Optimizing Stochastic and Multiple Fitness Functions

Joseph L. Breeden

SFI WORKING PAPER: 1995-02-027

SANTA FE INSTITUTE

Optimizing Stochastic and Multiple Fitness Functions[*]

Joseph L. Breeden
Santa Fe Institute
1399 Hyde Park Rd., Santa Fe, NM 87501
e-mail: breeden@santafe.edu

June 7, 1995

Abstract

How does one optimize a fitness function when the values it generates have a stochastic component? How does one simultaneously optimize multiple fitness criteria? These questions are important for many applications of evolutionary computation in an experimental environment. Solutions to these problems are presented along with discussion of situations where they arise, such as modeling and genetic programming. A detailed numerical example from control theory is also provided. In the process, we find that population-based search algorithms are well-suited to such problems.

1 Stochastic Fitness Functions

In the theoretical development of evolutionary computation[1], the fitness functions considered are almost invariably deterministic functions of a predefined parameter set. In the real-world application of optimization techniques, we must often account for stochastic fitness functions. A simple example of a stochastic fitness function is one measurement drawn from a distribution of readings obtained from an experimental apparatus. The fitness may be viewed as on realization of a random variable with a mean corresponding to the presumed true fitness. In some problems, the uncertainties can be significant.

Several approaches to optimizing stochastic fitness functions are possible, but the least successful method is to ignore the uncertainty. If we follow that path, we could easily be left at the end of the search with a parameter set which

[*] To appear in *Proceedings of the Fourth Annual Conference on Evolutionary Programming, 1995.*

[1] This paper assumes that the reader has a basic understanding of population-based stochastic search techniques, such as genetic algorithms [1, 2] (GA), evolutionary programming [3] (EP), or evolution strategies [4] (ES).

is suboptimal, but lucky enough in the single fitness realization to score higher than all other sets. To go beyond the simplistic approach, it helps to present some possible situations.

1.1 Alternative Criteria

In many applications, supplementary fitness criteria are available. For example, we can define a generic alternative criterion for noisy fitness functions similar to a signal-to-noise ratio by using the height of a peak relative to the surrounding region of the fitness landscape as a fitness measure. This is supplementary to the raw fitness of the peak. We will examine in the next section how to utilize multiple criteria.

1.2 Uncertainty Measures

Another common alternative is the standard deviation of the fitness over some a small neighborhood. We can generate a mean fitness and the uncertainty in that mean by computing the fitness over a small neighborhood in parameter space. (This assumes real-valued parameters or a fitness function which varies relatively smoothly over a range of integers.) We can then incorporate the uncertainty estimate into our fitness evaluation.

1.3 Segmented Fitness Functions

In some situations, we may want to segment the data and compute the fitness on each segment. This is an option in situations such as time series analysis. Rather than minimize the overall error in a parameterized model over a data set, we can segment the data so that we produce one fitness estimate for each data segment.

Segmentation increases the uncertainty in the fitness estimate of a single segment, but provides multiple measures instead of a single combined measure. This is most beneficial when we are concerned about drift in the fitness estimates as we scan along the data set. The classic example is non-stationarity in time series.[2]

As a more concrete example, imagine that we are conducting a scattering experiment and recording the exit angle for each particle scattered from the target. While collecting the data, the target is briefly perturbed causing a strong peak at one angle. If we average all the data together, we discover a dominant,

[2] From the author's point of view, non-stationarity is a property of a model describing a time series, not the data itself. Say we had a model for the water usage in San Diego on January 29, 1995, based upon the usage patterns earlier in the month. From our errors, we might conclude that the data is non-stationary, whereas a more insightful model (one which included the NFL Superbowl halftime) would have no such problem.

though erroneous, peak. By segmenting the data, the spurious nature of this peak becomes more apparent.

Segmentation is a natural way to produce models which are stable with respect to perceived non-stationarity. It is conceptually similar to cross-validation, except that in multi-stage cross-validation, the search considers each data segment sequentially. They are not optimized in parallel as we intend to do.

2 Multiple Fitness Functions

All of the approaches described above involve the use of vector-valued fitness functions. Thus, a secondary problem has been created. The original scalar fitness function has been replaced with a vector-valued fitness function which is no longer trivial to optimize. By addressing this problem, we expand the range of problems being addressed to any that require the simultaneous optimization of multiple criteria.

In the following sections we will discuss the pros and cons of various optimization methods. At some level, all these approaches are intended to produce a single scalar value such that logical comparison of different parameter sets is possible.

2.1 Significance Measures

When the vector fitness is composed of the raw fitness, f, and the uncertainty in that value, σ, they are commonly combined as $s = f/\sigma$ to produce a significance estimate in units of standard deviation. Such a quantity is useful, but provides different information from f. Optimizing s returns the most clearly defined peak, not necessarily the highest peak. In most situations, the two quantities, f and s, must still be simultaneously optimized.

2.2 Linear Combinations

When multiple fitness measures are available, linear combination is the most commonly suggested approach. It is also the most fragile. If the composite fitness is

$$F = \sum_{i=1}^{N} a_i f_i, \tag{1}$$

we must now solve the meta-optimization problem of defining the $\{a_i\}$. In some situations, a simple answer from the theory may be available. One obviously beneficial technique is to normalize the fitness distributions, so

$$f_i' = \frac{f_i - \mu_i}{\sigma_i}, \tag{2}$$

but this still does not indicate the relative importance of the various f_i'.

Non-Gaussian distributions are a major concern. If the distribution of f_1' is normal, but the distribution of f_2' is strongly skewed or has broad tails, F' will have an enhanced probability of being dominated by f_2'. This is part of the more general outlier problem. If the f_i' have a stochastic component, an outlier in one of the f_i' can dominate F'. In practice, the optimization routine becomes an outlier search, producing very few real solutions.

Yet another wrinkle in this approach is combining minimization and maximization. Maximization and minimization can always be interchanged by letting $f' = \frac{1}{f}$, but the resulting distribution may be strongly skewed. Significant thought must be given to how fitness functions are added.

2.3 Rank Statistics

The problems described above naturally lead to the consideration of rank, or nonparametric, statistics [5]. All the f_{ij} (the ith fitness function applied to the jth member of the population) are computed. Then each member is ranked relative to the rest of the population for each fitness function, producing a matrix of ranks, r_{ij}. Ranks avoid normalization and distribution problems.

To generate a scalar quantity for comparison, a net rank for the jth member, R_j is computed. If a priori knowledge of the relative importance or stochasticity of the f_i is available, a linear combination can again be considered,

$$R_j = \sum_{i=1}^{N} \alpha_i r_{ij} \tag{3}$$

When some of the f_i are very noisy, outliers will still be a source of concern. We can dispense with the outlier problem by taking R_j to be the median of the ranks $\{r_{ij}\}$,

$$R_j = \mathrm{med}_{i=1,N} \quad r_{ij} \tag{4}$$

This implicitly assumes a uniform weighting for the f_i, but a priori knowledge of the proper weights can still be incorporated by including certain f_i multiple times in the median calculation. We can also include diversity enhancement to prevent stagnation of the population during the search [6], and thus produce several candidate solutions to choose from.

The median rank approach is certainly the most robust of the techniques described. There are no real disadvantages, but there are some side-effects. The most obvious is that the fitness of member j is dependent upon all the members $k \neq j$. Fitness is now a relative rather than an absolute quantity. One might argue that relative fitness is more in keeping with biological reality, but such discussions are beyond the scope of this paper.

3 Vague Examples

The following examples are discussions of where these methods may be used advantageously. This is a representative sample, not an exhaustive list. A detailed numerical example is provided in the next section.

3.1 Modeling

The median rank technique just described can do much to improve the applicability of EP to data analysis. To go beyond rudimentary modeling requires the optimization of many attributes besides average residual error. Much of the current usefulness of neural nets in data analysis is due more to the proper addressing of these issues than to the basic neural net architecture. Regularization, multifold cross-validation, and visual inspection of secondary criteria to stop over-fitting are all part of a well-considered application of neural nets, yet none of these are specific to, nor invented by, the neural net community. In any modeling procedure, including neural nets, a median rank optimization of such quantities as minimum root-mean-square error, largest single error[3], fewest outliers, smoothness of approximation function, or any of these applied to segmented data (a la cross-validation) can be very beneficial. In some time series modeling, such as for control, minimizing the correlation between the error ϵ_i and $\epsilon_{i-\tau}$ (τ is some specified lag) or cumulative error over some window, $\sum_{i=1}^{N} \epsilon_i$, can be equally important. The number of possible criteria are as varied as the researchers needing answers, and all should be accommodated.

3.2 Genetic Programming

Genetic Programming (GP) typically involves using an evolutionary computation approach to evolve a computer program to perform a specific computation. For example, GP could be used to evolve a program which computes the digits of π. The standard fitness function is the distance from the algorithm's output to the desired value. With the median rank approach described above, other fitness criteria can be considered. This may sound unusual initially, but humans utilize multiple criteria every time they write code. Along with producing the correct result, algorithms should make efficient use of memory and cpu-time, be robust under extraordinary inputs, and be readable (using subroutines where appropriate). Many more domain specific criteria exist, but this list already demonstrates the advantages of considering multiple fitness functions. If the program to compute the digits of π were to be placed on a hand calculator, compromises must be made between numerical accuracy, cpu-time, and memory usage. Similarly, as networking moves toward intelligent agents, those agents could be evolved via GP, simultaneously optimizing many of the above criteria.

[3] RMS error is the L_2 norm, and largest single error is the L_∞ norm. Others are of course possible, particularly the L_1 norm.

The use of multiple fitness criteria should aid genetic programming in solving many real-world problems.

4 A Numerical Example—Open-Loop Control

Before concluding this paper, one detailed example should be provided. Control has been mentioned in this paper as an example of multiple fitness criteria [6]. Control examples come so readily to mind, because alternative criteria are so important.

Obviously, control error is to be minimized, but engineering realities provide many more constraints. We could minimize time to achieving control from initiation of control, maximize duration of control, minimize the cost of applying the control, maximize distance from hazardous operating regions, minimize complexity of applying the control with certain apparatus, and so forth. Considerations such as these are an ineradicable aspect of any control design. The objective is to bring these criteria into the optimization framework.

Consider the following problem in open-loop control. We have a system which is modeled as

$$x_{n+1} = M\left(x_n\right), \tag{5}$$

and we wish to entrain it to a period-2 goal dynamics $\{g_0, g_1\}$ by subtracting the undesirable dynamics and adding those desired

$$\begin{aligned} y_{n+1} &= M\left(y_n\right) + D_n \\ D_n &= g_n - M\left(g_n\right) \end{aligned}$$

Open-loop means no feedback is available from the system, so $M\left(y_0\right) \neq M\left(g_0\right)$. This procedure can work only in certain nonlinear dynamical systems which naturally dampen the control errors [7, 8, 9], and is sometimes referred to as entrainment control.

Now let

$$M\left(x_n\right) = \lambda x_n \left(1 - x_n\right), \quad \lambda = 3.8, \tag{6}$$

the Logistic Map in the chaotic regime. Suppose a period-2 goal $\{g_0, g_1\}$ is required that has a small control error,

$$\eta^2 = \sum_{i=1}^{N} \left(y_n - g_n\right)^2; \tag{7}$$

a small driving term,

$$\Delta^2 = \sum_{i=1}^{N} D_n; \tag{8}$$

that succeeds for many initial conditions; and has a large separation between g_0 and g_1. The vector fitness function is

$$\mathbf{f} = \{f_1 = \eta; f_2 = \Delta; f_3 = |g_1 - g_0|; f_{i+3} = T_i, i \in [1, N_i]\} \tag{9}$$

where N_i is the number of initial conditions tested, $T_i =$ the number of iterations for which $y^i_n \in [0, 1]$ starting with the initial condition

$$y^i_0 = \frac{2i - 1}{2N_i}. \tag{10}$$

f_1 and f_2 are averages over all successful trials, i.e., $T_i = N$. f_1 and f_2 have χ^2 distributions, f_3 has a roughly normal distribution, and f_{i+3} has a bimodal distribution composed of a binomial distribution and a solitary peak at N.

Because of these different distribution functions, a linear combination of the f_i would be marginally effective. Instead, the median rank approach described previously will be employed. Each candidate solution is be ranked according to each f_i with the median of these ranks being the overall fitness. We will also incorporate a weighting vector $w = \{w_1; w_2; w_3; w_{i+3}, \forall i \in [1, N_i]\}$ to allow the importance of the different fitness criteria to be adjusted.

The search algorithm is a simple EP approach with deterministic selection. The following table contains the results of the optimization for different weighting vectors.

Table 1: This table contains the best solution obtained for a variety of weights. The first three weights are each multiplied by N_i, so 1:1:1:1 indicates control error is as important as all the control times, T_i, combined. The fitness values shown are the control error, driving amplitude, and the separation between goals. The basin size is the number of initial conditions which resulted in entrainment. This is only a way to summarize the fitness values T_i. The individual T_i were used during optimization to aid the early generations of the search.

Weights	Solution	η	Δ	Separation	Basin
3:1:1:1	0.5009, 0.5013	4.1e-06	0.449	0.0005	13
3:2:1:2	0.5830, 0.5851	0.0036	0.339	0.0020	15
2:1:1:1	0.4726, 0.5425	0.0040	0.439	0.0699	13
2:1:2:2	0.1239, 0.6963	0.2394	0.218	0.5724	19
3:2:2:2	0.0783, 0.6910	0.2715	0.163	0.6127	22
1:1:1:1	0.0061, 0.7208	0.3549	0.033	0.7147	24

Which weights should have been chosen? Which fitness functions should have been included? The ones appropriate to the problem, of course. That seems like a simplistic answer, but it represents the heart of what we are attempting to achieve. The real world is messy. There are no simple, universal answers. No universal solution is provided in this paper. Instead, a collection of tools is offered which can be used in varying combinations to solve specific problems. The preceding example is a case in point. The person designing the controller would be the only one who would know which set of weights are correct or which fitness functions are pertinent. Without that domain knowledge, any solution obtained would be irrelevant. Too often in application, we have single-minded tools and multifaceted problems. Hopefully, that trend can be reversed.

5 Conclusion

The purpose of this paper is to raise awareness of the difficulties with stochastic and multiple fitness criteria, and to present solutions which have proven successful. Taken individually (uncertainty measures, segmentation, rank statistics), the ideas presented here are not original. But, evolutionary optimization techniques have seen little application in many fields because these problems have not been carefully considered with solutions spelled out for non-specialists. Hopefully, such problems will become part of the standard test suite when comparing and discussing optimization techniques.

We must also realize that the median rank fitness function allows more than just shoring up a weakness in the application of evolutionary computation. The median rank approach is only possible because of the existence of a population of candidate solutions. Further, it is a much cleaner solution than is possible with other optimization techniques. The whole is definitely more than the sum of the pieces mentioned above, and it presents an opportunity—an opportunity to solve via an automated procedure those problems which are certainly unpleasant by standard means.

References

[1] J. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

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

[3] D. B. Fogel, "An analysis of evolutionary programming", in: D. B. Fogel and W. Atmar, eds, *Proceedings of the First Annual Conference on Evolutionary Programming,* pages 1–16, San Diego, CA, 1992. Evolutionary Programming Society.

[4] T. Bäck, G. Rudolph, and H.-P. Schwefel, "Evolutionary programming and evolution strategies: Similarities and differences", in: D. B. Fogel and W. Atmar, eds, *Proceedings of the Second Annual Conference on Evolutionary Pro-*

gramming, pages 11–22, San Diego, CA, 1993. Evolutionary Programming Society.

[5] J. R. Thompson and R. A. Tapia, *Nonparametric Function Estimation, Modeling, and Simulation.* SIAM, Philadelphia, 1990.

[6] J. L. Breeden and N. Packard, "A learning algorithm for optimal representation of experimental data", *Bifurcations and Chaos*, **4**(2), 311–326, April 1994.

[7] A. Hübler and E. Lüscher, "Resonant stimulation and control of nonlinear oscillators", *Naturwissenschaften*, **76**, 67, 1989.

[8] E. Jackson and A. Hübler, "Periodic entrainment of chaotic logistic map dynamics", *Physica D*, **44**, 407–420, 1990.

[9] J. L. Breeden, "Open-loop control of nonlinear systems", *Physics Letters A*, **190**, 264–272, 1994.