An Alternative to Random Generation of the Initial Population for Genetic Algorithms.

University of Wales, Newport
School of Computing and Engineering
Allt-yr-yn Campus
PO Box 180
Newport Gwent UK
Telephone (01633) 432441
Fax (01633) 432442
shane.lee@newport.ac.uk
hefin.rowlands@newport.ac.uk

Abstract

An alternative to the traditional random method of generating the initial population of genetic algorithms is introduced. The orthogonal arrays of the Taguchi method are used to create a population of chromosomes. The performance of these populations is compared empirically with populations created in the conventional random way. Just one test function is consider here which adequately illustrates the strengths and weaknesses of the new techniques The addition of Gaussian noise to the new initialization routine is shown to further improve the results.

1. Introduction.

Genetic algorithms (GA)[Goldberg 1989] have proven to be a robust heuristic optimization technique based on natural selection[Holland 1975 ,Dawkins 1987]. In common with all heuristic methods, GAs cannot guarantee to locate the global optimum in any problem space in a reasonable time.

All search techniques[Rich & Knight 1991, M.Lee 1989], GAs included, are very demanding on computer resources in all but the most trivial of problems[Goldberg 1989, Holland 1975, Rich & Knight 1991] which can prohibit their use in applications where the resources of hardware and/or time are unavailable. In parallel with seeking the global optimum there exists the need to minimise the computer resources necessary to complete an optimization task in an acceptable time scale[Buckly et al 1997].
The two desires are at odds with one another. To increase the probability of finding the global optimum using a GA, all that need be done is to increase the size of the population, which will broaden the coverage of the problem space. However, any increase in the size of the population will mean extra demand for the resources of the computer i.e. the power of the processor, time and/or memory.

It follows that if the population of a genetic algorithm can be kept small then it is possible to reduce the computer resources used. This approach brings other problems; a small population may encapsulate too little information for the search to progress without premature convergence. Creating the initial population of a GA in the traditional random manner can in itself accelerate premature convergence if that population is small. The problem is clustering.

The method described in this paper shows that creating the initial population in a structured way can, in some instances, aid a genetic algorithm to converge more effectively to an optimum. In this paper the method of creating initial populations with orthogonal arrays will be described. The features of this technique and the results obtained will be shown in detail for the first Dejong problem (a three dimensional parabola). A summary of results for the other Dejong functions is also included.

2 Clustering

To demonstrate the clustering problem, consider a simple 2-dimensional space this is divided into 25 equal sub-areas. 25 pairs of random numbers are generated which create co-ordinates, the sub-area where the position which a pair of co-ordinates represents gives a count of +1 to that area. Figure 1, shows such an experiment; it is clear that the distribution of co-ordinates is not even. Some sub-areas are 'hit' more than once while many sub-areas are not hit at all. If this where a genetic algorithm and the random co-ordinates where used to create chromosomes, the population would leave large regions of the problem space unrepresented in the initial population. If the desired optimum is in one of these areas the GA will have difficulty finding it.

0	2	0	0	0
0	2	0	2	0
2	1	2	0	0
2	1	1	4	1
0	1	0	2	2

Figure 1, Demonstration of Clustering

The technique introduced in this paper uses the orthogonal arrays of Taguchi method to create the

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

initial population of a GA in an attempt to circumvent the clustering problem.

3. Taguchi Method and Orthogonal Arrays.

The 'Taguchi method' [Roy 1990] was developed after the second world war as a means of streamlining the experimental process and to increase the quality of manufactured products.

The orthogonal arrays are the portion of Taguchi method pertinent to this paper. The origin of the arrays is the use of 'Latin squares orthogonal arrays' for experimental design [Roy 1990].

In the Taguchi method the orthogonal arrays are used to select the parameters for a series of experiments in such a way that the ranges of each variable and combination of variables are well represented. In this way maximum information can be drawn from the experimental process with a minimum of effort. The technique developed in this paper treats the process of creating an initial population for a genetic algorithm as if a series of experiments were being planned.

An orthogonal array has three important features:-

1. *Factors*- The variables or parameters in an experiment.
2. *Levels*- The values assigned to a factor. A factor will have 2 or more levels in an array.
3. *Trials*- A combination of factors instantiated with levels to perform a single experiment.

There are many different orthogonal arrays [ASI] with varying numbers of trials, factors and levels. However each array has the property of 'Orthogonality'. This property is said to exist when any pair of factors has each possible combination of levels equally represented. It is the orthogonal nature of the arrays which is exploited in this method by supplying the initial population with a structured coverage of the problem space.

4. Initializing a Genetic algorithm Using Orthogonal Arrays.

4.1 Creating Chromosomes Using Orthogonal Arrays.

To create the initial population of a GA an orthogonal array, designated L_{25} (figure 2), with the ability to vary up to six factors each at five levels has been used throughout the experiments reported in this paper. The population size of 25 is at the higher end of what is deemed to be a small population [Reeves 1995] (<30). The orthogonal property of the array gives coverage of the two dimensions space considered in section 2 as shown in figure 3. Comparing figure 3 with figure 1 it can be seen that all the 25 sub-areas are now represented.

It should be noted that figure 3 is a projection of the geometric positions of the chromosomes in the problem space onto any two dimensions. The grid-like appearance of the distribution of the chromosomes disappears when more dimensions are considered. Regardless of the number of dimensions the number of chromosomes is constant.

Trials	Factors (dimensions)					
(chromosomes)	1	2	3	4	5	6
1	1	1	1	1	1	1
2	1	2	2	2	2	2
3	1	3	3	3	3	3
4	1	4	4	4	4	4
5	1	5	5	5	5	5
6	2	1	2	3	4	5
7	2	2	3	4	5	1
8	2	3	4	5	1	2
9	2	4	5	1	2	3
10	2	5	1	2	3	4
11	3	1	3	5	2	4
12	3	2	4	1	3	5
13	3	3	5	2	4	1
14	3	4	1	3	5	2
15	3	5	2	4	1	3
16	4	1	4	2	5	3
17	4	2	5	3	1	4
18	4	3	1	4	2	5
19	4	4	2	5	3	1
20	4	5	3	1	4	2
21	5	1	5	4	3	2
22	5	2	1	5	4	3
23	5	3	2	1	5	4
24	5	4	3	2	1	5
25	5	5	4	3	2	1

Figure 2, the L_{25} array

To make a population of chromosomes, the 'factors' of the orthogonal array should be considered as the dimensions of the problem space. The trials then become the individual chromosomes of the population.

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Figure 3, 2 Dimensional Problem Space Instantiated Using the Orthogonal Array Technique.

To illustrate the construction of a chromosome, consider any 2-dimensional problem space, where both dimensions have the range ±5.12. As an example the ninth chromosome in the population will be constructed using the first two factor columns of L_{25} (the choice of these two columns is arbitrary, any different pair will do). The chosen levels give an even spread across for a given dimension, they are as follows:-

Level 1 = +3.40, Level 2 = +1.70, level 3 = 0.00, Level 4 = -1.70 and level 5 = - 3.40.

From the L_{25} array the two parameters for chromosome 9, for dimension 1 becomes, 1.70 (level 2) and for dimension 2; -1.70 (level 4). These values would then be encoded.

4.2 The Demonstration Experiments.

4.2.1 The Fitness Function.

To illustrate the effects of employing the orthogonal array technique the results of a simple set of experiments will be discussed here. Important to the discussion is the choice of problem space. The first Dejong equation[Dejong 1975] is a three dimensional parabola as described by equation 1, with each dimension in the range ±5.12.

$$F(x)=\sum_{i=1}^{3} x_i^2 \quad \text{……… equation (1)}$$

From equation 1, equation 2 has been developed as a fitness function. A 2 dimensional version of the problem space created by equation 2 can be seen in figure 4.

$$G(x)=\frac{10000}{F(x)} \quad \text{…….. equation(2)}$$

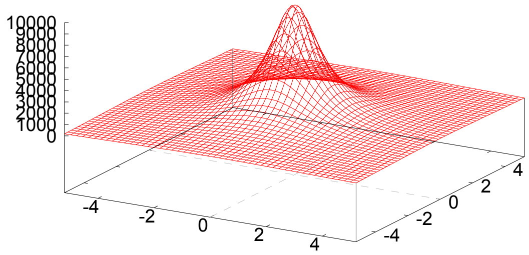

Figure 4, "dimensional version of equation 2

This problem space is simple, uni-modal with steep slopes leading to the global-optimum. These properties make an ideal toy problem with which to gauge the performance of a GA initialised with an orthogonal array except for 1 more problem. The very position of the global optimum may well favour the levels chosen for use in creating the chromosomes. As the position of the global optimum is at (0,0,0), if the value 0 is chosen for one of the levels it can be seen that that initial population will easily be able to converge of the global optimum as all its components are already present. So the experiments use a further modification to the fitness function as seen in equation 3.

$$H(x)=\frac{10000}{\sum_{i=1}^{3}(x_i+\Delta)^2} \quad \text{………… equation 3}$$

The additional factor 'Δ' allows the position of the global optimum to move between experiments. In the experiments reported here Δ is adjusted such that global optimum is moved from (-5,-5,-5) to (5,5,5) in increments of 0.1.

4. 2. 2 Common experimental parameters.

The experiments have been deliberately kept simple; the crossover rate was set to 1 and mutation rate to 0.005 throughout the experiments. The roulette wheel fitness proportional selection method has been used constantly. The population was consistently 25, dictated by the L_{25} array. Each experiment used one population generated using the Taguchi array and another produced in the traditional random manner as a bench mark. All experiments were run 60 times and the results aggregated and averaged. All the genetic algorithms ran for 130 generations.

4.2.3 Performance Measure.

A measure used throughout the experiment is the 'plateau value'. When a single GA is run and convergence is achieved the fitness value of convergence is determined by a straight line through a set of points.

4. 2.4 Experimental Results.

The graph in figure 5 shows the results of the experiments described above with traditional binary encoded chromosomes. The 'Y'-axis of the graph shows the plateau value of the maximum fitness while the 'X'-axis shows the 'Δ' factor used. This means that a point on either of the lines of the graph represents the maximum fitness found when a GA was run with the global optimum at (-Δ,-Δ,-Δ). The two lines represent the performance of the Taguchi & random initialised populations. It is clear that the Taguchi population has areas of good performance and areas of very bad performance. However of all the genetic algorithm runs the Taguchi population produced a better result 55% of the time but those good results are very dependent on the position of the global optimum and the levels used.

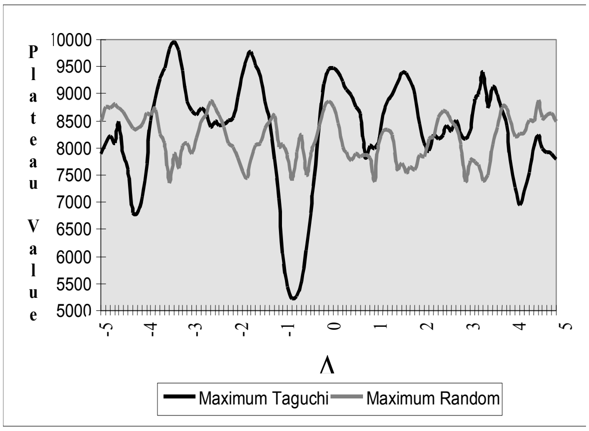

Figure 5, the result for binary encoding.

The results depicted in figure 6 are for the same experiment save that the GA has used floating point encoding; here the range of results is somewhat less dramatic but the peaks and troughs are still there.

For floating-point encoding the Taguchi populations performed better than the random 70% of the time.

4.2.5 Building Block Starvation & Noise.

The method of creating the initial population using orthogonal arrays is at odds with schemata theory Goldberg, 1989, Holland 1975, Neubauer 1997]. The initial Taguchi population is distributed in a manner that considers the geometry of the problem space. In fact the worst performances of the Taguchi populations tend to occur when the global optimum is at large Euclidean distance from the co-ordinates of the nearest chromosome in the population.

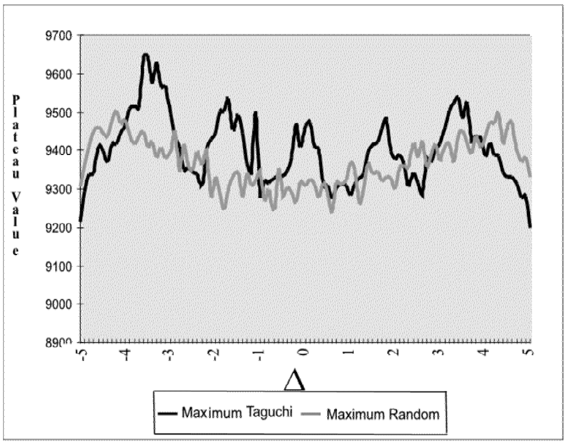

Figure 6, the result for floating-point encoding

However, no consideration has been given to the building blocks necessary for constructing a large variety of novel chromosomes. In an initial population created with the orthogonal array method each parameter will only be supplied with five different values with which to start the search. An initial random population with 25 chromosomes can reasonably expect 25 different values for the same parameter. To increase the number of distinct values and preserve most of the geometric distribution of the Taguchi arrays, experiments have been performed with Gaussian noise added to each value before it is instantiated into the orthogonal array. The range of the noise used was ± 1.50. With this modified arrangement with binary encoding the Taguchi initialised population out performed the random 61% of the time. When float-point encoding is used the Taguchi initialised population out performed the random 76% of the time. This is an increase in relative performance when compared to the results where Gaussian noise is not used.

5 Discussion

This paper has introduced and reported the results of experiments of a new initialisation technique for genetic algorithms. The results show a large degree of success for the new technique but by no means are they an unqualified success. Why does the structured approach not work under all circumstances for binary encoded chromosomes? It has clearly been shown that random generation does indeed result in clusters of chromosomes in some points of the problem space and that the space is more evenly covered using the structured orthogonal array method. However, consider the case of a experiment where the global optimum is not close to a point where its co-ordinates can be easily constructed by the values used by the orthogonal array. The orthogonal array population guarantees that there are no chromosomes close to the optimum. However a randomly generated population has a probability that one or more are close to the optimum or contain the building blocks to build a good chromosome through crossover. The question is “Can initialisation with orthogonal arrays be recommended as a replacement for traditional random initialisation?” The answer has to be no. However it can be a useful supplementary way to investigate a problem space with a genetic algorithm. That is to say, when the opportunity exists to use both types of initialisation, use orthogonal initialisation for some runs of the algorithm and traditional random initialisation for others and pick the best result. The results for floating-point encoded genetic algorithms are much more encouraging. As with the binary encoded genetic algorithms the better performance comes when the levels instantiated into the orthogonal array are close to the co-ordinates of an optimum in the problem space. However the performance of the algorithm does not degrade to the same extreme as it does for binary encoded genetic algorithms, making the orthogonal array initialisation routine far more reliable. It must also be noted that the difference of performance is not so marked between the standard genetic algorithm as it is for those initialising with orthogonal arrays, making this method a more reliable means for floating-point algorithms and a possible substitute for random initialisation.

6 Further Work.

The results of the experiments reported here have inspired techniques. The new techniques exploit the observation that if the chromosomes created by orthogonal arrays are near the global optimum they perform well. This has lead to a new reproduction operator [Lee & Rowlands 1998], intelligent mutation operator[Lee & Rowlands 2000] and a new (non genetic) search algorithm based solely on the iterative use of orthogonal arrays[Lee 2003].

7. References

ASI Quality Systems , Orthogonal Arrays and Linear Graphs, ASI Quality Systems, UK

K Buckly et al, “Solution of Inverse Problems of a Highly Kinematically Redundant Manipulator Using Genetic Algorithms”, 1997, Proceedings of GALESIA ‘97, IEE.

R. Dawkins, 1987, The Blind Watch Maker, W. W. Norton & Company.

K. A. Dejong, 1975, An Analysis of the Behaviour of a Class of Genetic Adaptive Systems, Doctoral Thesis, University of Michigan

D. E. Goldberg, 1989, Genetic Algorithms in Search, Optimization & Machine Learning, Addison & Wesley.

J. H. Holland, 1975, Adaptation in natural and artificial systems, The University of Michigan Press.

M.Lee, 1989, Intelligent Robotics, Open University Press.

S. Lee and H Rowlands, 1998, Reproduction in Genetic Algorithms Using the Refocusing Operator. Poster Proceedings of ACDM '98 PEDC, University of Plymouth.

S. Lee,2002, Genetic Algorithms, Orthogonal arrays And Diploid Chromosomes, Doctoral Thesis, University of Wales, chapter 7.

S. Lee and H. Rowlands, 2000, A Psuedo-Mutation Operator Using Orthogonal Arrays in Genetic Algorithms to Improve Local Search. Mendel 2000, University of BRNO.

A Neubauer, 'The Cyclical Schema Theorem For Genetic Algorithms And Two-Point Crossover', 1997, Proceedings Of GALESIA '97,IEE.

E. Rich & K. Knight, 1991, Artificial Intelligence 2nd edition, pp 3-102, Mcgraw & Hill.

R. Roy, 1990, A Primer on the Taguchi Method, Van Nostrand Reinhold.

C. R. Reeves, "Using Genetic Algorithms with Small Populations", 1995, Proceedings of the Fifth International Conference on Genetic Algorithms, pp 92-99, Morgan Kaufmann Publishers, Inc.

8. Biographies

Dr Shane Lee

Senior Lecturer, School of Computing and Engineering –UW,N

His research field is computational intelligence in particular evolutionary computation and covers areas related to traditional engineering applications, computer games and art. He is the programme leader and creator of the BSc (hons) Games Development and Artificial Intelligence.

Outside academia he is a promoter and performer of stand-up comedy.

Professor Hefin Rowlands

Head of Research & Enterprise, School of Computing and Engineering –UW,N

Professor Hefin Rowlands is responsible in promoting, developing and leading research and enterprise activities in the University College's School of Computing and Engineering.

His research field, in the area of Quality Systems, covers research into Quality Management and Strategies including Six-sigma and Business Excellence Models.

Research interests also cover process modelling and developing an integrated model for business systems.

He has been involved in may successful TCS and College & Business Partnership (CBP) projects which enables local businesses to benefit from the expertise of a recently qualified graduate and the knowledge base of the School of Computing and Engineering at the University College.