Efficient implementation of inverse approach for forecasting hydrological time series using micro GA

S. Y. Liong, K. K. Phoon, M. F. K. Pasha and C. D. Doan

ABSTRACT

This paper implements the inverse approach for forecasting hydrological time series in an efficient way using a micro-GA (mGA) search engine. The inverse approach is based on chaos theory and it involves: (1) calibrating the delay time (τ), embedding dimension (m) and number of nearest neighbors (k) simultaneously using a single definite criterion, namely optimum prediction accuracy, (2) verifying that the optimal parameters have wider applicability outside the scope of calibration, and (3) demonstrating that chaotic behaviour is present when optimal parameters are used in conjunction with existing system characterization tools. The first stage is conducted efficiently by coupling the Nonlinear Prediction (NLP) method with mGA using a lookup facility to eliminate costly duplicate NLP evaluations. The mGA-NLP algorithm is applied to a theoretical chaotic time series (Mackey–Glass) and a real hydrological time series (Mississippi river flow at Vicksburg) to examine its efficiency. Results show that: (1) mGA is capable of producing comparable or superior triplets using only up to 5% of the computational effort of all possible points in the search space, (2) the lookup facility is very cost-effective because only about 50% of the triplets generated by mGA are distinct, (3) mGA seems to produce more robust solutions in the sense that the record length required to achieve a stable optimum triplet is much shorter, and (4) the prediction accuracy is not sensitive to the parameter k. It is sufficient to use $k = 10$ in future studies. In this way, the 3D search space could be reduced to a much smaller 2D search space of m and τ.

Key words | chaotic time series, inverse approach, micro genetic algorithm, nonlinear prediction

S. Y. Liong (corresponding author)
K. K. Phoon
M. F. K. Pasha
C. D. Doan
Department of Civil Engineering,
Blk E1A, #07-03, 1 Engineering Drive 2, National University of Singapore,
Singapore 117576
Tel: +65 6874 2155;
Fax: +65 6779 1635;
E-mail: *cvelsy@nus.edu.sg*

NOTATION

k	number of nearest neighbors in state space
x_i	observed value in calibration/production set
$\bar{x}$	mean observed value in calibration/production set
$\hat{x}_i$	predicted value of x_i
m	embedding dimension
n	total number of data in calibration/production set
τ	delay time
N	population size in genetic algorithm
Δ	shift parameter for Mackey–Glass time series

ABBREVIATIONS

GA	Genetic Algorithm
mGA	Micro-Genetic Algorithm
NLP	Nonlinear Prediction
NRMSE	Normalized Root Mean Square Error
SSR	State Space Reconstruction
MG	Mackey–Glass Time Series
CIA	Correlation Integral Analysis

INTRODUCTION

State space reconstruction from a chaotic time series is fundamental to both system characterization and forecasting. One usually attempts to reconstruct an attractor from the observed scalar data that preserves the invariant characteristics of the original unknown attractor by using an appropriate embedding dimension, m, and delay time, τ (delay coordinate method). Forecasting can be carried out by

following the trajectories of k nearest neighbors in this state space and assuming that some weighted average of these trajectories will produce the desired future state (nonlinear prediction or NLP). While these ideas are generally accepted, numerical determination of the triplet (m, τ, k) in actual practice is exceedingly difficult because observed time series are necessarily limited in length, sampled at discrete intervals and possibly corrupted by noise. To the best of the authors' knowledge, only limited guidelines are available on the selection of (m, τ, k) under such conditions. More fundamentally, it is uncertain at present whether the embedding derived from attractor considerations will be the same as that derived from forecasting considerations.

In the absence of more general theoretical results that can account for the vagaries in observed data, Phoon *et al.* (2002) suggested that it is more sensible to determine the triplet (m, τ, k) during the forecasting stage rather than following the standard approach described above. The main objective of this study is to assess the suitability of using micro-GA (mGA) to optimize the highly nonlinear objective function in the (m, τ, k) parameter space established by this inverse approach for forecasting hydrological time series. Micro-GA is supported by a very small population size and this key feature can be exploited to great advantage in this study because costly NLP evaluations are kept to a minimum. Micro-GA also inherits all the advantages of simple GA, among which its amenability to parallel computing is the most important for the inverse problem. Although the primary reason for applying mGA is to expedite the search process and reduce computational cost in the first stage of the inverse approach, it is conceivable that mGA could also produce more robust solutions as a result of its global perspective. In this paper, the behavior of mGA with respect to cost and robustness is studied using the Mackey–Glass time series and the Mississippi river runoff time series at Vicksburg. The former is a theoretical chaotic time series that has been studied by Phoon *et al.* (2002). Hence, the performance of mGA could be compared with the brute force approach. The latter will serve to demonstrate the applicability of the proposed scheme on real data.

INVERSE APPROACH

The inverse approach proposed by Phoon *et al.* (2002) involves: (1) calibrating the delay time, embedding dimension and number of nearest neighbors simultaneously using a single definite criterion, namely optimum prediction accuracy, (2) verifying that the optimal parameters have wider applicability outside the scope of calibration, and (3) demonstrating that chaotic behavior is present when optimal parameters are used in conjunction with existing system characterization tools such as correlation integral analysis. This approach represents a novel departure from the standard approach since: (1) forecasting precedes system characterization (hence an "inverse" approach) and (2) state parameters are evaluated during forecasting rather than system characterization. It must be acknowledged that Casdagli (1989) proposed a similar idea to construct a robust predictive model directly from time series data. The author treats prediction as an "inverse problem" that can be used to establish a nonlinear map for forecasting purposes when the data are limited and noisy.

The first stage in the inverse approach essentially involves searching a three-dimensional parameter space for the triplet that produces the most accurate prediction. There are three practical computational issues to consider. First, evaluation of each triplet is very costly because it requires running NLP to predict a sufficient number of data points (typically hundreds) to get a robust estimate of prediction accuracy. Each NLP itself requires a tedious search for nearest neighbors in a state space containing a large number of delay vectors (typically thousands). Second, the number of possible triplets in the parameter space is potentially huge. Third, the objective function (prediction accuracy) is a highly nonlinear function of the triplet with many local optima. The first two issues could result in runtimes that are unacceptably long while the third issue could trap the search process in a local optimum. Phoon *et al.* (2002) circumvented the local optimum problem by applying a brute force search procedure. However, the procedure is extremely costly because a large number of triplets (and hence NLP runs) need to be examined. It is possible to expedite the procedure by sampling the triplets over a coarse grid but there is a significant possibility of missing the global optimum given the highly nonlinear objective function.

The cost per evaluation of the objective function is fixed unless a more efficient alternative to NLP is available and/or prediction accuracy can be estimated without making predictions explicitly. Therefore, the challenge is to locate

the global optimum with the least number of NLP runs, while avoiding local optima traps. The Genetic Algorithm (GA) is potentially very effective for this problem because of its global perspective and inherent parallelism (Deb 2001). In recent years, GA is gaining popularity as an optimization method in the field of water resources engineering. Wang & Zheng (1998) applied GA for groundwater management optimization and found that it is more efficient compared to other traditional methods. Jayawardena *et al.* (1997) used GA as the parameter optimization method for moisture and contaminant transport through partially saturated porous media. Liong *et al.* (2001) applied GA to derive the optimal values of catchment model parameters and then combine it with a neural network to derive desired information on the Pareto front. Babovic *et al.* (2000) employed a genetic algorithm to evolve an embedding for prediction of the water level at Punta Della Salute (Venice, Italy). Babovic *et al.* (2000) implemented an alternate efficient search procedure during *kd* trees. This implementation is comparable to a sorting procedure if k is small and the number of (m, τ) is of the same order as record length. Because the search is exhaustive, it is possible to locate the global optimum as well.

GENETIC ALGORITHM

The Genetic Algorithm (GA) is inspired by the process of natural selection in nature (Holland & Nafpliotis 1975). GA differs from other classical search and optimization methods in a number of ways. The main differences are: (1) it does not use gradient information, (2) it works with a set of solutions instead of one solution in each iteration, and (3) it works on a coding of solutions instead of the solutions themselves. GA provides a powerful alternative to problem solving based on random search – one in which solutions to the problem are evolved rather than the problems being solved directly.

Each parameter set, generally coded in binary, is called a chromosome. The fitness value associated with each chromosome depends on its performance. The total number of chromosomes in each iteration is known as the population size whereas each iteration is known as a generation. The size of a population is usually the same from one generation to the next. Although chromosomes in the first generation are usually generated randomly, chromosomes of the subsequent generations are generated through three basic mechanisms, namely selection, crossover and mutation. As the population evolves, new chromosomes will replace the older ones and the overall performance is supposed to improve. The chromosomes associated with "fit" individuals will be produced more often than the less fit chromosomes, following the well-known Darwinian principle of "survival of the fittest". This process is repeated until a prescribed stopping criterion based on fitness or number of generations is met.

Typically, the population size in GA ranges from 30–200. If the population size is too small, the genetic algorithm will converge too quickly. This might lead to a solution that is not near the global optima. However, a population with too many members results in a large number of iteration. Thus, a trial-and-error process is needed to determine the appropriate population size in GA. The mutation operator, which changes the bit value in the chromosome string randomly, is an operation to prevent premature convergence to local optima. A study by Schaffer *et al.* (1989) shows that population size and mutation rate play an important role in GA.

For problems where function evaluations are very expensive, many researchers have resorted to micro-GA with a small population size (Deb 2001). Micro-GA (mGA) derives its name from GA because it uses almost the same basic operations. It differs, however, from other GAs in two important aspects: (1) small population size and (2) no implementation of conventional mutation. The idea of mGA was suggested by some theoretical results obtained by Goldberg (1989). The micro-GA procedure can be summarized as follows:

1. Generate randomly a small population size (usually from 5–10).
2. Calculate the fitness of each chromosome. The fittest chromosome is automatically carried to the next generation (elitism). The mating pool is formed by the remaining chromosomes.
3. Produce offspring from the mating pool based on tournament selection.
4. Check the convergence of the population. If the population is converged, repeat the procedure from step 1, keeping the best chromosome and generating the other population randomly. Should the population have not converged, repeat the procedure from step 2.

Krishnakumar (1989) was the first to report the implementation of mGA. He compared his mGA against simple GA (with a population size of 50, a crossover rate of 0.6 and a mutation rate of 0.001). The author reported that mGA could achieve faster and better results on two stationary functions and a real world engineering control problem (a wind-shear controller task). The relatively small population is restarted, whenever they converge, by randomly selecting new parameter sets to replace the previous generation. However, the best parameter set from the previous population is also included in the new generation. This technique of keeping the best parameter set is known as elitism.

PROPOSED mGA-NLP WITH LOOK-UP FACILITY

In this study, mGA (Carroll 1996) is coupled with NLP to determine the triplet (m, τ, k) that optimizes prediction accuracy. The fitness function, or the objective function, used in this study to guide the mGA search is the normalized root mean square error and it is expressed as follows:

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}} \quad (1)$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (2)$$

where x_i = observed value, $\hat{x}_i$ = predicted value, $\bar{x}$ = mean of observed values and n = number of data points to be predicted in the calibration/production set. Note that prediction is perfect (i.e. $x_i = \hat{x}_i$) if *NRMSE* is zero. On the other hand, predictions are inferior to the constant mean value if *NRMSE* is greater than one. Hence, triplets producing low *NRMSE* over the calibration/production set are "fit". Equation (1) clearly shows that *NRMSE* is costly to evaluate because it involves n NLPs and the predicted value $\hat{x}_i$ produced by each NLP involves searching k nearest neighbors in a state space containing thousands of delay vectors. It is of practical interest to ascertain the smallest number of data points in the state space reconstruction set and calibration set that is required to achieve stable optimum triplets. A triplet is considered to be "stable" when m and τ converge to some values with increasing number of data points in the state space reconstruction and calibration sets. This important computational issue is systematically studied in the following two sections.

Figure 1 is a flowchart summarizing the operation of the coupled mGA-NLP. Note that NLP is implemented within mGA using an important lookup facility. This involves storing the fitness value associated with each unique triplet (m, τ, k) in a table so that costly duplicate NLP runs are avoided during mGA iterations. The lookup table is created "on the fly", i.e. new entries are inserted whenever mGA churns out triplets that are not present in previous generations. This novel feature is cost-effective because: (1) NLP is very costly in comparison to a search operation, (2) the number of unique triplets is finite because the

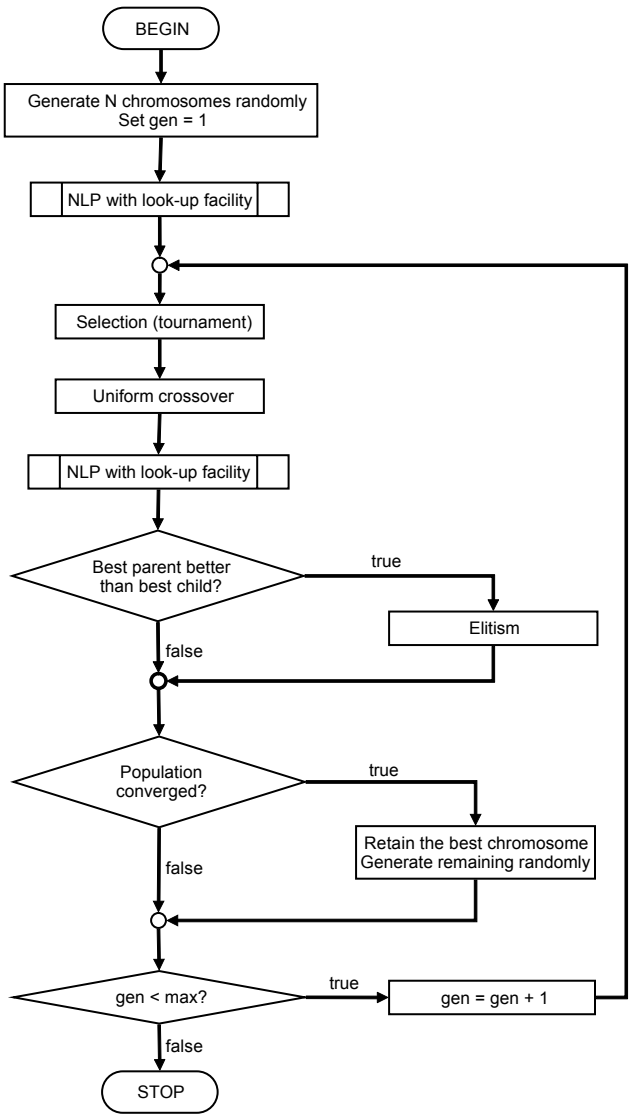

Figure 1 | Flowchart of mGA-NLP with lookup facility.

parameter space is discrete (integer) and bounded, and (3) a small population size converges very quickly to the same triplet between restarts in mGA. The cost savings are significant as shown in the following sections for the Mackey–Glass time series and the Mississippi river runoff time series.

The population size used in this study is $N = 10$. Each triplet is encoded using 45 binary bits. The tournament method is used for selection and a uniform crossover is applied with a crossover rate = 0.5. Elitism is practiced when the best offspring in the new generation is less fit than the best parent in the preceding generation. The best parent replaces one of the offspring randomly. The population is considered to have "converged" when more than 95% of the bits are identical or 9 out of 10 chromosomes result in the same fitness value. As shown in Figure 1, the best candidate is retained while the remaining 9 chromosomes are randomly generated afresh to re-start the procedure. The entire mGA-NLP algorithm is terminated when the number of generations exceeds 100 and 150 for the Mackey–Glass time series and the Mississippi river flow time series, respectively.

MACKEY–GLASS TIME SERIES

The well-known Mackey–Glass (MG) (Mackey & Glass 1977) model is given by

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-\Delta)}{1+x^{10}(t-\Delta)} \quad (3)$$

where Δ is the shift parameter. The equation can be solved numerically using the fourth-order Runge–Kutta method with the initial condition $x(t = 0) = 1.2$ and $x(t - \Delta) = 0.0$ for $0 \le t < \Delta$. A time step of 0.01 is used to solve the equation numerically. A moderately chaotic MG time series corresponding to $\Delta = 30$ is selected for study as shown in Figure 2. Note that the first 2000 data points, which represent the transient response, have been truncated from the time series under study.

To apply the inverse approach, the time series is divided into a state space reconstruction (SSR) set, a calibration set and a production set. The mGA-NLP algorithm discussed in the preceding section is used to tune the prediction parameters (m, τ, k) so that the calibration set can be predicted from the state space reconstruction set with maximum accuracy and minimum computational cost. The robustness of the solutions will be investigated by studying the effect of time series length on the optimized parameters. The performance of the optimized parameters at various lead times is then evaluated objectively using the production set. Finally, the Correlation Integral Analysis (CIA) is adopted to partially verify if chaotic behavior is present.

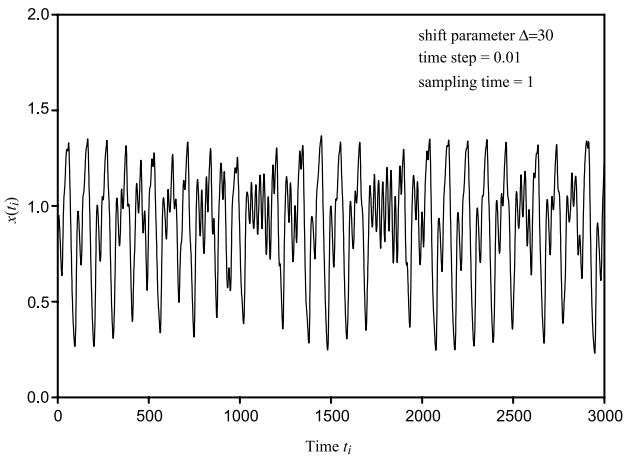

Figure 2 | Mackey–Glass time series, $x(t_i)$.

In the study reported by Phoon *et al.* (2002), the embedding dimension m is varied from 2 to 10 in steps of 1, delay time τ from 1 to 18 in steps of 1 and from 20 to 30 in steps of 2, and the number of nearest neighbors k from 1 to 10 in steps of 1 and from 15 to 50 in steps of 5. There are a total of 3888 ($9 \times 24 \times 18$) combinations in the search space. The same ranges of parameters are selected in this study but it is not necessary to use uneven step sizes for cost savings. In principle, mGA is capable of searching the entire grid in steps of 1, i.e. ($10 \times 30 \times 50$) = 15,000 combinations (note: m starts from 1 for mGA). Hence, mGA should locate a more superior triplet, but it is uncertain if it could locate it under 3888 *NRMSE* evaluations (only roughly 25% of 15,000) and if the superior triplet is more robust. Robustness is studied by varying the size of the SSR set while keeping the calibration set constant and vice versa. Investigation was conducted on prediction lead times of 1, 10, 30 and 50.

The results of the study are presented in Table 1 for lead time 10. Note that a fixed calibration set consisting of 250

Table 1 | Sensitivity of record lengths of state space reconstruction (SSR) set and calibration set on Mackey–Glass time series with different search engine (lead time 10)

State space reconstruction set		Calibration set	(m, τ, k) with minimum *NRMSE* mGA-NLP				Brute force–NLP (Phoon *et al.* 2002)		
Time series (1)	Length (2)	Length (3)	First optimal chaotic set (m, τ, k) (4)	Ranking among solutions (5)	*NRMSE* (6)	Number of *NRMSE* evaluations* (7)	First optimal chaotic set (m, τ, k) (8)	*NRMSE* (9)	Number of *NRMSE* evaluations (10)
State space reconstruction length study keeping calibration length constant									
$x_{8251}-x_{9000}$	750	250	(2, 24, 4)	1	0.195	525	(2, 24, 4)	0.195	3888
$x_{8201}-x_{9000}$	800	250	(3, 27, 15)	1	0.201	470	(2, 24, 2)	0.192	3888
$x_{7501}-x_{9000}$	1500	250	(3, 27, 18)	1	0.162	513	(2, 24, 4)	0.174	3888
$x_{6001}-x_{9000}$	3000	250	(3, 27, 18)	1	0.132	426	(2, 24, 25)	0.163	3888
$x_{4501}-x_{9000}$	4500	250	(3, 27, 20)	1	0.124	461	(3, 26, 20)	0.159	3888
$x_{3001}-x_{9000}$	6000	250	(3, 27, 3)	1	0.104	507	(3, 12, 7)	0.132	3888
$x_{1501}-x_{9000}$	7500	250	(3, 27, 2)	1	0.104	520	(3, 12, 7)	0.135	3888
$x_{1}-x_{9000}$	9000	250	(3, 27, 23)	1	0.099	484	(3, 12, 7)	0.135	3888
Calibration length study keeping state space reconstruction length constant									
$x_{1501}-x_{9000}$	7500	250	(3, 27, 2)	1	0.104	520	(3, 12, 7)	0.135	3888
$x_{1501}-x_{9000}$	7500	750	(3, 27, 2)	1	0.094	526	(3,12, 10)	0.118	3888
$x_{1501}-x_{9000}$	7500	1000	(3, 27, 2)	1	0.084	509	(3,12, 9)	0.105	3888

*Note that due to repetition of some combinations, the total numbers of evaluations are less than 1000 (for generation 100 and population size 10 total evaluation should be 100 × 10 = 1000).

data points from x_{9001} to x_{9250} is used in the first part of the study. The size of the SSR set is varied from 750 to 9000 data points. Since the two data segments must be connected for forecasting purposes, the last data point in all the state space reconstruction sets is x_{9000}. In the second part of the study, a fixed SSR set consisting of 7500 data points is used. The size of the calibration set is varied from 250 to 1000 data points. The performance of the optimized parameters outside the scope of the calibration set can be evaluated objectively using an independent third segment, which is called the production set. In the third part of the study, an independent production set consisting of 1000 data points (x_{10001} to x_{11000}) is used for this purpose together with a new SSR set of 8500 data points (x_{1501} to x_{10000}). A summary of the results obtained from the various lead times 1, 10, 30 and 50 are presented in Table 2. Note that the computational cost of each *NRMSE* evaluation is not a constant, but varies according to the lengths of the SSR set, calibration set and/or production set.

A comparison of columns 6 and 9 in Table 1 clearly shows that mGA is able to produce comparable or smaller *NRMSE* than brute force search. For prediction lead time 10, the optimum triplets at maximum SSR set (3, 27, 23) or at maximum calibration set (3, 27, 2) are superior to the corresponding ones derived using brute force search because $\tau = 27$ is left out of the brute force search grid (τ from 1 to 18 in steps of 1 and from 20 to 30 in steps of 2).

Note that brute force search requires 3888 *NRMSE* evaluations while mGA requires at most roughly 520 *NRMSE* evaluations (column 7). In other words, mGA is capable of producing comparable or superior triplets using less than 15% of the computational effort expended in brute force search. When compared with the theoretical maximum number of *NRMSE* evaluations required to search the denser mGA grid exhaustively (15,000), the cost saving is greater than 95%. The lookup facility incorporated in mGA is also very cost-effective. For a stopping criterion of 100 generations and a population size of 10, the number of

Table 2 | Prediction errors (*NRMSE*) using various approaches on Mackey–Glass time series

	mGA–NLP and brute force search methods				
	Optimal (m, τ, k) set from calibration study				**Optimal (m, τ, k) from standard approach**
	lead time 1 (m, τ, k)	**lead time 10 (m, τ, k)**	**lead time 30 (m, τ, k)**	**lead time 50 (m, τ, k)**	**(m, τ, k)**
Lead time to forecast production set x_{10001} to x_{11000}	**(2, 30, 6) (2,30,5)***	**(3, 27, 2) (3, 12, 9)***	**(7, 3, 11) (8, 3, 9)***	**(8, 4, 6) (4, 8, 6)***	**(6, 17, 7)**
1	0.014	0.043	0.067	0.064	0.072
	0.014*	0.059*	0.109*	0.058*	
10	0.478	0.057	0.232	0.073	0.173
	0.439*	0.074*	0.139*	0.077*	
30	0.421	0.378	0.061	0.076	0.151
	0.488*	0.173*	0.062*	0.073*	
50	0.748	0.336	0.245	0.116	0.210
	0.766*	0.240*	0.159*	0.123*	

*Resulting from brute force search conducted by Phoon *et al.* (2002).

NRMSE evaluations should be 1000. However, column 7 in both Tables 1 and 2 clearly shows that only about 50% of the 1000 chromosomes are distinct and a lookup table of the type implemented in this study will eliminate costly redundant NLP runs within each *NRMSE* evaluation. In addition, even lesser NLP runs are possible in mGA if the number of generations is reduced and a slightly less "fit" triplet is acceptable. This is possible because the evolutionary nature of GA applies a unidirectional pressure towards increasingly "fitter" descendant triplets (in a statistical sense) that would result in *NRMSE* decreasing roughly monotonically with generation count. A fully systematic or fully random brute force search will not possess this highly desirable monotonicity property.

For reasons not entirely clear at present, mGA also seems to produce more robust solutions in the sense that the record length (i.e. number of data points in one time series) required to achieve a stable optimum triplet is much shorter. Phoon *et al.* (2002) noted that the parameter k is not expected to be an invariant because it is not required in state space reconstruction. In addition, the impact of k on prediction accuracy is quite minimal as illustrated in Figure 3 for four embeddings. These results seem to confirm a brief remark made by Phoon *et al.* (2002) and potentially could make the inverse approach even more practical by eliminating the k parameter from the search grid. This reduces the search grid from 3D to 2D, which has a tremendous impact on computational cost. For example, the number of possible combinations for the dense mGA grid will reduce from $(10 \times 30 \times 50) = 15{,}000$ combinations to only $(10 \times 30) = 300$ combinations, a saving of 98% in this particular instance. Table 1 clearly shows that stable embeddings (m, τ) are achieved using only 800 data points in the SSR set. In contrast, brute force search requires 6000 data points in the SSR set to achieve stable embeddings at lead time 10. The ability to achieve stable embeddings at short record length is of practical interest because actual data is costly to collect and long historical time series are rare. As noted previously, a shorter SSR set and/or calibration set are more computationally efficient for *NRMSE* evaluation because there are fewer points to check in the search for nearest neighbours and there are fewer NLP runs, respectively.

The robustness of the solutions can also be evaluated by examining the performance of the optimized parameters outside the scope of the calibration set. Table 2 shows the *NRMSE* incurred in forecasting an independent production set at various prediction lead times. The values marked with asterisks are the results reported by Phoon *et al.* (2002) using the brute force method. Note that the leading diagonal (grey cells) contains *NRMSE* values for the special case in which the prediction lead time coincides with the calibration lead time used to optimize the embeddings. It is reassuring that the mGA results reflect the same trends as

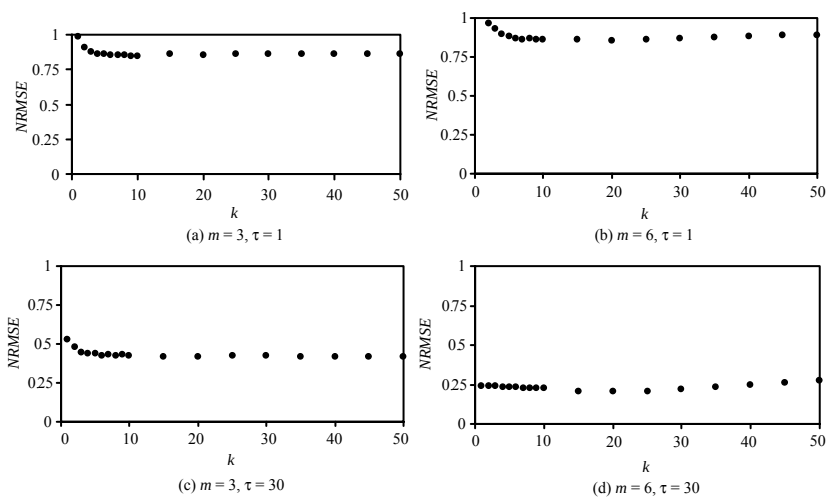

Figure 3 | Sensitivity of number of nearest neighbors (k) with varying embedding dimensions (m) and time delays (τ) at lead time = 10 for Mackey–Glass time series.

those exhibited by brute force search. In particular, Table 3 reaffirms the key observation made by Phoon *et al.* (2002) that the optimal embedding for forecasting is a function of lead time, even for an ideal noise-free, densely sampled and long chaotic time series. This observation is not merely of academic interest. For example, the optimal embedding of (3, 27, 2) calibrated using lead time 10 produces a very small $NRMSE = 0.057$ when the forecasting lead time is coincident but degrades roughly six-fold to $NRMSE = 0.336$ when the forecasting lead time is 50. This is clearly of practical engineering significance. It must be emphasized that the results in Table 3 are not merely a repetition of those produced by Phoon *et al.* (2002). They furnished significantly stronger support for previous observations made by Phoon *et al.* (2002) because the search grid for mGA is fully dense.

The third stage of the inverse approach, namely demonstrating that chaotic behavior is present in the optimal embeddings, is evaluated outside the mGA-NLP program using the correlation integral analysis (CIA) as shown in Figure 4. A clear scaling region and saturation of the correlation exponent are considered to be indicative of chaos. In this study, the mGA solutions were verified one at a time from the "fittest" solution downwards until fairly strong evidence of chaos is obtained using CIA. Hence, all the optimal embeddings reported in Tables 1 and 2 will exhibit the required system characteristics but, in principle, may not be the highest ranked solution in mGA. Their rankings are given in column 5 of Table 1. It is clear that the "fittest" solution (rank 1 solution) is almost always chaotic as well for this theoretical Mackey–Glass time series. Research is on-going at present to ascertain if this verification of system characteristics can be incorporated into mGA by augmenting the fitness function. In this way, the "fittest" solution would automatically provide the most accurate predictions and produce the most consistent chaotic behavior as well. Babovic & Keijzer (1999) and Babovic *et al.* (2000) made an implicit assumption that NLP is meaningful even if the time series is not chaotic.

Table 3 | Sensitivity of record lengths of state space reconstruction set and calibration set on Mississippi river time series (at Vicksburg) using mGA-NLP (lead time 1)

			(m, τ, k) with minimum *NRMSE*			
State space reconstruction (SSR) set		**Calibration set**	**mGA–NLP**			
Time series (1)	**Length (2)**	**Length (3)**	**First optimal chaotic set (m, τ, k) (4)**	**Ranking among solutions (5)**	***NRMSE* (6)**	**Number of *NRMSE* evaluations* (7)**
State space reconstruction length study keeping calibration length constant						
1984–1989	2192	1 year	(2, 1, 3)	1	0.04593	609
1981–1989	3287	1 year	(2, 1, 5)	1	0.04319	514
1978–1989	4383	1 year	(2, 1, 4)	1	0.04060	619
1975–1989	5479	1 year	(2, 1, 5)	1	0.03940	647
Calibration length study keeping state space reconstruction length constant						
1975–1989	5479	1 year	(2, 1, 5)	1	0.03940	647
1975–1989	5479	1.5 years	(2, 1, 5)	1	0.04120	601
1975–1989	5479	2 years	(2, 1, 5)	1	0.03564	660

*Note that due to repetition of some combinations, the total numbers of evaluations are less than 1500 (for generation 150 and population size 10 total evaluation should be 150 × 10 = 1500).

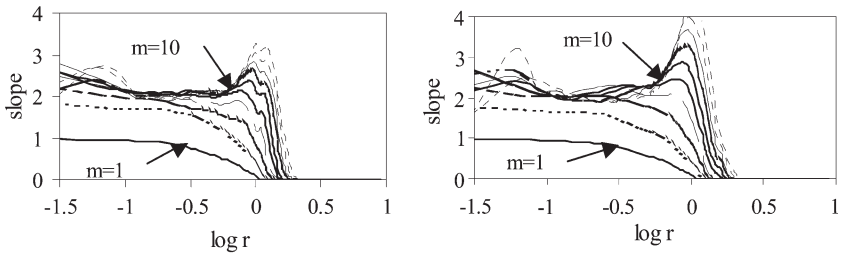

Figure 4 | Correlation integral analysis for Mackey–Glass time series.

MISSISSIPPI RIVER FLOW TIME SERIES AT VICKSBURG

The time series under study is the daily flow data from the Mississippi river at Vicksburg, station no. 07289000 (Hydrologic Region 08 of USGS). The data used for this study cover the period from 1 January 1975 to 31 December 1993. The river flow data at this station has the following statistics: mean flow = 18,456.54 m^3/s, standard deviation = 9727.72 m^3/s, maximum flow = 52,103.00 m^3/s and minimum flow = 3907.72 m^3/s. Following the study on Mackey–Glass time series, the entire data set is divided into three segments: (1) data from 1975–1989 are used for the sensitivity study of the state space reconstruction set length, (2) data from 1990–1991 are for the sensitivity study of the calibration set length, and (3) data from 1992 and 1993 are reserved for the production set. The search space is much larger in this example: m is varied from 1 to 10, τ from 1 to 60 and k from 1 to 200. The total number of potential combinations is $10 \times 120 \times 200 = 240{,}000$.

Three prediction lead times of 1, 3 and 5 days were studied. However, only results obtained from lead time 1 are shown in great detail (Table 3). It is interesting to note that, for lead times 1, 3 and 5, a stable embedding could be achieved at a relatively short record length of 6 years for the SSR set and 1 year for the calibration set. The stable embedding ($m = 2$, $\tau = 1$) is not a function of the lead time in this example, although the number of nearest neighbors k is different. This is different from that shown in Table 2. As to be expected, the smallest *NRMSE* increases from 0.0356 at lead time 1 to 0.1938 at lead time 5. Depending on the length of the SSR set and calibration set, the prediction accuracy in terms of *NRMSE* is between 0.1938 and 0.2237 for lead time 5. This is a notable achievement for a real-time series. For comparison, the prediction accuracy achieved at lead time 10 for the theoretical Mackey–Glass is at best between *NRMSE* = 0.084 and 0.195 (Table 1).

As indicated above, a stable embedding is not a function of lead times in the case of Mississippi river flow data. The lead times considered, however, clustered about a narrow range. For comparison, the corresponding embeddings for the Mackey–Glass time series were evaluated. The stable (m, τ) is (2, 30), (3, 30) and (3, 29) for lead times 1, 3 and 5, respectively. Although they are not numerically identical, they are practically identical because the stable (m, τ) for each lead time is ranked not lower than 9 when applied to other lead times. It would appear that the stable embeddings do not fluctuate rapidly with lead time.

The maximum number of *NRMSE* evaluations required for prediction lead times 1, 3 and 5 is only 688, which is less than 0.3% of the theoretical maximum in an exhaustive brute force search (240,000 combinations). Further significant cost savings are realized within mGA by the lookup table facility because the number of *NRMSE* evaluations could have been 1500 if every chromosome in the population size of 10 in all 150 generations is unique. Similar to the observation obtained in Mackey–Glass time series, the Mississippi river flow also shows that *NRMSE* is not sensitive to the parameter k after a certain value. The values of the parameter k for lead times 1, 3 and 5 are 5, 7 and 9, respectively. These values are comparable with the optimum k values in the Mackey–Glass time series, which range from 5–9 as shown in Table 2. Hence, it is probably sufficient to use $k = 10$ in future studies. In this way, the search space could have been reduced to $10 \times 120 = 1200$, which is only 0.5% of the original 3D search space with k as a free parameter.

The robustness of the optimal embedding derived from mGA-NLP is assessed objectively using the production set (maximum flow of 52,103 m^3/s) as shown in Table 4. It is noteworthy that the *NRMSE* from mGA-NLP is significantly lower than that using the standard approach and naïve shift. A more detailed comparison between the observed and the predicted runoff (with Standard Approach) using scatter plots is given in Figure 5. Note that τ is evaluated using the mutual information method and m is evaluated using Correlation Integral Analysis in the Standard Approach (Phoon *et al.* 2002). Figure 5 shows that the mGA-NLP approach consistently outperforms the standard approach for all lead days over the full range of runoffs. This is to be expected because the standard embedding would have emerged from the search process if it were to be optimal. More surprisingly, the standard approach fails to outperform the naïve shift as shown in Table 4. mGA-NLP is able to provide a more accurate prediction than naive shift, especially at higher runoffs. These results seem to demonstrate that it is not easy to outperform the naive shift even if the data is chaotic, unless an appropriate embedding is selected for NLP.

The results obtained above are supported by similar analyses conducted on other river flow time series (e.g. Kootenai and Salmon rivers in Idaho, USA). However, there are time series in which the inverse approach does not yield a significant improvement in prediction accuracy (e.g. Yukon river in Alaska). The chaotic model may not be appropriate for such time series. Note that the inverse approach is capable of rejecting the chaotic model hypothesis more decisively than the standard approach because practically all reasonable embeddings would have been tested and it is not possible to argue that a better embedding could be obtained using a different system characterization technique.

CONCLUSIONS

This paper implements the inverse approach proposed by Phoon *et al.* (2002) in an efficient way using a micro-GA (mGA) search engine. The Nonlinear Prediction (NLP) method is coupled to mGA using a lookup facility. This involves storing the fitness value associated with each unique triplet (m, τ, k) in a table so that costly duplicate prediction accuracy evaluations (in terms of normalized root mean square error or *NRMSE*) are avoided during mGA iterations. The lookup table is created "on the fly", i.e. new entries are inserted whenever mGA churns out triplets that are not present in previous generations. This feature is cost-effective because: (1) *NRMSE* evaluation (involving hundreds of NLPs) is very costly in comparison to a search operation, (2) the number of unique triplets is finite because the parameter space is discrete (integer) and bounded, and (3) a small population size converges very quickly to the same triplet between restarts in mGA. Further reduction in NLP runs is possible if the number of generations is reduced and a slightly less "fit" triplet is acceptable. The mGA-NLP algorithm with lookup facility is applied to a theoretical

Table 4 | Prediction errors (*NRMSE*) using various approaches for Mississippi river flow at Vicksburg

	mGA–NLP search method				
	Optimal (m, τ, k) set from calibration study			**Standard approach**	
	lead time 1 (m, τ, k)	**lead time 3 (m, τ, k)**	**lead time 5 (m, τ, k)**	**(m, τ, k)**	
Lead time to forecast production set (1992–1993)	**(2, 1, 5)**	**(2, 1, 7)**	**(2, 1, 9)**	**(6, 13, 7)**	**Naïve prediction**
1	0.0452	0.0442	0.0437	0.2064	0.0771
3	0.1533	0.1477	0.1437	0.3342	0.2162
5	0.2782	0.2685	0.2622	0.4835	0.3392

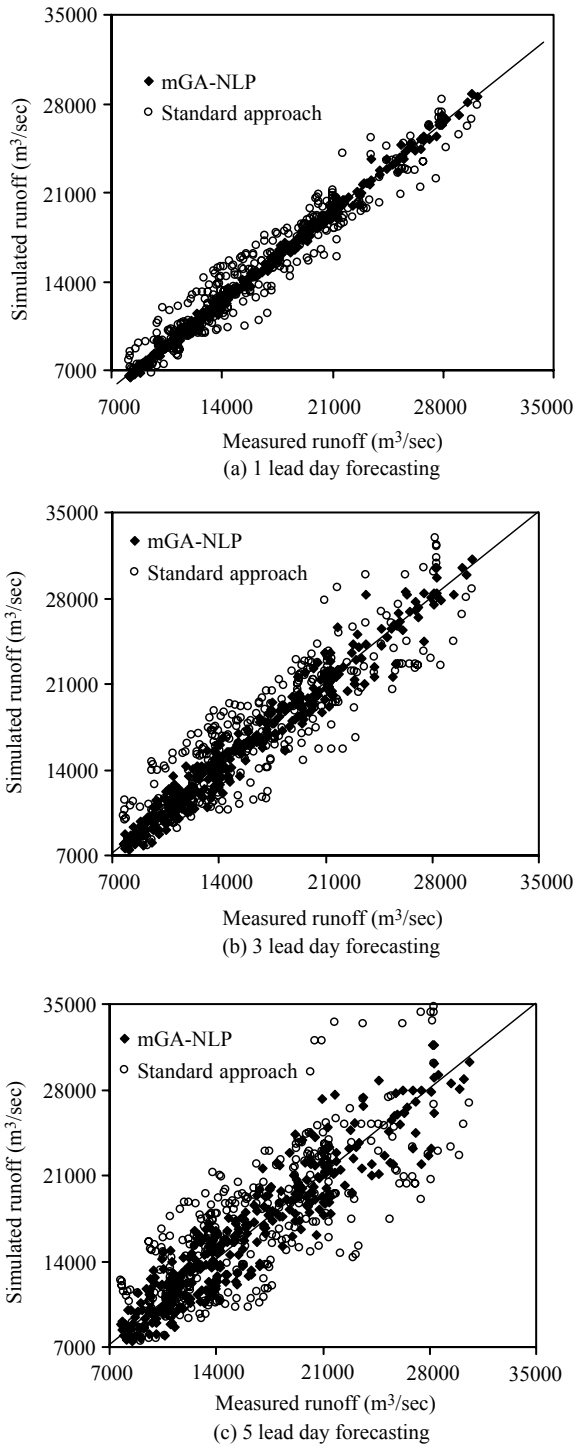

Figure 5 | Forecasting of daily Mississippi river runoff data (at Vicksburg) using mGA-NLP and standard approaches: production set.

chaotic time series (Mackey–Glass) and a real hydrological time series (Mississippi river flow at Vicksburg).

For the Mackey–Glass time series, mGA is capable of producing comparable or superior triplets using less than 15% of the computational effort expended in a brute force search conducted by Phoon *et al.* (2002) over a coarse grid. When compared with the theoretical maximum number of *NRMSE* evaluations required to search the denser mGA grid exhaustively (15,000), the cost saving is greater than 95%. The lookup facility incorporated in mGA is also very cost-effective. For a stopping criterion of 100 generations and a population size of 10, the number of *NRMSE* evaluations should be 1000. However, results show that only about 50% of the 1000 chromosomes are distinct and a lookup table of the type implemented in this study will eliminate costly redundant NLP runs.

For the Mississippi river runoff time series, a stable embedding of ($m = 2$, $\tau = 1$) could be achieved at a relatively short record length of 6 years for the SSR set and 1 year for the calibration set. Note that a shorter SSR set and/or calibration set are more computationally efficient for *NRMSE* evaluation because there are fewer points to check in the search for nearest neighbors and there are fewer NLP runs, respectively. Depending on the length of the SSR set and calibration set, the prediction accuracy in terms of *NRMSE* is between 0.1938–0.2237 for lead time 5. For comparison, the prediction accuracy achieved at lead time 10 for the theoretical Mackey–Glass is at best between *NRMSE* = 0.084–0.195.

The maximum number of *NRMSE* evaluations required is only 688, which is less than 0.3% of the theoretical maximum in an exhaustive brute force search (240,000 combinations). Hence, mGA is indeed an extremely efficient search technique. Further significant cost savings are realized within mGA by the lookup table facility because the number of *NRMSE* evaluations could have been 1500 if all the chromosomes were unique.

REFERENCES

Babovic, V. & Keijzer, M. 1999 Forecasting of river discharges in the presence of chaos and noise. In *Proc. Coping with Floods: Lessons Learned from Recent Experiences. NATO ARW Series* (ed. J. Marsalek), pp. 405–419. Kluwer, Dordrecht.

Babovic, V., Keijzer, M. & Stefansson, M. 2000 Optimal embedding using evolutionary algorithms. In *Proc. 4th International Conference on Hydroinformatics, Iowa City, IA*. University of Iowa, IA, CD-ROM.

Carroll, D. L. 1996 Chemical laser modeling with genetic algorithms. *AIAA J.* **34** (2), 338–346.

Casdagli, M. 1989 Nonlinear prediction of chaotic time series. *Physica D* **35**, 335–356.

Deb, K. 2001 *Genetic Algorithms for Optimization*. *KanGAL Report Number 2001002*.

Goldberg, D. E. 1989 *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Holland, J. & Nafpliotis, N. 1975 *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Jayawardena, A. W., Fernando, D. A. K. & Dissanayake, P. B. G. 1997 Genetic algorithm approach of parameter optimization for the moisture and contaminant transport problem in partially saturated porous media. In *Proc. 27th Congress of the International Association for Hydraulics Research*, vol 1, pp. 761–766. ASCE, San Francisco.

Krishnakumar, K. 1989 Micro-genetic algorithms for stationary and non-stationary function optimization. *SPIE: Intelligent Control Adaptive Syst.* **1196**, 289–296.

Liong, S. Y., Khu, S. T. & Chan, W. T. 2001 Derivation of Pareto Front with genetic algorithm and neural network. *J. Hydrol. Engng., ASCE* **6** (1), 56–61.

Mackey, M. & Glass, L. 1977 Oscillation and chaos in physiological control systems. *Science* **197**, 287–289.

Phoon, K. K., Islam, N., Liaw, C. Y. & Liong, S. Y. 2002 A practical inverse approach for forecasting nonlinear hydrological time series. *J. Hydrol. Engng., ASCE* **7** (2), 116–128.

Schaffer, D., Caruana, R. A., Eshelman, L. J. & Das, R. 1989 A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms* (ed. J. D. Shaffer), pp. 51–61. Morgan Kaufman, Los Altos, CA.

Wang, M. & Zheng, C. 1998 Ground water management optimization using genetic algorithms and simulated annealing: Formulation and comparison. *J. Am. Wat. Res. Assoc.* **34** (3), 519–530.