Beyond Encoding: Rotationally Invariant Operators for Evolutionary Computation

Arnold L. Patton
pattona@cse.msu.edu

Erik D. Goodman
goodman@egr.msu.edu

William F. Punch III
punch@cse.msu.edu

Genetic Algorithms Research and Applications Group (GARAGe)
Dept. of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
http://garage.cse.msu.edu

Abstract

Ideally the character of an evolutionary search algorithm should be predicated on the shape of the underlying search landscape. Unfortunately, a number of operators used in evolutionary computation are very sensitive to rotation of the encoding space. The potential bias of the orientation of the encoding for these operators indicates that both the alignment of the encoding as well as the search problem influence the performance of search algorithms using these operators. This paper explores two potential mechanisms for removing rotational bias in a recombinative operator (2-point crossover) as well as a mutative operator as applied to real-encoded function optimization.

1 INTRODUCTION

The majority of operators utilized in evolutionary computation are applied along the axes of the original encoding. That is, the basis set that determines the alignment of the probability density of the offspring for a given operator is typically the original axes of encoding. This is especially true of operators which treat individual solution parameters, or *fields*, as non-decomposable or *atomic* entities, such as those applied to real-valued functions; though with certain limitations it also applies to lower level operators, such as bit-wise operators.

The alignment of the probability density of certain axially aligned atomic (AAA) operators varies greatly in respect to the current population density when the population shows strong correlation between solution parameters. We label such operators rotationally biased AAA operators (RB-AAA). If we assume that, for a system with reasonable diversity, the population distribution is predicated on the shape of the landscape, the inconsistency in the search character of these operators under rotation implies that the effectiveness of the search is directly linked to the alignment of the selected encoding rather than only the shape of the underlying landscape. That is, these operators are not rotationally invariant.

One fairly obvious solution to this problem would be to apply the operators using a rotated basis set. That is, the selected individuals of the population are rotated into the new basis and then the operator is applied to these rotated individuals after which the offspring are counter-rotated and returned to the original encoding space. However, *a priori* selection of a consistent rotation is not possible without prior knowledge about the shape and alignment of the problem space.

If we accept the proposition that under fairly common situations in evolutionary computation that the population distribution is largely predicated on the shape of the landscape, then we may be able to use the current distribution of the population to induce a consistent alignment for the landscape. Such an approach has the dangers of being predicated on the diversity of the population, as well as being recursive in nature. (The distribution of the population determines the distribution of the offspring which directly influences the distribution of the population for the next generation, etc.) However, these same limitations apply to all recombinative operators.

If we do not accept the proposal that the form of the population follows the landscape, it may still be possible to remove the rotational bias of RB-AAA operators by using uniformly randomly selected rotations for successive operator applications. This approach would tend to average out the rotational biases over large numbers of operator applications; however, the number of operator applications required to reach the same level of rotational sampling increases exponentially with the number of degrees of freedom of the problem space. Thus we might expect this approach to become less stable linearly as population size or application rate decreases and exponentially as the problem size increases.

This paper presents the results of several experiments in attempting to reduce the rotational bias of crossover and a form of AAA mutation. First, the rotational bias of the axially aligned forms of these operators is examined, after which the rotationally invariant variants are presented. Next, the experimental method is outlined and results are presented. Finally, conclusions are drawn from the results of the experiment.

2 OBSERVATIONS

Numerous recombinative operators have been proposed and examined in evolutionary computation. The classical form of recombination in genetic algorithms is crossover, where individual fields (normally bits) of two contributing parents are separated and recombined, creating two new target solutions. Other popular forms of recombination include BLX-α and similar variants (for examples, see [Eshelman 93], [Ono 97]) which place offspring near the bounding box of the parents or near the line between the parents. Intermediate recombination and dominant recombination are two common ES operators [Salomon 98]. Intermediate recombination reinitializes the population around the current population mean, while dominant recombination samples individual parameters from multiple parents (with the addition of noise). Each of these operators has a unique geometric interpretation in the search space and each has the potential for certain biases.

For this study, we choose to restrict or analysis to the standard GA approach to crossover, since it has a large rotational bias which is well known [Salomon 96]. Further, we choose to treat individual solution parameters atomically in order to sidestep artifacts of binary operators when used with parameters represented as IEEE reals (such as the high attraction toward zero).

In general, we introduce the concept of operator bias as being the ratio of local as opposed to global exploration. Local exploration is defined on a population level as exploration near an existing member of the population (not necessarily near the progenitors or a given solution). Thus, we may use the following expectation as a *global search bias* (*GSB*) metric:

$$E\left[\sqrt{\sum_i (x_i - y_i)^2}\right]$$

where x is the offspring produced by a given operator, and y is the nearest point to x in the parent generation.

Note that the GSB value for a given operator does not reflect on the appropriateness or intrinsic value of a search operator; it merely reflects the basic bias of the operator from a population oriented framework. Creating a proper balance between local and global exploration is one of the open problems of evolutionary computation. (Indeed, the NFL theorems imply that no single algorithm is capable of creating such a "proper" balance for all possible search problems.) Though the relative benefit of high or low local search bias may be debated, ideally the GSB of an operator for a given population distribution should be rotationally invariant. That is, it should remain the same regardless of the actual linear encoding used. (Note that non-linear encodings effectively warp the underlying search space when viewed from the resultant encoding, and thus are much closer to changing the search space. Throughout this paper, alternate encodings refers to alternate *linear* encodings.)

We may define the *rotational bias* of a given operator as the variance of the GSB across all possible rotations. Rotationally invariant operators will have zero rotational bias.

While the GSB and rotational bias metrics are useful conceptual tools, both are difficult to calculate for many operators since even calculating the probability density for a given operator for an entire population is often a very complex operation. Nonetheless, it is often possible estimate if a rotational bias exists (i.e. is non-zero) through inspection of the operation of the operator on rotated strongly covariant populations.

2.1 POPULATION LEVEL EFFECTS OF FIELD BASED CROSSOVER

In examining the effects of atomic GA crossover (as opposed to bit-wise crossover), the most common approach evaluates the geometric interpretation of the operator. If we create an axially aligned hypercube with two of the diametrically opposed corners occupied by the

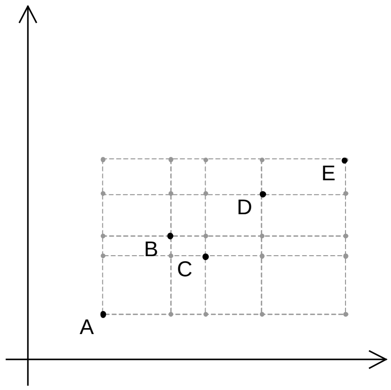


Figure 1. Potential child locations from a standard crossover operation.

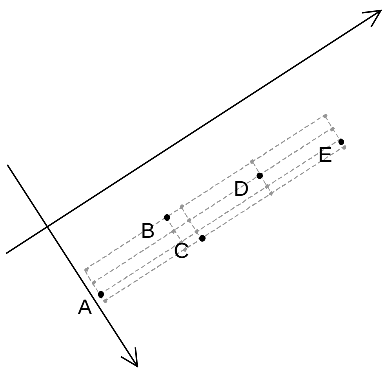


Figure 2. Potential child locations from crossover using GSB minimizing rotation.

original parents, the crossover operator chooses offspring at some pair of diametrically opposing corners of this hypercube (potentially even the original parent solutions). [Holland 92] originally proposed this evaluation of crossover. A two-dimensional graphic representation of a sample crossover situation is demonstrated in figure 1. The potential offspring are located at any of the intersection points on the grid shown.

2.1.1 Covariance Reduction and Assumptions of Parameter Independence

As an alternate form of analysis, consider the statistical effects of crossover on the distribution of individuals in the search space. Crossover exchanges values between two candidate solutions in a population, producing two new solutions, and placing the new offspring into the next generation. Note that since no individual parameter values are modified or removed, the mean of the new solutions must be the same as the mean of the original solutions. In fact, if we consider the resulting population in the next generation (in the absence of other operators), the population mean for each parameter remains exactly the same regardless how many times crossover is employed. The same principal also holds true for the variance of each parameter.

Crossover neither affects the mean nor the variance of individual parameters. However, it strongly affects the covariance between parameters. On average crossover negates (reverses the sign) of half of the existing parameter covariance between two independently random parent solutions. This remains true for one-point, two-point, as well as uniform crossover. (However, the assumption of independently random parents fails over time; thus, the actual average amount of covariance disturbance typically decreases as the search progresses. Also, the level of independence between parents over time remains higher as the number of crossover points increases.) Since the negated covariance tends to offset the covariance of the original parents (or similarly correlated solutions), the net result is to reduce the overall level of parameter correlation in the population. If the population is currently strongly correlated, crossover will tend to reduce the level of correlation in the next population. Viewed another way, the net effect of crossover is always to assert the independence of solution parameters, since independent parameters have zero covariance. This being the case, we might expect crossover to work better for problems where parameters operate independently, and not as well in situations which require large amounts of linkage between parameters. This does not preclude the possibility of crossover having positive effects even with highly correlated problems; however, it does indicate that the level of parameter correlation will directly affect the character of crossover. Empirical evaluation of the effects of rotation on crossover in a breeder genetic algorithm (BGA), which is essentially uniform field-based crossover, is examined in [Salomon 96], with similar conclusions.

2.1.2 Correlated vs. Independent Parameter Encoding

The contrast between the effects of crossover on populations that exhibit strong parameter correlation is demonstrated graphically in figures 1 and 2. In figure 1., the current population exhibits a strong correlation along the line X = 2Y for the chosen encoding parameters. In figure 2., the same population is shown; however, the chosen encoding is rotated from that in Figure 1. Neither the shape of the problem landscape, nor the distribution of the population have changed between figures 1 and 2, yet the distribution of points available through crossover changes dramatically. In particular, the area over which the offspring are distributed is much larger in figure 1 than that demonstrated in figure 2. The level of parameter correlation under the chosen parameter encoding directly affects the GSB of crossover. This is especially troubling given that the shape of the landscape to be examined is usually unknown, let alone the level of parameter correlation. In general, it is probably undesirable for an operator to be strongly dependent on the alignment of the original encoding.

2.2 FIELD-BASED MUTATION

The form of mutation used by evolutionary programming treats each parameter as a single entity. Individual fields are mutated according to a given (usually strongly central) probability distribution such as the Normal, Cauchy, or Laplace distribution. [Yao 97] All fields are mutated simultaneously to produce a single offspring. In general, such a direct statistical approach toward mutation is more easily amenable to evaluation when encoding parameters as floating point values than the typical bit-wise mutation under the standard genetic algorithm approach. Further, since the form of crossover under study also treats each parameter as an indivisible atomic entity, the form of mutation used on the individual fields should not be at issue. (Although whether it is better to simultaneously mutate multiple fields or individual fields between a parent and offspring in the presence of crossover is unknown.)

Typical evolutionary programming approaches evolve the variance parameters for the size of the mutation simultaneously with the parameter values for a given solution. [Bäck 93] Each individual encodes both the current parameter value as well as the base mutation size for each parameter. Both the parameter value as well as the mutation value for that parameter is mutated for each offspring.

Previous research [Patton 98] has shown that good results when using field-based crossover may be obtained over a number of test problems by using population sampled variance estimates to drive the size of the mutation. Under this approach, a sample or *cohort* from the current population is selected (without replacement), and the variance along each parameter is calculated. These variances are then used as the variance for the subsequent mutation. This *cohort driven* operator is dubbed Guided Gaussian Mutation (GGM). This is the form of mutation we have selected for use in this study, since it is easily adaptable to application under rotation.

2.2.1 Mutation hyper-ellipsoid is axially aligned

Regardless of the manner in which the variance for the mutation operator is obtained, the shape of the probability density surrounding the parent will always be a hyper-ellipsoid that is axially aligned. The GSB of the search conducted by such a mutation, as with crossover, varies with the level of parameter correlation in the population under the given encoding. As parameter correlation increases, the GSB of an axially aligned hyper-ellipsoid of mutation increases in respect to a hyper-ellipsoid of mutation that is aligned along the axes of highest correlation.

3 USING ROTATED AXES FOR OPERATOR APPLICATION

One possible solution in overcoming the bias introduced by the alignment of the chosen encoding axes with the population distribution is to use rotated axes for operator application. However, selection of an appropriate basis set may be difficult. This section offers two proposals for selection of appropriate axes for operator application. For the first, axes are chosen on a continuous random basis. The second uses partial analysis of the statistical distribution of the population as an estimate of the tightest fit basis set.

3.1 FREELY ROTATED CROSSOVER AND MUTATION

Continuously reselecting random basis sets for operator application may negate or average out any alignment bias. Of course, such an approach is expected to degrade performance over standard operators when the initial encoding is correctly aligned with the problem landscape (i.e. when the parameters are independent under the original encoding).

3.1.1 Crossover becomes hyper-spherical

Geometrically, randomly rotated crossover (RRX) locates children on the n-dimensional hypersphere that is centered at the mean between the parents and intersects both parents. Note that since this is essentially the union of the offspring produced under all possible rotations, this is also the *a priori* distribution of offspring from crossover when the alignment of the encoding is unknown. That is, without knowledge of the encoding, we know that the outcome of a crossover between two individuals will be somewhere on this hypersphere.

Unlike standard crossover applied across the axes of encoding, RRX affects both the variance as well as the covariance of the population. However, the total variance in the population remains constant. In other words, the RRX allows the variance in the population to shift between parameters.

Randomly rotated GGM (RR-GGM) is much more difficult to analyze geometrically since the probability density of all mutations for the population depend both on the placement of individuals in the population as well as the amount of variance in the total population along each

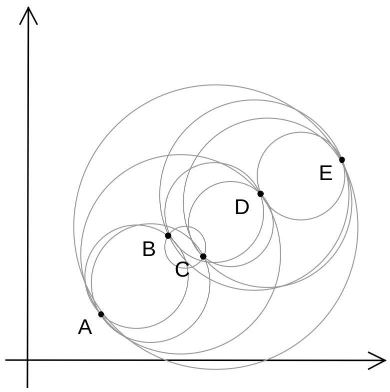


Figure 3. Potential child locations from freely rotated crossover.

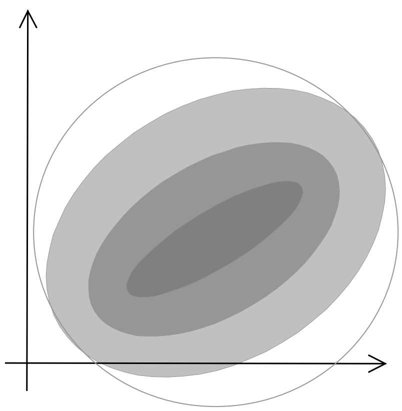


Figure 4. Potential child locations from freely rotated mutation.

chosen axis of the rotated bases. In general, the shape of the distribution will be a hypersphere with the highest density tending to run parallel to the vector of greatest variance through the population, and the lowest density along the outer edges of the vector of lowest variance. Figure 4 shows a rough estimate of the probability density of RR-GGM when the population is normally distributed with maximal variance along the line X = 2Y.

3.2 USING POPULATION DISTRIBUTION TO ESTIMATE APPROPRIATE ROTATION

Randomly rotated axes for operator application may offer an opportunity to reduce, or at least normalize, any alignment bias. However, ideally we would like to be able to select a *correctly* aligned basis set, or at least one with known properties, such as minimizing the GSB (i.e. reducing the area of distribution for the offspring). An optimal rotation would be one which aligns the longest dimension of the hyper-ellipsoid with the dimension having the largest variance for the population, and which aligns the second longest dimension of the hyper-ellipsoid with the dimension having the next largest variance which is also orthogonal to the dimension already selected, and so on. Note that this basis set may be located by solving for the eigenvectors of the covariance matrix and ordering the eigenvectors in descending order by the absolute value of their associated eigenvalue. This procedure is identical to that used in creating principal component projections for data viewing. [Jain 88]

3.3 PRINCIPAL COMPONENT PRESERVING CROSSOVER

Principal Component Preserving Crossover (PCPC or PC^2) uses the eigenvector solution of the covariance of a sample of the population to specify a rotated basis under which to apply crossover. The sample is chosen uniformly without replacement from the population. After a sample is selected, the covariance matrix, and the eigenvalues and eigenvectors of the covariance are calculated. The eigenvectors of the covariance matrix are sorted in descending order by their associated eigenvalue. The parent solutions are selected as usual, and are rotated by multiplication with the sorted eigenvector matrix. Standard field-based crossover (one-point, two-point, or uniform crossover) is then applied to the rotated parent solutions. The resulting offspring are counter-rotated by multiplication with the inverse of the sorted eigenvector matrix (which is the same as the transpose of the sorted eigenvector matrix), and placed into the population for the following generation. The covariance may be resampled as often or seldom as desired; however, more frequent sampling should result in overall better agreement between the estimated and actual variance alignments in the population.

In many ways, the operation of PC^2 resembles the concept of inversion. Inversion is an operator that attempts to reorder the parameters on the chromosome in order to modify the probabilities of preserving certain relationships or *linkages* among the parameter. Typically, inversion is used with 1-point or 2-point crossover. [Goldberg 89] PC^2 also attempts to preserve existing relationships within the population; however, the focus is on preserving linear relationships within the encoding space. Similarly parallels may be drawn between the proposed operators and delta coding [Mathias 94] in that both have the ability to modify the representation as the search proceeds; however, unlike delta coding, the encoding may be dynamically reselected for each operator application.

3.4 PRINCIPAL COMPONENT GUIDED GAUSSIAN MUTATION

Principal Component Guided Gaussian Mutation (PC-GGM) like PC^2 uses the eigenvector solution of the covariance of a sample, without replacement, of the population to specify a rotated basis under which to apply the mutation. The eigenvectors of the covariance matrix are used to rotate the selected parent solution. Each parameter in the rotated solution is mutated by addition of a sample from the normal distribution with a variance equal to the eigenvalue associated with the eigenvector that produced the given rotated parameter. The mutated offspring is then counter-rotated by multiplication with the inverse of the eigenvector matrix (the transpose of the eigenvector matrix), and placed into the population for the following generation.

4 EMPIRICAL EVALUATION OF SELECTED ROTATED OPERATORS

The effectiveness of both the randomly shifted basis selection algorithm and population sampled covariance rotation algorithm have been tested on several common real-valued function optimization problems. These functions are not intended as a realistic domain for application of these techniques, but rather as a simplified test-bed, which will hopefully allow some fruitful comparisons to be drawn. Several operator combinations were selected for testing as outlined below. Results for these search systems were then obtained for multiple runs, both with and without rotation of the standard encoding axes for each given optimization problem.

4.1 SELECTED ALGORITHMS

In an effort to evaluate both the effectiveness of rotation of operator axes as well as the effectiveness of using the sampled covariance matrix to derive the operator axes, several operator combinations were tested. As a baseline system, a GA using tournament selection with tournament size 2, 2-point field-based crossover, and guided Gaussian mutation (GGM) was selected, using a crossover rate of 0.8 and a mutation probability of 0.2 per individual (as opposed to the typical bit-wise probabilities). The second system tested uses randomly rotated 2-point crossover (RRX-2), and randomly rotated GGM (RRGGM). The third system tested used cohort sampling to estimate the population covariance to calculate the axes of rotation. This system used both principal component preserving crossover (PCPC) and principal component GGM (PCGGM). A final variant was used to measure the impact of crossover. This last system employed only PCGGM. For the first three variants, the crossover and mutation application rates were held constant at 0.8 and 0.2 per selected individual respectively. For the mutation only system, the mutation rate remained 0.2, and the crossover rate was effectively zero.

4.2 TEST PROBLEMS

Three representative problems were selected for these experiments. First, the simple sphere function was selected, since rotating the encoding space should not modify the apparent search landscape. Second, Rosenbrock's banana was selected due to its non-symmetric, highly covariant, and unimodal characteristics. Finally, Griewank's optimization function was selected as a representative difficult multimodal function. The formulae for these functions are given in Table 1.

Table 1: Selected test problems

Function	Name	Initial Range
$\sum x_i^{\,2}$	Sphere	$-100 \le x_i \le 100$
$1+\sum \frac{x_i^{\,2}}{4000} - \prod\left[\cos\left(\frac{x_i}{\sqrt{i}}\right)\right]$	Griewangk	$-600 \le x_i \le 600$
$\sum_{i=1}^{n-1}[100*(x_i^{\,2}-x_{i+1})+(1-x_i)^2]$	Rosenbrock	$-15 \le x_i \le 15$

Certain operators have an affinity for the encoding origin, especially when IEEE reals are being used. In order to avoid such encoding artifacts, the standard target for each test function was shifted +10.0 in each dimension. (Shifting was applied before rotation.) Note that while it is possible that this shift may cause some rounding error, or limit the precision of the answer, such effects remain constant across all tested systems. Since initialization of the population remained centered about the encoding origin, as opposed the shifted function origin, this shifting also provides some assurance against artifacts caused by center tending operators.

4.3 TESTING PROCEDURE

The performance of each system was measured over 100 trials using rotated encodings, and 30 trials using the standard, non-rotated, encoding. Each trial used a population size of 500 and continued for 1000 generations. The population best value and number of evaluations over the rotated and non-rotated trials was recorded and averaged for each system.

Rotations were determined by selecting $\mathbf{d^2/2}$ 2-dimensional rotations randomly, where **d** is the number of

Figure 5. : Sphere function (10 Dimensions)

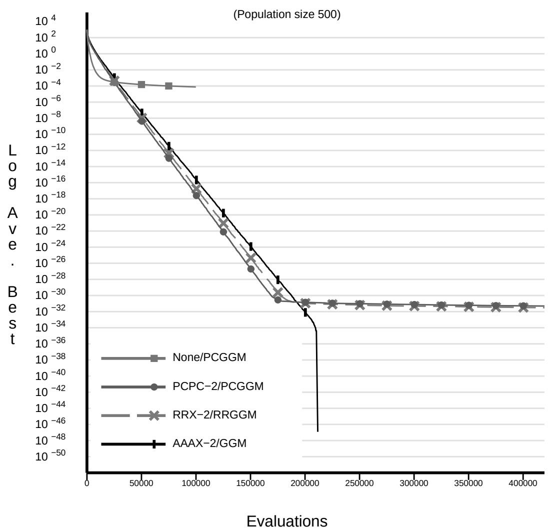


Figure 6. : Rotated Sphere function (10 Dimensions)

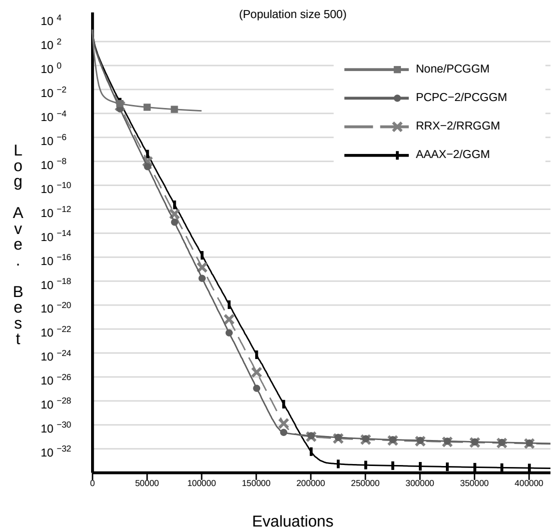


Figure 7. : Rosenbrock function (10 Dimensions)

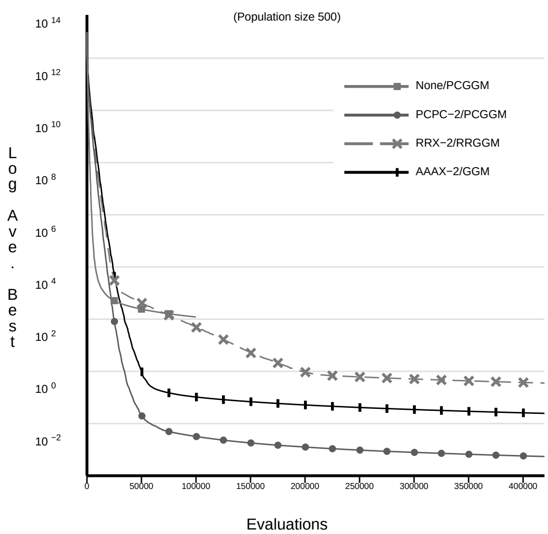


Figure 8. : Rotated Rosenbrock function (10 Dimensions)

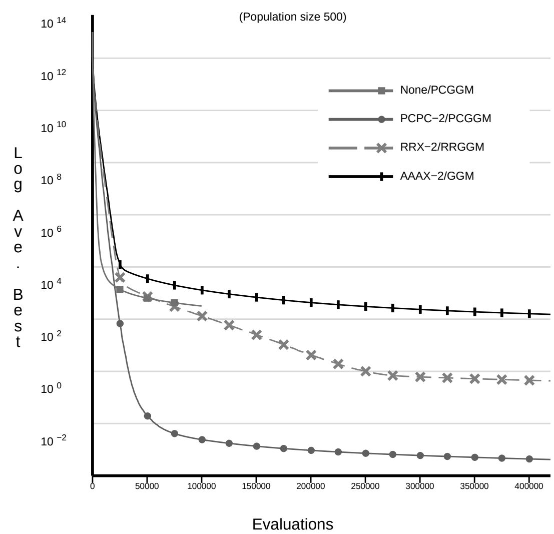


Figure 9. : Griewangk function (10 Dimensions)

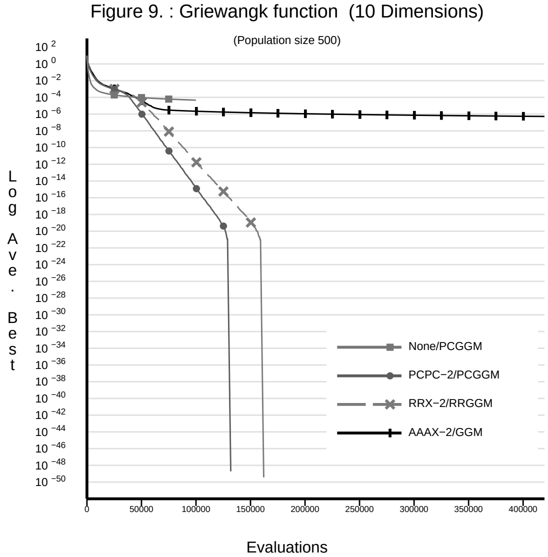
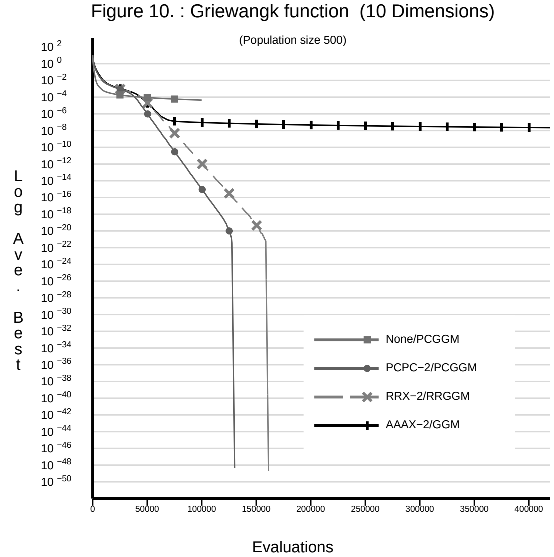


Figure 10. : Griewangk function (10 Dimensions)



dimensions in the original encoding space of the problem function. For each 2D rotation, an originating axis, target axis, and rotation angle are selected, and the rotation matrix for rotating the originating axis toward the target axis while holding all other axes fixed is calculated. Successive 2D rotation matrices are multiplied, thereby producing the final rotation matrix. The selected rotation remained fixed throughout a single run. During evaluation of an individual, the raw parameters encoded by the individual are multiplied by the rotation matrix, after which the origin offset is applied (by subtracting 10.0 from each parameter). The resulting values are passed to the objective function for evaluation. The net effect is an apparent translation and rotation of the problem space from the point of view of the search engine.

All systems were tested on the same sets of random encoding space rotations. Every system was tested on ten rotation matrices. Rotation matrices were held constant for ten runs for each system. Reported averages are for all 100 rotated test runs.

4.4 RESULTS

Figures 5, 7, and 9 show the results for the average best function value of the four tested systems over the three test functions without rotation of the encoding space. Figures 6, 8, and 10 show the average best function value for the rotated test cases. The tested systems used the following operators: 2-point field-based crossover and guided Gaussian mutation (AAX-2/GGM), randomly rotated 2-point crossover and randomly rotated GGM (RRX-2/RRGGM), principal component preserving crossover and principal component GGM (PCPC/PCGGM), and PCGGM alone.

5 CONCLUSION

In analysis of the results as presented here, several trends are evident which will be examined in greater depth in this section. First, it can be noted that the system which employed operators which are applied solely along the presented axes of encoding showed marked degradation when the encoding space was rotated, while the performance of the others was remarkable unperturbed. Second, it appears that in addition to being rotationally invariant, the system using principal component preserving operators actually slightly outperformed the other systems even under non-rotated circumstances.

The issue of computational cost for the proposed operators is also addressed. Finally, some attempt is made to address the implications of this research on the broader topic of encoding selection.

5.1 ROTATIONAL INVARIANCE

By comparing the performance of the various systems on identical rotated and non-rotated landscapes, it is apparent that the system employing the standard operator forms suffers a degradation of search quality when the landscape is rotated. For both the Rosenbrock and the Griewangk functions, the system using only axially aligned operators (AAX-2/GGM) consistently converged to an answer several orders of magnitude worse when the encoding space was rotated. In contrast, all systems that employed operators that were not limited to the given encoding showed remarkably consistent results between rotated and non-rotated tests. As expected, none of the systems showed any marked change under rotation with the rotationally invariant sphere function. (Although the changes of behavior at the actual point of convergence are interesting.)

5.2 IMPROVED SEARCH QUALITY

An interesting side effect of the use of rotated operators appears to be improved performance for certain functions. The contrast between the use of rotated operators on the Griewangk function is consistent and especially large, though the reason behind this radical change in behavior is not currently known.

Additionally, we observe that the use of principle component preserving operators appears to improve search quality for both the Rosenbrock and the Griewangk functions even in non-rotated test cases. A possible explanation is that the increased focus on maximizing population local search causes an increase in the hill-climbing capacity of the system. If such is the case, then this distinction in behavior is not very interesting, since a simpler hill-climbing agent could perform the same task with less cost.

A more favorable explanation of the increased behavior would be that the population distribution does indeed indicate productive search directions. However, without further analysis the current data does not necessarily allow us to reach such a conclusion.

5.3 COMPUTATIONAL COST

One of the drawbacks to measuring the population covariance is the computational cost. The cost of calculating the covariance for a sample size **n** of **d** dimensional individuals is $\mathbf{O(nd^2)}$. Using routines from [Press 92], calculating the eigenvectors and eigenvalues of a **dxd** matrix costs $\mathbf{O(d^3)}$. If we resample often, the cost can become quite large as **d** increases. Therefore, this technique is not a good candidate for use when the evaluation of an individual solution is relatively inexpensive, as is usually the case in real-valued function optimization. However, as the cost of evaluation rises, the cost of sampling the covariance becomes less prohibitive.

The use of randomly rotated axes also demonstrated rotational invariance and improved performance on the Griewangk function. Computing random rotational transformations is relatively simple and requires only $\mathbf{O(d^2)}$ operations to complete. Therefore, this approach could be used even with fairly inexpensive cost functions; however, it does show less favorable performance than the use of standard axially aligned operators under certain landscape alignments as can be witnessed through examination of Figure 9.

5.4 BEYOND ENCODING

On of the persistent difficulties in evolutionary computation, especially when using encoding sensitive operators, as is typically the case with genetic algorithms, is locating an appropriate encoding which will facilitate the search process. This is as well as it should be, since as [Wolpert 97] points out, the success of a particular search algorithm is predicated to a great extent as to whether the assumptions made by the particular system in question align well with the actual landscape. However, this implies a great deal of *a priori* information on the part of the problem encoder. To the extent that we are unable to know how a given encoding will modify the presentation of a given landscape, the development of transformationally invariant operators may allow us to move beyond the encoding selection process for transformationally equivalent encodings.

References

Mathias, K. E. and L. D. Whitley (1994). "Changing representation during search: a comparative study of delta coding," *Evolutionary Computation*, Vol. 2., No. 3, pp. 249-278.

Eshelman, L. J. and J. D. Schaffer (1993) . "Real-coded genetic algorithms and interval-schema," *Foundations of Genetic Algorithms 2*, Morgan Kaufmann, pp. 187-202.

Salomon, R. (1998). "Short notes on the schema theorem and the building bock hypothesis in genetic algorithms," *Evolutionary Programming VII*, Springer-Verlag, pp. 113-122.

Salomon, R. (1996). "Performance degradation of genetic algorithms under coordinate rotation," *Evolutionary Programming V*, MIT Press, pp. 155-169.

Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems*, 2nd Edition. MIT Press.

Yao, X., G. Lin and Y Liu, (1996). "An analysis of evolutionary algorithms based on neighborhood and step sizes", *Evolutionary Programming VI*, Springer-Verlag, pp. 297-307.

Ono, I., and S. Kobayashi, (1997). "A real coded genetic algorithm for function optimization using unimodal normal distribution crossover," *Proceedings of the Seventh International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 246-253.

Bäck, T. and H.-P. Schwefel, (1993). "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, Vol. 1., No. 1., pp. 1-23.

Patton, A., T. Dexter, E. D. Goodman, and W. F. Punch III, (1998). "On the Application of Cohort-Driver operators to continuous Optimization Problems Using Evolutionary Computation," *Evolutionary Programming VII*, Springer-Verlag, pp. 671-681.

Jain, A. K. and R. C. Dubes, (1988). *Algorithms for Clustering Data*, Prentice Hall.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley.

Press, W. H., S. A. Teokolsky, W. T. Vetterling and B. P. Flannery (1992). *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Edition. Cambridge University Press.

Wolpert, D. H. and W. G. Macready, (1997). "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 1., No. 1., pp. 67-83.