

GENETIC ALGORITHMS: WHAT FITNESS SCALING IS OPTIMAL?

Vladik Kreinovich, Chris Quintana

Computer Science Department

University of Texas at El Paso, El Paso, TX 79968, USA, vladik@cs.ep.utexas.edu

Olac Fuentes

Department of Computer Science

Texas A&M University, College Station, TX 77840, fuentes@cs.tamu.edu

Abstract. Genetic algorithms are now among the most promising optimization techniques. They are based on the following reasonable idea. Suppose that we want to maximize an objective function $J(x)$. We somehow choose the first generation of “individuals” x_1, x_2, \dots, x_n (i.e., possible values of x) and compute the “fitness” $J(x_i)$ of all these individuals. To each individual x_i , we assign a survival probability p_i that is proportional to its fitness. In order to get the next generation we then repeat the following procedure k times: take two individuals at random (i.e., x_i with probability p_i) and “combine” them according to some rule. For each individual of this new generation, we also compute its fitness (and survival probability), “combine” them to get the third generation, etc. Under certain reasonable conditions, the value of the objective function increases from generation to generation and converges to a maximal value.

The performance of genetic algorithms can be essentially improved if we use *fitness scaling*, i.e., use $f(J(x_i))$ instead of $J(x_i)$ as a fitness value, where $f(x)$ is some fixed function that is called a *scaling function*. The efficiency of fitness scaling essentially depends on the choice of f . So what f should we choose?

In the present paper we formulate the problem of choosing f as a mathematical optimization problem and solve it under different optimality criteria. As a result, we get a list of functions f that are optimal under these criteria. This list includes both the functions that were empirically proved to be the best for some problems, and some new functions that may be worth trying.

1. INTRODUCTION TO THE PROBLEM.

Genetic algorithms: basic idea. The idea to simulate biological evolution for optimization dates back to 1960 [Bremermann 1962; Bremermann, Rogson, Salaff 1966]. Genetic algorithms, in their present form, were invented by John Holland (his pioneering work, starting from [Holland 1962, 1965], was summarized in [Holland 1975]), are now among the most promising optimization and machine learning techniques (see, e.g., [Goldberg 1989]). They are based on the following reasonable idea.

Suppose that we want to maximize an objective function $J(x)$ on a set X . We must then do the following:

- 1) First of all, we choose an integer n that is called a *population size*, and then we choose n elements $x_i^{(1)}$ from X . These elements are called *individuals of the first generation*.
- 2) For each of the individuals $x_i^{(k)}$ (the variable k is called a *generation number*; initially, $k = 1$), we compute a value $F(x_i^{(k)})$ called its *fitness* as $F(x_i^{(k)}) = J(x_i^{(k)})$.

Comment. The origin of this term is simple: In the analogy with Darwinism that underlies this whole approach, we want to make $J(x)$ as big as possible so that from our viewpoint, the bigger the $J(x)$ for some x , the more this x fits us. Hence, we would like to simulate an “evolution” and thus guarantee the “survival of the fittest”, i.e., in this case, the elements x with maximal possible $J(x)$.

- 3) Then for each of the individuals $x_i^{(k)}$ we compute the *survival probability* $p_i^{(k)}$ as

$$p_i^{(k)} = \frac{F(x_i^{(k)})}{\sum_{j=1}^n F(x_j^{(k)})}, \quad (1)$$

and then design a “random individual generator” that generates each individual $x_i^{(k)}$ with the probability $p_i^{(k)}$.

Comment. The formula for $p_i^{(k)}$ actually means that survival probability is proportional to fitness $p_i^{(k)} = CF(x_i^{(k)})$. The normalizing factor C is then determined by the condition that the sum $\sum_j p_j^{(k)}$ of all these probabilities is 1.

- 4) Now we are ready to compute (one by one) n individuals $x_i^{(k+1)}$ of the next generation. Namely, to get one new individual, we run a random individual generator twice, get two individuals $x_j^{(k)}$ and $x_l^{(k)}$ and “combine” these two individuals according to some combination rule.

In order to do this, we must have initially fixed some *recombination operator*, i.e., some algorithm that being applied to two elements of X , generates a new element (their *recombination*). This algorithm can also use random number generators to generate a combined individual. After we repeat this n times, we get an entirely new generation.

- 5) Now we can start the whole process 2)-4) anew, with the individuals of the new generation, etc.

Under certain reasonable conditions the value of the objective function increases from generation to generation and converges to a maximal value (a precise result is formulated in [Holland 1975, De Jong 1975, Goldberg 1986]).

Why scaling? There are at least three reasons why it is necessary to modify the procedure described above [Goldberg 1989]:

- 1) If the value of the objective function is negative for some x , we cannot apply this procedure, because then formula (1) leads to senseless negative values of probabilities.
- 2) Empirical studies of genetic algorithms, performed as early as [Bagley 1967, Rosenberg 1967, De Jong 1975], showed that in the beginning, the previously described algorithm often leads to the appearance of a few “superindividuals” who dominate the selection process and therefore slow it down. At the end, when the population consists largely of the individuals x , for which $J(x)$ is close to maximum, the competition is practically absent, which again slows down the process.
- 3) In some situations, there is no objective function at all. Namely, we want to find the parameters for which some system performs in the best possible way. We can figure out when the performance is better and when it is worse, so in reality what we have is a *ranking* of all possible behaviors. So for some individuals x and \bar{x} , we can say that x is better than \bar{x} ($x > \bar{x}$) or x is worse than \bar{x} ($x < \bar{x}$). Such problems are called *order problems* in [Sirag, Weisser 1987] (the problems for which an objective function is known are called *value problems*).

Of course, it is possible to find a function $J : X \rightarrow R$ from X into the set R of real numbers that is *consistent* with $>$ in the sense that $x > \bar{x}$ if and only if $J(x) > J(\bar{x})$, and then apply the genetic algorithm to this J . But the problem is that this function J is not uniquely defined: for any monotone function $f(z)$ from real numbers into real numbers the objective function $\bar{J}(x) = f(J(x))$ is also consistent with $>$. Also, the results of applying the genetic algorithm to J and $f(J)$ can be radically different: For some J , the genetic algorithm will converge quickly, and for some other \bar{J} it will be extremely slow. How do we choose J ?

This empirical fact that turning from J to $f(J)$ can drastically improve the algorithm prompted the idea that such a change can solve problems 1) and 2). So we arrive at the modification of the above procedure that is called *fitness scaling* [Goldberg 1989] and was proposed first in [Bagley 1967] and [Rosenberg 1967]. Instead of taking fitness equal to the value of the objective function $F(x) = J(x)$ (as above), we take $F(x) = f(J(x))$, where $f(z)$ is some monotone function from real numbers into real numbers (called a *scaling*).

In case of an order problem, when the objective function is not given, we can take $F(x_i^{(k)}) = f(r_i)$, where $f(z)$ is a scaling and r_i is a rank of the individual $x_i^{(k)}$ in the set of all individuals $x_j^{(k)}$ of this generation (the best individual has rank 1, the second best rank 2, etc, and the worst has rank n).

What scaling mechanisms are now used? A survey of such procedures is presented in [Forrest 1985] and briefly reproduced in [Goldberg 1989]. They include:

- 1) *linear scaling*, where $f(z) = az + b$. The constants a and b are chosen either at the beginning of the procedure, or recomputed for each generation [Forrest 1985]. For order problems such a procedure was proposed and used in [Baker 1985].
- 2) *power law scaling*, where $f(z) = z^\alpha$ for some constant α . This type of scaling was proposed by [Gillies 1985], who showed that for machine-vision applications the best value for α is 1.005; for other problem other values of α can lead to better results [Goldberg 1989, Ch. 4].
- 3) *exponential scaling*, where $f(z) = \exp(-\beta z)$ for some β . This type of scaling is motivated by an analogy with simulated annealing (started by [Hopfield 1982, Kirkpatrick et al 1983]; for a recent theoretical survey see [Romeo et al 1991]). As an example of its usage see [Sirag, Weisser 1987]. This type of scaling is the most commonly used in robotics [Davidor 1991].

Formulation of the problem and why it is important. Choosing an appropriate scaling is very important because the performance of genetic algorithms can be essentially improved by using scaling. However, as Goldberg notices in [Goldberg 1989, Ch. 4, p. 124], the existing procedures are somewhat *ad hoc*, i.e., they lack deep theoretical motivation. Therefore, it is desirable to figure out whether the scaling mechanisms that are now actually used are really the best possible ones or whether one can come out with some better ones, which can help if all other known scaling procedures do not help?

What we are planning to do. We formulate the problem of choosing the best scaling function f as a mathematical optimization problem and solve it under different optimality criteria. As a result, we get a list of functions f that are optimal under different criteria. This list includes

both the functions that were empirically proved to be the best for some problems, and some new functions that might be worth trying.

2. MOTIVATIONS OF THE PROPOSED MATHEMATICAL DEFINITIONS

Why is this problem difficult? We want to find a scaling function f that is the best in some reasonable sense, that is, for which some characteristic I attains the value that corresponds to the best performance of the genetic algorithm. For example, an average running time or the percentage of failures are minimal. The problem is that even for the simplest linear functions, we do not know how to compute any of these possible characteristics. How can we find f for which $I(f)$ is optimal if we cannot compute $I(f)$ even for a single f ? There does not seem to be a likely answer.

However, we will show that this problem is solvable (and give the solution).

The basic idea of our solution stems from the above-mentioned analogy between genetic algorithms and neural networks, where the similar problem of choosing the best non-linearity is of great importance. That problem has been solved in [Kreinovich, Quintana 1991] (using the ideas from [Kreinovich 1990]), and we are planning to solve our problems by appropriately modifying the mathematical formalism that was developed there.

We must choose a family of functions, not a single function. The expression (1) for the probability $p(x_i^{(k)})$ does not change if we multiply all the fitness values by a constant C , i.e., if we use $\tilde{F}(x) = CF(x)$ instead of $F(x)$. Therefore, the functions $f(z)$ and $\tilde{f}(z) = Cf(z)$, where C is a constant, lead to the same probabilities and hence to the same results.

We spoke about choosing a function $f(z)$, but from mathematical viewpoint, it is better to speak about choosing a *family* of functions f . Therefore, it is reasonable to suggest that if a function $f(z)$ belongs to this family, then this family must contain a function $\tilde{f}(z) = Cf(z)$ for every positive real number C .

In the simplest case, the family is $\{Cf(z)\}$ for some function $f(z)$. The next step is to consider *m-dimensional families* of functions, i.e., all functions of the type $f(z) = \sum_{i=1}^m C_i f_i(z)$ for some basis $f_i(z)$, where C_i are arbitrary constants.

Which family is the best? Among all such families, we want to choose the best one. In formalizing what “the best” means, we follow the general idea outlined in [Kreinovich 1990] and applied to neural networks in [Kreinovich, Quintana 1991]. The criteria to choose may be computational simplicity, efficiency of optimization, or something else. In mathematical optimization problems, numeric criteria are most frequently used, where to every family we assign some value

expressing its performance and choose a family for which this value is maximal. However, it is not necessary to restrict ourselves to such numeric criteria only. For example, if we have several different families that have the same average running time T , we can choose between them the one that has the minimal percentage of failures P . In this case, the actual criterion that we use to compare two families is not numeric, but more complicated: *a family Φ_1 is better than the family Φ_2 if and only if either $T(\Phi_1) < T(\Phi_2)$ or $T(\Phi_1) = T(\Phi_2)$ and $P(\Phi_1) < P(\Phi_2)$* . A criterion can be even more complicated. What a criterion *must* do is to allow us for every pair of families to tell whether the first family is better with respect to this criterion (we'll denote it by $\Phi_2 < \Phi_1$), or the second is better ($\Phi_1 < \Phi_2$) or these families have the same quality in the sense of this criterion (we'll denote it by $\Phi_1 \sim \Phi_2$).

The criterion for choosing the best family must be consistent. Of course, it is necessary to demand that these choices be consistent, e.g., if $\Phi_1 > \Phi_2$ and $\Phi_2 > \Phi_3$ then $\Phi_1 > \Phi_3$.

The criterion must be final. Another natural demand is that this criterion must be *final* in the sense that it must choose a *unique* optimal family (i.e., a family that is better with respect to this criterion than any other family). The reason for this demand is very simple. If a criterion does not choose any family at all, then it is of no use. If several different families are “the best” according to this criterion, then we still have a problem choosing the absolute “best” family. Therefore, we need some additional criterion for that choice. For example, if several families turn out to have the same training ability, we can choose among them a family with minimal computational complexity. So what we actually do in this case is abandon that criterion for which there were several “best” families, and consider a new “composite” criterion instead: Φ_1 is better than Φ_2 according to this new criterion if either it was better according to the old criterion or according to the old criterion they had the same quality, and Φ_1 is better than Φ_2 according to the additional criterion. In other words, if a criterion does not allow us to choose a unique best family it means that this criterion is not ultimate; we have to modify it until we come to a final criterion that will have that property.

The criterion must be reasonably invariant. The numerical value of the objective function $J(x)$ depends on what units we use to represent it. For example, in a traveling salesman problem, when we are minimizing the total length $J(x)$ of the route x , we can express this length in miles or in kilometers. If $J(x)$ is the length in miles, then the length in kilometers is $cJ(x)$, where $c = 1.6\dots$ is the number of kilometers in 1 mile. Suppose now that we first used miles, compared two different scaling functions $f(z)$ and $\tilde{f}(z)$, and it turned out that $f(z)$ is better (or, to be more precise, that the family $\Phi = \{Cf(z)\}$ is better than the family $\tilde{\Phi} = \{C\tilde{f}(z)\}$).

It sounds reasonable to expect that the relative quality of the two scaling functions should not depend on what units we used. So we expect that when we apply the same methods, but with

the lengths in kilometers, the results of applying f will still be better than the results of applying f' . But the fitness $F(x) = f(cJ(x))$ that results from applying f to the length in kilometers coincides with the result of applying a new scaling function $f_c(z) = f(cz)$ to the length in miles. So we conclude that if f is better than \tilde{f} , then f_c must be better than \tilde{f}_c , where $f_c(z) = f(cz)$ and $\tilde{f}_c(z) = \tilde{f}(cz)$. This must be true for every c because we could use not only miles or kilometers, but arbitrary units as well.

Another reasonable demand is related to the possibility of choosing different starting points for the objective function. For example, in solving a financial problem we can maximize either the net revenue $R(x)$ or the profit $P(x)$. The difference between them, crudely speaking, is that the profit is obtained by subtracting all the expenses from the revenue, i.e., if the expenses are fixed, $P(x) = R(x) - a$ for some constant a .

From a mathematical viewpoint, whether to maximize $R(x)$ or to maximize $P(x) = R(x) - a$ is the same problem. These two functions attain their maxima for the same individual x . Therefore, it is reasonable to expect that if a scaling function $f(z)$ is better than some function $\tilde{f}(z)$ when we apply them to revenues, it will still be better if we apply them both to profits. But the result $F(x) = f(R(x) - a)$ of applying $f(z)$ to $R(x) - a$ is equal to the result to applying a new function $f_a(z)$ to $R(x)$, where $f_a(z) = f(z - a)$. So we conclude that if $f(z)$ is better than $\tilde{f}(z)$, then the scaling $f_a(z)$ must be better than the scaling $\tilde{f}_a(z)$, where $f_a(z) = f(z - a)$ and $\tilde{f}_a(z) = \tilde{f}(z - a)$.

Now we are ready for the formal definitions.

3. DEFINITIONS AND THE MAIN RESULT FOR 1-DIMENSIONAL FAMILIES

Definitions. By a *scaling function*, we mean a smooth (differentiable) monotone function from real numbers into real numbers. We say that two scalings $f(z)$ and $\tilde{f}(z)$ are *equivalent* if $\tilde{f}(z) = Cf(z)$ for some positive constant C .

Comment. As we have already mentioned, if we apply a genetic algorithm with two equivalent scalings, we get the same result in both cases.

By a *1-dimensional family* of functions (or a *family* for short) we mean the set of functions $\{Cf(z)\}$, where $f(z)$ is a fixed scaling and C runs over all positive real numbers. The set of all 1-dimensional families will be denoted by S_1 .

A pair of relations $(<, \sim)$ is called *consistent* [Kreinovich 1990; Kreinovich, Kumar 1990; Kreinovich, Quintana 1991] if it satisfies the following conditions:

- (1) if $a < b$ and $b < c$ then $a < c$;
- (2) $a \sim a$;
- (3) if $a \sim b$ then $b \sim a$;
- (4) if $a \sim b$ and $b \sim c$ then $a \sim c$;
- (5) if $a < b$ and $b \sim c$ then $a < c$;
- (6) if $a \sim b$ and $b < c$ then $a < c$;
- (7) if $a < b$ then $b < a$ or $a \sim b$ are impossible.

Assume a set A is given. Its elements will be called *alternatives*. By an *optimality criterion* we mean a consistent pair $(<, \sim)$ of relations on the set A of all alternatives. If $b < a$, we say that a is *better* than b ; if $a \sim b$, we say that the alternatives a and b are *equivalent* with respect to this criterion. We say that an alternative a is *optimal* (or *best*) with respect to a criterion $(<, \sim)$ if for every other alternative b either $b < a$ or $a \sim b$.

We say that a criterion is *final* if there exists an optimal alternative, and this optimal alternative is unique.

Comment. In the present section we consider optimality criteria on the set S_1 of all families.

By the *result of adding a* to a function $f(z)$ we mean a function $f_a(z) = f(z + a)$. By the *result of adding a* to a family Φ we mean the set of the functions that are obtained from $f \in \Phi$ by adding a . This result will be denoted by $\Phi + a$. We say that an optimality criterion on S_1 is *shift-invariant* if for every two families Φ and $\tilde{\Phi}$ and for every number a , the following two conditions are true:

- i)* if Φ is better than $\tilde{\Phi}$ in the sense of this criterion (i.e., $\tilde{\Phi} < \Phi$), then $\tilde{\Phi} + a < \Phi + a$.
- ii)* if Φ is equivalent to $\tilde{\Phi}$ in the sense of this criterion (i.e., $\Phi \sim \tilde{\Phi}$), then $\Phi + a \sim \tilde{\Phi} + a$.

Comment. As we have already remarked, the demands that the optimality criterion is final and shift-invariant are quite reasonable. The only problem with them is that at first glance they may seem rather weak. However, they are not, as the following Theorem shows:

THEOREM 1. *If a 1-dimensional family Φ is optimal in the sense of some optimality criterion that is final and shift-invariant, then every function $f(z)$ from Φ is equivalent to $\exp(-\beta z)$ for some β .*

Comments. Thus we explain that the exponential scaling can be optimal.

(The proofs are given in Section 6).

Definitions. By a *result of a unit change* in a function $f(z)$ to a unit that is $c > 0$ times smaller we mean a function $f_c(z) = f(cz)$. By the *result of a unit change* in a family Φ by $c > 0$ we mean the set of all the functions that are obtained by this unit change from $f \in \Phi$. This result will be denoted by $c\Phi$. We say that an optimality criterion on S_1 is *unit-invariant* if for every two families Φ and $\tilde{\Phi}$ and for every number $c > 0$ the following two conditions are true:

i') if Φ is better than $\tilde{\Phi}$ in the sense of this criterion (i.e., $\tilde{\Phi} < \Phi$), then $c\tilde{\Phi} < c\Phi$.

ii') if Φ is equivalent to $\tilde{\Phi}$ in the sense of this criterion (i.e., $\Phi \sim \tilde{\Phi}$), then $c\Phi \sim c\tilde{\Phi}$.

THEOREM 2. *If a family Φ is optimal in the sense of some optimality criterion that is final and unit-invariant, then every scaling f from Φ is equivalent to $f(z) = z^\alpha$ for some α .*

Comments.

1. This Theorem explains the power law scaling.
2. By comparing the results of Theorems 1 and 2 one can conclude that a unit-invariant criterion cannot be shift-invariant; indeed, in this case we could apply Theorem 2, so f must be described by the power law. But all these functions are different from the exponential functions from Theorem 1, and so due to Theorem 1 this criterion is not shift-invariant.

So if we want our criterion to be both shift- and unit-invariant, we cannot restrict ourselves to 1-dimensional families of scalings, and we must consider multi-dimensional families instead. A natural way to define a finite-dimensional family of functions is to fix finitely many functions $f_i(z)$ and consider their arbitrary linear combinations $\sum_i C_i f_i(z)$.

4. DEFINITIONS AND THE MAIN RESULT FOR m -DIMENSIONAL FAMILIES

Definition. Let's fix an integer m . By a *basis*, we mean a set of m smooth, linearly independent functions $f_i(z)$, $i = 1, 2, \dots, m$. By an *m -dimensional family* of functions, we mean all functions of the type $f(z) = \sum_{i=1}^m C_i f_i(z)$ for some basis $\{f_i(z)\}$, where C_i are arbitrary constants. The set of all m -dimensional families will be denoted by S_m .

Comment. "Linearly independent" means that all these linear combinations $\sum_i C_i f_i(z)$ are different. If the functions $f_i(z)$ are not linearly independent, then one of them can be expressed as a linear combination of the others, and so the set of all their linear combinations can be obtained

by using, not the whole basis, but its subset consisting of less than m functions. From the well known algebraic fact that every linear space has a basis, we conclude that for any set of functions $f_i(z)$ the set of all linear combinations $\sum_i C_i f_i(z)$ either forms an m -dimensional family, or it forms a l -dimensional family for some $l < m$.

Our definitions of the optimality criterion, final criterion, shift-invariant and unit-invariant criteria can be applied to these families.

THEOREM 3. *If an m -dimensional family Φ is optimal in the sense of some optimality criterion that is final, shift- and unit-invariant, then this family coincides with the set of all polynomials $f(z) = a_0 + a_1 z + a_2 z^2 + \dots + a_{m-1} z^{m-1}$ of order $\leq m - 1$.*

Comments.

1. In particular, for $m = 2$ we conclude that optimal scalings must be linear $f(z) = az + b$. Thus this result explains why linear scaling turned out to be so efficient.
2. This Theorem, however, only gives the family from which we can choose an optimal scaling and does not explain how to choose a and b for a specific situation. Some methods of choosing these parameters are described in [Forrest 1985, Goldberg 1989].
3. This Theorem also tells what scalings to use if linear scalings are not sufficient: it is reasonable to use quadratic, cubic, and higher order polynomial scalings.

5. m -DIMENSIONAL FAMILIES: AUXILIARY RESULTS

If we consider m -dimensional families, but demand only shift-invariance, or only unit-invariance, then we get the following more general families of functions:

THEOREM 4. *If an m -dimensional family Φ is optimal in the sense of some optimality criterion that is final and shift-invariant, then every function $f(z)$ from Φ is equal to a linear combination of the functions of the type $z^p \exp(\alpha z) \sin(\beta z + \phi)$, where p is a non-negative integer, α , β and ϕ are real numbers.*

Comment. In particular, for $p = 0, 1, \alpha = \beta = 0$ we get linear functions; for $p = \beta = 0$ we get exponential functions, etc.

THEOREM 5. *If an m -dimensional family Φ is optimal in the sense of some optimality criterion that is final and unit-invariant, then every function $f(z)$ from Φ is equal to a linear combination of the functions of the type $(\ln z)^p z^\alpha \sin(\beta(\ln z) + \phi)$, where p is a non-negative integer, α , β and ϕ are real numbers.*

Comment. In particular, for $p = \beta = 0$ we get power law scalings, and if in addition we assume that $\alpha = 0$ or $\alpha = 1$, we get linear scalings.

6. PROOFS.

Proof of Theorem 1. The idea of this proof is as follows: first we prove that the optimal family is shift-invariant (in part 1), and from that, in part 2 we conclude that any function f from Φ satisfies a functional equation, whose solutions are known.

1. Let us first prove that the optimal family Φ_{opt} exists and is *shift-invariant* in the sense that $\Phi_{opt} = \Phi_{opt} + a$ for all real numbers a . Indeed, we assumed that the optimality criterion is final, therefore there exists a unique optimal family Φ_{opt} . Let's now prove that this optimal family is shift-invariant (this proof is practically the same as in [Kreinovich 1990] or [Kreinovich Quintana 1991]). The fact that Φ_{opt} is optimal means that for every other Φ , either $\Phi < \Phi_{opt}$ or $\Phi_{opt} \sim \Phi$. If $\Phi_{opt} \sim \Phi$ for some $\Phi \neq \Phi_{opt}$, then from the definition of the optimality criterion we can easily deduce that Φ is also optimal, which contradicts the fact that there is only one optimal family. So for every Φ either $\Phi < \Phi_{opt}$ or $\Phi_{opt} = \Phi$.

Take an arbitrary a and let $\Phi = \Phi_{opt} + a$. If $\Phi < \Phi_{opt} = \Phi_{opt} + a$, then from the invariance of the optimality criterion (condition *ii*) we conclude that $\Phi_{opt} < \Phi_{opt} - a$, and that conclusion contradicts the choice of Φ_{opt} as the optimal family. So $\Phi = \Phi_{opt} + a < \Phi_{opt}$ is impossible, and therefore $\Phi_{opt} = \Phi = \Phi_{opt} + a$, i.e., the optimal family is really shift-invariant.

2. Let us now deduce the actual form of the functions f from the optimal family. If $f(z)$ is such a function, then the result $f(z + a)$ of adding a to this function f belongs to $\Phi + a$, and so, due to 1., it belongs to Φ . But all the functions from Φ can be obtained from each other by multiplying by a constant, so $f(z + a) = C(a)f(z)$ for some C (this C depends on a). So we arrive at a functional equation for f . This equation is well known: it was first solved in [Pexider 1903], and its most general monotone solution is [Aczel 1966, Section 3.1.1] $f(z) = C \exp(-\beta z)$ for some C and β . Q.E.D.

Proof of Theorem 2. Just like in the proof of Theorem 1, we conclude that for every c there exists a $C(c)$ such that $f(cz) = C(c)f(z)$ for all z . This functional equation has also been solved in [Pexider 1903], and its solution is [Aczel 1966, Section 3.1.1] $f(z) = Cz^\alpha$ for some α . Q.E.D.

Proof of Theorems 3 – 5. Let's first prove Theorem 4. As in the proof of Theorem 1, we come to a conclusion that the optimal family Φ_{opt} exists and is shift-invariant. In particular, for every i the result $f_i(z + a)$ of shifting $f_i(z)$ must belong to the same family, i.e.,

$$f_i(z + a) = C_{i1}(a)f_1(z) + C_{i2}(a)f_2(z) + \dots + C_{im}(a)f_m(z)$$

for some constants C_{ij} , depending on a . Let us prove that these functions $C_{ij}(a)$ are differentiable. Indeed, if we take m different values z_k , $1 \leq k \leq m$, we get m linear equations for $C_{ij}(a)$:

$$f_i(z_k + a) = C_{i1}(a)f_1(z_k) + C_{i2}(a)f_2(z_k) + \dots + C_{im}(a)f_m(z_k),$$

from which we can determine C_{ij} using Kramer's rule. Kramer's rule expresses every unknown as a ratio of two determinants, and these determinants polynomially depend on the coefficients. The coefficients either do not depend on a at all ($f_j(z_k)$) or depend smoothly ($f_i(z_k + a)$) because f_i are smooth functions. Therefore these polynomials are also smooth functions, and so is their ratio $C_{ij}(a)$.

We have an explicit expression for $f_i(z + a)$ in terms of $f_j(z)$ and C_{ij} . So when $a = 0$, the derivative of $f_i(z + a)$ with respect to a equals to the derivative of this expression. If we differentiate it, we get the following formula: $f'_k(y) = c_{i1}f_1(z) + c_{i2}f_2(z) + \dots + c_{im}f_m(z)$, where by $c_{ij} = C'_{ij}(0)$ we denoted the derivative of $C'_{ij}(z)$ at $z = 0$. So the set of functions $f_i(z)$ satisfies the system of linear differential equations with constant coefficients. The general solution of such system is well known [Bellman 1970], so we get the desired expressions. Q.E.D.

Now let's prove Theorem 5. Just like in Theorems 1 and 4, we conclude that an optimal family exists and is unit-invariant. From this we conclude that for every i , the result $f_i(cz)$ of changing the unit for z must belong to the same family, i.e., $f_i(cz) = C_{i1}(c)f_1(z) + C_{i2}(c)f_2(z) + \dots + C_{im}(c)f_m(z)$ for some constants C_{ij} , depending on c .

This functional equation is almost the same as for shift-invariance (in the proof of Theorem 4), the only difference is that we have a product instead of a sum. In order to reduce it to the precious case, let's recall that if we turn to logarithms, then product turns into the sum: $\ln(ab) = \ln a + \ln b$ for all a and b . So let's introduce a new variable $Z = \ln z$ (so that $z = \exp(Z)$), and new functions $F_i(Z) = f_i(\exp(Z))$ (so that $f_i(z) = F_i(\ln z)$). Then for these new functions this functional equation takes the form $F_i(Z + A) = \bar{C}_{i1}(A)F_1(Z) + \dots + \bar{C}_{im}(A)F_m(Z)$. This is precisely the system of functional equations that we already know how to solve (see the proof of Theorem 4). So we can conclude that $F_i(Z)$ is a linear combination of the functions $Z^p \exp(\alpha Z) \sin(\beta Z + \phi)$ from the formulation of Theorem 4. When we substitute $Z = \ln z$, we conclude that

$$f_i(z) = F_i(Z) = F_i(\ln z)$$

is a linear combination of the functions $(\ln z)^p \exp(\alpha \ln z) \sin(\beta \ln z + \phi)$. Since

$$\exp(\alpha \ln z) = (\exp(\ln z))^\alpha = z^\alpha$$

, we get the desired expression for f_i . Q.E.D.

Let us finally prove Theorem 3. Since the optimality criterion is both shift- and unit-invariant, both Theorems 4 and 5 are applicable here. Therefore, for each of the functions f_i we have two different expressions obtained from the demands of shift-invariance and unit-invariance. When can a function f_i satisfy both conclusions, i.e., belong to both classes? If it contains terms with logarithms, it cannot be a linear combination of the functions from Theorem 4, because there are no logarithms among them. The same if it contains sines of logarithms. So the only case when a linear combination of the functions $(\ln z)^p z^\alpha \sin(\beta \ln z + \phi)$ is at the same time the linear combination of the functions $z^{\bar{p}} \exp(\bar{\alpha}z) \sin(\bar{\beta}z + \bar{\phi})$ is when $p = \beta = 0$. In this case the above expression turns into z^α , and from the equality of these expressions we conclude that $\alpha = \bar{p}$. But \bar{p} is necessarily a non-negative integer, and therefore α is non-negative integer as well. So $f_i(z)$, which is equal to a linear combination of such terms, is equal to the linear combination of the terms z^α for non-negative integers α , i.e., each of the functions $f_i(z)$ is a polynomial. Therefore, every function f from the optimal family, which is a linear combination of the polynomials, is a polynomial itself.

Let us now prove that every function $f(z)$ from the optimal family is a polynomial of order $m - 1$ or smaller. Suppose that f is an arbitrary polynomial from Φ and its order equals to p . Let's prove that $p \leq m - 1$.

The fact that f is of order p means that $f(z) = a_p z^p + \dots$, where $a_p \neq 0$. Since Φ is a set of all linear combinations, it contains $Cf(z)$ with any function $f(z)$. In particular, if we take $C = 1/a_p$, we conclude that Φ contains a polynomial $g_p(z) = z^p + a_{p-1}z^{p-1} + \dots$

Since the optimal family is invariant with respect to shifts (see the proof of Theorem 4) and contains $g_p(z)$, it must also contain $g_p(z + \varepsilon)$ for any real $\varepsilon > 0$. Since Φ is a set of all linear combinations, it is a linear space. In particular, this means that with any two functions it contains their difference $\bar{g}_p = g_p(z + \varepsilon) - g_p(z)$. Substituting the above expression for $g_p(z)$ into this expression for \bar{g}_p , we conclude that $\bar{g}_p(z) = ((z + \varepsilon)^p - z^p) + a_{p-1}((z + \varepsilon)^{p-1} - z^{p-1}) + \dots$. Each term $(z + \varepsilon)^k - z^k$, if we substitute the binomial expression for $(z + \varepsilon)^k$, turns into

$$z^k + k\varepsilon z^{k-1} + \dots - z^k = k\varepsilon z^{k-1} + \text{lower order terms}.$$

Therefore, the terms $(z + \varepsilon)^k - z^k$ with $k < p$ lead only to terms z^l with $l \leq k - 1 < p - 1$. The only term proportional to z^{p-1} stems from $(z + \varepsilon)^p - z^p$ and is equal to $p\varepsilon z^{p-1}$.

So $\bar{g}_p(z) = p\varepsilon z^{p-1} + \text{lower order terms}$. Similarly to the transition from $f(z)$ to $g_p(z)$ we can now divide $\bar{g}_p(z)$ by the coefficient $p\varepsilon$ at z^{p-1} and thus obtain a function

$$g_{p-1}(z) = z^{p-1} + b_{p-2}z^{p-2} + \dots$$

that also belongs to Φ .

So from the fact that Φ contains a polynomial $g_p(z) = z^p + \dots$ whose expansion starts with z^p we concluded that Φ contains a function $g_{p-1}(z) = z^{p-1} + \dots$ whose expression starts with z^{p-1} . Applying the same arguments to $g_{p-1}(z)$, we conclude that Φ contains a function $g_{p-2}(z) = z^{p-2} + \dots$, which, in its turn, implies that Φ contains functions

$$g_{p-3}(z) = z^{p-3} + \dots, g_{p-4}(z) = z^{p-4} + \dots, \dots, g_k(z) = z^k + \dots, g_1(z) = z^1 + \dots, g_0(z) = z^0 = 1.$$

So we found $p + 1$ functions $g_k(z)$ in Φ .

Let's prove that these functions $g_k(z)$ are linearly independent. Linear independence means if a linear combination $C_0g_0(z) + C_1g_1(z) + \dots + C_pg_p(z)$ is equal to 0, then all the coefficients C_i are equal to 0. Indeed, let's substitute the above expressions for $g_k(z)$ into the equation $C_0g_0(z) + C_1g_1(z) + \dots + C_pg_p(z) = 0$. The polynomial on the left-hand side of this equation is identically 0, therefore all its coefficients must be equal to 0. Since each of $g_k(z)$ is a polynomial of k -th order, the highest term in the left-hand side is proportional to z^p or to a smaller power of z . The only term proportional to z^p in the left-hand side can stem from $g_p(z)$, and since $g_p(z) = z^p + \dots$, the coefficient at z^p must be equal to C_p . But on the other hand, the coefficient at z^p in the left-hand side must be 0, therefore $C_p = 0$. If we substitute $C_p = 0$ into the this equation, we conclude that $C_0g_0(z) + C_1g_1(z) + \dots + C_{p-1}g_{p-1}(z) = 0$. Similar arguments now lead to $C_{p-1} = 0$, and consequently to $C_{p-2} = c_{p-3} = \dots = C_k = \dots = C_1 = C_0 = 0$. So all the coefficients are 0, and therefore the functions $g_k(z)$ are linearly independent.

So in an m -dimensional linear space Φ there are $p + 1$ linear independent elements $g_0(z), g_1(z), \dots, g_p(z)$. Since in an m -dimensional space there can be at most m independent elements, we conclude that $p + 1 \leq m$, hence $p \leq m - 1$. So every polynomial from Φ is indeed of order $\leq m - 1$.

Therefore Φ is an m -dimensional linear subspace of an m -dimensional linear space of all the polynomials of order $\leq m - 1$. But any m -dimensional linear space has only one m -dimensional linear subspace: itself. So Φ coincides with the set of all polynomials of order $\leq m - 1$. Q.E.D.

Acknowledgments. This research was supported by NSF grant No. CDA-9015006, NASA Research grant NAG 9-482 and a grant from the Institute for Manufacturing and Materials Management. One of the authors (V.K.) is thankful to Patrick Suppes (Stanford) and Larry Shepp (AT& T Bell Labs) for discussing the relevant mathematics.

REFERENCES

- Aczel, J. *Lectures on functional equations and their applications*. Academic Press, NY-London, 1966.
- Baker, J. E. *Adaptive selection methods for genetic algorithms*, Proceedings of an International Conference on Genetic Algorithms and their Applications, Pittsburgh, PA, 1985.
- Bellman, R. *Introduction to matrix analysis*. McGraw-Hill, N. Y., 1970.
- Bremermann, H. J. *Optimization through evolution and recombination*, In: Yovits, Jacobi and Goldstein (eds.) *Self-Organizing Systems*, Spartan Books, Washington, D. C., 1962.
- Bremermann, H. J., M. Rogson, and S. Salaff. *Global Properties of Evolution Processes*, In: H. H. Pattee, E. A. Edelsack, Luois Fein and A. B. Callahan (eds.) *Natural Automata and Useful Simulations*, Spartan Books, Washington, D.C., 1966.
- Davidor, Y. *Genetic algorithms and robotics: A heuristic strategy for optimization*. World Scientific, Singapore, 1991.
- De Jong, K. A. *An analysis of the behavior of a class of genetic adaptive systems*, Doctoral Dissertation, University of Michigan. *Dissertation Abstracts International*, Vol. 44, No. 10, p. 3174B. University Microfilms No. 8402282.
- Forrest, S. *Documentation for PRISONERS DILEMMA and NORMS programs that use the genetic algorithm*, Unpublished manuscript, University of Michigan, Ann Arbor, MI, 1985.
- Gillies, A. M. *Machine learning procedures for generating image domain feature detectors*, Doctoral Dissertation, University of Michigan, Ann Arbor, MI, 1985.
- Goldberg, D. E. *Simple genetic algorithms and the minimal deceptive problem*, University of Alabama, Tuscaloosa, The Clearinghouse for Genetic Algorithms, TGGA Report No. 86001, 1986.
- Goldberg, D. E. *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, MA, 1989.
- Holland, J. *Outline for a logical theory of adaptive systems*, Journal of the Association of Computing Machinery, 1962, Vol. 3, pp. 297–314.
- Holland, J. *Some practical aspects of adaptive systems theory*, In: A. Kent and O. E. Taulbee, Editors, *Electronic Information Handling*, Spartan Books, Washington, DC, 1965, pp. 209–217.

Holland, J. *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, MI, 1975.

Hopfield, J. J. *Neural networks and physical systems with emergent collective computational abilities*, Proceedings of the National Academy of Sciences, 1982, Vol. 79, pp. 2554–2558.

Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi, *Optimization by simulated annealing*, Science, Vol. 220, May 1983, pp. 671–680.

Kreinovich, V. *Group-theoretic approach to intractable problems*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Vol. 417, 1990, pp. 112–121.

Kreinovich, V. and S. Kumar, *Optimal choice of $\&$ - and \vee - operations for expert values*, Proceedings of the 3rd University of New Brunswick Artificial Intelligence Workshop, Fredericton, New Brunswick, Canada, 1990, pp. 169–178.

Kreinovich, V. and C. Quintana. *Neural networks: what non-linearity to choose*, Proceedings of the 4th University of New Brunswick Artificial Intelligence Symposium, Fredericton, New Brunswick, 1991, pp. 627–637.

Pexider, J. V. *Notiz uber Funktionaltheoreme*, Monastch. Math. Phys., 1903, Bd. 14, S. 293–301.

Romeo, F. and A. Sagniovanni-Vincetelli. *A theoretical framework for simulated annealing*, Algorithmica, 1991, Vol. 6, No. 3, pp. 302–345.

Sirag, D. J. and P. T. Weisser, *Toward a unified thermodynamic genetic operator*, in *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, Cambridge, MA, 1987, Lawrence Elbaum publ., Hillsdale, NJ, 1987.