Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.

Conference Paper · January 1985

Source: DBLP

1 author:

J. David Schaffer

Binghamton University

83 PUBLICATIONS   **7,839** CITATIONS

Some of the authors of this publication are also working on these related projects:

ontology and machine intelligence to develop Bayesian network process for threat detection / response View project

evolution of spiking neural networks for machine intelligence applications View project

Kuchinski, M.J. (1985), "Battle Management Systems Control Rule Optimization Using Artificial Intelligence", Technical Note, Naval Surface Weapons Center, Dahlgren, VA.

Kullback, S. (1959), *Information Theory and Statistics*, John Wiley and Sons, New York.

J. David Schaffer
Department of Electrical Engineering
Vanderbilt University
Nashville, TN 37235

ABSTRACT

Genetic algorithms (GA's) have been shown to be capable of searching for optima in function spaces which cause difficulties for gradient techniques. This paper presents a method by which the power of GA's can be applied to the optimization of multiobjective functions.

1. Introduction

There is currently considerable interest in optimization techniques capable of handling multiple non-commensurable objectives. Many practical problems are of this type where, for example, such factors as cost, safety and performance must be taken into account.

A class of adaptive search procedures known as genetic algorithms (GA's) have already been shown to possess desireable properties [3,10] and to out perform gradient techniques on some problems, particularly those of high order, with multiple peaks or with noise disturbance [4,5,6]. This paper describes an extension of the traditional GA which allows the searching of parameter spaces where multiple objectives are to be optimized. The software system implementing this procedure was called VEGA for Vector Evaluated Genetic Algorithm.

The next section of this paper will describe the basic GA and the vector extension. Then some properties are described which might logically be expected of this method. Some preliminary experiments on some simple problems are then presented to illuminate these properties and finally, VEGA is compared to an established multiobjective search technique on a set of more formidable problems.

2. A Vector Genetic Algorithm

Unlike many other search techniques which maintain a single "current best" solution and try to improve it, a GA maintains a set of possible solutions called a population. This population is improved by a cyclic two-step process consisting of a selection step (survival of the fittest) and a recombination step (mating). Each cycle is usually called a generation. More detailed descriptions of these operations may be found in the literature [3,4,5,6,10].

The question addressed here is, how can this process be applied to problems where fitness is a vector and not a scalar? How might survival of the fittest be implemented when there is more than one way to be fit? We exclude scalarization processes such as weighted sums or root mean square by the assumption that the different dimensions of the vector are non-commensurable.

When comparing vector quantities, the usual concepts employed are those proposed by Pareto [11,13]. For two vectors of the same size, the equality, less-than and greater-than relations require that these relations hold element by element. Another relation, partially-less-than, is defined as follows: vector $X = \{x1, x2, \ldots, xn\}$ is said to be partially-less-than vector $Y = \{y1, y2, \ldots, yn\}$ iff $xi <= yi$ for all i and for at least one value of i, $xi < yi$. Assuming that minima are sought, if X is partially-less-than Y, then Y is said to be inferior to or dominated by X. The objective of a search for minima in a vector-valued space is, then, a search for the set of non-inferior members, or the members not dominated by any others. At least one member of this Pareto-minimal set will dominate each vector outside the set, but among themselves, none is dominated.

With these concepts in mind, a simple vector survival of the fittest process was implemented. The selection step in each generation became a loop, each time through the loop the appropriate fraction of the next generation was selected on the basis of another element of the fitness vector. This process, illustrated in figure 1, protects the survival of the best individuals on each dimension of performance and, simultaneously, provides the appropriate probabilities for multiple selection of individuals who are better than average on more than one dimension.

3. Some Anticipated Properties of VEGA

3.1 Multiple Solutions

One potential advantage of VEGA over other optimization searches should now be clear. Since the object of the search is a set of solutions, a GA has a built-in advantage by working with a population of test solutions. By comparing each individual in a population to every other, those who are dominated by any other/s can be flagged as inferior. The set of non-inferior individuals in each generation is the current best guess at the
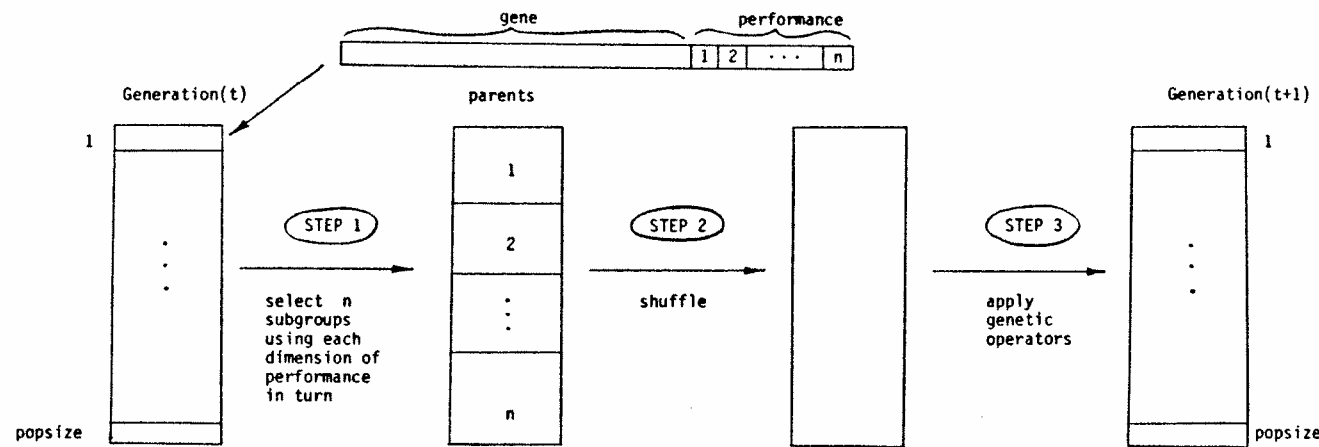
Figure 1. Schematic of VEGA Selection

Pareto-optimal (PO) set. By presenting a number of non-inferior solutions, VEGA provides the user with an idea of the tradeoffs required by his problem if a single solution must be selected. It should be noted that VEGA's view of non-inferiority is strictly local; it is limited to the current population. While a locally dominated individual is also globally dominated, the converse is not necessarily true. An individual who is non-dominated in one generation may become dominated by an individual who emerges in a later generation.

3.2 Possible Speciation

There is a potential problem with this vector selection process. Survival pressure is applied favoring extreme performance on at least one dimension of performance. If a utopian individual (i.e. one who excels on all dimensions of performance) exists, then he may be found by genetic combinations of extreme parents, but for many problems this utopian solution does not exist. For these problems, the location of the Pareto-optimal set or front is sought. This front will contain some members with extreme performance on each dimension and some with "middling" performance on all dimensions. Frequently, these compromise solutions are of most interest, but there may be danger of their not surviving VEGA's selection process. This might give rise to the evolution of "species" within the population which excel on different aspects of performance. This danger is expected to be more severe for problems with a concave PO front than for those with a convex one. See figure 2.

Two methods for combating this potential property of VEGA were conceived. One trick would be to provide a heuristic selection preference for non-dominated individuals in each generation. This would provide extra protection for the "middling" individuals.

Another, not necessarily exclusive, approach would be to try to encourage crossbreeding among the "species" by adding some mate selection

heuristics. In a traditional GA, mates are selected at random. On the assumption that utopian individuals are more likely to result from crossbreeding than inbreeding, such heuristics might speed the search.

4. Preliminary Experiments

4.1 The Test Functions

In order to test the properties of VEGA search, a set of three simple functions (f1, f2 & f3) was selected.

F1 was a single-valued quadratic function of three variables. (i.e. $f1(x1,x2,x3) = x1^{**}2 + x2^{**}2 + x3^{**}2$). This function was run to test whether VEGA reverts to a traditional GA when the performance vector has only one dimension.

F2 was a two-valued function of one variable (i.e. $f21(x) = x^{**}2$; $f22(x) = (x-2)^{**}2$). The initial random population for the search on this function is illustrated in figure 3. In addition to the locations of x, f21 and f22, this figure also shows the dominated flag for each x (1 if dominated, 0 if not). The PO region is $0<=x<=2$.

F3 was another two-valued function of one variable, but with two disjoint PO regions $0<=x<=2$ and $4<=x<=5$.

4.2 Heuristics

In order to mitigate the anticipated loss of "middling" individuals a heuristic was tested which gave an extra selection preference to locally non-dominated individuals. This preference took the form of numeric adjustments to the performance measures which were required by the selection algorithm to sum to zero across the population. Therefore, a small penalty was deducted from each inferior individual and the sum of these penalties was divided among the non-inferior individuals.

Experiments were also conducted to see if the search for the PO front could be improved by mate-selection heuristics which encouraged crossbreeding. Inbreeding, in this context, means a
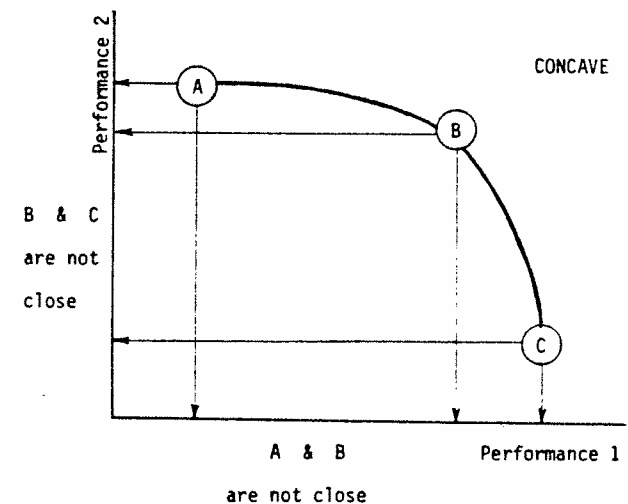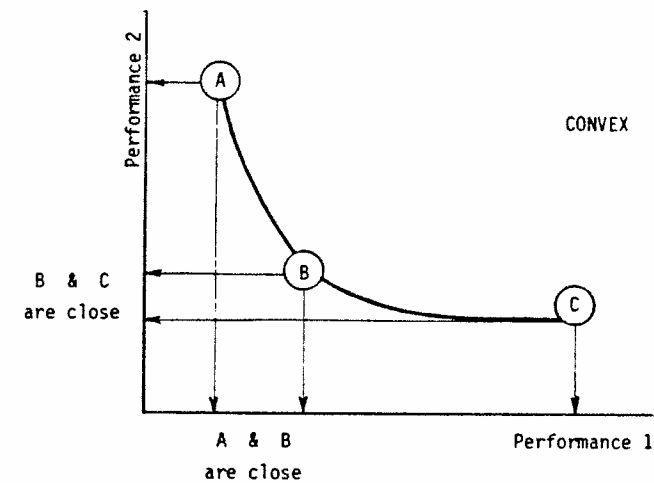

Figure 2. A Concave and Convex Pareto-Optimal Front

mating between two individuals whose high performance is on the same dimension. Two such heuristics were tested, both attempting to improve upon the performance of VEGA with random mating. Random mating was implemented by shuffling the population and mating pairs from the top, shown as step three in figure 1. Each heuristic proceeded by selecting a individual at random and then selecting a mate whose distance in performance space was maximum. Two distance measures were tested, Euclidian distance and "improvement" distance which was computed ignoring those dimensions on which the proposed mate performed worse.

4.3 Results

All of these experiments were conducted with populations of 30 individuals per dimension of the performance vector, and crossover and mutation rates of .95 and .01 respectively. This represents a smaller population size and higher rates of
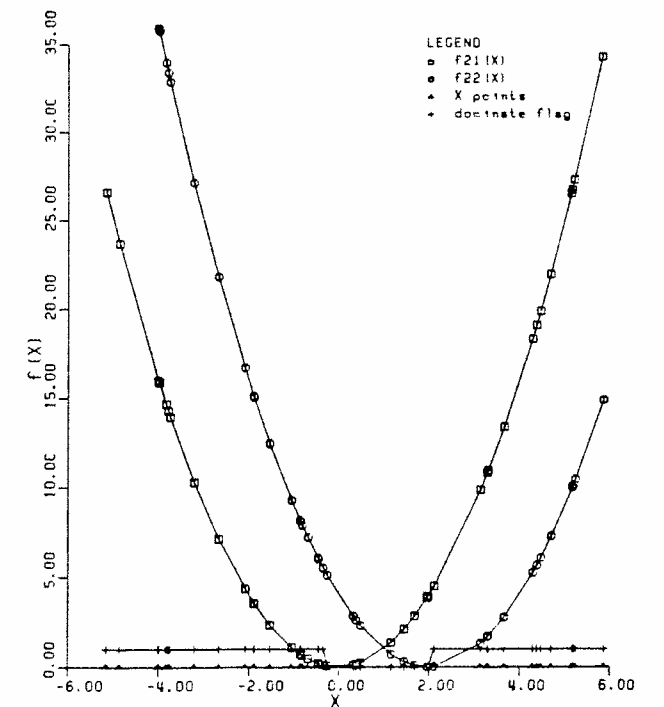

Figure 3. F2 Generation Zero

application of the genetic operators than has been traditional [3,5]. These setting were, however, suggested by the work of Grefenstette [8].

On f1, VEGA replicated a search previously conducted on this function by a traditional GA [7] when started with the same random seeds. Thus, VEGA does appear to be a vector generalization of a scalar GA.

On f2, VEGA evolved the population illustrated in figure 4 in just three generations. While not all the individuals are in the PO region ($0<=x<=2$), those which are outside are known to be dominated. This result, combined with similar performance by VEGA on f3 yielded some confidence in the soundness of the VEGA approach.

However, during these experiments, a dangerous property of the heuristic selection preference for non-dominated individuals was discovered. It had a tendency to produce sudden premature convergence of the population to a suboptimal solution. This occurred when, in an early generation, only one or two individuals managed to be non-dominated. Then, the sum of the dominated penalties was large and, when divided among very few, gave them an overwhelming selection advantage. This lead to subsequent generations consisting only of offspring of a few parents with too little genetic diversity. After this observation, this heuristic was removed. VEGA has, so far, not exhibited the anticipated loss of the "middling" individuals from the PO set. Perhaps concave PO fronts are not a characteristic of many practical problems.

The mate-selection heuristics faired no better. Random mating proved superior to both of them. This was an encouraging finding for two-
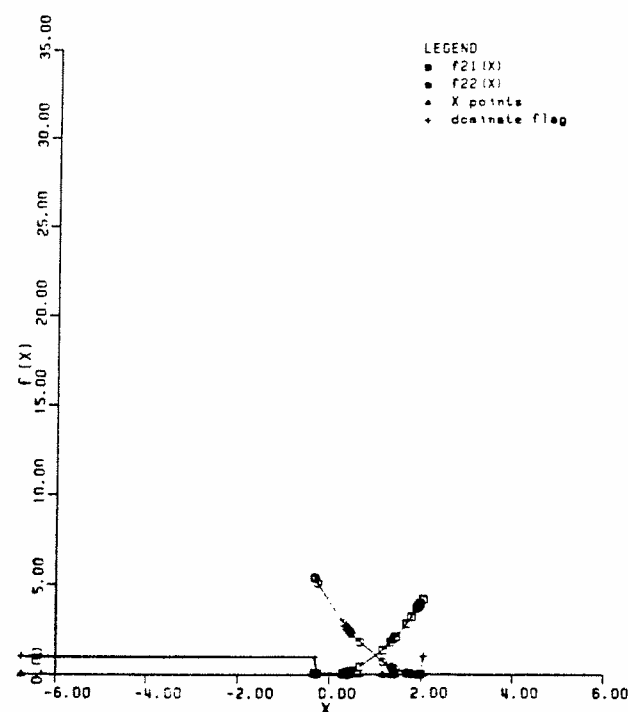
Figure 4. F2 Generation Three

valued problems, since the probability of inbreeding with random mating decreases as the number of dimensions of performance increases. All subsequent experiments utilized neither of these heuristics.

5. Comparison of VEGA with ARSO

Once some confidence was acquired that VEGA was able to conduct a genetic search in spaces with multiple objectives, it was desired to compare the performance of VEGA with that of an established technique for multiple objective search.

5.1 The ARSO Technique

For comparison purposes, the Adaptive Random Search Optimization (ARSO) procedure, pioneered by Beale [1,2], was selected. ARSO requests a starting point in the parameter space to be searched and proceeds to try to improve upon it by randomly perturbing the parameters. Statistics (mean & variance) are maintained for all perturbations which produce improvements (defined as a new solution which dominates the old one), and these statistics are used to guide the future perturbations. Random perturbation techniques have been shown to solve a large class of optimization problems faster than gradient techniques when the number of parameters exceeds four, and furthermore, the convergence time seems to increase only linearly with this number. ARSO had already exhibited high performance in problems of the sort tested here.

5.2 Some Methodological Problems

The comparison of two search procedures presents some methodological problems which are complicated when there are multiple objectives. One approach is to run each procedure until the solution is within some tolerance of a known solution and then compare the computational effort. This approach was rejected since the true solution was not known for the test problems. It was desired to compare the methods on problems whose solutions were not known so as to include in the comparison, the stopping criterion of each method. ARSO has a threshold on the number of perturbations tried without finding an improvement which forces a halt to the search. VEGA has no such preset stopping criterion and is stopped by the user when no further improvement is evident.

Another approach is to run both procedures for the same amount of computational effort and then compare the quality of the solutions. Comparing vector solutions is probably best done by checking if any are dominated by those provided by the other procedure. If not, then a tie must be declared. This approach may be unfair since ARSO reports only a single solution while VEGA may report several.

The approach adopted was to run each procedure to its natural stopping criterion. All proposed non-dominated solutions were then compared and, if any were found to be inferior, they were rejected. (Included in this set were solutions provided by Hartley [9] who used a variant of ARSO, but solved the scalar problem of the equally weighted sum of the errors on all dimensions.) Then, the number of "ultimately" non-dominated solutions found by each procedure was plotted against computational effort (number of function evaluations).

5.3 The Test Functions

A set of three problems, drawn from the domain of control engineering was contributed by a colleague, Hartley [9]. All involved the simulation of a system with a different integration operator for each of the system state variables. The systems were of orders 2, 3 and 7 respectively. The object of the search was an optimal set of integrators, each characterized by three parameters, making the dimensions of the parameter search spaces 6, 9 and 21, respectively. The performance measures were the rms error of the simulated solution from a known solution, one for each of the state variables making the dimensions of the performance spaces 2, 3 and 7.

All searches were conducted using the same GA parameters as were used for the preliminary problems. The integrator parameter sets were gray coded (see Schaffer[12] or Bethke[3]) to 12 bit precision, making the binary search spaces $2^{**}(12^*6) = 4.7 * 10^{**}21$, $2^{**}(12^*9) = 3.5 * 10^{**}32$ and $2^{**}(12^*21) = 7.2 * 10^{**}75$ for the three systems, respectively.

5.4 Results

While the true system behavior was assumed known, the object of the search was for optimal integrators for the simulations, and these were not known. Thus the problem of when to stop searching had to be faced. To illustrate, a scatter plot of performance of the initial random generation for the second order system is shown in figure 5. Figure 6 shows that considerable improvement had been achieved in three generations. Figure 7 shows the leading edge of the population after 49 generations. Note that the axes have been expanded three orders of magnitude. After running VEGA to generation 110 no substantial increase in performance was evident. See figure 8. There are however, several more points on what appears to be the PO front. Thus, a decision to stop such a search must be a judgement call based on a belief that the PO front has been located and that further search effort would be wasted.

The experiences were similar for the third and seventh order systems, but scatter plots for these high order systems could not be drawn.

Before proceeding to the comparison of VEGA with ARSO, it may be instructive to illustrate one of the ARSO searches in the second order system problem. Figure 9 traces the improvements in the solution found by ARSO and is presented on the same axes scales used for figures 5 to 8. ARSO found a solution which was judged "ultimately" non-dominated in 607 evaluations. ARSO's stopping criterion halts if no improvement is located after 1000 consecutive evaluations and so this run continued until 1607 evaluations and halted.

A second run of ARSO was initiated with one of the two non-dominated individuals from the initial population generated by VEGA. This run halted after about 1300 evaluations, but its solution was inferior. VEGA, on the other hand, did not locate its first "ultimately" non-dominated solution until 2621 evaluations and by 6000 it had found eight. These results are shown in figure 10.
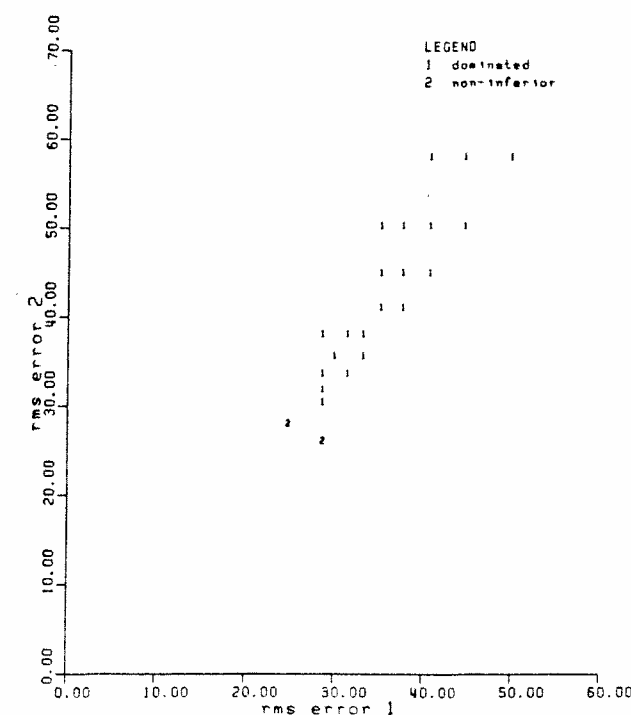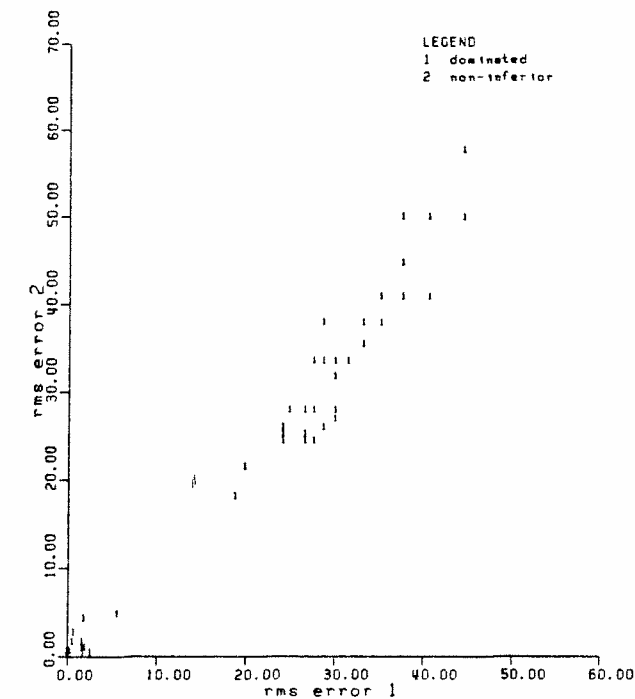

Figure 6. Second Order System -- Generation Three
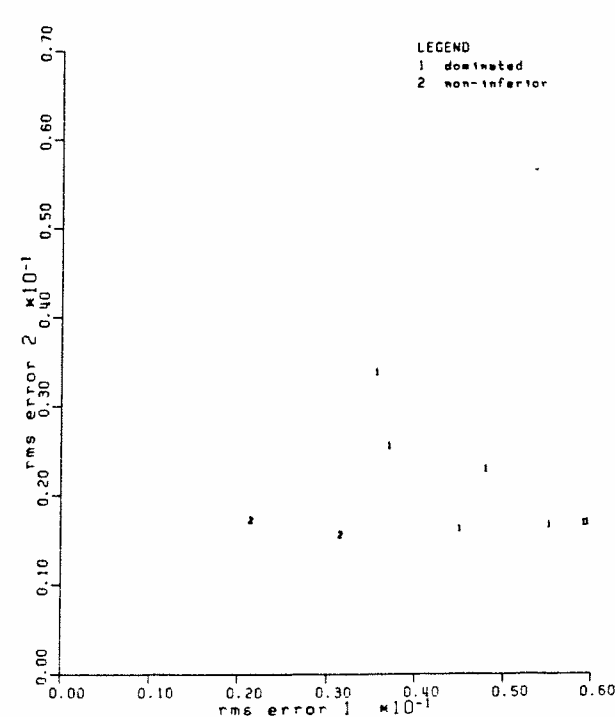

Figure 5. Second Order System -- Generation Zero

Figure 7. Second Order System -- Generation 49

Figure 8. Second Order System — Generation 110

Figure 9. Trace of ARSO Search, Second Order System
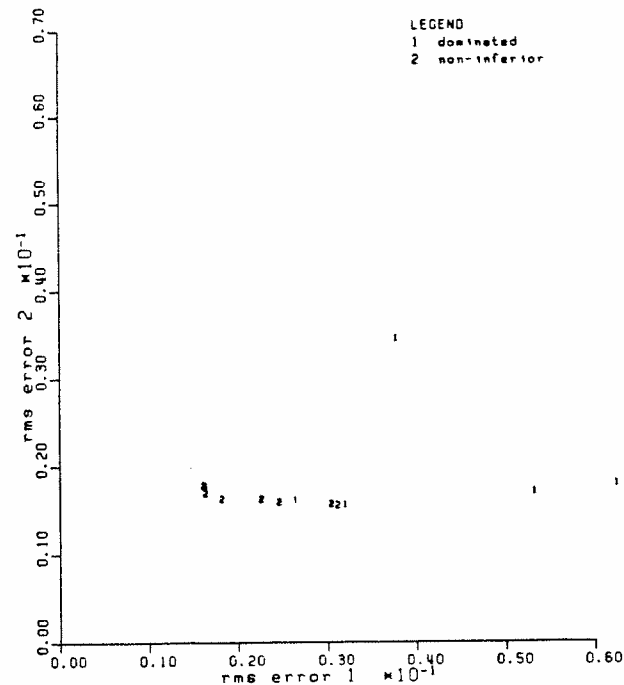
The tentative conclusion from these runs is that ARSO is fast, but may get trapped on local extrema. VEGA is slower, but more robust.

Four searches were made on the 3rd order system, two with VEGA and two with ARSO. VEGA was initiated with a random population for one run and given a whole population of clones of Hartley's solution for the other. ARSO was started with a reasonable starting point analogous to the starting point that lead to success on the 2nd order problem, and also the Hartley solution. The results of these tests are presented in figure 11. ARSO again found a good solution in under 2000 evaluations on its first run. When given a PO solution, it could only try for 1000 evaluations to improve it and then halt. VEGA found a non-dominated solution quite early in its search (415 evaluations), but because it was not sufficiently extreme on any one dimension of performance, it did not survive into future generations. More good solutions emerged later with VEGA having five after 10000 evaluations. VEGA, unlike ARSO, when given a PO solution, quickly located many variants of it.

The same four searches were run on the 7th order system. This time neither VEGA nor ARSO located any solutions which were not dominated by the Hartley solution. Hartley had used his knowledge of the problem to start his search at a close-to-PO point, but both VEGA and ARSO were started without this prior knowledge. The stopping criteria for ARSO had been relaxed to lengthen the search and a variance parameter had also been relaxed so as to broaden the search, but after almost 12000 evaluations no non-dominated solutions had been found. Its best solution at that time was

also known to be unstable. VEGA searched for almost 36000 evaluations without locating any solutions not dominated by Hartley's, however many of them were stable. Again, when told where to look, VEGA generated many more PO solutions.

The tentative conclusion, then, seems to have been supported by the higher order searches.

6. Discussion

The major finding of this research was that vectorization of performance feedback and the selection process of a GA can be successfully done. This opens the domain of multiobjective optimization problems to the already established power of genetic search.

Heuristic modifications of the traditional method to give selection preference to non-dominated members of a population and to try to improve on random mating proved to be inferior to the traditional method. The possibility that VEGA may have a weakness in the central region of a concave PO front cannot be eliminated, but empirical evidence to date suggests that it may not be serious.

The comparisons of VEGA with ARSO contain no small amount of "apples versus oranges." The methods differ in the number of solutions presented and in the way their searches are normally halted. However, both contain stochastic elements, both conduct multidimensional search and both are halted when no further improvement is apparent. Both may be started with random information, or may take advantage of prior knowledge the user possesses about his search space. In the comparison runs VEGA
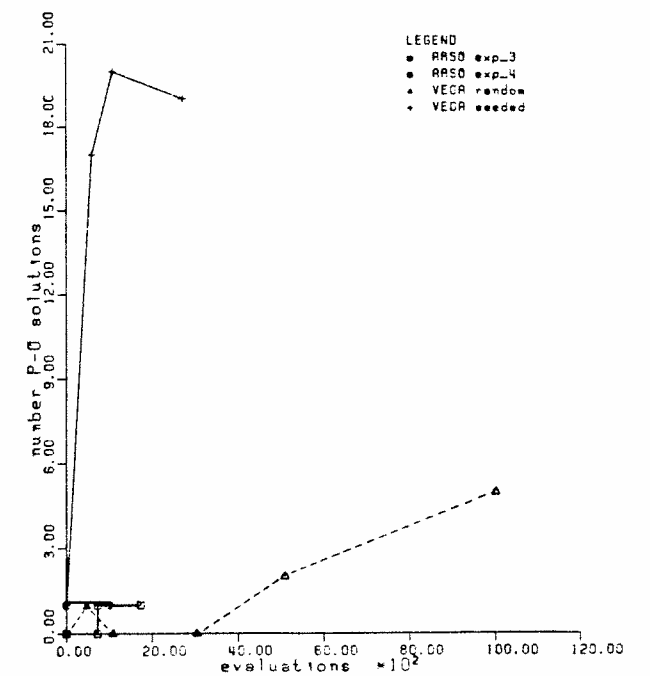

Figure 11. ARSO vs VEGA on Third Order System

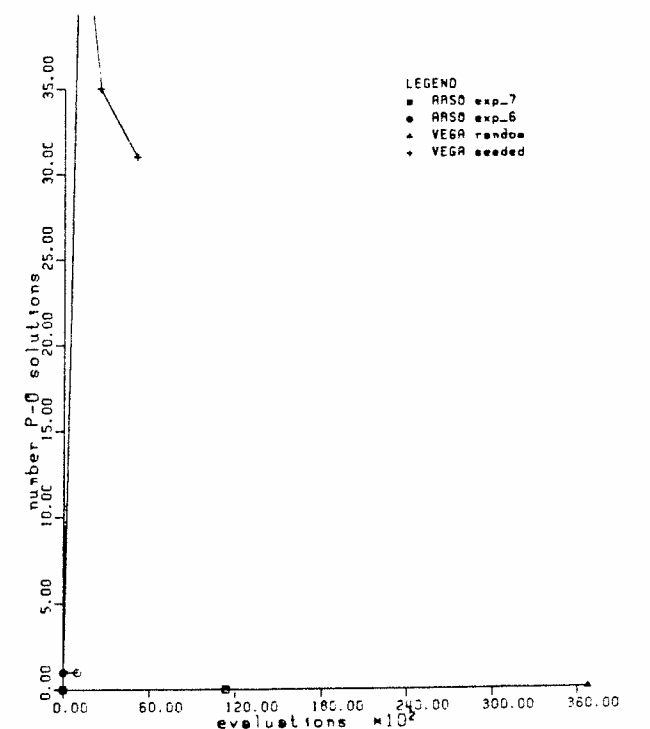Figure 10. ARSO vs VEGA on Second Order System

Figure 12. ARSO vs VEGA on Seventh Order System

was given several times more computational effort than was ARSO, due largely to differences in the methods for stopping each search.

The general conclusion of the comparison was that ARSO is capable of very quickly locating solutions to complex multidimensional problems, but its performance may be less robust than VEGA's. VEGA, on the other hand, takes longer to locate the good regions of complex search spaces, but seems to be able to do so more reliably. This conclusion is not dissimilar to previous results from comparison of scalar genetic search with gradient techniques [5,6].

Finally, a simple method has been conceived which may improve both VEGA and ARSO. By maintaining a data structure "off to the side" containing all non-dominated solutions encountered in the search, VEGA would be protected against the loss of good but not extreme individuals, as occurred in the search on the 3rd order problem. Similarly, ARSO would then have the power to report a number of solutions instead of only one. Furthermore, by monitoring the adding and subtracting of members to this set, both techniques might be given a more rational stopping criterion. Work on this addition to both methods will commence in the near future.

REFERENCES

1. Guy O. Beale Optimal Aircraft Simulator Development by Adaptive Random Search Optimization, Ph.D. Dissertation, University of Virginia, Charlottesville, Virginia, May 1977.

2. Guy O. Beale and Gerald Cook, "Optimal digital simulation of aircraft via random search techniques," J. Guidance and Control, Vol. 1, no. 4, July-Aug. 1978.

3. Albert Donally Bethke Genetic Algorithms as Function Optimizers, Ph.D. Dissertation, University of Michigan, Ann Arbor, Michigan, Jan 1981.

4. Anne Brindle Genetic Algorithms for Function Optimization, Ph.D. Dissertation, University of Alberta, Edmonton, Alberta, Canada, Jan 1981.

5. Kenneth DeJong, Analysis of the behavior of a class of genetic adaptive systems, Ph. D. Dissertation, University of Michigan, Ann Arbor, Mich., 1975.

6. Kenneth DeJong, "Adaptive System Design: A Genetic Approach," IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-10 No. 9, Sept 1980.

7. John Grefenstette, "A user's guide to GENESIS," Tech. Report CS-83-11, Computer Science Dept., Vanderbilt University, Nashville, Tenn., Aug. 1983.

8. John Grefenstette, "Genetic algorithms for multilevel adaptive systems," IEEE Trans. on Systems, Man and Cybernetics, in press.

9. Thomas Hartley, Parellel Methods for the Real Time Simulation of Still Nonlinear Systems, Ph.D. Dissertation, Vanderbilt University, Nashville, Tenn., Aug. 1984.

10. John H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan 1975.

11. V. Pareto, Cours d'Economie Politique, Rouge, Lausanne, Switzerland, 1896.

12. J. David Schaffer Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms, Ph.D. Dissertation, Vanderbilt University, Nashville, Tennessee., Dec 1984.

13. Thomas L. Vincent and Walter J. Grantham, Optimality in Parametric Systems, John Wiley and Sons, New York, 1981.

Adaptive Selection Methods for Genetic Algorithms

James Edward Baker

Computer Science Department
Vanderbilt University

Abstract

Premature convergence is a common problem in Genetic Algorithms. This paper deals with inhibiting premature convergence by the use of adaptive selection methods. Two new measures for the prediction of convergence are presented and their accuracy tested. Various selection methods are described, experimentally tested and compared.

1. Introduction

In Genetic Algorithms, it is obviously desirable to achieve an optimal solution for the particular function being evaluated. However, it is not necessary or desirable for the entire population to converge to a single genotype. Rather the population needs to be diverse, so that a continuation of the search is possible. The loss of an allele indicates a restriction on the explorable search space. Since the full nature of the function being evaluated is not known, such a restriction may prevent the optimal solution from ever being discovered. If convergence occurs too rapidly, then valuable information developed in part of the population is often lost. This paper deals with the control of rapid convergence.

Three measures are typically used to compare genetic algorithms. They are: the *Online Performance,* the average of all individuals that have been generated; the *Offline Performance,* the average of the Best Individuals from each generation; and the *Best Individual,* the best individual that has been generated. We attempt to optimize functions and therefore use the Best Individual measure for comparison. In order to improve this measure, we promote diversity within the population and control rapid convergence. Increased diversity detrimentally affects the Online Performance measure and inhibited convergence detrimentally affects the Offline Performance measure. Improving these two performance measures is not in the scope of this paper.

Methods for the prediction of a rapid convergence are the topics of section 2. Section 3 will describe various algorithms with which to slow down convergence, and section 4 will present their results. A conclusion section will follow the results.

2. Prediction of Rapid Convergence

There are two different aspects to the control of rapid convergence. First, how can one tell that it has occurred and second, how can one predict when it will occur.

Recognizing rapid convergence after it has occurred is rather straightforward. By its very meaning, a rapid convergence will result in a dramatic rise in the number of lost and converged alleles. A *lost allele* occurs whenever the entire population has the same value for a particular gene. Thus subsequent search with that gene is impossible. A *converged allele,* as defined by DeJong [1], is a gene for which at least 95% of the population has the same value. However, the effects of rapid convergence are not limited to only those alleles which are indicated by these measures. A rapid take over of the population will cause all genes to suddenly lose much of their variance. We define *bias* as the average percent convergence of each gene. Thus for binary genes, this value will range between 50, for a completely uniform distribution (in which for each gene there are as many individuals with a one as a zero) and 100, for a totally converged population (in which each gene has converged to a one or a zero). The bias measure provides an indication of the entire population's development without the disadvantage of a threshold, such as the one suggested by DeJong to indicate a converged allele. A threshold does not indicate the amount by which individuals exceed it or the number of individuals which fall just short. We can therefore monitor the sudden jumps in the lost, converged or bias values to determine when a rapid convergence has occurred.

PROCEEDINGS OF
AN INTERNATIONAL CONFERENCE ON
GENETIC ALGORITHMS
AND THEIR APPLICATIONS

July 24-26, 1985
at
Carnegie-Mellon University
Pittsburgh, PA

Sponsored By

Texas Instruments, Inc.

U.S. Navy Center for Applied Research
in Artificial Intelligence
(NCARAI)

John J. Grefenstette
Editor