Ant Colony Optimization for Determining the Optimal Dimension and Delays in Phase Space Reconstruction

Wei-neng Chen and Jun Zhang (Corresponding Author)

Department of Computer Science, Sun Yat-sen University

Key Laboratory of Digital Life, Ministry of Education

Key Laboratory of Software Technology, Education Dept. of Guangdong Province, P.R. China

junzhang@ieee.org

ABSTRACT

The selection of parameters in time-delay embedding for phase space reconstruction is crucial to chaotic time series analysis and forecasting. Although various methods have been developed for determining the parameters of embedding dimension and time delay for uniform embedding, the study of parameter selection for non-uniform embedding is progressed at a slow pace. In a non-uniform embedding which enables different dimensions in the phase space to have different time delays, the optimal selection of time delays presents a difficult optimization problem with combinatorial explosion. To solve this problem, this paper proposes an ant colony optimization (ACO) approach. The advantages of ACO for the embedding parameter selection problem are in two aspects. First, as ACO builds solution in an incremental way, it does not need to use a fixed embedding dimension as the encoding length of a solution. Instead, both the embedding dimension and the time delays can be optimized together. Second, ACO enables the use of problem-based heuristics. Therefore heuristics designed based on the original observed time series can be used to accelerate the search speed of ACO. Experimental results show that the proposed algorithm is promising.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control and Search – *Heuristic methods*; G.3 [**Probability and Statistics**]: Time Series Analysis;

General Terms: Algorithms

Keywords: Attractor embedding, time series forecasting, phase space reconstruction, non-uniform embedding, ant colony optimization (ACO)

1. INTRODUCTION

A time series is a chronological sequence of data points that are measured on a particular variable at successive times. Applications of time series can be found in many areas. The research into time series analysis and forecasting has attracted a considerable amount of research effort during the last decades [1]-[3]. To predict future values of time series variables based on previous observed data, one has to identify the nature of the phenomenon that the observed data represent. Usually, time series are driven by some underlying dynamical systems, in which the state changes with time as a function of its current state [4]. As the dynamical system evolves, the set of attracting states forms an attractor. If the dynamical system is linear, the corresponding attractor exhibits a regular appearance. Thus it is straightforward to extract meaningful statistics to characterize the time series. However, in many real-world observed time series, the intrinsic dynamics is nonlinear [5][6]. The states of a nonlinear dynamical system may evolve to a chaotic attractor, exhibiting highly irregular geometrical pattern and becoming sensitive to initial conditions [7]. As such, analysis of the chaotic time series derived from nonlinear dynamical systems has become a challenge in many fields of science and engineering [8].

One of the most important techniques for the investigation of chaotic time series is time delay embedding [1][9]. The technique is theoretically supported by the celebrated embedding theorem of Takens [10] and its extensions [11]. The theorems showed that for a time series of scalar observations of a dynamical system, it is possible to reconstruct its underlying dynamics in a phase space by a time delay embedding with sufficiently large dimension. To correctly reconstruct the attractor in phase space, a key issue is to find suitable embedding parameters, i.e., the embedding dimension m and the time delay τ. Various methods for optimal selection of embedding parameters have been proposed in the literature. Methods for selecting an embedding dimension include the false nearest neighbor (FNN) method [12], G-P method [13], and Cao's method [14]. Methods for selecting a time delay include the correlation method [15] and the averaged mutual information (AMI) method [16]. In addition, some methods that set the values of embedding dimension and time delay simultaneously have also been proposed [18]. However, even with these methods, it is found that no single method outperforms all others in all situations [1][5], and the research on designing better methods for phase space reconstruction continuous [17].

In the above mentioned methods for time delay embedding, there is an assumption that the time delay in each dimension grows uniformly by the same time delay value τ. Therefore, this traditional kind of time delay embedding is called uniform embedding. Although uniform embedding is simple and effective

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'11, July 12–16, 2011, Dublin, Ireland.

Copyright 2011 ACM 978-1-4503-0557-0/11/07...$10.00.

for classic chaotic dynamical systems such as the Lorenz and Rossler systems, Judd and Mees [19] pointed out that uniform embedding has difficulties in dealing with the time series with multiple strong periodicities with greatly differing timescales. To overcome the deficiency of uniform embedding, they proposed the non-uniform embedding technique [19]. Different from the traditional uniform embedding technique, non-uniform embedding allows different dimensions to use different time delays. As a result, it manages to deal with the time series with multiple timescales and provides more accurate predictions [19]-[21]. Non-uniform embedding has attracted increasing attention in recent years [20]-[24].

Although non-uniform embedding is more flexible and performs better on chaotic time series forecasting, it brings in a new problem. That is, not a single time delay value τ but a group of time delay values τi (i=2,3,…,m) has to be optimally selected. Suppose the maximum value of a time delay is τmax, the number of all possible combinations for (τ2, τ3, …, τm) is $(\tau_{\max})^{m-1}$. In other words, the selection of time delays in non-uniform embedding is a problem where the solution space grows exponentially with the increase of the embedding dimension m. In this case, the traditional embedding parameter selection methods for uniform embedding fail to work. By now, there is still lack of standard methods for selecting dimension and time delays for non-uniform embedding. In Small [22], a minimum description length (MDL) method was proposed to evaluate the performance of an embedding and a simple binary genetic algorithm (GA) was developed. Another GA approach was designed by Vitrano and Povinelli [23]. But the performance of these algorithms is still not satisfactory. For the problem of time series forecasting with fuzzy inference systems, some specialized deterministic or stochastic methods for selecting embedding dimension and time delays also exist [4][8][24].

To improve the performance of non-uniform embedding, this paper aims at proposing an ant colony optimization (ACO) algorithm for the optimal selection of embedding parameters. ACO is a swarm intelligence method proposed by Dorigo [25][26] in inspiration of the foraging behavior of ant colonies. Since its proposal in the early 1990s, ACO has evolved into an important optimization technique and has been successfully applied to a wide range of combinatorial optimization problems [27]-[29]. The potential advantages of ACO on the considered optimization problem are in the following aspects. First, different from GA which encodes the time delays as a fixed-length chromosome, ACO builds solutions based on a construction graph and the encoding length of a solution can adapt to the actual need of the optimization problem. Therefore, ACO provides a more flexible way for solving the considered embedding parameter selection problem as we do not need to first arbitrarily fix the embedding dimension as the length of the chromosome. Instead, the embedding dimension m can also be optimized along with the time delays in the algorithm. Second, ACO enables the use of problem-based heuristics for accelerating the search process [30]. Thereby, instead of selecting time delays in a completely random way, it is possible to extract useful statistical information from the original time series to provide guidance for the selection of suitable time delays.

The rest of the paper is organized as follows. In section 2, the background of time delay embedding is introduced. Section 3 proposes the ACO algorithm for optimal selection of embedding dimension and time delays. Experimental results are given in section 4. The conclusion finally comes in section 5.

2. BACKGROUND

2.1 Uniform Time Delay Embedding

Time delay embedding is important for chaotic time series analysis. Let $\{x(t)\}_{t=1}^{N} = \{x(1), x(2), \cdots, x(N)\}$ be a scalar time series of N observations. The uniform time delay embedding method is to find an embedding dimension m and a time delay τ to obtain a set of vectors

$$v_t = (x(t), x(t-\tau), x(t-2\tau), \cdots, x(t-(m-1)\tau)), t = N, N-1, ... \quad (1)$$

If $\{x(t)\}_{t=1}^{N}$ is a chaotic time series, the trajectory of v_t in the phase space usually behaves with quasi-periodicity. In addition, for a properly selected τ, if the number of observations N and the embedding dimension m are sufficiently large, Takens theorem [10] guarantees that the trajectory of v_t in the phase space is topologically equivalent to the original dynamical system of the scalar time series. In this sense, we can reconstruct the dynamics of the time series in the phase space based on the set of vectors v_t and use it to forecast the future variable values of the time series.

2.2 Non-Uniform Time Delay Embedding

The uniform time delay embedding technique has been found to be very suitable for the reconstruction of some classic chaotic systems such as the Lorenz and the Rossler system. However, for the time series with multiple periodicities, the performance of uniform embedding becomes poor [19]-[22]. This is because the uniform time delay τ usually captures only a single quasi-periodicity. If the underlying dynamical system of the time series contains multiple periodicities, uniform embedding may fail to capture all the periodicities and thus its performance becomes poor. Judd and Mees [19] suggested that non-uniform embedding is a better and more general approach for the reconstruction of dynamics of chaotic time series. Different from uniform embedding, the non-uniform embedding technique uses different time delays in different dimensions. Suppose the time delay for the j^{th} dimension is τ_j, the vector v_t reconstructed in the phase space becomes

$$v_t = (x(t), x(t-\tau_2), x(t-\tau_3), \cdots, x(t-\tau_m)) \quad (2)$$

In this way, non-uniform time delay embedding enables a more flexible way to reconstruct the dynamics and is able to overcome the deficiency of uniform embedding [20][21].

In non-uniform embedding, there is a combinatorial explosion of the possible settings for (τ_2, τ_3,…, τ_m) as the embedding dimension m increases. Therefore, the traditional methods for selecting m and τ in uniform embedding are not applicable in non-uniform embedding. The selection of embedding parameters in non-uniform embedding becomes an even more important and difficult task. Several works have been done in recent years to tackle this intractable parameter optimization problem. In [22], a GA approach was proposed by Small. The algorithm works by specifying a maximum embedding window $\tau_{\max}$ and encoding the solution as a binary string with the length of $\tau_{\max}$. For the j-th binary digit of the string, "1" means that j is selected as a time delay and "0" means the opposite. The simple binary GA is applied to optimize the parameters. The main deficiency of this GA approach is that its performance is seriously affected by the maximum embedding window τ_{max}. If τ_{max} is small, the encoding length of the GA is short. Thus the GA performs well. However, if the time series requires a large τ_{max}, the GA usually produces solutions with a large embedding dimension. As a result, the performance of the GA becomes poor and the solutions built by the algorithm are unusable in chaotic time series forecasting. Another GA approach for the embedding parameter selection problem was proposed by Vitrano and Povinelli [23]. The algorithm fixes the length of a chromosome as a predefined maximum embedding dimension and uses eight bits for the representation of time delays. As such, the definition domain of time delays is limited to [0, 127] and the presentation scheme is not flexible for different chaotic time series.

In order to further improve the performance of non-uniform time delay embedding, this paper aims to propose an ACO approach for the embedding parameter selection problem.

3. THE ACO APPROACH

3.1 Basic Idea of the Algorithm

The basic idea of ACO is to simulate the foraging behavior of ant colonies. When a group of ants set out from their nest to search for a path to food, they use a special kind of chemical which is called pheromone to communicate with each other. Once the ants discover a path to food, they deposit pheromones on the path to record their previous search experience. By sensing pheromones on the path, an ant can follow the trails of other ants to find the food source. As this process continues, the shortest path to the food source gradually accumulates more and more pheromones, attracting more and more ants to choose. In this way, though the capability of a single ant is limited, the whole ant colony is able to cooperate to find the best path to the food source. Inspired by such intelligent behavior of ants, Dorigo [25] proposed ACO in the early 1990s and ACO has now become an important swarm intelligence technique.

Different from other computational intelligence techniques, ACO has two main features. First, the artificial ants in ACO build solutions to the problem step by step incrementally. In this way, for the parameter selection problem in non-uniform embedding, we do not need to first fix the embedding dimension m as the encoding length. Instead, we can select time delays incrementally until the delays of all required dimensions have been chosen and the number of dimensions m can also be optimized along with the process of the algorithm. In other words, ACO provides a more flexible way for the encoding of the parameters m and τ_j (j=2,3,…,m). Second, ACO enables the use of some problem-based heuristics to guide the search direction of artificial ants. Therefore, some useful heuristic information from the original time series can be extracted to further accelerate the search speed of the algorithm.

Overall, the proposed ACO algorithm for the non-uniform embedding parameter selection problem involves the following steps:

Step 1) Initialization – The pheromones and heuristics of the algorithm are initialized.

Step 2) Solution Construction – During each generation, a group of ants set out to find suitable embedding parameters for the non-uniform time delay embedding problem.

Step 3) Evaluation – The solutions built by ants are evaluated based on a fitness function.

Step 4) Pheromone Management – The pheromone values are updated based on the local pheromone updating rule and the global pheromone updating rule.

Step 5) Terminal Test – If a predefined number of iterations have been processed, the algorithm ends. Otherwise, the algorithm continues to run *steps 2) -4)* iteratively.

3.2 Solution Construction

In each iteration of the proposed ACO algorithm, a group of M ants is dispatched to build M solutions to the problem. We denote the solution built by the i-th ant as

$$S(i) = (m(i), \tau_1(i), \tau_2(i), \tau_3(i), \cdots, \tau_{m(i)}(i)) \quad (3)$$

where $m(i)$ is the embedding dimension and $\tau_1(i), \tau_2(i), \tau_3(i), \cdots, \tau_{m(i)}(i)$ are the time delays. Note that the value of $\tau_1(i)$ is 0 as we usually use $x(t)$ as the first dimension of the reconstructed vector in the phase space. To build a solution, the i-th ant first selects the embedding dimension $m(i)$ based on some pheromones. Then the ant selects $m(i)$-1 time delays from the embedding window to compose the time delay vector $(\tau_2(i), \tau_3(i), \cdots, \tau_{m(i)}(i))$. In this way, both the embedding dimension $m(i)$ and the time delays $(\tau_2(i), \tau_3(i), \cdots, \tau_{m(i)}(i))$ can all be optimized along with the process of the algorithm. The procedure of solution construction is described in detail as follows.

3.2.1 Selection of Embedding Dimension

To construct an embedding in the phase space, the ant first needs to determine the embedding dimension $m(i)$. In the studies of time delay embedding [23], because a large embedding dimension usually causes much higher computational burden in time series analysis and forecasting, building an embedding with a very large dimension is meaningless. Therefore, it is reasonable to limit the embedding dimension to a maximum value $MaxD$. As such, the selection of embedding dimension is actually to assign $m(i)$ to a value from {2,3,…, $MaxD$}.

In the proposed ACO algorithm, the selection is based on pheromones. Pheromones represent the previous search experience of ants. We define the pheromone of setting the embedding dimension to k ($k \in \{2,3,...,MaxD\}$) as pd_k. A larger value for pd_k means that the previous search experience of ants regards $m(i)$=k as a good choice and vice versa. The rule for the selection of embedding dimension $m(i)$=k is then given by

$$P(m(i)=k) = \begin{cases} \dfrac{pd_k \cdot \dfrac{1}{\sqrt{k}}}{\sum_{l=2}^{MaxD} pd_l \dfrac{1}{\sqrt{l}}}, & \text{if } 2 \le k \le MaxD \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $P(m(i)$=$k)$ is the probability of setting $m(i)$=k. Equation (4) actually shows that the probability of setting $m(i)$ to k is in positive proportion to the value of $pd_l / \sqrt{l}$. Because an embedding with a smaller embedding dimension is more convenient for time series analysis, the reason of adding the term $1/\sqrt{l}$ is to favor smaller dimension numbers.

3.2.2 Selection of Time Delays

After determining the embedding dimension $m(i)$, the ant selects $m(i)$-1 time delays to complete the construction of an embedding. The time delays should satisfy

$$1 \le \tau_2(i) < \tau_3(i) < \cdots < \tau_{m(i)}(i) \le \tau_{\max} < N \quad (5)$$

where $\tau_{\max}$ is the upper bound of the embedding window and N is the size of the time series. Usually, $\tau_{\max}$ can be arbitrarily set to a sufficiently large number that satisfies $\tau_{\max} > m\tau$ [22], where m and τ are the embedding dimension and time delay in the uniform embedding technique obtained by traditional methods such as FNN [12] and AMI [16]. To select $m(i)$-1 time delays, the ant repeats the following steps for $m(i)$-1 times.

Step a) For k=1,2,…, $\tau_{\max}$, the value of $pt_k ht_k^\beta$ is evaluated. Here, pt_k is the pheromone of selecting k as a time delay, ht_k is the heuristic of selecting k as a time delay, and β is a parameter of the ACO algorithm to weigh the importance of heuristics. As has been mentioned before, in ACO, pheromones represent previous search experience and heuristics are some problem-based useful information.

Step b) The time delay is selected by

$$\text{time\_delay} = \begin{cases} \arg\max_{k=1,2,\ldots\tau_{\max}} pt_k ht_k^\beta \cdot \Omega(k), \\ \text{ if } ran < q_0 \\ \text{using the roulette wheel selection scheme,} \\ \text{otherwise} \end{cases} \quad (6)$$

where the function $\Omega()$ is defined as

$$\Omega(k) = \begin{cases} 0, & \text{if } k \text{ has been selected} \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

and $q_0 \in (0,1)$ is a parameter which means the probability of choosing the feasible delay with the largest value of $pt_k ht_k^\beta$ directly. To perform (6), a random number *ran* uniformly distributed in (0,1) is generated and is compared with the parameter q_0. If $ran<q_0$, the time delay that maximizes the value of $pt_k ht_k^\beta$ will be selected. Otherwise, the roulette wheel selection scheme (RWS) is applied. The RWS is to select a time delay based on the probabilities defined in (8). For k=1,2,…, $\tau_{\max}$, the probability of selecting k as a time delay is given by

$$P(\tau_j = k) = \frac{pt_k ht_k^\beta \cdot \Omega(k)}{\sum_{l=1}^{\tau_{\max}} pt_l ht_l^\beta \cdot \Omega(l)} \quad (8)$$

After running these steps for $m(i)$-1 times, $m(i)$-1 different time delays are finally selected. We reorder these selected time delays in ascending order and assign them to $\tau_2(i), \tau_3(i), \cdots, \tau_{m(i)}(i)$. In this way, all embedding parameters are set by the ant and the time delays are guaranteed to satisfy $1 \le \tau_2(i) < \tau_3(i) < \cdots < \tau_{m(i)}(i) \le \tau_{\max} < N$.

3.3 Definition of Heuristics

One unique characteristic of ACO is that it enables the use of heuristics to accelerate the search. Therefore, we can extract some useful information from the original time series to define heuristics and use them in equation (6) and (8) to direct the selection of time delays. In the proposed algorithm, we define two types of heuristics, i.e., the two-dimensional-based heuristic (2D heuristic) and the three-dimensional-based heuristic (3D heuristic). The underlying ideas of these heuristics are derived from the concept of nearest neighborhood in time delay embedding.

For k=1,2,…, $\tau_{\max}$, the 2D heuristic of selecting k as a time delay is defined as

$$ht_k = \frac{1}{\sqrt{k}} \cdot \frac{1}{\min_{l=1,2,\ldots,N-k-1}([x(N)-x(N-l)]^2 + [x(N-k)-x(N-l-k)]^2)} \quad (9)$$

The equation actually shows that the heuristic ht_k is in inverse proportion to the square of the distance between the vector $(x(N), x(N-k))$ and its nearest neighborhood vector $(x(N-l), x(N-l-k))$ in the two-dimensional phase space. In addition, to make the embedding more convenient, short time delays are more preferred [22]. Therefore the $1/\sqrt{k}$ item is added to the heuristic to favor short time delays. In fact, if $(x(N), x(N-k))$ has a very close neighborhood vector, it implies that k may possibly be a promising time delay that can lead to a good embedding in terms of geometry. Therefore, the definition of heuristics encourages artificial ants to select the time delays that can find closer nearest neighborhood points of $(x(N), x(N-k))$.

We also need to notice that the nearest neighborhood of $(x(N), x(N-k))$ in the two-dimensional phase space may be a false neighborhood point. In other words, $(x(N), x(N-k))$ and its nearest neighborhood point in the two-dimensional phase space may not be close to each other any more when the embedding is unfolded to the phase space with higher dimension. In this case, the 2D heuristic defined in (9) may induce a misleading effect. To reduce such effect, we also extend the 2D heuristic to a 3D version as follows

$$ht_k = \frac{1}{\sqrt{k}} \cdot \frac{1}{\min_{l=1,2,\ldots,N-2k-1} ([x(N)-x(N-l)]^2 + [x(N-k)-x(N-l-k)]^2 + [x(N-2k)-x(N-l-2k)]^2)} \quad (10)$$

In (10), the heuristic ht_k is in inverse proportion to the square of the distance between the vector ($x(N)$, $x(N\text{-}k)$, $x(N\text{-}2k)$) and its nearest neighborhood vector ($x(N\text{-l})$, $x(N\text{-l-}k)$, $x(N\text{-l-}2k)$) in the three-dimensional phase space. As we consider one more dimension, the chance of the emergence of false neighborhood points is reduced, and thus the 3D heuristic is more reliable. On the other hand, the computational cost of the 3D heuristic is also higher than the 2D heuristic. In general, we find that the 3D heuristic is already enough to facilitate the search of the algorithm without significantly reducing its search speed. Thus we do not need to define the heuristics in higher dimensions.

Finally, it is important to note that the heuristics defined in (9) and (10) are all static. In other words, once the values of these heuristics are initialized, the values remain unchanged during the whole process of the algorithm. Therefore, we only need to initialize the heuristics for all k (k=1,2,…, $\tau_{\max}$) once at the beginning of the algorithm. In this sense, the evaluation of heuristics is not time-consuming.

3.4 Evaluation

Evaluating the performance of the solutions built by the algorithm is an important step in swarm intelligence techniques. In this paper, we adopt the MDL method for fitness evaluation.

The MDL principle proposed by Rissanen [31] is an important concept in information theory. The idea behind the MDL principle is that we can use the regularity in a given set of data to construct a model and use the model to compress the data. In this sense, the best model is the one that describes the set of data in the most concise way. Judd and Mees [19] introduced the MDL principle to evaluate the performance of a model for a time series. In terms of this principle, if a model has the most concise description of a time series, (i.e., the data size required for the description of model parameters and prediction errors is the smallest,) the model is considered to be the best for the time series.

Based on this concept, Small and Tse [20][22] set up an approximation formula for the description length of an embedding and showed that the optimal embedding is the one that minimizes

$$\frac{d}{2}\ln\left[\frac{1}{d}\sum_{j=1}^{d}(x(j)-\bar{x})\right]+d+DL(d)+\frac{N-d}{2}\ln\left[\frac{1}{N-d}\sum_{j=d+1}^{N}e(j)^2\right]+DL(P) \tag{11}$$

where d is the length of the actual embedding window of the embedding, $\bar{x}$ is the mean of the data within the embedding window, $DL(d)=\lceil \log d\rceil+\lceil \log\lceil \log d\rceil\rceil+\cdots+0$ is the description length of an integer d, N is the number of data in the time series, $e(j)$ is the prediction error of $x(j)$, and $DL(P)$ is the description length of the parameters in the model. Given the embedding parameter $(\tau_2,\tau_3,\cdots,\tau_m)$, we can obtain $d=\tau_m$

$$\bar{x}=\frac{1}{\tau_m}\sum_{j=1}^{\tau_m}x(j) \tag{12}$$

In addition, according to [22], for the class of local constant models, we have $DL(P)=0$ and $e(j+1)=x(j+1)-x(l+1)$ where $l\in\{1,2,...,N\}\setminus\{j\}$ is the one that minimizes $|x(j)-x(l)|$.

3.5 Pheromone Management

During the process of ACO, pheromones are updated frequently to record the previous search experience of artificial ants. In the proposed algorithm, we follow the mechanism of the ant colony system (ACS) algorithm proposed by Dorigo [26] to design the pheromone management rules.

First, at the beginning of the algorithm, all pheromones (including the pheromones for embedding dimension selection and the pheromones for time delay selection) are initialized by

$$pd_k = p_{initial}, \qquad k=2,3,...,MaxD \tag{13}$$

$$pt_l = p_{initial}, \qquad l=1,2,...,\tau_{\max} \tag{14}$$

where $p_{initial}$ is the initial value of pheromones. In the proposed algorithm, we set $p_{initial}=1$.

Second, after the i-th artificial ant selects an embedding dimension $m(i)$ based on equation (4), the local pheromone updating rule for pd given in (15) is applied.

$$pd_{m(i)}=(1-\rho)\cdot pd_{m(i)}+\rho\cdot p_{initial} \tag{15}$$

Here, $\rho\in(0,1)$ is a parameter of the algorithm. Similarly, after the i-th artificial ant selects a time delay $\tau_l(i)$ based on equation (6), the local pheromone updating rule for pt given in (16) is applied.

$$pt_{\tau_l(i)}=(1-\rho)\cdot pt_{\tau_l(i)}+\rho\cdot p_{initial} \tag{16}$$

In fact, the function of the local pheromone updating rules (15) and (16) is to reduce the pheromones on the selected components so that the chances for the following ants to choose a different unexplored component are increased. In other words, the local pheromone updating rule is designed to increase the search diversity of the algorithm.

Finally, at the end of each iteration, after all ants in the colony have completed their solutions, additional pheromones are added to the components of the best-so-far solution. Note that the performance of a solution is evaluated by the MDL criterion defined in Section 3.4. Suppose $(m(i)\,|\,0,\tau_2(i),\tau_3(i),\cdots,\tau_{m(i)}(i))$ is the best-so-far solution, the pheromones on $m(i)$ and $\tau_l(i)$ (l=2,3,…,$m(i)$) are updated by the global pheromone updating rule as follows

$$pd_{m(i)}=(1-\rho)\cdot pd_{m(i)}+\rho\cdot MaxD \tag{17}$$

$$pt_{\tau_l(i)}=(1-\rho)\cdot pt_{\tau_l(i)}+\rho\cdot\tau_{\max} \tag{18}$$

The function of the global updating rule is to increase the pheromones on the components of the best-so-far solution to make them more attractive.

Based on the above-mentioned procedures, the overall flowchart of the proposed ACO algorithm is summarized in Fig. 1.

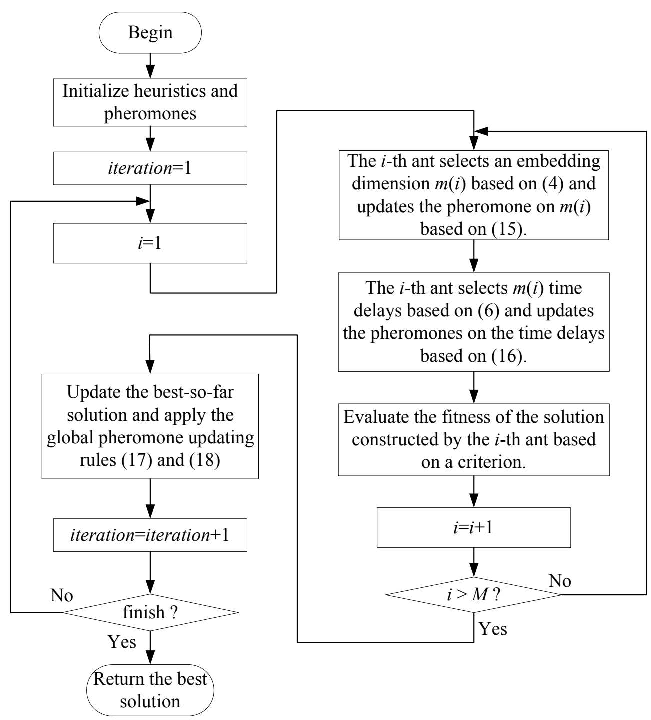

Fig. 1 Flowchart of the proposed ACO algorithm

4. EXPERIMENTAL RESULTS

4.1 Parameter and Heuristic Configurations

The proposed ACO algorithm has the following parameters: the number of artificial ants M, the parameter β in (6) and (8) to weigh the importance of heuristics, the parameter q_0 in (6), and the pheromone updating rate ρ in (15)-(18). In the experiment, these parameters are set as M=10, β=1, q_0=0.5 and ρ=0.1. Basically, the parameters are set according to the original ACO algorithm [26] and we slightly adjust the values of β and q_0 empirically. Overall, these parameter settings are found to contribute to the good performance of the proposed ACO algorithm.

One unique characteristic of ACO is that it enables the use of heuristics. Therefore, we define the 2D and 3D heuristics in (9) and (10) based on the geometry topology of the embedding to accelerate the search speed of ACO. To test if the heuristics can really improve performance, we test the algorithm on the Mackey-Glass time series. The Mackey-Glass time series is a well-known chaotic time series [32] and is expressed as

$$\frac{dx}{dt} = \frac{0.2x(t-\lambda)}{1+x^{10}(t-\lambda)} - 0.1x(t) \tag{19}$$

We adopts the Mackey-Glass time series with λ=17 in the experiment. The length of the time series is N=1000. The performance of the ACO versions with the 2D heuristic, the 3D heuristic, or without using heuristics is compared. In each single run of the algorithm, 100 iterations are executed. For each version, 20 independent trials are run based on the MDL criterion and the best result achieved by the algorithm in each run is recorded.

Table 1 compares the performance of different versions in terms of two-sample t-tests and the t values are tabulated. It can be seen that the difference between the performance of the algorithm with the 2D heuristic and the one without using heuristics is not significant. This is because in the two-dimensional phase space the proportion of FNNs is too large. This phenomenon induces a misleading effect and causes poor performance. On the other hand, when the dimension of the phase space is extended to three, the proportion of FNNs becomes much smaller, and thus the 3D heuristic is much more meaningful and manages to yield significantly better results. Overall, these results demonstrate that the proposed 3D heuristic is effective to improve the performance of the ACO algorithm for the embedding parameter selection problem.

Table 1. Comparison of Different Heuristics Based on Two Sample t-tests on the Mackey-Glass Time series

	Heuristic 2D	Heuristic 3D	No Heuristic
Heuristic 2D		**19.571#**	-1.371
Heuristic 3D	**-19.571***		**-16.301***
No Heuristic	-1.371	**16.301#**	

* The results obtained by the ACO version of the corresponding row are significantly smaller (better) than those obtained by the ACO version of the corresponding column at the 0.05 level.

\# The results obtained by the ACO version of the corresponding row are significantly larger (worse) than those obtained by the ACO version of the corresponding column at the 0.05 level.

Table 2. Comparison of the Optimization Performance of GA and ACO in Terms of Two-Sample t-tests

	M-G	sunspot
$\tau_{\max}$	30	70
ACO-GA	**-2.247#**	**-4.628#**
Dimension of ACO	4	4
Dimension of GA	5	30

4.2 Comparison with Other Approaches

4.2.1 Experimental Settings

To further demonstrate the performance of the proposed algorithm, we compare the proposed ACO with the GA proposed by Small [22] in terms of optimization performance based on the MDL criterion.

In the experiment, the parameters of GA are configured as follows: crossover rate px=0.7, mutation rate pm=0.1, and the population size is 20. These parameters are commonly used in the classic simple GA. The parameters of the proposed ACO are set based on the discussions provided in Section 4.1. For the traditional uniform embedding technique, the time delay τ is yielded by the AMI method [16] and the embedding dimension m is determined by the FNN method [12]. The experiments are run on the following time series:

- *The Mackey-Glass time series*

The Mackey-Glass time series has been defined in (19). In the experiment, the data $\{x(t)\}_{t=1}^{1000}$ will be used as the training set and the forecasting window is t=1001, 1002, …, 1200.

- *The sunspot time series*

The sunspot time series is a time series of the numbers of sunspots in May from 1700-2009. The time series exhibits quasi-periodicity and we usually call the periodicity as sunspot cycles. In the experiment, we use the sunspot data from 1700-1960 as the training set and the forecasting window is 1961-2009.

4.2.2 Comparison in Terms of Optimization Performance

Both algorithms are run on the training sets of the above two time series based on the MDL criterion and the best result obtained in each single run is recorded. In each run of these algorithms, 1000 solutions are generated. For each algorithm, 20 independent runs are executed. The maximum embedding window $\tau_{\max}$ used for each time series are defined in Table 2. Small [22] suggested that $\tau_{\max}$ should satisfy $\tau_{\max} > \tau m$ where τ and m are the time delay and the embedding dimension, respectively, for uniform embedding. The values of τ and m can be determined by the AMI and the FNN method. Therefore, the value of $\tau_{\max}$ in Table 2 is given by the smallest integer that is larger than $m\tau$ and is divisible by 10. For the proposed ACO, the maximum embedding dimension $MaxD$ is set equal to $\tau_{\max}$.

Table 2 compares the optimization performance of ACO and GA in terms of t-tests. According to the t-values, ACO achieves significantly better results than GA. The results produced by ACO generally have smaller objective function values in terms of the MDL criterion given in (11). In addition, the results produced by ACO have smaller embedding dimensions. Especially in the case of the sunspot time series, the embedding dimension of the best solution generated by ACO is only four, while the dimension of the best solution produced by GA is 30. In fact, an embedding with such a large dimension is very time-consuming for analyzing chaotic time series. Because we define heuristics to prefer smaller dimensions and smaller time delays in the proposed ACO, artificial ants tend to build solutions with relatively small embedding dimensions and short embedding windows. Therefore, according to Table 2, the results found by ACO are not only better-optimized, but also more practical for time series analysis.

5. CONCLUSION

An ACO algorithm has been proposed for the optimal selection of embedding parameters in non-uniform embedding for analyzing chaotic time series. The main advantages of the proposed algorithm is that it provides a flexible way for optimizing the embedding dimension and time delays together and enables the use of heuristics to accelerate the search speed. Experimental results show that the proposed ACO algorithm is promising.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (NSFC) No. 61070004, and by NSFC Joint Fund with Guangdong under Key Project U0835002, and by the Scholarship Award for Excellent Doctoral Student granted by Ministry of Education, China. The authors are with the Key Lab of Digital Life, MoE, China, and the Key Lab. of Softw. Tech, Edu. Depart of Guangdong Province. The corresponding author is Jun Zhang, email: junzhang@ieee.org

7. REFERENCES

[1] H. Abarbanel, *Analysis of Observed Chaotic Data*, Springer-Verlag, 1996.

[2] G.P. Zhang and D.M. Kline, "Quarterly time-series forecasting with neural networks," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1800-1814, 2007.

[3] J. Zhang, H. S.-H. Chung, and W-L. Lo, "Chaotic time series prediction using a neuro-fuzzy system with time-delay coordinates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 7, pp. 956-964, 2008.

[4] Y. Manabe and B. Chakraborty, "A novel approach for estimation of optimal embedding parameters of nonlinear time series by structural learning of neural network," *Neurocomputing*, vol. 70, pp. 1360-1371, 2007.

[5] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, Cambridge Univ. Press, 2004.

[6] C. Gao and J. Qian, "Evidence of Chaotic Behavior in Noise from Industrial Process," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2877-2884, 2007.

[7] L. Chen and G. Chen, "Fuzzy modeling, prediction, and control of uncertain chaotic systems based on time series," *IEEE Transactions on Circuits and Systems, Part I*, vol. 47, no. 10, pp. 1527-1531, 2000.

[8] M. Ragulskis and K. Lukoseviciute, "Non-uniform attractor embedding for time series forecasting by fuzzy inference systems," *Neurocomputing*, vol. 72, pp. 2618-2626, 2009.

[9] T. Matsumoto, et al., "Reconstructions and predictions of nonlinear dynamical systems: a hierarchical Bayesian approach," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 2138-2155, 2001.

[10] F. Takens, "Detecting strange attractors in turbulence," *Dynamical Systems and Turbulence, Lecture Notes in Mathematicians*, no. 898, Springer, 1981.

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

[12] M.B. Kennel, R. Brown and H.D.I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A*, vol. 45, no. 6, pp. 3403-3411, 1992.

[13] P. Grassberger, "An optimized box-assisted algorithm for fractal dimensions," *Physics Letters A*, vol. 148, pp. 63-68, 1990.

[14] L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Physica D: Nonlinear Phenomena*, pp. 43-50, 1997.

[15] T. Schreiber, "Interdisciplinary applications of nonlinear time series methods," *Physical Reports*, vol. 308, pp. 2-64, 1999.

[16] A.M. Fraser, H.L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A*, vol. 33, no.2, pp. 1134-1140, 1986.

[17] N. Xie and H. Leung, "Reconstruction of piecewise chaotic dynamic using a genetic algorithm multiple model approach," *IEEE Transactions on Circuits and Systems, Part I*, vol. 51, no. 6, pp. 1210-1222, 2004.

[18] H.S. Kim, R. Eykholt, J.D. Salas, "Nonlinear dynamics, delay times, and embedding windows," *Physical D*, vol. 12, pp. 48-60, 1999.

[19] K. Judd and A. Mees, "Embedding as a modeling problem," *Physica D*, vol. 120, pp. 273-286, 1998.

[20] M. Small and C.K. Tse, "Optimal selection of embedding parameters for time series modeling," *European Conference on Circuits Theory and Design*, European Circuit Society and the Institute of Electrical and Electronic Engineers, Krakow, Poland, 2003.

[21] M. Small and C.K. Tse, "Optimal embedding parameters: a modeling paradigm," *Physica D*, vol. 194, pp. 283-296, 2004.

[22] M. Small, "Optimal time delay embedding for nonlinear time series modeling," arXiv:nlin/0308114v1 [physics.data-an], 2003.

[23] J.B. Vitrano and R.J. Povinelli, "Selecting dimensions and delay values for a time-delay embedding using a genetic algorithm", *in Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1423-1430, 2001.

[24] K. Lukoseviciute and M. Ragulskis, "Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems," *Neurocomputing*, vol. 73, pp. 2077-2088, 2010.

[25] M. Dorigo, V. Maniezzo, A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems Man, and Cybernetics - part B,* vol. 26, pp. 29-41, 1996.

[26] M. Dorigo, L. M. Gambardella, "Ant colony system: A cooperative learning approach to TSP", *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 53–66, 1997.

[27] W.-N. Chen and J. Zhang, "An ant colony optimization approach to a Grid workflow scheduling problem with various QoS requirements", *IEEE Transactions on System, Man, and Cybernetics, Part C*, vol. 39, no. 1, pp. 29-43, 2009.

[28] W.-N. Chen, J. Zhang, H. S.-H. Chung, R.-Z. Huang and O. Liu. "Optimizing discounted cash flows in project scheduling – an ant colony optimization approach," *IEEE Transactions on Systems, Man, and Cybernetics – Part C*, vol. 40, no. 1, pp. 64-77, 2010.

[29] X.-M. Hu, J. Zhang, H.S.-H. Chung, O. Liu and J. Xiao, "An intelligent testing system embedded with an ant colony optimization based test composition method," *IEEE Transactions on Systems, Man and Cybernetics, Part C,* vol. 39. no. 6, pp. 659-669, 2009.

[30] M. Dorigo, T. Stützle, *Ant colony optimization*, MIT Press, 2004.

[31] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, vol. 11, no. 2, pp. 416-431, 1983.

[32] E. Ott, T. Sauer, and J.A. York, *Coping with Chaos: Analysis of Chaotic Data and the Exploitation of Chaotic Systems*, pp. 1-13, John Wiley & Sons, 1994.

[33] J.D Framer, J.J. Sidorowich, "Predicting chaotic time series," *Physical Review Letters*, vol. 59, no. 8, pp. 845-848, 1987.