A Distance Measure Comparison to Improve Crowding in Multi-Modal Optimization Problems

3rd International Symposium on Resilient Control Systems

D. Todd Vollmer
Terence Soule
Milos Manic

August 2010

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

Idaho National
Laboratory

A Distance Measure Comparison to Improve Crowding in Multi-Modal Optimization Problems

D. Todd Vollmer
Idaho National Laboratory
National and Homeland Security
Idaho Falls, Idaho 83402

Terence Soule
University of Idaho
Department of Computer Science
Moscow, Idaho 83844

Milos Manic
University of Idaho at Idaho Falls
Department of Computer Science
Idaho Falls, Idaho 83402

Abstract— **Solving multi-modal optimization problems are of interest to researchers solving real world problems in areas such as control systems and power engineering tasks. Extensions of simple Genetic Algorithms, particularly types of crowding, have been developed to help solve these types of problems. This paper examines the performance of two distance measures, Mahalanobis and Euclidean, exercised in the processing of two different crowding type implementations against five minimization functions. Within the context of the experiments, empirical evidence shows that the statistical based Mahalanobis distance measure when used in Deterministic Crowding produces equivalent results to a Euclidean measure. In the case of Restricted Tournament selection, use of Mahalanobis found on average 40% more of the global optima, maintained a 35% higher peak count and produced an average final best fitness value that is 3 times better.**

Evolutionary computation, genetic algorithms, multimodal optimization, niching methods

I. Introduction

Problems in which a number of points are potentially good solutions, while not necessarily optimal, are defined as multi-modal problems. Genetic Algorithms (GA) including crowding approaches such as Deterministic Crowding (DC) and Restricted Tournament Selection (RTS) have been developed to maintain sub-populations that track these multi-modal solutions. For example, multi-modal GA's have been used in the design of a nuclear reactor core [1]. In addition, two surveys highlight the multiple uses of GA's in control systems and power engineering tasks [2],[3]. These tasks include optimization for controller design and model identification, fault diagnosis, reliable systems, robustness analysis and robot control.

The basic idea of multi-modal GA's is to encourage the evolution of subsets representing diverse solutions in a single population during the evolutionary process. In order to measure this diversity, distance measures are employed. Given better distance measures, improved results may be realized. Thus, we examine the effectiveness of Mahalanobis distance in comparison with Euclidean distance in two real value encoded Genetic Algorithm solutions.

A Genetic Algorithm (GA) is a heuristic search technique inspired by concepts of evolutionary biology. They became popular with the published work of John Holland in the 1970's. Generally, for an algorithm to be categorized as a GA it needs a population representation of possible solutions, variation operators, selection and replacement mechanisms. The actual details, such as population representation and distance measure, of a GA can vary greatly and represent one of the challenges of an actual implementation.

In a GA an individual is a candidate solution out of a set of solutions called a population. This individual may be represented by a genotype where in turn each genotype maps to a phenotype [4]. For example, in a population of integers an integer value of 10 (phenotype) could be represented by a binary code of 1010 (genotype). Following this definition to a logical conclusion, it is critical to a robust solution that the genotype is capable of fully representing the optimal individual solution characteristics. An evaluation function is used to determine the 'fitness' of individuals in a population. This fitness is a measure indicating how well the individual is solving a given problem and used for decision making in the evolutionary process.

When optimizing multi-modal functions a conventional GA's population tends to converge to just one of the optimal, or near optimal points. This characteristic occurs because of *Genetic Drift* and is an artifact of the application of random selection processes to finite populations [5]. This convergence to only one solution is undesirable in multi-modal optimization of real problems. Because a GA utilizes a population of many (hundreds, thousands or more) possible solutions, modifications to the algorithm can enable maintaining several optima.

One such modification, Fitness Sharing, lowers each individual's fitness by an amount relative to the number of similar individuals in the population [6]. Similarity is determined by evaluating a distance measure between population elements.

Another GA modification, Deterministic Crowding, is an improved version of De Jongs Crowding. After crossover and mutation, each resulting child individual replaces the most similar parent used to create it if it has a higher fitness value.

Similarity is typically computed using phenotypic distance [7]. Restricted Tournament Selection, similarly to DC, creates two new children [8]. These children then compete with a fixed number of randomly chosen individuals from the population; the number of competing individuals is defined by the Crowding Factor (CF). The nearest individual chosen for the competition is selected from the CF population using a distance measure.

Distance measures as a means of determining similarity or closeness are a common algorithmic feature for many GA implementations such as those used in Crowding. The distance measure used is dependent on the population definition but typically solutions like Euclidean distance or Hamming distance are implemented. In addition, these distance measures can occur in different domains of the problems set i.e. Phenotypic, Genotypic and Fitness [9].

II. RELATED WORK

This section reviews research efforts in multi-modal GA's and the use of Mahalanobis distance metric. Independent support for the use of this distance metric in computer vision is summarized. Finally the appropriate use of real coded genetic algorithms in a lengthy paper is presented.

Recent work has introduced the Mahalanobis distance metric in the framework of Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [10]. This metric was a natural fit as the covariance matrix of the distribution, a necessary component of Mahalanobis, was already being utilized. Specifically, the authors explored using the distance metric in a self adapting niche radius and compared it to an implementation using the Euclidean distance. Several test functions including Ackely, Rastrigin, and Griewank were evaluated. (Some of these functions were also selected for use in this paper.) The niching routines based on the Mahalanobis achieved the goal of improving the niching process in terms of obtaining on average higher quality sub-optima.

The Mahalanobis distance metric has been shown to be effective in other subject areas such as computer facial recognition. In [11] a statistically significant improvement was found when compared to City-block, Euclidean and Angle routines. McNemar's test was used to test the statistical significance of the results. This paper extended a previous work done on Principle Component Analysis in facial recognition on the FERRET database. In addition to singular distance values, combinations of the values were compared with no improvements found. The result of this paper provides independent evidence of Mahalanobis as an effective distance measure.

Herrera, et al. reviewed the features of real-coded genetic algorithms (RCGA) [12]. They found that the most important feature of RCGAs is their capacity to exploit local continuities. The experimental results indicated the most suitable genetic operators for building RCGAs are nonuniform mutation, BLX-α, logical fuzzy connectives based and linear crossover. It is possible for these operators to generate values outside the range of the problem domain so this must be considered in an implementation. Another advantage of using real coding is that the solution representation is close to the natural formulation of many problems. In fact, there may be no difference between the genotype and the phenotype. Therefore, the coding and decoding processes that are needed in binary coded versions of GA's are avoided, thus increasing speed. In addition, it was pointed out that the natural representation of genes for optimization problems of parameters with variables in the continuous domain is directly as real numbers. The use of real parameters also makes it possible to use large domains, which is difficult in binary implementations where increasing the domain could mean sacrificing precision. The importance of this information, in regards to the work in this paper, is the appropriate use of RCGAs in continuous function domains such as those used experimentally and the selection of genetic operators.

III. EXPERIMENTAL APPROACH

This section describes the implementation details of the Genetic Algorithm and the two population diversity algorithms. Equations for the two distance metrics and their computational complexity are discussed. The final section discusses the test functions used for evaluation.

A. Genetic Algorithm Description

For this project, solutions to minimize five benchmark functions with a range of different dimensions were evolved utilizing two forms of a crowding Genetic Algorithm (GA). A pseudo-code implementation of a GA is shown in Fig. 1 below and lines with numbers are described more fully later.

One of the first tasks in building a GA is to decide upon a representation of the solution population (line 1 in Fig. 1) and create a number of individuals in that population. All of the functions defined for this project make use of real values as inputs. This made for a natural definition for the population representation and floating point numbers were used to represent each of the n positions in a given individual solution. Therefore an individual can be seen as a vector $\vec{v}_i = (x_0, \ldots, x_n)$ of floating point values x. The total population is a set of vectors $P = \{\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_T\}$. For each test function the initial population size was set to 60. The initial generation of individuals is populated by randomly generating values uniformly in the domain range of the given function. The test function section provides the ranges of each function used in testing.

After an initial population is created a series of variations

```
1 Create an initial population
  For Each Selection Process
  2 LOOP while below execution count
    3 select individuals as parents
    4 create children from parent
       (crossover/mutation)
    5 select and replace individuals with children
    6 update fitness values
    END LOOP
  End For Each
```

Figure 1. Genetic Algorithm

and selections must take place on the population individuals. This can continue until some acceptable solution or predefined resource limit is reached. For this project, each function has an associated max number of iterations value and is shown in Table 1 as iterations. When this number was reached, after incrementing by one each pass through the loop (line 2 in Fig. 1) and initially starting at zero, processing terminated and the fittest individual in the population was isolated as the final best solution. The fitness in this case consisted of the value returned from exercising the test function with the real value parameters represented by the individual. The fittest individual produced the smallest function output.

A steady state population model was implemented. This means that, for each iteration through the loop, only a small part of the original population is changed. This is in contrast to a generational model where the entire population is replaced by the offspring. In this case, two individuals were selected as parents and two offspring were created from them. Subsequently, two more individuals were then selected for replacement by the new offspring creating a new generation.

The two candidate parent individuals were randomly selected independent of any fitness or distance measure (line 3 in Fig. 1). After crossover of the parents occurs, the two resulting offspring are evaluated to replace candidates in the population. This is a critical step for the maintenance of a muti-modal solution set. The determination of how replacement individuals are selected is discussed later in the section on replacement strategy.

Crossover between the two parents to create two new offspring utilized whole arithmetic recombination (line 4 in Fig. 1). Each position of the child is a new value created from the values at the same position of the parent vectors. This new value lies between that of the two parents and is created for each child using equations (1) and (2). Where x and y are the values at ith position in the parent vectors and α is a weight adjustment. The weight adjustment value α used for all functions is 0.6.

$$Child1 = \alpha * x_i + (1 - \alpha) * y_i \qquad (1)$$

$$Child2 = \alpha * y_i + (1 - \alpha) * x_i \qquad (2)$$

After crossover, mutation occurs on both child individuals every iteration prior to replacement selection (line 4 in Fig. 1). This mutation is non-uniform with a fixed Gaussian distribution which means most of the changes made will be small. For each position in the selected individual, a value drawn randomly from a Gaussian distribution with mean zero and user defined standard deviation of 0.1 is added to it. If this operation results in a value outside of the acceptable function range, defined in the test function section, the value is set equal to the closest boundary value.

Finally the fitness values for the mutated individuals are updated (line 6 in Fig. 1). This is the last step before the next

iteration is performed following the steps already outlined with a population set that contains the new individuals.

This entire process is repeated for both replacement selection algorithms on a copy of the same population. An exact copy of the *initial* population is carried over from one replacement strategy trial to the next ensuring the same starting data sets are used. In addition to the multiple iterations of both selection routines on the same data set, an outer loop is executed 100 times (i.e. 100 independent trials). Because of the stochastic nature of crossover and mutation this is done to obtain a higher confidence in the performance measures.

B. Replacement Strategy

In order to maintain a diverse solution set within the population two methods of choosing replacement candidates were considered: Restricted Tournament Selection and Deterministic Crowding. As mentioned in the introduction, both of these techniques are capable of maintaining a diverse population of solutions. The implementation of line 5 in Fig. 1 makes use of only one of the replacement selection techniques at a time. For comparison purposes, both were implemented and exercised in independent trials.

Uni-modal GA solutions using tournament selection randomly pick two individuals for replacement. These values are replaced if the new children have better fitness values. Restricted Tournament Selection (RTS) instead picks a candidate replacement individual that is closest to the new child from a subset of the population of window size w. The size of the population contained in w is defined by empirical testing (30 for this project in all cases) and each member is drawn from the original population using a uniformly random selection process. Closeness is determined by a distance function. After determination of the closest individual to the candidate child a competition is held based on fitness between the child and selected individual. The one with the best fitness is selected for inclusion into the solution population.

Deterministic crowding introduces competition between the children and the individuals used to create them. After crossover and mutation each child replaces the nearest parent if it has a higher fitness. Given two parents (P1, P2) and two related children (C1, C2), two of the four possible tournaments are executed. Selection of the tournaments is determined by the smallest distance value between a parent and child. The pseudo algorithm for this procedure is shown in Fig. 2 where F is a fitness function and D is the distance function. It should be noted this is for optimizing a minimization problem where a smaller fitness value is better.

```
IF (D(Pi,Ci) + D(Pj,Cj)) <= (D(Pj,Ci)+(D(Pi,Cj))
        If (F(Cj) < F(Pj) then replace Pj with Cj;
        If (F(Ci) < F(Pi) then replace Pi with Ci;
ELSE
        If (F(Ci) < F(Pj) then replace Pj with Ci;
        If (F(Cj) < F(Pi) then replace Pi with Cj;
```

Figure 2. DC Tournament

As just described, both RTS and DC require a distance measure. This distance measure occurs in the genotype domain in our experiments. The two distance measures (Euclidean and Mahalanobis) that were evaluated are presented in the following section.

C. Distance Measures

Gray coding, Hamming distances and similar algorithms can be used when binary encodings are utilized. However these type of distance measures are not directly applicable to real value encodings. In general a Euclidean or Hamming distance measured is used in genetic algorithms whenever a distance measure is needed [12],[13]. This paper focuses on two distance measures: Euclidean and Mahalanobis.

The Euclidean distance is the familiar geometric distance based on the Pythagorean formula. This distance measure is relatively simple to calculate using the following formula where *x* and *y* are n dimensional vectors representing points:

$$d(x,y) = \sum_{i=1}^{n} \sqrt{(x_i - y_i)^2} \qquad (3)$$

This distance measure has a straightforward geometric interpretation, is computationally inexpensive and simple to code. However it does have two drawbacks expanded upon in the following paragraphs.

First, in geometric problem domain variables are typically measured utilizing the same units of length. Data values from real world problems may have different scales. For example a regression problem making use of class information such as age, test scores and time are all on a different scale and therefore not directly comparable. The Euclidean distance is sensitive to the scales of the variables involved and may not perform optimally. This problem can be overcome by a standardized or weighted Euclidean distance which incorporates variances but not covariances. A Mahalanobis distance incorporates both variances and covariances.

Second, the Euclidean distance does not compensate for correlated variables. Given a test data set containing multiple variables where one variable set is an exact duplicate of another set, these sets are highly correlated. The Euclidean distance calculation will weight the duplicate variables more heavily than the others. It has no method of accounting for the fact that the duplicate provides no new information.

Mahalanobis distance was introduced by P.C. Mahalanobis in 1936. It is based on both the mean and variance of the variables in addition to the covariance matrix. The iso-surface formed around the mean is an ellipse in two dimensional space or an ellipsoid or hyper-ellipsoid when more variables are used. It is a multivariate quantitative method that can solve for multiple dimensions simultaneously. The covariance among the variables is taken into account when calculating the distance. Because of this, the problems of scale and correlation inherent in the Euclidean distance are not an issue. Given an individual as a vector $\vec{x}_i = (x_0,\ldots,x_n)$ of floating point values *x*, a vector representing the mean of a data set $\vec{\mu} = (\mu_0,\ldots,\mu_n)$ and a covariance matrix C of size *n x n* representing the covariance values between all dimensions *n*, the Mahalanobis distance is calculated with the given formula:

$$md(\vec{x}_i) = (\vec{x}_i - \vec{\mu})C^{-1}(\vec{x}_i - \vec{\mu})^T \qquad (4)$$

This function produces a distance value for the $\vec{x}_i$ vector. This vector is either a parent or child individual. The steady state population is used to compute the mean $\vec{\mu}$. In effect the distance measure utilized is not the distance between two vectors but the distance of a vector from the GA population. Hence the population is used as a reference point for all distance measures.

D. Algorithmic Complexity

Computational complexity of the Mahalanobis Distance measure is $O(n^2)$ for n dimensional data vectors in the solution population domain [14]. Without any optimizations, the Euclidean distance computational complexity is $O(n)$. In this paper, the Euclid distance was computed in the genotype domain. Given a minimization problem with 0.0 being the global optimal solution this computational complexity is reduced to a constant time in one dimension.

E. Test Function Descriptions

Five functions were used to evaluate performance. These functions have been used frequently in GA evaluations. For completeness these function are described next. The functions were evaluated with 2, 3 and 5 dimension value sets.

Sphere (5) is a continuous, convex uni-modal, n-dimensional function constrained to real values -5.12, 5.12. A global minimum occurs at 0.

$$F_s(x) = \sum_{i=1}^{n} x_i^2 \qquad (5)$$

The generalized Rastrigin (6) function is *n*-dimensional function with a large number of local minima whose value increases with the distance to the global minimum. The function was constrained to real values in the range -1.5, 0.5 where *A*=10 and *w*=2π. This limited the number of optima to 5 including the global.

$$F_{ra}(x) = nA + \sum x_i^2 - A\cos(wx_i) \quad (6)$$

Ackley (7) is a highly multi-modal n-dimensional function. A large number of local minima are spread evenly over the space. One global minimum occurs at 0.

$$F_{ak}(x) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)$$
$$- \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) \quad (7)$$

Greiwangk (8) has a product term that introduces interdependence among the variables. It is a continuous multi-modal function that has a global optimum at 0.0. In addition it has four relatively large optima at $[\pm\pi, \pm\pi * 1.414, 0.0, \ldots]$. It is constrained to -600,600.

$$F_g(x) = 1 + \sum_{i=1}^{n}\frac{x_i^2}{400} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (8)$$

M6 (9) is called the Shekels Foxhole problem [5]. This is a 2-dimensional problem with 25 optima. The two variables are restricted to the range -65.536, 65.535. The maxima are located at (16i, 16j) where i and j are integers in the range [-2,2]. They are all of differing heights with a global optimum at (-32,-32). M6 is defined below:

$$F_{m6} = 500 - \frac{1}{.002 + \sum_{i=0}^{24}\frac{1}{1 + i + (x - a(i))^6 + (y - b(i))^6}} \quad (9)$$

where $a(i) = 16[(i \bmod 5) - 2]$ and $b(i) = 16(\lfloor i/5 \rfloor - 2)$.

IV. RESULTS

The intent of this project is to gauge the performance of the distance measures when used in GA crowding multi-modal solutions. The performance of the test runs was measured by three metrics: 1. Peak count = Average number of peaks found. 2. The number of times the global optimum was found in the 100 repeated runs. 3. The average best fitness of the final solution for the 100 repeated runs.

For a solution to be considered as an optima it was not necessary to match the exact value. It should be noted that the goal was minimization and most of the test functions had a global minimum at 0. If all the points of the data vector fell within a small offset from the correct value it was considered to have been a match. Table 1 labels this value as niche for each test function.

In order to empirically show that the replacement selection method, using two distance measures, was choosing different individuals the specific individual selection results were tracked during a trial run of RTS. At each point in the algorithm where a distance measure was required, both distance measures were calculated for the two children. In a run with 200 iterations, the individuals selected were identical 10% of the time. 46% of the selections had one individual in common. In the remaining 44% the individuals chosen were unique.

A similar process for evaluating replacement selection in DC was implemented. The test consisted of 400 iterations on the five test functions. Given that DC can select to replace at a maximum two parents the total number of possible replacement considerations the algorithm had to make was 4,000. The Mahalanobis distance made a change 2,314 times and Euclidean 2,320. Of these changes all but 152 were identical replacements. This shows that while there were differences in the distance measures they were in agreement on 96% of the decisions.

Fig. 3 contains graphs of the Peak Count for both replacement solutions with a higher value indicating more optimal solutions are found. The data is from the two dimensional test sets. It shows that Mahalanobis matches or outperforms Euclidean for RTS. Performance results for the DC implementation showed on average the performance for both distance measures was equivalent and consequently the results are not shown in the figure.

The number of global optimum discovered in each of the 100 independent runs is found in Table 2. The bold face numbers indicate the better score for each distance measure. If they performed equally well then both values for the distance measure are in bold unless both values are zero. Results for all five test functions in the three different dimensions are available. Out of the fifteen RTS test runs, Mahalanobis scored better eleven times and equivalent once with two instances of both finding no optimums. In the fifteen DC runs, five runs were equivalent, Euclid scored better five times, Mahalanobis scored better five times and there was one instance where no optima were found. This is the second result that shows little difference in performance between the two distance measures in regard to DC output.

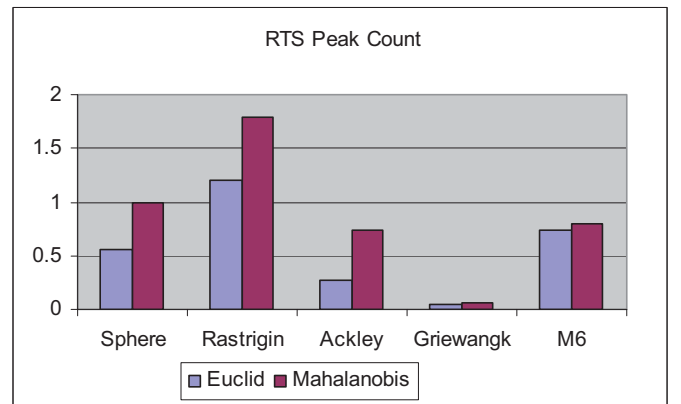Finally Fig. 4 depicts the average final best fitness value of

Figure 3. RTS Peak Count

TABLE II. GLOABAL OPTIMUM COUNT

| Functions | DC | | RTS | | |
	Euclid	Mahal	Euclid	Mahal	Dim.
Sphere	**100**	**100**	97	**100**	2
Rastrigin	**98**	**98**	**91**	89	2
Ackley	**100**	**100**	70	**95**	2
Griewangk	**20**	11	0	**1**	2
M6	**3**	2	3	**4**	2
Sphere	**100**	**100**	61	**100**	3
Rastrigin	**49**	42	**35**	**35**	3
Ackley	**98**	93	17	**73**	3
Griewangk	0	**1**	0	0	3
M6	2	**3**	2	**3**	3
Sphere	**100**	**100**	5	**97**	5
Rastrigin	0	**1**	1	**3**	5
Ackley	35	**36**	0	**33**	5
Griewangk	0	0	0	0	5
M6	0	**3**	2	**5**	5

the 100 independent trials of the RTS selection function for five dimension solutions. The fitness value is the result of executing a test function with the values of an individual of the population. For this experiment the goal is to find a minimum value. Consequently smaller fitness values are better. Mahalanobis consistently produced better (smaller) results in the four test functions. The M6 test function is defined for only two dimensions and therefore is not shown.

It was observed in the DC implementation that only small differences existed in all of the experiments with the two distance measures. The DC algorithm uses the distance measure (in addition to fitness) as part of a replacement strategy to choose between two parents and two children. The conjecture is that this small population choice accounts for the similar performance when utilizing the different distance functions. In RTS the crowding factor provides a larger population (30 individuals) for distance comparisons. As was shown in the previous section, the two distance measures used in RTS select at least one different individual 90% of the time. However for DC the difference in selection was only 4%.
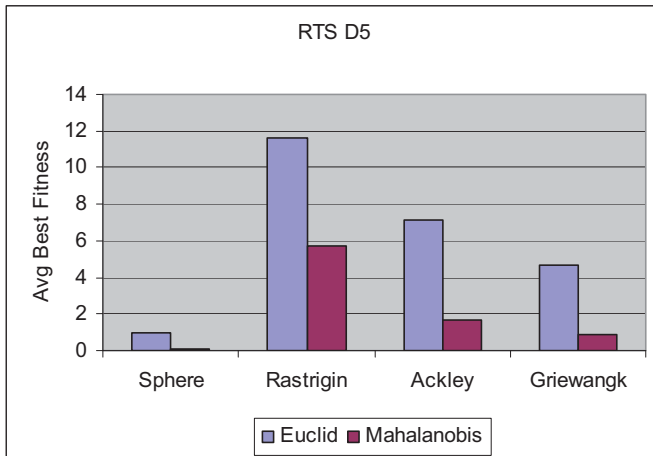

Figure 4. 5 Dimension RTS

V. CONCLUSION

Two crowding type Genetic Algorithm multi-modal solutions with real coded values were chosen for implementation: Restricted Tournament Selection and Deterministic Crowding. Both Mahalanobis and Euclidean distance measures were utilized by these routines in selection determination. Five frequently used test functions were implemented and used to evaluate the performance of the selection routines.

Mahalanobis is computationally more expensive but, within the parameters of this project, appears to be superior to Euclidean distance in Restricted Tournament selection and equivalent in Deterministic Crowing. The use of Mahalanobis distance, in the case of Restricted Tournament Selection, found 40% more of the global optimum, maintained a 35% higher average peak count and produced an average final fitness that was 3 times better (lower). As is often the case, this paper depicts a tradeoff of computational complexity versus performance when choosing a distance algorithm to implement.

REFERENCES

[1] A. Mishra, M.D. Pandey, "Regulation of nuclear power plants a multi objective approach,", Trans. SMiRT 19, Toronto, August 2007.

[2] N. Rajkumar, T. Vekara, J.T. Alander, "A review of genetic algorithms in power engineering," Report 03-04, Univ. of Vassa, 2008.

[3] P.J. Fleming, R.C. Purshouse, "Evolutionary algorithms in control systems engineering: a survery", Control Engineering Practice 10, 1223-1241, Elsevier Science Ltd, 2002.

[4] A.E. Eiben, J.E. Smith, "Introduction to evolutionary computing", SpringerVerlag, 2003.

[5] K.A. DeJong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems." PhD Thesis, University of Michigan, 1975.

[6] D.E. Goldberg, J. Richardson, "Genetic algorithms with sharing for multimodal function optimization", in Proc. of the 2nd International Conf. on Genetic Algorithms and Applications, Cambridge, MA., 1987.

[7] S.W. Mahfoud, "Niching methods for genetic algorithms", Ph.D. dissertation, Dept. Comp. Sci., Univ. of Illinois, 1995.

[8] G.R. Harik, "Finding multimodal solutions using restricted tournament selection," in Proc. of the 6th International Conf. on Genetic Algorithms, San Francisco, CA, USA, pp. 24-31, 1995.

[9] F. Natalia, A. Hugo, G. Raul, "Crowding under diverse distance criteria for niche formation in multimodal optimization", in Jour. of computer science and technology, October 2000.

[10] O. M. Shir, M. Emmerich and T. Bäck, "Self-Adaptive Niching CMA-ES with Mahalanobis Metric," in Proc. of IEEE-CEC. Singapore, IEEE Press, 2007.

[11] W. Yambor, B. Draper and R. Beveridge, "Analyzing PCA-based face recognition algorithms: eigenvector selection and distance measures," in Empirical Evaluation Methods in Computer Vision, World Scientific Press, Singapore, 2002.

[12] F. Herrera, M. Lozano and J.L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis," in Artificial intelligence review 12, Kluwer academic publishers, 1998, pp. 265-319.

[13] R. Dreżewski and L. Siwik, "Techniques for maintaining population diversity in classical and agent-based multi-objective evolutionary algorithms," In proc. of the 7th International Conf. on Computational Science, Beijing, China, May 2007.

[14] R. R. Pinho, J. Manuel, R.S. Tavares and M.V. Correia, "Efficient approximation of the Mahalanobis distance for tracking with the Kalman filter," in CompIMAGE, 2006, pp. 84-92.