Selecting Dimensions and Delay Values for a Time-Delay Embedding Using a Genetic Algorithm

James B. Vitrano
Department of Electrical and
Computer Engineering
Marquette University
Milwaukee, WI 53201 USA
jimv@jimv.net

Richard J. Povinelli
Department of Electrical and
Computer Engineering
Marquette University
Milwaukee, WI 53201 USA
richard.povinelli@marquette.edu

Abstract

This paper describes a novel technique for determining a useful dimension for a time-delay embedding of an arbitrary time series, along with the individual time delays for each dimension. A binary-string genetic algorithm is designed to search for a variable number of time delays that minimize the standard deviation of the distance between each embedded data point and the centroid of the set of all data points, relative to the mean distance between each data point and the centroid. The geometric transformations of rotation and scaling are added to the algorithm to allow it to identify attractors that are not aligned with the data axes. Several artificial and real-world attractors and time series are analyzed to describe the types of attractors favorable to the use of this technique.

1 INTRODUCTION

Time-delay embedding, or establishing a phase space representation of a system using current and delayed values from a sampled time series, is a useful technique for characterizing nonlinear behavior of a system (Abarbanel, 1995; Povinelli, 1999). Takens (1981) showed that an embedding of dimension greater than twice the dimension of a smooth manifold containing an attractor is a true embedding; i.e., the phase space is topologically equivalent to the state space of the attractor. Sauer and Yorke (1993) extended Takens' continuous-time work into discrete time and found that in many circumstances a lower embedding dimension is sufficient to represent the dynamics of the system.

When performing a time-delay embedding of a sampled time series, the two key questions to be answered are (1) how many embedding dimensions are required, and (2) what are the proper time delays, or lags, to use for each dimension? As described above, Takens, Sauer, and Yorke have theoretical answers to the first question. However, when facing a system with an attractor of unknown dimensionality, their theorems provide only general guidance. To test the adequacy of a particular embedding dimension, the false nearest neighbors technique (Kennel, Brown, and Abarbanel, 1992) examines the relative location of neighboring data points in the next higher dimension to determine whether the neighboring points remain neighbors in the higher dimension. Even with these techniques, selecting the proper embedding dimension for a particular time series seems to be as much art as it is science (Abarbanel, 1995).

Some more specific techniques are available to help answer the second question, finding the individual time delays for each dimension. Zeros or minima of the autocorrelation function of the time series have been mentioned as useful choices for time delays (Kantz and Schreiber, 1997), along with the first minimum of the time-delayed mutual information function (Fraser and Swinney, 1986). However, if these delays do not produce a useful embedding, little additional guidance is available.

This paper proposes the use of a binary-string genetic algorithm (GA) to search for the dimensionality and individual delay values for an embedding that best fits a given criterion – in this case, the minimum standard deviation of estimates of the radius of the attractor, compared to the mean of those radius estimates. While the GA amounts to a solution by trial and error, it represents an improvement in that it is an automated and directed trial-and-error solution.

2 CHARACTERISTICS OF THE GENETIC ALGORITHM

A genetic algorithm (GA) (Dumitrescu *et al.*, 2000), designed to emulate the natural principles of evolution, is an iterative technique for searching a large set of possible solutions to a problem for an optimal solution. In most GAs, a population of random solutions is generated, and the "fitness" of each solution in the population is calculated. Based on the fitness of each solution, a new generation of solutions is created such that the "fittest" solutions survive and combine into new possible solutions. Typically, some level of mutation is introduced into the new population to help prevent the GA from converging to a solution that is only locally optimal. This process is then repeated until a stopping criterion is met (e.g., a fixed number of generations, exceeding a fitness threshold, or domination of the population by one particular solution).

In a binary-string GA, each solution is represented by a series of binary digits, known as a "chromosome". After decoding each chromosome, evaluating the fitness of each solution, and selecting two "parents" to be combined, the combination is often performed using a "crossover" technique, where a portion of one parent's chromosome is combined with a portion of the other parent's chromosome. Mutations are usually performed by inverting one or more bits within the chromosome.

The binary-string GA used in this paper was selected because a software implementation of the GA was already available to the authors. The GA uses a fixed, predetermined population size and number of generations.

The most dramatic difference from the "standard" GA described above is in the method of selecting and combining parents. A preset number of the least-fit members of the population are not allowed to be selected as parents. A preset number of the most-fit members of the population are copied directly into the next generation, and in addition can be selected as parents. Members in the remaining "middle-fit" portion of the population are also able to be selected as parents. For each slot in the new population not occupied by the copies of the most-fit members, two parents are selected at random (with equal probability) from the set of eligible parents, and a byte-wise crossover is performed where each byte of the child's chromosome has a 50% probability of being copied from either parent. Single-bit mutations are also placed in the child's chromosome randomly at a preset rate.

The specific GA implementation was not studied in much detail. This may be an area for future research and improvement. In particular, a more efficient breeding strategy may result in more rapid convergence to an optimal solution.

3 THE TIME-DELAY EMBEDDING CHROMOSOME

The chromosome used with the GA is designed to be simple to decode into its corresponding time-delay embedding. The chromosome contains a fixed number of possible embedding dimensions. These dimensions are combined with a fixed first dimension, which (when not rotated as described later) corresponds to $x(t)$, the current sample from the time series. The maximum number of possible dimensions is preset by the user. This represents one of the methods of limiting the dimensionality of the set of possible solutions.

The format of the chromosome is shown in Figure 1 below:

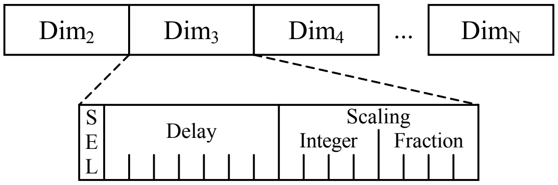

Figure 1: The format of the time-delay embedding chromosome (without rotation).

Each embedding dimension contains a single "selector" bit, which controls whether the dimension is considered when the chromosome is decoded into a time-delay embedding. Another seven bits are allocated for the time delay value corresponding to that dimension, allowing each dimension to contain a delay of between 0 and 127 samples. Another eight bits contain a scaling factor along that dimension's axis, ranging from 1/16 to 15 15/16 in increments of 1/16. This allows, for example, a properly aligned, oval-shaped, two-dimensional attractor to be expanded or compressed along the two embedding axes to nearly form a circle, which the fitness function described below recognizes as optimal.

The chromosome generation and decoding routines used with the GA may also be configured to allow the time-delay embedding to be rotated in space. In the scaling example above, the oval-shaped two-dimensional attractor needed to be "properly aligned", i.e., its major and minor axes needed to be roughly parallel to the coordinate axes. Allowing the GA to search through possible rotations allows the GA to rotate a misaligned oval so that it is properly aligned, then scale it to be roughly circular, thus producing a nearly optimal fitness value.

A rotation operation affects only two coordinates of a point, regardless of the number of dimensions (Burbanks, 1996). If rotation is enabled, an additional 8-bit field is appended to the chromosome for each possible pair of dimensions, resulting in $n_d(n_d\text{-}1)/2$ possible rotations, where n_d is the maximum number of dimensions allowed in the embedding. Only those rotations where both dimensions in the pair are enabled by their respective selector bits are performed. The 8-bit field allows for 256 possible rotations in the dimension pair, resulting in resolution of approximately 1.4 degrees. The rotation is performed by multiplying a transformation matrix by the coordinate vector of the data point (Hoggar, 1992). For example, Equation 1 shows a rotation in dimensions 1 and 3 of a 5-dimensional point:

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ x_5' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \qquad (1)$$

If rotation is enabled, it is performed before the scaling operation. This was done with the misaligned oval-shaped attractor in mind: performing the scaling along the coordinate axes before the rotation would have made the transformation from oval to circle impossible. There may be other cases where performing the scaling first would provide a benefit. Providing for both a pre-rotation and a post-rotation scaling may be another possible improvement to this technique.

None of the field sizes chosen for this chromosome appear to be "magical", i.e., they can most likely be varied to suit an individual application without harming the ability of the GA to find a useful embedding. If a user has reason to believe that, for example, a scaling factor larger than 16 may be needed in some dimension, the chromosome can certainly be modified to allow this. Adding multiple selector bits in each dimension, which are XOR'ed together to determine whether a given dimension is included, may also provide interesting results by taking increased advantage of the mutation feature of the GA.

4 THE FITNESS FUNCTION

The fitness function is a key component of the GA: it controls which members of the population are represented in the next generation. Because the "most fit" members are selected most often for reproduction, the GA tends to find the maximum of the fitness function over many generations (Dumitrescu *et al.*, 2000). If a minimization is needed instead, a simple approach is to make the fitness function the negative of the original function.

The fitness function used in this technique assumes that all data points lie near an attractor in phase space, and that the attractor can be rotated and scaled to produce a roughly constant radius in all dimensions. The GA locates the centroid of the data points in phase space, calculates the Euclidean distance between each data point and the centroid, and then uses statistical properties of the distance values *d* to provide a fitness judgment:

$$f(d, n_d) = -\frac{\sigma_d}{\mu_d} \cdot b^{n_d} \qquad (2)$$

In Equation 2 above, σ_d represents the standard deviation of the distances, and μ_d represents the mean of the distances. The standard deviation is scaled by the reciprocal of the mean so that the GA does not favor smaller attractors over larger ones. The n_d parameter represents the number of dimensions, and *b* is a constant bias (≥1) toward a smaller number of dimensions. The bias causes a lower-dimensionality embedding to be rated as more fit than a higher-dimensionality embedding that is otherwise slightly more fit. This behavior may be desirable, for example, when seeking to view an embedding in two or three dimensions, or when working with the resulting embedding with limited computing resources. The bias causes the GA to add dimensions only when the added dimensions result in a fitness improvement. Values of *b* of 1.05 and 1.2 were used for the examples in this paper, and appeared to yield good general-purpose results.

The fitness function is negative to cause the GA to seek embeddings that minimize the relative standard deviation in the distance measurements.

Based on the description above, it is clear that the attractor geometry for which this technique is ideally suited is a hypersphere. With a sufficient number of noise-free samples, the centroid will be calculated at the center of the hypersphere, and thus all samples will have an equal distance from the centroid, yielding an optimal fitness value of 0. However, the technique is not necessarily limited to attractors that are hyperspheres. Many other geometric shapes and real-world attractors have roughly uniform radii, as shown in Table 1 below. The noise-free fitness values in Table 1 were calculated by randomly placing 1,000 points on or near the surface of each attractor, and removing the dimensionality bias shown in Equation 2. The noisy fitness values in Table 1 were calculated similarly, except that random Gaussian noise with RMS magnitude $0.1 \bullet \mu_d$ was added to each point:

Table 1: Partial fitness values ($-\sigma_d/\mu_d$ ratios) of several geometric and non-time-delay real-world attractors.

Dimension	Attractor Description	Noise-Free Fitness	Noisy Fitness
2	Circle	-0.0130	-0.0885
2	Hexagon	-0.0510	-0.0869
2	Square	-0.1093	-0.1353
2	Van der Pol oscillator limit cycle (Vidyasagar, 1993)	-0.1723	-0.1849
3	Sphere	-0.0094	-0.0638
3	Cube	-0.1229	-0.1386
3	Torus (o.d.=2•i.d.)	-0.2265	-0.2316
3	Lorenz attractor (Abarbanel, 1995)	-0.5453	-0.5445
3	Rössler attractor (Frazer and Swinney, 1986)	-0.4086	-0.4060

The Lorenz and Rössler attractors both exhibit a “folded” geometry, i.e., most of the samples fall near one of two planes that intersect at nearly right angles. Because this geometry is quite different than the spherical geometry that this technique was designed to seek, the fitness values for these two attractors are quite low. These examples point out one of the limitations of this technique: it searches for a time-delay embedding that best meets its goal of a uniform radius between the samples and the centroid, even if an attractor with a different geometry is responsible for the dynamical behavior of the system. However, if a reasonable guess about the geometry of the attractor can be made, a different fitness function that favors that particular geometry can be used.

5 RESULTS

To illustrate the technique, a simple test pattern was devised. A two-dimensional time-delay embedding of a sinusoidal signal should produce a circle if a proper delay (for example, ¼ of the oscillation period) is chosen. To test the technique, 1,000 samples of the time series shown in Equation 3 were presented to the GA to find an embedding with a maximum dimension of 7:

$$x(t) = \sin\left(\frac{2\pi}{\sqrt{2455}}t\right) + noise \qquad (3)$$
$$noise \in N(0, 0.1)$$

Table 2 below shows the two-dimensional embedding found by the GA.

Table 2: The time-delay embedding parameters found by the GA for a noisy sinusoidal time series.

	Dim. 1	Dim. 2
Delay value	0	15
Rotation vs. dim. 1 (radians)	N/A	2.90
Scaling factor	1	0.9375
Overall fitness value	-0.1665	
$-\sigma_d/\mu_d$ ratio	-0.1156	

Figure 2, a plot of the time-delay embedding, shows that the GA did indeed find a circular attractor. The dots in the plot represent the time series samples, and the cross represents the calculated centroid:

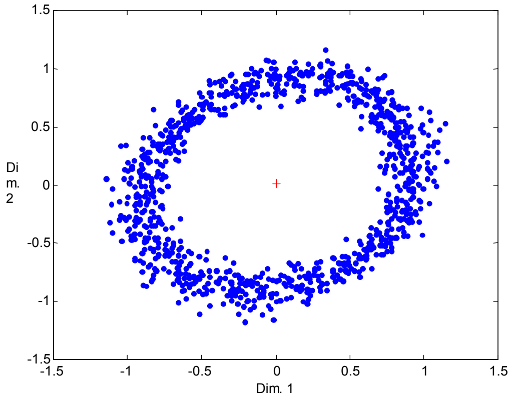

Figure 2: Plot of the time-delay embedding found by the GA for a noisy sinusoidal time series.

Interestingly, another run of the sinusoidal time series with noise recalculated from the same distribution found a three-dimensional solution shown in Table 3 and Figure 3, with rotation and scaling that produce a ring-shaped attractor in the three-dimensional phase space:

Table 3: Parameters for a three-dimensional time-delay embedding found by the GA for a noisy sinusoidal time series.

	Dim. 1	Dim. 2	Dim. 3
Delay value	0	25	62
Rotation vs. dim. 1 (radians)	N/A	4.52	5.82
Rotation vs. dim. 2 (radians)	N/A	N/A	0.07
Scaling factor	1	3.8750	1.5625
Overall fitness value		-0.1742	
$-\sigma_d/\mu_d$ ratio		-0.1008	

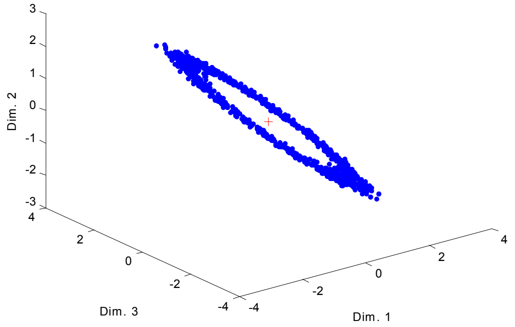

Figure 3: Plot of the three-dimensional time-delay embedding described in Table 3.

The three-dimensional result points out another characteristic of this technique: noise can cause the GA to find a more complex embedding (e.g., higher dimensionality or non-intuitive time delay values, rotation, or scaling) than might be required for a particular data set. The b parameter in the fitness function can be varied to compensate for the effect of noise on the dimensionality of the embedding found by the GA. Since this technique does not provide a similar mechanism for constraining rotation or scaling, minimizing measurement noise makes the GA more likely to find an attractor that is based on the actual dynamics of the system instead of the noise.

A more complex test was also presented to the GA, a simulation of a system governed by two attractors. An interleaved sinusoidal time series was developed by generating a time series using Equation 3 and doubling every second $x(t)$ value. Depending on the time delay chosen, an embedding of this series may either separate or combine the two attractors. In this series, a time delay of an even number of samples results in an embedding that appears as two concentric circles, thus allowing the attractors to be separated visually. A delay of an odd number of samples gives an embedding that combines the attractors into one shape. The GA determined that the combined attractor had a more uniform radius than the union of the two separated attractors, and thus found an odd embedding delay:

Table 4: Parameters for a two-dimensional time-delay embedding found by the GA for the dual interleaved sinusoidal time series.

	Dim. 1	Dim. 2
Delay value	0	13
Rotation vs. dim. 1 (radians)	N/A	3.14
Scaling factor	1	0.8125
Overall fitness value		-0.3730
$-\sigma_d/\mu_d$ ratio		-0.2590

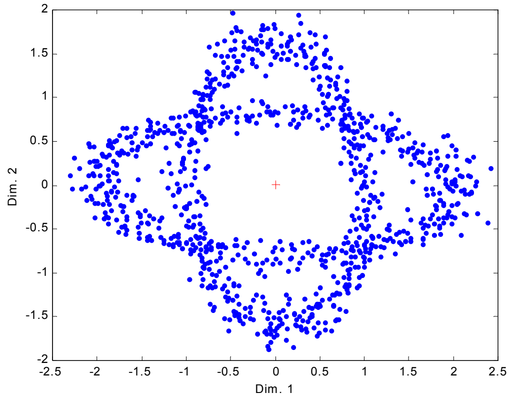

Figure 4: Plot of the time-delay embedding described in Table 4.

Having produced reasonable results with test data, this technique was then applied to several real-world attractors. One example is a time series representing the temperature of a room heated with a boiler and radiator. The time series is a set of 1,000 samples taken every two minutes during a simulation of a nonlinear model of the heating system. The time series was provided to the GA with a maximum dimensionality of 7. The GA reported the results in Table 5 and Figure 5:

Table 5: Parameters for a two-dimensional time-delay embedding found by the GA for the boiler/radiator time series.

	Dim. 1	**Dim. 2**
Delay value	0	88
Rotation vs. dim. 1 (radians)	N/A	0.61
Scaling factor	1	5.5
Overall fitness value	-0.5515	
$-\sigma_d/\mu_d$ ratio	-0.3830	

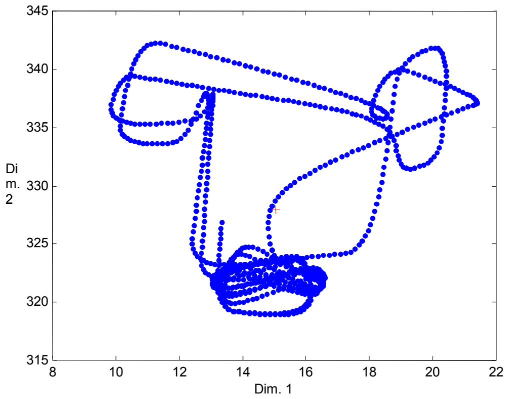

Figure 5: Plot of the boiler/radiator time-delay embedding described in Table 5.

In this case, the GA has found an interesting two-dimensional attractor. Three lobes appear in the plot, one of which is more densely populated than the others. While the attractor found by the GA is not circular, a dominant radius range does exist: a clear majority (693 of the 1,000) of the data points fall within a radius range of between 4 and 10 units. Figure 6 shows a histogram of the radii calculated for the data points. More importantly, the attractor does have some potentially useful structure.

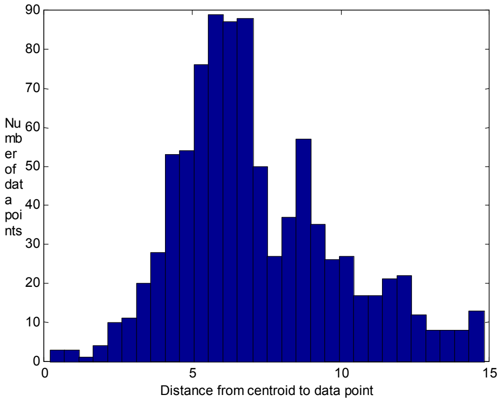

Figure 6: Histogram of radii between data points and centroid for the boiler/radiator time delay embedding described in Table 5.

Another interesting, and potentially lucrative, problem is to attempt to produce a model of the price of a stock. A 1,263-point time series containing the daily percentage price changes in the common stock of General Electric Co. between November 27, 1995, and November 24, 2000, was assembled and provided to the GA to find an attractor with a maximum dimensionality of 20. The results, shown in Table 6 and Figure 7 below, are essentially a three-dimensional cloud that indicates that the GA did not truly find an attractor:

Table 6: Parameters for a three-dimensional time-delay embedding found by the GA for a stock price time series.

	Dim. 1	**Dim. 2**	**Dim. 3**
Delay value	0	76	108
Rotation vs. dim. 1 (radians)	N/A	2.63	4.54
Rotation vs. dim. 2 (radians)	N/A	N/A	3.02
Scaling factor	1	12.75	14.75
Overall fitness value		-1.1257	
$-\sigma_d/\mu_d$ ratio		-0.6514	

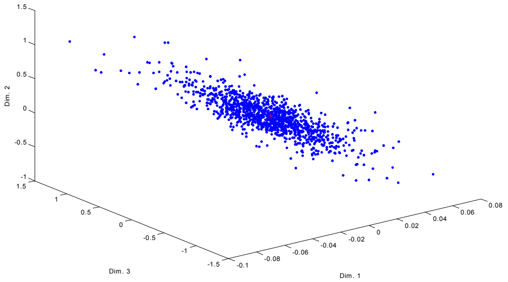

Figure 7: Plot of the stock price time-delay embedding described in Table 6.

Because the GA always reports the solution that it found to be best, there is no guarantee that the GA's solution is meaningful. In this case, it clearly is not. Unfortunately, the GA gives little guidance as to why it did not produce a meaningful result: perhaps the attractor requires more than 20 dimensions to become apparent, or perhaps the real attractor in this system has a geometry that is significantly different than a sphere. The higher-dimensionality case can be tested by allowing a larger maximum dimension, although there seems to be a point of diminishing returns in increasing dimensionality. The different geometry case can also be tested with a set of fitness functions tailored to different attractor geometries.

In some ways, this is a lucky result in that the three-dimensional view clearly shows that no attractor was truly found. This raises the question of how to detect a failure of this technique in higher dimensions, where a plot becomes infeasible. Two possible methods emerge from this example. One is to examine the fitness value of the GA's best solution. Knowing the fitness value, the dimensionality bias *b*, and the number of dimensions found by the GA, the ratio $-\sigma_d/\mu_d$ can be back-calculated and used as a measure of the uniformity of the attractor radius. This ratio can be compared directly to the values shown in Table 1. Based on the values in Table 1 and the examples in this paper, a ratio value of above approximately -0.4 tends to suggest that a meaningful attractor was found, and a ratio value of below -0.6 tends to suggest failure. Ratio values between these cutoffs are probably inconclusive, but may very well indicate success with noisy data.

Another possible method for determining the meaningfulness of a result is to examine the scaling factors on the various dimensions. If the scaling factors are dissimilar in magnitude, as in the stock price example, this suggests that the GA may have maximized the $-\sigma_d/\mu_d$ ratio with a meaningless combination of rotations and scaling.

One final example attempts to find a model for seismic activity. A 1,000-sample time series was taken from the east-west, broadband, high-gain sensor at the MA2 seismograph station at Magadan, Russia, during an earthquake of magnitude 6.4 that occurred on July 30, 2000, south of the Japanese island of Honshu (33.92 degrees north, 139.28 degrees east) (IRIS web site). Because previous investigation by the authors suggested that the attractor is likely to be of high dimensionality, the dimensionality bias b was relaxed to a value of 1.05 so as not to excessively penalize higher-dimensionality solutions. A maximum dimensionality of 20 was set.

The GA produced an 8-dimensional solution, summarized in Table 7. The rotations between the eight dimensions are omitted for clarity. Because of the high dimensionality, it is impractical to plot the data in the 8-dimensional phase space.

Table 7: Selected parameters for an eight-dimensional time-delay embedding found by the GA for a seismic time series.

	Delay value	**Scaling factor**
Dim. 1	0	1
Dim. 2	7	8.4375
Dim. 3	19	1.1875
Dim. 4	24	9.5
Dim. 5	26	9.625
Dim. 6	41	10.5625
Dim. 7	107	6.6875
Dim. 8	110	6.875

Overall fitness value	-0.6136
$-\sigma_d/\mu_d$ ratio	-0.4153

When examining this result, the natural question to ask is whether the result is meaningful. Because of the high dimensionality, the simplest tool to help determine this, plotting the data in the phase space, is not practical.

Examining the $-\sigma_d/\mu_d$ ratio and applying the benchmarks described earlier in this paper provide some guidance. The ratio value of -0.4153 falls in the inconclusive range, but it is very near the ratio found with the boiler/radiator system. Since the time series was sampled during an earthquake, presumably with a fair amount of noise present from sources such as seismic wave reflections from the earth's surface, the GA's result seems to be believable. The dimensionality of 8 also seems plausible, based on earlier investigation by the authors that showed no clear or emerging pattern in a number of embeddings in two and three dimensions.

The other tool developed in the stock price example, examining the scaling factors on each dimension for similarity, tends to lend some believability to the meaningfulness of the embedding: six of the eight dimensions have scaling factors between 6.6875 and 10.5625. The remaining two dimensions with scaling factors of 1 and 1.1875 may prove to be unnecessary. However, for now, all that can be said about this embedding is that it is an interesting candidate for further study.

6 CONCLUSIONS

Based on the examples in this paper, the GA technique appears to be a viable method for identifying appropriate time-delay embeddings for certain types of attractors. The technique works well when the attractor has a relatively uniform radius in phase space.

The main factor to consider when using this technique is that the GA finds an embedding that optimizes its fitness function, given the time series provided. Noise in the data can cause the GA to converge to an embedding that is more complex than necessary. In cases where the true attractor has a radius that is far from uniform, the GA still converges to the embedding that it found gives the data the most uniform radius. In some cases, this embedding represents the data as a random jumble of points, yielding no useful information. This paper discusses some techniques to help determine if this has occurred on a given data set, but at some point the user must decide whether to pursue higher dimensionality, different attractor geometries, or an entirely different technique.

The area of alternate attractor geometries appears to be an interesting area for future research. To change this technique to operate on a different attractor geometry, a new fitness function that detects that particular geometry is needed. It is even conceivable that the GA could select among a number of geometries for a given embedding, assuming that the fitness functions can be balanced so that a particular fitness value represents a the same level of fitness across the different geometries.

Another interesting area for future research is optimization of the GA itself. Faster convergence and speed optimization translate directly to decreased time to reach a result. The GA used in this paper is a simple, generic GA; it is likely possible to tailor the GA for faster convergence in this application.

While this technique is certainly not a "silver bullet" to find the optimal time-delay embedding for any time series, it is another tool that, in many circumstances, can provide useful results.

References

Henry D. I. Abarbanel, *Analysis of observed chaotic data*. New York: Springer-Verlag, 1995.

Andy Burbanks, *Gallery of mathematics: hyperspace structures – the hypercube*, Oct. 28, 1996, http://info.lboro.ac.uk/departments/ma/gallery/hyper/cube.html.

D. Dumitrescu, B. Lazzerini, L. C. Jain, and A. Dumitrescu. *Evolutionary computation*. Boca Raton, Fla.: CRC Press, 2000.

A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information", *Physical Review A*, no. 33, p. 1134-1140, 1986.

S. G. Hoggar, *Mathematics for computer graphics*. Cambridge, U.K.: Cambridge University Press, 1992.

Incorporated Research Institutions for Seismology, http://www.iris.washington.edu/.

Holger Kantz and Thomas Schreiber, *Nonlinear time series analysis*. Cambridge, U.K.: Cambridge University Press, 1997.

M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining minimum embedding dimension using a geometrical construction", *Physical Review A*, no. 45, p. 3403-3411, 1992.

Richard J. Povinelli, *Time series data mining: Identifying temporal patterns for characterization and prediction of time series events*. Doctoral dissertation. Milwaukee, Wis.: Marquette University, 1999.

T. Sauer and J. A. Yorke, "How many delay coordinates do you need?", *International Journal of Bifurcation and Chaos*, no. 3, 1993.

F. Takens, "Detecting strange attractors in turbulence", *Lecture Notes in Mathematics*, no. 898, 1981.

M. Vidyasagar, *Nonlinear systems analysis*, 2nd ed. Upper Saddle River, N.J.: Prentice-Hall, 1993.