Finding Multimodal Solutions Using Restricted Tournament Selection

Georges Harik
The Illinois Genetic Algorithms Laboratory (IlliGAL)
University of Illinois at Urbana-Champaign
117 Transportation Building, 104 South Mathews Avenue
Urbana, Illinois 61801-2996 USA
harikg@gal7.ge.uiuc.edu

Abstract

This paper investigates a new technique for the solving of multimodal problems using genetic algorithms (GAs). The proposed technique, Restricted Tournament Selection, is based on the paradigm of local competition. The paper begins by discussing some of the drawbacks of using current multimodal techniques. The paper then presents the new technique along with an analysis of a class of sets of solutions it preserves and locates. This presentation researches the new technique's restriction on competition from the viewpoint of calculating probability distributions for its tournaments as well as its various niche takeover times. Empirical observations are then presented as evidence of the technique's abilities in a wide variety of settings. Finally, this paper explores the future trajectory of multimodal GA research.

Finding Multimodal Solutions Using Restricted Tournament Selection

Abstract

This paper investigates a new technique for the solving of multimodal problems using genetic algorithms (GAs). The proposed technique, Restricted Tournament Selection, is based on the paradigm of local competition. The paper begins by discussing some of the drawbacks of using current multimodal techniques. The paper then presents the new technique along with an analysis of a class of sets of solutions it preserves and locates. This presentation researches the new technique's restriction on competition from the viewpoint of calculating probability distributions for its tournaments as well as its various niche takeover times. Empirical observations are then presented as evidence of the technique's abilities in a wide variety of settings. Finally, this paper explores the future trajectory of multimodal GA research.

1 Introduction

There is, in the realm of optimization, quite often a need to identify several good solutions for a problem, as opposed to one optimal solution. This need arises from several sources. Often, an optimization problem is a simplification of a real world problem that in part requires human or inquantifiable judgment. In these types of problems, an optimizer needs to suggest various possible alternatives that can later be judged by a human expert. In other situations, a better understanding of a search space is desired in terms of the location of its various optima.

Genetic Algorithms (GAs) can be and are being used as general optimization techniques for a number of problems in various domains. They often produce good solutions to these problems quickly, while requiring little or no problem specific information. GAs are thus useful in domains that are not well understood, as well as domains in which the use of a complete model of the underlying problem is computationally infeasible. A GA capable of finding multiple solutions in real world optimization problems would thus be a valuable asset.

The purpose of this document is to add a tool to the repertoire of those seeking to solve these sorts of problems. We will investigate this tool, Restricted Tournament Selection or RTS, as to its merits as a multimodal GA technique as well as for its ability to overcome some of the problems faced by other multimodal techniques.

2 Background

This presentation assumes that the reader is familiar with the basic genetic algorithm (GA). Other readers may want to refer to Goldberg (1989) for an introduction to GAs before proceeding with the reading. The summary we will view below is a brief introduction to the multimodal aspects of solving problems with GAs.

2.1 Multimodal Genetic Algorithms

An even cursory consideration of all the techniques available for multimodal GA optimization is beyond the reach of this document. The reader is referred to Harik (1994) for a review of many such techniques. We will briefly review the more thoroughly proven techniques in this area, while paying particular attention to fitness-sharing (Goldberg & Richardson, 1987).

Two general techniques and their descendants form the basis for current multimodal GA optimization. These are the techniques of crowding (De Jong, 1975) and fitness-sharing Both techniques define a distance metric over the search space and use that metric to preserve multiple solutions in the GA's population. Crowding uses that metric to force individuals newly entering a population to replace similar individuals. Fitness-sharing uses the metric to force individuals in a population to share their fitness assignments with neighboring individuals.

The crowding technique and its descendants were analyzed by Mahfoud (1992). The basic crowding technique was shown to be ineffective in multimodal optimization. We should note however that it was designed only for the less demanding task of diversity-preservation at which it succeeds. That study also introduces a variant of crowding, called deterministic crowding. This variant proved more capable than the original of multimodal optimization. However, it failed to maintain certain sought out optima when those could recombine to form more fit optima in the search space. This failure, along with the difficulty of interpreting the set of solutions returned by this algorithm represent the current difficulty of the most advanced of the crowding methods.

A study by Goldberg et. al. (1992) highlighted many of the strengths and weaknesses of the fitness-sharing algorithm. In that study, the fitness-sharing algorithm was used successfully to extract 32 global solutions from a problem with over 1 million local deceptive optima. This study also exposed the difficulty encountered in setting this algorithm's parameters. To successfully solve the above problem, the authors had to assign the distance over which the fitness sharing was to operate as well as a level of exponential scaling to assign to the fitness function. Heuristics developed in earlier studies (Deb & Goldberg, 1989) proved inadequate for the choice of the distancing parameter. The proximity of many deceptive solutions to each global solution, which forced the exponential fitness score scaling, underscored the algorithm's dependence on raw fitness scores to determine its steady-state. Both parameters thus had to be set and reset through a period of empirical observation. Better guidelines to setting fitness-sharing's parameters are still obviously required.

A variation on this algorithm was proposed by Beasly et. al. (1993). In the variant, called the sequential niche technique, multiple solutions to a problem are found serially. When a particular solution has been found, a derating function is applied to its neighboring solutions, preventing the GA from searching again that part of the space and forcing it thus to locate other solutions. This algorithm also faces the same distancing parameter problem observed with fitness-sharing. However, this algorithm overcomes the problem of having to exponentially scale the its fitness function. This algorithm forbids the transfer of building block information from the finding of one solution to the next. In situations where this kind of transfer is advantageous, this property will hinder down the GA's powerful search engine. The massively multimodal problem solved by fitness sharing is one such example because it repeatedly uses a small set of building blocks to define all its global solutions. This technique needs to be further investigated for its effects on search before becoming a viable option in multimodal GA optimization. We proceed now to the introduction of the new multimodal technique.

3 Restricted Tournament Selection

Tournament selection is, as the name suggests, one way to select the collection of individuals out of one generation that are to survive to the next and reproduce. In regular binary tournament selection, tournaments are held between pairs of individuals chosen at random from the population. The winners of the tournaments then get to move on to the next generation. This makes perfect sense when solving unimodal problems. The winners of the tournaments are deemed to be the better or more fit individuals and they get to move on to be exploited more by the genetic algorithm. However, this selection method tends to make less sense if the algorithm wishes to preserve and find multiple solutions or needs to take advantage of the schema found in multiple local solutions in order to reach a particular global solution.

The inherent problem in the multimodal situation for tournament selection can be viewed in two ways. Algorithmically, the takeover time for tournament selection is logarithmic in the population size (Goldberg, 1991). Therefore, most all the niches in a population will tend to disappear relatively rapidly. Another way to view this problem is that tournament selection compares apples and oranges. For a selection scheme to maintain two relatively different solutions, it would have to create niches for both solutions so they could peacefully coexist. This implies that it should prevent members of one niche from competing with members of another niche. Yet that competition which should be forbidden is precisely what is occurring in regular tournament selection.

We will investigate a modification to tournament selection based on the concept of direct local competition. This modification works as follows: two elements A and B are selected at random from the population. These elements are crossed and mutated to form two new elements, A' and B'. A' and B' are then to be placed into the population as in a steady state GA. For each of A' and B', w (windowsize) more members of the population are scanned and the closest among the group to A' or B' is saved for further processing. Let the two elements found be called A'' and B''. A' then competes with A'' for a spot in the population and a similar competition is held between B' and B''. This form of tournament should restrict an entering element from competing with others too different from it.

Like other multimodal GA techniques this method requires the imposition of a distance metric over the GA's search space. As particularly high-fitted niches overtake the GA's population, this method should force more and more competitions for spots to be held between members of the same niche, thus allowing other niches to flourish at the same time. The next few sections explore exactly what we mean by 'niche' in this context as well as some of the theoretical and empirical properties of this algorithm.

3.1 A large class of sets preserved by RTS

To investigate the meaning that can be assigned to solutions returned by this algorithm, we will try to characterize a large class of sets of solutions that is well-preserved by RTS. These states will be very resistant to change under RTS and thus can be considered pseudo-absorbing states for the algorithm.

More formally, let Z be the search space in question. Let $d : Z \times Z \rightarrow R$ be a distance metric over Z. Let $S = \{S_0..S_N\}$ be a set of solutions in $P(Z)$[1]. S is defined to be an *optimal set* of solutions under d if $\forall x$ in Z $\exists i : \forall j\ d(x, S_i) \leq d(x, S_j) \Rightarrow f(x) \leq f(S_i)$. In short, the fitness value of any other solution has to be less than the fitness value of the solution in the optimal set it is closest to in the given distance metric. Thus each of the solutions in S is forced to be a local optimum. To see this for S_0, let d_0 be the minimum distance from S_0 to any other solution in S. Any element within $B(S_0, d_0/2)$[2] has to have a fitness less than S_0. Thus S_0 is a locally optimal point in the space. Without a loss of generality so are all the other points in S.

This concept facilitates the definition of niches. A *niche* in a GA's population can be defined as the part of the search space that is nearest to any particular one element of the GA's current optimal set. The various niches embodied in a GA's population thus cooperate to form a covering of the entire search space.

We will now see how optimal sets are very resistant to change under RTS. Let's consider a situation where a GA has found and placed an equal number of points at a number of peaks throughout the search space that form an optimal set. Let the number of peaks found be s and the window size be w. Let $w > s : w = cs, c > 1$. When a new individual is formed, to enter the population, it has to compete with the closest individual to it out of a set of w elements chosen at random from the population. Assume that this individual, call it A, is closest to the solution S_0 in the optimal set. By the definition of an optimal set then, the fitness of A is less that that of S_0. Thus if a copy of S_0 is present among the w individuals picked in the window, A will be forced to compete against S_0 and A will not enter into the new population.

We have from our basic assumptions:

S_0 occupies a proportion equal to $1/s$ of the population
P(a copy of S_0 present among w elements without replacement) $>$
P(a copy of S_0 present among w elements with replacement) $=$

[1] The power set of Z
[2] An open ball centered at S_0 of radius $d_0/2$.

1 - P(no copies of S_0 present among w elements without replacement) =
$1-(1/s)^w = 1-(1/s)^{cs} > 1-e^{-c}$ (since $e^{-c} > (1-1/s)^{cs}$)

Therefore, the probability of replacing some element out of this population with A is very small. In fact exponentially decreases as the window size increases as a multiple of the number of peaks to be maintained. If the window size is a large enough multiple of the number of solutions to be preserved, then the error should be sufficiently small that the underlying probability distribution of the population should not change for a very long time. Optimal sets thus form the pseudo-absorbing class of sets we were looking for with relation to RTS.

3.2 How Are Competitions Restricted Using RTS?

To understand how RTS restricts tournaments in a GA we define a function F on pairs of elements from the population. Let $F(x,y)$ be the proportion of the population that is phenotypically closer to x than y[3]. When a new individual, call it A, is created using genetic operators, it has to enter a competition before being allowed into the current population. Thus, it will have to compete with some individual chosen from the population. We can view $F(A$, chosen element$)$ as a random variable X defined over $[0,1]$. To extract the important elements of this analysis, we assume that the population is large enough that X can be viewed as a continuous random variable over [0,1]. We then wish to characterize X's probability density function. Regular binary tournament selection randomly picks an element's competitor from the population at hand. Thus the p.d.f.[4] of F using regular tournament selection is uniform over [0,1]. We now wish to investigate the probability density functions of X under RTS with different values of the window size w.

To pick a competitor for the new element A, RTS randomly selects w elements from the population and picks the element closest to A. For each of the w picked elements, we define a random variable X_i as $F(A$,ith element picked$)$. We note that as each element is picked randomly, the pdf of X_i is thus a constant and equal to 1 over $[0,1]$. Of these, RTS then chooses the element closest to A. This corresponds to the element with the smallest r.v. X_i since the r.v. X_i equals the proportion of the population that is closer to A than the ith element picked. This is seen to be the first order minimum statistic on w elements from a uniform distribution on $[0,1]$. That random variable's probability density function can be calculated as follows :

First look at X's distribution function, let j be : $0 < j < 1$
$P(X \leq j) =$ P(at least one of $X_w \leq j$) = 1 - P(All of the $X_w > j$) $= 1-(1-j)^w$
Therefore the p.d.f. of X is equal to $dP(X \leq j)/dj = w(1-j)^{w-1}$
Verifying this is indeed a p.d.f $\int_0^1 w(1-j)^{w-1} = |_0^1 -(1-j)^w = 1$

Qualitatively, we can thus say that RTS changes the GA's underlying joint distribution of elements picked together for competition. Quantitatively, by observing the form of the density function itself, we can say even more. We can calculate the probability that an individual A will have chosen for it for competition an element that is within the $1/w$th closest elements of the population to it. This is simply equal to $1-(1-1/w)^w$. This number is bounded from below by $1-e^{-1}$. Thus at least a constant proportion equal to $1-e^{-1}$ of the mates picked for competition with A will be proportionally within the closest $1/w$th of the population to A. Similarly, $1-e^{-c}$ of the comparisons will be within the closest c/wth of the population to A. From this analysis, we can conclude that this simple windowing mechanism efficiently limits comparisons to nearby members of the population. Additionally, this analysis combined with that of the immediately preceding section, suggests a strong and direct correlation between the window size parameter w, and the number of niches RTS will maintain in the population. Thus we are assured that by having the window size be some constant multiple of the number of niches sought out most of the

[3] Note that this definition is not symmetric because we may be working in a multidimensional space.
[4] Probability density function

competitions that occur will be within the niches themselves. This represents a heuristic for setting RTS' w parameter based on the number of different solutions sought out.

3.3 Analysis Of RTS Absorption Times

We can now further study how a persistently dominant individual begins to take over a population under RTS. Of particular interest is the time for an individual to take over a proportion equal to $1/(w+1)$ of the population as well as the time it takes an individual to take over one half of the population. In order to compare these times to takeover times of generational binary tournament selection, we adopt a generational view of RTS. A generation under RTS is defined to be the time needed to complete N tournaments where N is the population size in question.

We would like for RTS to exhibit takeover time equivalent to that of tournament selection when working within niches. This will ensure that a GA using RTS maintains proper selective pressure when no individual threatens the takeover of the entire population. Thus in a population of size N, we would like the first calculated takeover time to be on the order of $log(N/w+1)$ generations (Goldberg & Deb, 1991). We would also like for RTS to delay for an inordinately long period of time an individual trying to take over a large part, say one half, of the search space. This corresponds to the multimodal capabilities of the algorithm. Thus we would like the second calculated number to grow very quickly in terms of w.

To study these times we determine the probability that the dominant individual in the population receives one more copy after one steady state competition. Let N be the size of our population and let C be the number of copies that a dominant individual has in the population at a particular time.

P (dominant member receives one more copy) =
P(dominant picked initially) * P (no copies of dominant in w choices | dominant picked)

P (dominant picked initially) = C/N
We use a binomial distribution[5] with w trials and a success rate $(N-C)/N$
P (no copies of dominant in w choices | dominant member picked initially) = $((N-C)/N)^w \Rightarrow$
P (dominant receives one more copy) = $(C/N)((N-C)/N)^w$

Thus the expected number of reproduction attempts before the dominant element gets one more member is $1/P$(dominant gets one more copy) which equals $(N/C)(N/N-C)^w$.

Now we can calculate the expected time before a dominant individual takes over a proportion equal to 1/(w+1) of the population. This is simply equal to $\sum_{I=1}^{I=N/(w+1)}(N/I)(N/(N-I))^w$ which equals $N^{w+1}\sum_{I=1}^{I=N/(w+1)} 1/I(N-I)^w$. We now note that over the range of the summation $1/I(N-I)^w$ is monotonic. We do this by calculating its derivative and setting it to 0:

$$(N-I)^w + i(-w(N-I)^w - 1) = 0 \Rightarrow$$
$$(N-I) - Iw = 0 \Rightarrow I = N/(w+1)$$

Thus the derivative changes once only at $I = N/(w+1)$ and we can approximate the sum accurately by looking at $N^{w+1}\int_{I=1}^{I=N/(w+1)} 1/I(N-I)^w \, dI$. We see that over the range of the integration $N^w/e < (N-I)^w < N^w$. Thus we can bound the integral expression from above by $N * e\int_{I=1}^{I=N/(w+1)} 1/I \, dI$ which equals[6] $e * Nlog(N/(w+1))$. This gives us niche takeover time of the same order as $Nlog(N/(w+1))$ reproduction steps or $log(N/(w+1))$ generations. This implies that before taking over a proportion equal to $1/w+1$ of the population, the best individual receives little resistance from RTS. Thus RTS still allows

[5]This is an approximation that becomes more accurate as the population size get larger but is quite accurate in any case

[6]This factor of e in front if a rough estimate and can probably be reduced to something closer to 1. The intent of the author was to show that this factor was bounded by a constant number.

for the exploitation of information early on in the run in a manner similar to tournament selection with a tournament size of 2.

Now using the same technique we can show how RTS slows down the takeover of a fixed proportion of the population for a variable w. Let's look at the time it takes for an individual to take over one half of the population. As above, this is approximately[7] $N^{w+1}\int_{I=1}^{I=N/2} 1/I(N-I)^w \, dI$. This integral can be evaluated by partial fraction expansion and the whole expression equals

$$NlogI - Nlog(N-I) + N^{w+1}\sum_{K=1}^{K=w-1} 1/(KN^{w-k}(N-I)^k)$$

At I=N/2 the first two terms vanish and this evaluates to
$$N^{w+1}\sum_{K=1}^{K=w-1} 1/(KN^{w-k}(N/2)^k) = N^{w+1}\sum_{K=1}^{K=w-1} 2^K/(KN^w) \geq N2^w/w$$

At I=1 the first term vanishes and this approximately[8] evaluates to
$$-NlogN + N^{w+1}\sum_{K=1}^{K=w-1} 1/(KN^{w-k}(N-1)^k) \approx -NlogN + N\sum_{K=1}^{K=w-1} 1/K \approx Nlogw - NlogN$$

Thus the convergence time is greater than $N(2^w/w + log(N/w))$ The number of generations needed to take over half of the population is then greater then $2^w/w + log(N/w)$. This calculation shows that increments in w radically affect how quickly an individual can take over a large part of the population. For example with $w = 10$, even with a small population of about 100, it would take on average more than 100 generations for an individual to take over half the population. Thus RTS provides the GA with much time to do its search before allowing a specific part of the search space to dominate all the other parts of the space.

3.4 The Effects Of RTS On Search

Frequently in the analysis of niching algorithms, the effect of an algorithm on the GA's search takes a back seat to the analysis of the algorithm's stability. However, GAs are primarily search mechanisms and proper steps must be taken to insure that whatever mechanism is added to a GA to allow it to solve multimodal problems does not impede the GA's search strategy. What we have done above is to try to characterize the effects of RTS during a GA's run. We see indications that in the beginning of a run, it allows rapid exploitation of schema information. When an individual or a niche becomes prominent in the population, RTS then slows down its expansion to the point where other niches are allowed coexist with its niche.

RTS has other subtler effects on a GA's search strategy. A regular GA quickly converges to one solution and begins narrowing its search area early on in the run. Since RTS preserves multiple solutions, it will still allow crossovers between different types of solutions well into the run. It is not entirely clear whether this is desirable or not. In view of the assumption that a GA works to form new solutions using building blocks from other solutions it would seem that allowing different solutions to cross would be a good idea. However, lethals can also result from the crossover of different types of solutions. This area bears further investigation both individually in terms of customized GA applications and in general in terms of characterizing approaches to limited forms of mating restrictions which might be needed later on in a run.

4 Empirical Testing of RTS

The above analysis indicated that RTS might be a useful tool in solving multimodal problems. It then remained to test RTS on a variety of problems that have been suggested before in the literature. RTS was tested on the 5 functions used by Deb (1989) in his masters thesis. Additionally, RTS was evaluated on

[7]Since the derivative of $1/I(N-I)^w$ changes only once, we are not missing much in the approximation by using the integral.
[8]Approximating N-1 by N.

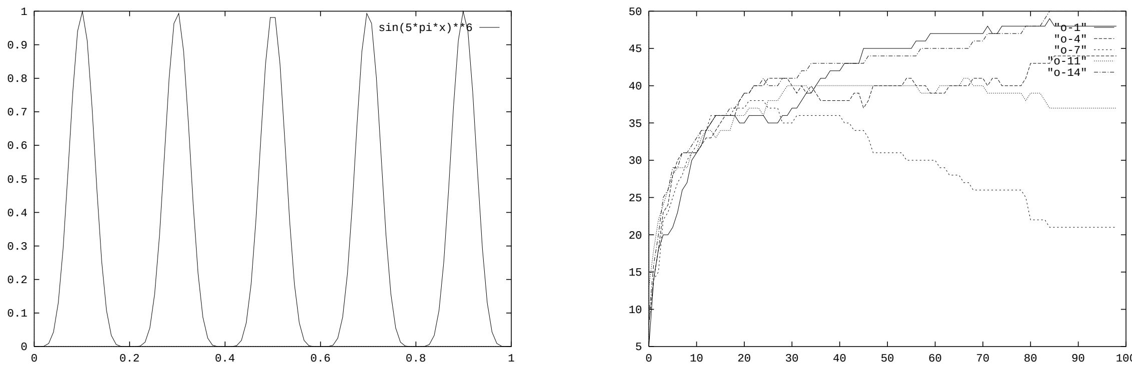

Figure 1: Function 1, left, and RTS working on Function 1, right.

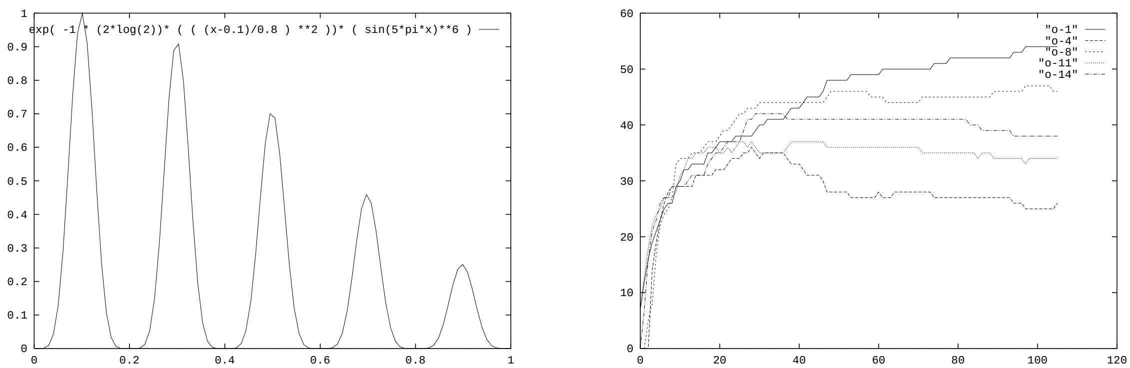

Figure 2: Function 2, left, and RTS working on Function 2, right.

the massively multimodal problem (Goldberg et. al., 1992). Following are the function descriptions and the results of using RTS on these problems.

4.1 Parameter Setting And The Reporting Of Results

In each of the problems attempted the window size was set at four times the number of peaks to be found. The reported results show the number of individuals at each peak as a function of the number of generations the GA has run. An individual is considered to be resident at a peak if its fitness exceeds 99% of the maximum fitness at that peak. In the first five problems a population size of 200 was used. For real variables that were coded as bits, string lengths of 15 were used. A crossover rate of 0.4 was used in all the runs. In each of the multimodal problems attempted, RTS was able to locate and maintain all of the peaks of the problem. Thus in each problem, we show the number of copies of individuals at all the peaks and how they are maintained for 100 generations.

4.2 Function 1

This is a function defined on [0,1] with five evenly spaced peaks of equal height. The function is defined by f(x) = $sin^6(5\pi x)$. The five peaks occur at x = 0.1, 0.3, 0.5, 0.7, 0.9 and all have a height of 1.0.

4.3 Function 2

This function is also defined on [0,1]. It has five evenly spaced peaks of unequal height. It is defined by f(x) = $e^{-2ln2((x-0.1)/0.8)^2} sin^6(5\pi x)$. The peaks occur at the same locations as Function 1 but have respective heights of approximately 1.0, 0.917, 0.707, 0.458 and 0.250.

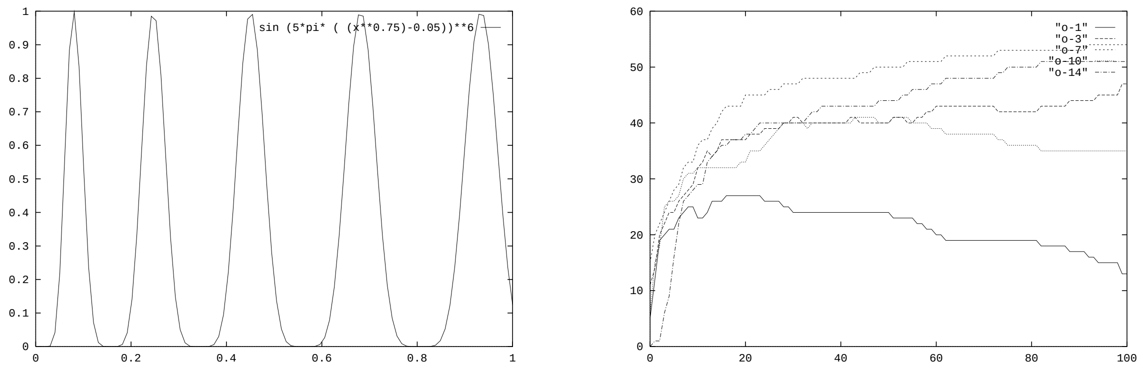

Figure 3: Function 3, left and RTS working on Function 3, right.

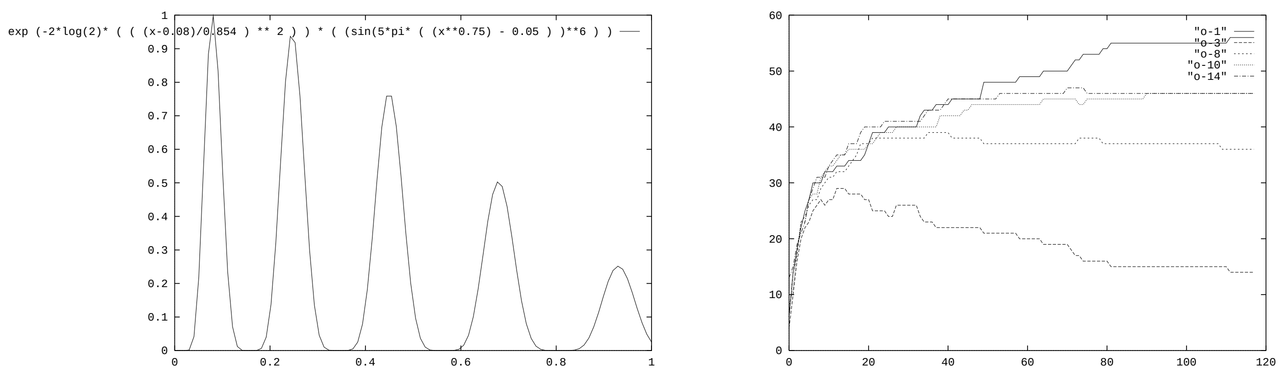

Figure 4: Function 4, left and RTS working on Function 4, right

4.4 Function 3

This function is also defined on [0,1]. Its five peaks are unevenly spaced over [0,1] but are of equal height. It is defined by f(x) = $sin^6(5\pi(x^{3/4} - 0.05))$. The five peaks each with a height of 1.0 occur at x = 0.246, 0.450, 0.681 and 0.934.

4.5 Function 4

This function defined on [0,1] has five peaks of unequal size unevenly spaced over [0,1]. It is defined by f(x) = $e^{-2ln2((x-0.1)/0.8)^2} sin^6(5\pi(x^{3/4} - 0.05))$.

4.6 Function 5

This is the two-dimensional function that is a variant of Himmelbau's function (Reklaitis et. al.,1983). Himmelbau's function is defined over [-6,6]×[-6,6] as f(x,y) = $(x^2 + y - 11)^2 + (x + y^2 - 7)^2$. This is a minimization problem. The function values are multiplied by -1 to convert it into a maximization problem[9].

4.7 Function 6

This is the massively multimodal problem posed by Goldberg et. al. (1992). It is based on 6 bit subfunctions of unitation[10]. In this case, 6 bit bimodal deceptive functions are used and 5 of them are concatenated to form a 30 bit problem. This function has over a million local optima and 32 global optima. The popu-

[9] This is simply an implementation requirement and since tournament selection is used the actual constant of scaling does not matter.

[10] Where the function value depends only on the number of 1s in a substring

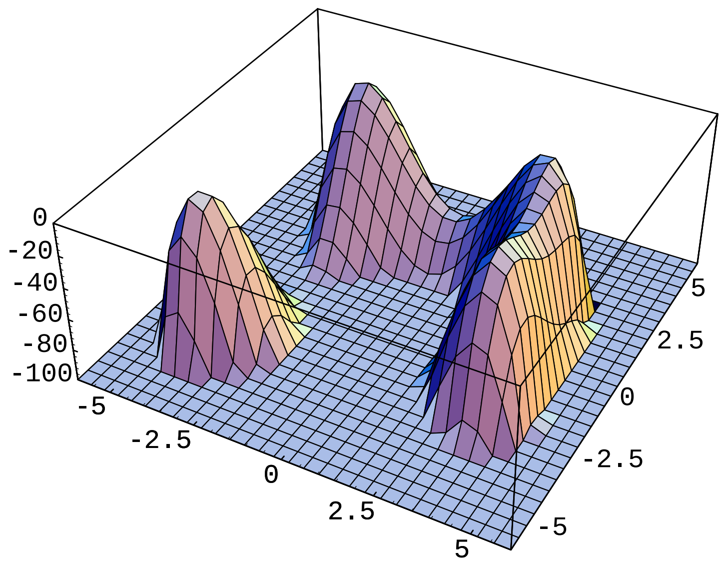

Figure 5: Function 5

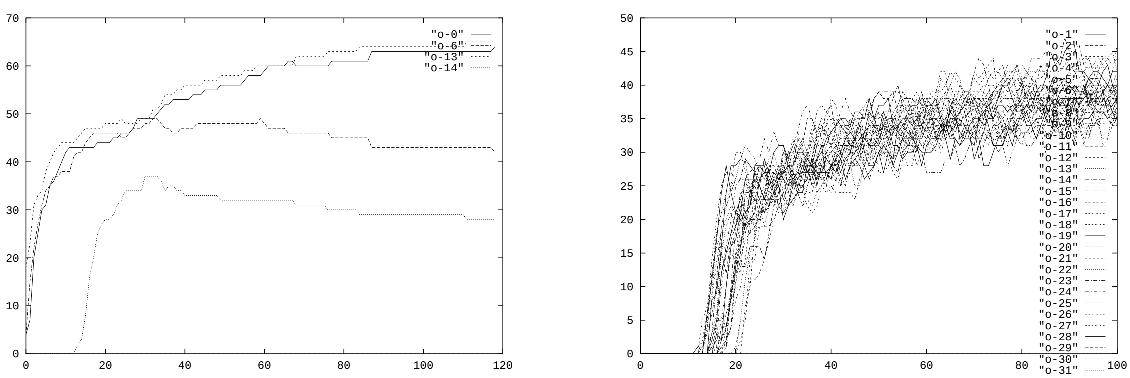

Figure 6: RTS on Functions 5, left and 6, right

lation size used here was 1280 and the window size used was 32. An individual was considered a solution only if it was one of the 32 global optima[11].

5 Summary

This document has introduced a new technique, RTS, for multimodal problem solving using GAs. It has described a general class of sets, *optimal sets*, that this algorithm attempts to return to the user as well as an interpretation of RTS' returned solutions. It has presented theoretical analysis indicating that this algorithm's population takeover properties are favorable. It has analyzed the restriction on competition that the GA is forced to undergo by this technique and thus suggested a heuristic for choosing w, RTS' only parameter. Finally, it has presented empirical results that verify the intuition derived from the above analysis that this algorithm *works*[12] both effectively and efficiently.

[11] With fitness of 50.0.

[12] for lack of a better word!

6 Conclusion

The class of worthwhile techniques and that of practical algorithms are not wholly identical. A technique can be useful beyond its immediate ability to provide concrete solutions to problems by suggesting directions for further research. On the other hand, practical algorithms are judged by the much harsher measures of speed, efficiency, efficacy and ease of use. As of now, the set of the better multimodal techniques available would have to be described as worthwhile but not completely practical. As practical algorithms, most lack one or more desirable features. For example, of the most useful techniques the following questions come immediately to the practitioner's mind: fitness sharing - "How do I set it's many parameters?"; deterministic crowding - "Just what is it returning?"; sequential niching "Could I have done better by keeping the previous solutions and building blocks around?"; RTS - "What if I get the wrong optimal set?"

Two basic desireables are unavailable with all current multimodal GA techniques. First, there is not general procedure or algorithm defined for assigning distance metrics to search problems. Second, there is to date little work in exploring the effects that these algorithms have on the GA's search capability. The first of these problems may actually have no solution independent of problem type. On the other hand, the second of these problems seems assailable but has not yet been tackled. There is to date, not even a simple model of what multimodal problems suitable for GA optimization look like. On the other hand, models of simple GAs based on the building block hypothesis have been around for a long time and have guided much of the development of GA research. This model has to be extended into the multimodal setting. Before this is undertaken, any exploration into the comparison of these various multimodal techniques as practical algorithms would be premature.

References

[1] K. A. De Jong. An analysis of the behavior of a class of genetic adaptive systems. *Dissertation Abstracts International*, 36(10):5140B, 1975. (University Microfilms No. 76-9381).

[2] K. Deb. Genetic algorithms in multimodal function optimization. TCGA report no. 89002 and master's thesis, University of Alabama, The Clearinghouse for Genetic Algorithms, Tuscaloosa, 1989.

[3] D. E. Goldberg and C. L. Bridges. An analysis of a reordering operator on a GA-hard problem. *Biological Cybernetics*, 62:397–405, 1990. (Also TCGA Report No. 88005).

[4] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. 1991. (Also TCGA Report 90007).

[5] D. E. Goldberg, K. Deb, and B. Korb. Messy genetic algorithms revisited: Studies in mixed size and scale. *Complex Systems*, 4:415–444, 1990.

[6] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3(5):493–530, 1989. (Also TCGA Report 89003).

[7] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J. J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49. 1987.

[8] K. Goldberg, D. E. amd Deb and J. Horn. Massive multimodality, deveption, and genetic algorithms. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 37–46. 1992.

[9] G. Harik. Finding multiple solutions in problems of bounded difficulty. IlliGal report no. 94002, University of Illinois, The Illinois Genetic Algorithms Laboratory, Urbana-Champaign, 1994.

[10] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[11] S. W. Mahfoud. Crowding and preselection revisited. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 27–36. 1992.

[12] G. V. Rekalitis, A. Ravindran, and K.M. Ragshell. *Engineering optimization: Methods and applications.* New York: Wiley, 1983.