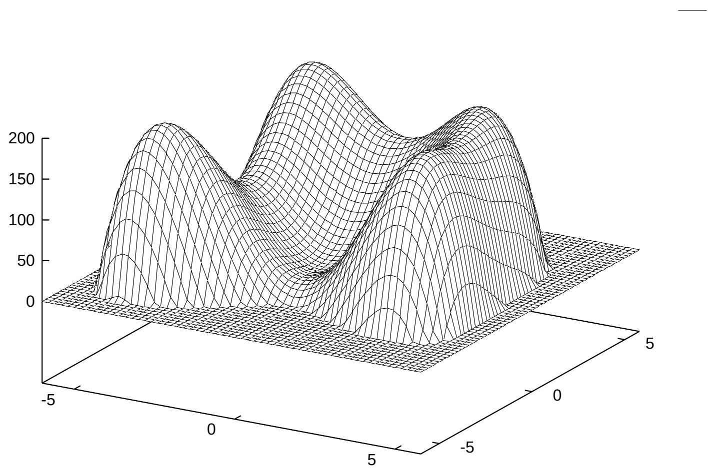

Figure 3. Modified Himmelblau's Function.

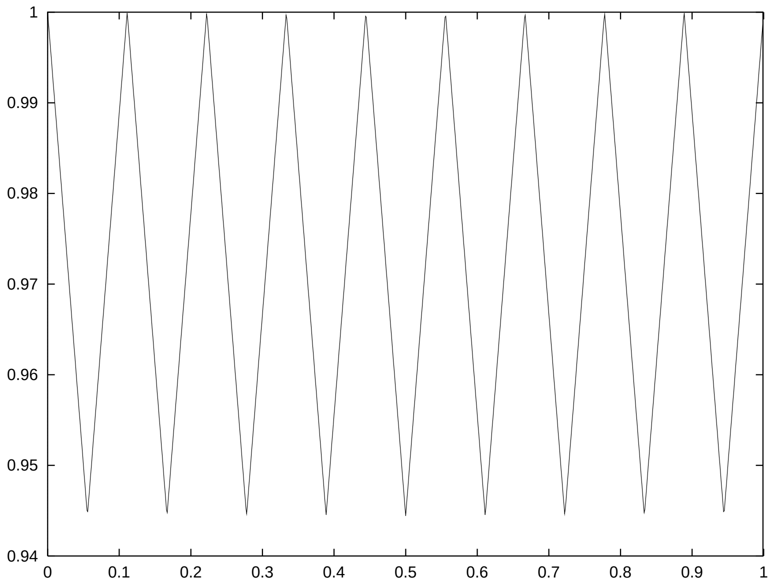

Figure 4. Location of racial perfects in solution landscape for functions F1 - F4.

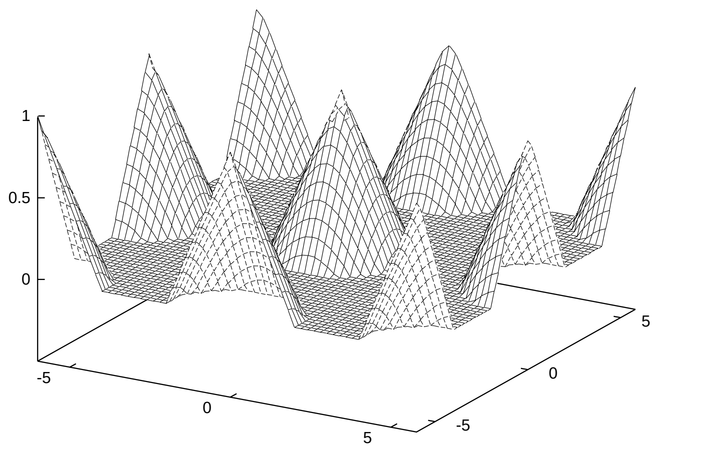

Figure 5. Nine races evenly divided for function F5.

evenly across the races, in the ratio 2:2:2:2:2. Occasionally one peak dominated three races, and, while some times the sole race left for the peak from which the first was taken was able to retain it's own cultural identity and evolve in its own direction, it was often the case that the remaining race was assimilated by another neighbouring race, resulting in a distribution along the lines of 2:2:3:0:3, and leading to the failure of that run to produce a solution.

If it were possible for the location of a race to change during a run the performance might be improved upon, this possibility is discussed further in section 8. At present, RGA concentrates only on static races.

Because function F5 is a two-variable function, the races had to be laid out in two dimensions. The first approach used nine races, just over twice the number of peaks, arranged evenly throughout the landscape as in figure 5. However, there was no overlap of races and many individuals were quite a distance from any race, and it was found that using 16 races, four times the number of peaks, yielded far superior results. Again, no race ever appeared at the same location as a peak. Those races which found peaks were simply near those peaks.

Inbreeding, outbreeding and the problem of clones

A meta-GA as described in section 2.2 was used to control the level of inbreeding taking place. For the sake of comparison, a variety of approaches were taken:

1. Control inbreeding through meta-GA.
2. Always inbreed.

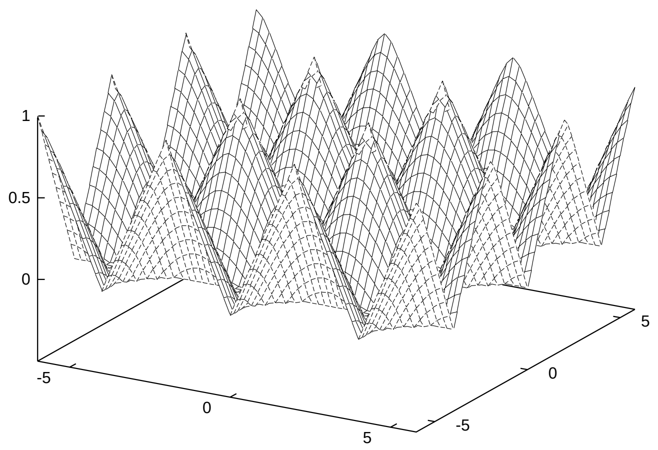

Figure 6. Sixteen races evenly divided for function F5.

3. Meta-GA controlled inbreeding with clones disallowed.
4. Always inbreed with clones disallowed.

Disallowing clones was simply allowing no clones *in the same race*. It is still possible for several clones of an individual to be present in the population by virtue of the fact that that individual is a member of several races. The effects of disallowing clones is described in the following section.

6 RESULTS

As mentioned in section 4, measuring how long RGA can maintain stable populations at several peaks is meaningless, as once discovered, a peak will not be lost. The first set of results compare the various implementations of RGA, while the second compare RGA with Beasley's (Beasley, D. et. al. 93) sequential niche technique. Beasley's results are used in the comparison because they are the most readily available. Beasley also demonstrated that the Sequential Niche Technique greatly outperformed a standard GA on the same problem set.

Table 2. Comparison of RGA results.

	Fixed	Evolve	Fixed, no clones	Evolve, no clones
F1	99%	97%	98%	98%
F2	100%	100%	100%	100%
F3	99%	98%	99%	99%
F4	100%	100%	100%	100%
F5	13%	77%	21.8%	90%

For functions F1-F4, the most striking aspect is how well RGA performed on F2 and F4. The reason RGA found those functions easier is because of the lower peaks. Ironically, it is for precisely this reason that other algorithms would find these functions the most difficult to optimize, as lower peaks are often deserted in favour of higher, more rewarding peaks.

Function F5 is perhaps the most illuminating of the five, as it is with this that not only is the advantage of evolving parameters shown, but also the advantage of disallowing clones is clear. The advantages of these are not so clear with the other functions for the reason that once a peak is located, optimizing it is much simpler when there is only one variable involved.

When clones are allowed, they tend to choke the exploratory power of a race. By forcing all members of a race to be different, inbreeding in that race can still improve the optimization of that peak. Varying the amount of inbreeding by evolution also improves performance. By breeding between races, cooperation is allowed between races evolving towards the same peak. Although in two cooperating races it is possible for clones to appear, there can only be one copy of each individual in each race, so the level of clones is kept acceptable.

In general then, the best method is to disallow clones, and allow a meta-GA to decide on the level of outbreeding. The results produced this way are now compared to Beasley's.

Table 3. Comparison of the Sequential Niche Technique and RGA.

	SNT succ.	RGA succ.	SNT RMS	RGA RMS
F1	99%	98%	.0043	.0028
F2	90%	100%	.0075	.0031
F3	100%	99%	.0039	.0023
F4	99%	100%	.0041	.0024
F5	76%	90%	.20	.10

For functions F1-F4, the performance of both algorithms is quite similar, although probability of success for the Sequential Niche Technique varies from 90% to 100%, while the variance for RGA is much more consistent, varying only from 98% to 100%. Another difference is that the RMS for RGA is much smaller, at least half that of the Sequential method. Again, the most interesting result is the most difficult function, F5, and the performance of RGA is much better in this case, both in success rate and RMS. It is significant also, that SNT *requires* knowledge of the number of peaks in the landscape, while no such knowledge is needed for RGA.

7 SUMMARY AND CONCLUSIONS

A new algorithm, *RGA*, has been presented and has been shown to be able to locate all maxima of a multimodal function with little or no prior knowledge of the number or nature of the maxima. RGA explores the fitness landscape through the use of different races of individuals, each of which has its own cultural identity. In the case where there are many more races than peaks in the landscapes, several races will have a similar identity, and, under control of a meta-GA, can crossbreed.

Previous methods, such as sharing, crowding or sequential niche techniques, all require prior knowledge of a problem. Sharing and crowding are concerned to a large degree with the preservation of knowledge, and are rated by how stable their populations are when spread across several peaks. RGA's populations are inherently stable and, once discovered, a peak will never be lost. The sequential niche technique is more concerned with the discovery of all peaks of a function, and RGA has been shown to outperform it at this.

This paper is very much an introduction to the use of races in multimodal functions, and there are several directions which RGA can go from this point. Deceptive functions are somewhat similar to multimodal functions in that they have two or more peaks in the fitness landscape, and RGA would seem to be ideally suited for finding all of these peaks, and thus identifying the true maximum.

While the use of crossbreeding has been shown to improve the performance of RGA considerably, particularly in the more difficult function, more work remains to be done. Currently, crossbreeding individuals can breed with any other race, but if they were restricted to crossbreeding only with individuals of a similar cultural identity - individuals on the same peak, it is reasonable to expect another increase in performance.

In functions with an unfeasibly large search space, it could well be possible for races to move about the landscape, searching for a new cultural identity when it becomes clear that their current one is not particularly rewarding.

For problems in which the test cases vary during a run, the use of races could well prove useful in remembering solutions to test cases that were presented early in a run, with different races concentrating on different test cases during a run.

Acknowledgements

Thanks to Dermot Ryan and Maura O'Halloran for their comments on this paper. Thanks also to Dr. Zbignew Michalewicz for presenting this paper for me.

References

Beaslery et. al (1993). A Sequential Niche Technique for Multimodal Function Optimization. *Evolutionary Computation 2:101-125.*

Deb, K. (1989). *Genetic Algorithms in multimodal function optimization.* Masters thesis. The University of Alabama, Dept of Engineering Mathematics.

Deb, K. & Goldberg D.E. (1989). An investigation of niche and species formation. In *Proceedings of the Third International Conference on Genetic Algorithms*, ed. J.D. Schaffer, 42-50. San Mateo, CA: Morgan Kaufmann.

De Jong, K. (1975). *An analysis of the behaviour of a class of genetic adaptive systems.* Doctoral Dissertation. University of Michigan.

Goldberg, D.E. & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms* ed. J.J. Greffenstette, 41-49. NJ: Lawrence Erlbaum Associates.

Goldberg, D.E. et. al (1992). Massive multimodality, deception, and genetic algorithms. *Illigal technical report No. 92005.*

Hillis, D. (1992). Co-evolving Parasites Improves Simulated Evolution as an Optimization Procedure. In *Artificial Life II.* Ed. C.G. Langton, 228-234. Santa Fe: Addison-Wesley.

Kinnear, K. (1993) : Generality and difficulty in GP : Evolving a sort. In *Proceedings of the Fifth International Conference on Genetic Algorithms.* ed. S. Forrest. San Mateo, CA: Morgan Kauffman.

Koza, J. (1992) : *Genetic Programming.* Cambridge : M.I.T. Press.

Mahfoud, S.W. (1993). Simple Analytical Models of Genetic Algorithms for Multimodal Function Optimization. *IlliGAL Report No. 93001.*

Ryan, C (1994). Pygmies and Civil Servants. In *Advances in Genetic Programming.* ed. K. Kinnear. Massachusetts : MIT Press.

Ryan, C (1994). Racial Harmony in Genetic Algorithms. In *Proceedings of the KI94 Workshop on Evolutionary Algorithms.* ed. J. Hopf. Saarbrucken : Springer-Verlag. *To appear*

Ryan, C (1994). The Degree of Oneness. In *Proceedings of the ECAI94 Workshop on Genetic Algorithms.* ed. A.E. Eiben. Amsterdam : Springer-Verlag. *To appear*

Siegel, E. (1994). Competitively evolving decision trees against fixed training cases for natural language processing. *Advances in Genetic Programming.* ed. K. Kinnear. Massachusetts : MIT Press.

Spears, W. M. (1994). Simple subpopulation schemes. In *Proceedings of the Third Annual Conference on Evolutionary Programming,* eds. A.V. Sebald and L.J. Fogel, River Edge, NJ: World Scientific, 296-306.

Syswerda G. (1989). Uniform Crossover in Genetic Algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms,* ed. J.D. Schaffer, . San Mateo, CA: Morgan Kaufmann.