Expert Systems with Applications 37 (2010) 4198–4203

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization

Leandro dos Santos Coelho *, Diego Luis de Andrade Bernert

Industrial and Systems Engineering Graduate Program, LAS/PPGEPS, Pontifical Catholic University of Paraná, PUCPR, Imaculada Conceição, 1155, 80215-901 Curitiba, Paraná, Brazil

ARTICLE INFO

Keywords:
Chaotic synchronization
Optimization
Ant colony optimization

ABSTRACT

Optimization algorithms inspired by the ants' foraging behavior have been initially used for solving combinatorial optimization problems. Since the emergence of ant algorithms as an optimization tool, some attempts were also made to use them for tackling continuous optimization problems. In recent years, the investigation of synchronization and control problem for discrete chaotic systems has attracted much attention, and many possible applications. The optimization of a proportional–integral–derivative (PID) controller based on a modified continuous approach of ant colony optimization combined with a differential evolution method (MACO) for synchronization of two identical discrete chaotic systems subject the different initial conditions is presented in this paper. Numerical simulations based on optimized PID control of a nonlinear chaotic model demonstrate the effectiveness and efficiency of MACO approach. Simulation results of the MACO to determine the PID parameters are compared with other metaheuristics including a classical ant colony optimization approach, genetic algorithm and evolution strategy. The proposed approach of PID tuning based on MACO can be generalized to other chaotic systems.

1. Introduction

Chaos, an interesting phenomenon in nonlinear dynamical systems, has been developed and thoroughly studied over the past two decades (Bagheri & Moghaddam, 2009; Chang, Yang, Liao, & Yan, 2008; Cheng, Huang, Cheng, & Yan, 2009; Hanbay, Turkoglu, & Demir, 2008; Mullin, 1993; Parker & Chua, 1989; Peitgen, Jürgens, & Saupe, 2004; Strogatz, 2000).

A chaotic system is a nonlinear deterministic system that displays complex, noisy-like and unpredictable behavior. Many fundamental features can be found in a chaotic system, such as excessive sensitivity to initial conditions, broad spectra of Fourier transforms, and fractal properties of the motion in phase space.

Due to its powerful applications in engineering systems, both control and synchronization problems of chaotic dynamical systems have recently attracted a significant attention within the engineering community, since Pecora and Carroll (1990) presented the chaos synchronization method to synchronize two identical chaotic systems with different initial values. Following this work, a considerable interest in the notion of synchronization of complex or chaotic systems has arisen in the last years (Guan, Hill, & Yao, 2006; Li & Liao, 2006; Park, 2006; Zhou, Chen, & Xiang, 2006).

Synchronization means agreement or correlation of different processes in time. Generally speaking, the synchronization phenomenon has the following feature: the output of the drive system is used to control the response system so that the output of the response system follows the output of the drive system asymptotically. Among others, potential applications of chaos synchronization are secure or private communications, biological applications (heart beat, brain activity, neural activity), and engineering (coordinated robot motion, time-series analysis, etc.).

In other hand, recently, many complicated nonlinear controllers have been also proposed in the literature for synchronization of chaotic systems, such as adaptive control (Li & Liao, 2006; Wang & Wang, 2007), robust control (Chen & Zhang, 2007; Huang, Li, & Zhong, 2006; Wu & Wang, 2006), variable structure control (Feki, 2009; Yan, Liao, Lin, & Cheng, 2006), neural control (Lam & Seneviratne, 2007), fuzzy control (Yau & Shieh, 2008) among many others (Chen & Dong, 1998).

Nevertheless, it is obvious that the use of PID (proportional–integral–derivative) control has a long history in control engineering and is acceptable for most of real applications because of its simplicity in architecture. The key of designing a PID controller is to determine three PID control gains, i.e., proportional gain K_p, integral gain K_i, and derivative gain K_d (Chang, 2007).

Overschee and De Moor (2000) report that 80% of PID type controllers in the industry are poorly/less optimally tuned. Over the years, many techniques have been suggested for the tuning of the PID parameters (Åström & Hägglund, 1995, 2001; Cohen &

* Corresponding author. Tel.: +55 41 3271 13 33; fax: +55 41 3271 13 45.
E-mail addresses: leandro.coelho@pucpr.br (L.S. Coelho), dbernert@gmail.com (D.L.A. Bernert).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.11.002

Coon, 1953; Cominos & Munro, 2002; Yamamoto, Indue, & Shah, 1999; Ziegler & Nichols, 1942; Wojsznis & Blevins, 2002). Moreover, many design and optimization methods have been investigated for the synchronization of chaotic systems via PID control such as linear matrix inequality (Wen, Wang, Lin, Li, & Han, 2007) and observers (Aguilar-López & Martinez-Guerra, 2007).

Recently, as an alternative to the classical mathematical approaches, modern heuristic optimization techniques such as evolutionary programming (Hung, Lin, Yan, & Liao, 2008), particle swarm optimization (Chang, 2009), genetic algorithms (Alfaro-Cid, McGookin, & Murray-Smith, 2006; Chang, 2007; Vlachos, Williams, & Gomm, 2002), artificial neural network combined with genetic algorithms (Sakaguchi & Yamamoto, 2002), and fuzzy systems (Hu, Mann, & Gosine, 1999) have been given much attention by many researchers due to their ability to find an almost global optimal solution in PID design and optimization.

One of these modern heuristic optimization paradigms is the ant colony inspired algorithm. The origins of ant colony inspired algorithm are in a field called swarm intelligence, which is assigned to study the behavior of some species and use certain properties of them to realize some tasks like optimization. Ant colony inspired algorithms draw an analogy between the optimization process and the foraging behavior of real ants. Based upon observations of real ant colonies, it was found that ants have the ability to find the shortest path between their nest and a food source. The main idea of ant colony inspired algorithm is that a population of artificial ants repeatedly builds and improves solutions to a given instance of a combinatorial optimization problem. From one generation to the next a global memory is updated that guides the construction of solutions in the successive population. The best solutions found so far by the ants are used for the memory update. After the construction phase of the algorithm usually a local search is applied to improve the solutions of the ants.

To date, most of these algorithms are confined to applications on discrete optimization (i.e., combinatorial) problems (Dorigo & Stützle, 2004). Bilchev and Parmee (1995) were the first to extend ant colony optimization to continuous space. Another notable works to problems with continuous variables were presented by various researchers (Blum and Socha, 2005; Coelho & Mariani, 2008; Dreo & Siarry, 2002; Jayaraman, Kulkarni, Gupta, Rajesh, & Kusumaker, 2001; Mathur, Karale, Priyee, Jayaraman, & Kulkarni, 2000; Monmarche, Venturini, & Slimane, 2000; Socha, 2004; Socha & Dorigo, 2008).

Inspired by the ant colony inspired algorithm proposed by Socha and Dorigo (2008), this work presents a modified ant colony optimization combined with a differential evolution method (MACO) in PID controller optimization to achieve the synchronization of discrete chaotic systems. Differential evolution (DE) algorithm is a population-based evolutionary method for global optimization. Storn and Price (1995) first introduced the DE algorithm a few years ago. DE uses a rather greedy and less stochastic approach to problem solving compared to other evolutionary algorithms, such as classical ant colony optimization, genetic algorithms (Goldberg, 1989), evolutionary programming (Fogel, 1995), and evolution strategies (Bäck & Schwefel, 1993).

Numerical simulations based on optimized PID control of a nonlinear chaotic model demonstrate the effectiveness and efficiency of MACO approach. Simulation results of MACO to determine the PID parameters are compared with other metaheuristics including genetic algorithm and evolution strategy.

The remainder of this paper is organized as follows: Section 2 describes the problem, while Section 3 explains the concepts of MACO approach. Section 4 presents the simulation results of PID optimization and chaotic synchronization. Finally, Section 5 outlines a brief conclusion about this study.

2. Problem description

2.1. PID controller

The PID controller is simple and easy to implement. It is widely applied in industry to solve various control problems. PID controllers have been used for decades. During this time, many modifications have been presented in the literature (Åström & Hägglund, 1995; Cominos & Munro, 2002). As modelled in this paper, the transfer function of PID controller is described by the following equation in the continuous s-domain (Laplace operator)

$$G_{\mathrm{PID}}(s) = P + I + D = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d \cdot s, \tag{1}$$

$$\text{or } G_{\mathrm{PID}}(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right), \tag{2}$$

where $U(s)$ and $E(s)$ are the control (controller output) and tracking error signals in s-domain, respectively; K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain. T_i is the integral action time or reset time and T_d is referred to as the derivative action time or rate time.

In this context, output of the PID controller in time domain is given by

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau)d\tau + K_d \cdot \frac{de(t)}{dt}, \tag{3}$$

where $u(t)$ and $e(t)$ are the control and tracking error signals in time domain, respectively. Using trapezoidal approximations for Eq. (3) to obtain the discrete control law, we have

$$\begin{aligned} u(k) = {} & u(k-1) + K_p \cdot [e(k) - e(k-1)] + K_i \cdot \frac{T_s}{2} \cdot [e(k) - e(k-1)] \\ & + K_d \cdot \frac{T_s}{2} \cdot [e(k) - 2e(k-1) + e(k-2)], \end{aligned} \tag{4}$$

where T_s is the sampling period. The proportional part of the PID controller reduces error responses to disturbances. The integral term of the error eliminates steady state error and the derivative term of error dampens the dynamic response and thereby improves stability of the system. How to solve these three gains to meet the required performance is the most key in the PID control system. However, it is difficult to find the optimal set of PID gains for nonlinear dynamical systems.

2.2. Nonlinear discrete chaotic system

In this study, two identical delayed discrete chaotic systems are considered to be synchronized using the proposed PID control. The master system is given by the following difference equation (Peng, 2004; Peng, Bai, & Lonngren, 2004):

$$x(k+1) = x(k) - \frac{\delta}{m}x(k-m) + \frac{\varepsilon}{m}x^3(k-m), \tag{5}$$

where δ and ε are positive constants, m is the delay term, and x is the master state. The delayed discrete system admits decaying, oscillatory, and chaotic behavior relying on settings of system parameters. On the other hand, the corresponding slave system is described by

$$y(k+1) = y(k) - \frac{\delta}{m}y(k-m) + \frac{\varepsilon}{m}y^3(k-m) + u(k), \tag{6}$$

where y is the slave state and u is the external control force that adopts the PID control of Eq. (4). For two identical delayed discrete chaotic systems (5) and (6) without control u, the state trajectories of these two chaotic systems will quickly separate each other if their initial conditions are not the same. However, the state trajectories can approach synchronization for any initial conditions if an appropriate controller is utilized (Chang, 2009). Hence the purpose of this paper is to apply the MACO algorithm to find out the PID control gains such that chaos synchronization for two identical delayed discrete chaotic systems is achieved.

For simplicity, the objective function F used in the study is defined as

$$F = \sum_{k=1}^{N} |x(k) - y(k)| = \sum_{k=1}^{N} |e(k)|, \tag{7}$$

where $e(k)$ is the error signal between the master and slave states and N is the total number of sampling. The optimization problem involves finding $g^* = [K_p^*, K_i^*, K_d^*]$ such that the F objective function of the system is minimized.

3. Fundamentals of ant colony optimization

Ant colony optimization system is a competing meta-heuristic for large-scale and difficult combinatorial optimization problems. It is based on the ant system – first defined by Colorni and Maniezzo (1991), Dorigo (1992) and Dorigo, Maniezzo, and Colorni (1996) in early 1990s – that imitates the foraging behavior of the real ants. The main idea of ant colony optimization is that a population of artificial ants repeatedly builds and improves solutions to a given instance of a combinatorial optimization problem.

The main idea behind ant colony optimization is that when the ants search for food, they initially explore the area surrounding their nest randomly. When one finds a food source, it evaluates it, take some food and goes back to the nest. As they move back, they deposit on the ground a chemical substance called pheromone, which is detectable by other ants. The amount of pheromone that is deposited varies depending on the quantity and quality of the food, and leads other ants to that food source. By the use of this property, the ants can find the shortest path between their nest and the source (Socha & Dorigo, 2009).

Algorithms based on ant colony optimization have been successfully applied to a variety of combinatorial optimization problems such as the traveling salesperson problem, the quadratic assignment problem, different variants of the vehicle routing problem, the graph colouring problem and different variants of machine scheduling problems (Dorigo & Stützle, 2004).

Since the first utilization of ant colony optimization for discrete problems, many researchers have tried to extend it to the continuous domains, developing some algorithms extended to continuous domains, according to Blum and Socha (2005).

3.1. Ant colony optimization for continuous domains

Socha (2004) observed that, since, in case of continuous optimization problems, the domain changes from discrete to continuous; the logical adaptation would be to move from a discrete probability distribution of pheromones to a continuous one – the Probability Density Function (PDF). One of the most popular functions that is used as a PDF is the normal (or Gaussian) function. In Socha (2004), this density function is – for each solution construction – produced from a population of solutions (ants) that the algorithm keeps at all times.

The normal PDF has some clear advantages, such as a fairly easy way of generating random numbers from it, but it also has some disadvantages. A single normal PDF is not able to describe a situation where two disjoint areas of the search space are promising, as it only has one maximum (Swaminathan, 2006). Due to this fact, Socha (2004) proposed the use of a mixture of normal PDF's and called it "mixture of normal kernels".

The ant colony optimization algorithm proposed in Socha (2004) and evaluated by Blum and Socha (2005) and Dorigo and Socha (2008) consists on three main steps, the ant based solution construction, the pheromone update and the daemon actions. These steps are described as follows.

3.1.1. Ant based solution construction

On this first step, the main goal is creating incrementally the construction of solutions based on a probabilistic choice of the solution components. In the case of the ACO for continuous domains, this is accomplished by the use of a PDF. A PDF can be represented as any function $P(x) > 0$, $\forall x$ that meets the requirement, $\int_{-\infty}^{\infty} P(x)dx = 1$.

One Gaussian function $g_l^i(x)$ is summed along with several others, to deal with the situation when the problem has two or more promising areas inside its search space, creating a Gaussian kernel$G^i(x)$, denoted as:

$$G^i(x) = \sum_{l=1}^{k} \omega_l g_l^i(x) = \sum_{l=1}^{k} \frac{1}{\sigma_l^i \sqrt{2\Pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^{i2}}}, \quad i = 1, \ldots, n, \tag{8}$$

where μ is the mean, ω is the weight, and σ is the standard deviation.

Based on the decision variables X_i, $i = 1, 2, \ldots, n$, each ant constructs a solution performing n steps. At an iteration i, an ant chooses a value for the variable X_i. After this, a Gaussian kernel must be created for this iteration. The number of functions used for its creation is equal to the size k of the solution archive T. At construction step i, the only information available for use is about the ith dimension. In this way, at each step i the resulting Gaussian kernel PDF $G^i(x)$ is a different one.

As it can be observed on Eq. (8), three parameters are needed to create a Gaussian kernel $G^i(x)$, the mean μ, the weight ω^i and the standard deviation σ^i, all with the cardinality k. In this context, the first two can be found directly by the use of the solution archive, where

$$\omega_l = \frac{1}{qk\sqrt{2\Pi}} e^{-\frac{(l-1)^2}{2q^2k^2}}. \tag{9}$$

By Eq. (9) one can presume that the weight is a value of the Gaussian function with parameter l, mean 1 and standard deviation qk, with q as an algorithm parameter. The means can be found just with the solutions s in the archive, where

$$\mu^i = \{\mu_1^i, \ldots, \mu_k^i\} = \{s_1^i, \ldots, s_l^i\}. \tag{10}$$

After that, the standard deviation can be found by the sampling of one of the Gaussians summed to create the Kernel. The choice for the one used is done probabilistically, by the use of the weights ω of each of the individuals dimension Gaussians as follows:

$$p_l = \frac{\omega_l}{\sum_{r=1}^{k} \omega_r}. \tag{11}$$

After this selection, the standard deviation σ_l^i, in the step i, can be calculated by the average distance from the chosen solution s_e^i to other solutions in the archive, resulting in:

$$\sigma_l^i = \varepsilon \sum_{e=1}^{k} \frac{|s_e^i - s_l^i|}{k-1}. \tag{12}$$

The parameter ε has the effect to raise or decrease the speed convergence of the algorithm. The greater the ε, lower is the convergence speed.

The whole process described in this section is repeated for each dimension $i = 1, 2, \ldots, n$, and for each the standard deviation is calculated with just one of the summed Gaussians. This fact can assure that the algorithm is able to adapt to several linear transformations of the problem.

3.1.2. Pheromone update

The second step of the algorithm is the pheromone update. All the pheromone information is stored in a solution archive T. For each solution to a problem of n dimensions, the algorithm stores in T the values of its n variables, besides the value of the objective function $f(s)$. All the solutions on this archive is evaluated and then ranked. By this rank they can be sorted.

The solution archive T has a size k, always equal or greater than the number of dimensions of the problem. Initially, the archive is created generating k solutions by uniform random sampling. To accomplish the pheromone update, the new generated solutions are added in the archive in the place of the worst solutions, assuring that the size of it doesn't change and all the best ones are kept, guiding the ants to a better solution at each iteration.

3.1.3. Daemon actions

The third and last step is just to update the best solution found, in order that it can be shown when the stop conditions met.

3.2. Continuous ant colony optimization combined with a differential evolution method

Inspired by the ant colony inspired algorithm proposed by Socha and Dorigo (2008), this work presents a modified ant colony optimization based on a DE method.

DE is a population-based stochastic function minimizer (or maximizer) relating to evolutionary algorithms, whose simple yet powerful and straightforward features make it very attractive for numerical optimization. Using a few control parameters, DE exhibits a fast convergence for a wide range of benchmark functions. A number of recent studies comparing DE with other meta-heuristics, such as genetic algorithms and particle swarm optimization, regarding real-world and artificial problems indicate superiority of DE in single-objective, noise free, numerical optimization (Krink, Filipic, Fogel, & Thomsen, 2004; Ursem & Vadstrup, 2003; Vesterström & Thomsen, 2004). In short, DE is now generally considered as a reliable, accurate, robust and fast optimization technique (Noman & Iba, 2008).

Various schemes are typically used for DE when creating the trial vectors (Price, Storn, & Lampinen, 2005; Storn & Price, 1995). Each scheme generates trial vectors by adding the weighted difference between other randomly selected members of the population. Meanwhile, each strategy is dependent on three factors; the solution to be perturbed, number of different solutions considered for perturbation and the type of recombination used. The different schemes of DE are classified using the following notation: DE/α/β/δ, where α indicates the method for selecting the parent chromosome that will form the base of the mutated vector, β indicates the number of difference vectors used to perturb the base chromosome, and δ indicates the recombination mechanism used to create the offspring population. The *bin* acronym indicates that the recombination is controlled by a series of independent binomial experiments.

The particular version subject to our investigation in modified ant colony optimization optimization design called MACO is the *DE/best/1/bin*-version.

In the MACO design, the generation of solutions uses Eq. (8) or Eq. (13) of mutation operation (differential operation) of *DE/best/1/bin*-version given by

$$X_i(t+1) = X_{best}(t) + MF \cdot [X_{i_2}(t) - X_{i_3}(t)]. \tag{13}$$

In the above equations, t is the time (generation), $MF > 0$ is a real parameter, called *mutation factor*, which controls the amplification of the difference between two individuals so as to avoid search stagnation. The mutation operation selects the best vector $X_{best}(t)$. Then, two individuals $X_{i_2}(t)$ and $X_{i_3}(t)$ are randomly selected with $best \neq i_2 \neq i_3$, and the difference vector $X_{i_2}(t) - X_{i_3}(t)$ is calculated. The utilization of Eq. (8) or Eq. (13) is given by a mutation probability, mp. The adopted value for mp is 0.30 in this work, i.e., 30% for use Eq. (13) and 70% for Eq. (8).

4. Results of numerical simulation

In this section, we illustrative the synchronization PID controller design for the above two systems given by Eqs. (5) and (6) with different initial value conditions, $x_i = 0.3$ and $y_i = -0.2$ ($i = -m,\_m+1,\ldots,0$). The parameters $\delta = 3.6$, $\varepsilon = 1$ and $m = 10$ are used in this example. We solve the optimization problem with $N = 70$, $k_f = 80$, $T_s = 1$, and $b = 0.05$.

In this section, to verify and demonstrate the effectiveness of the proposed method, we discuss the simulation results of synchronization between the master and slave states for different optimization approaches including the classical ant colony optimization (ACO), MACO, genetic algorithm (GA), and evolution strategy (ES).

Each optimization method was implemented in Matlab (MathWorks). All the programs were run on a 3.2 GHz Pentium IV processor with 2 GB of random access memory. In each case study, 30 independent runs were made for each of the optimization methods involving 30 different initial trial solutions for each optimization method. In this paper, the optimization approaches are adopted using 25,000 objective function evaluations in each run. The lower and upper bounds of the search space used in optimization methods were $(K_p, K_i, K_d) \in [0, 2]$.

In this case study, the population size was 25 ants, the stopping criterion (maximum number of generations evolved) was 1000 generations for ACO and MACO approaches. However, MACO uses the mutation factor $MF = 0.5$ on the differential operation setup.

Other particular parameters and procedures used in these optimization methods were:

- GA: binary representation with an individual length of 16 bits for each design variable, population size of 25 chromosomes, mutation probability of 0.15, crossover probability of 0.80, and roulette wheel selection with elitism.

Table 2
Best results of PID controller gains in 30 runs.

Parameter	GA	ES	ACO	MACO
K_p	0.5928	0.4126	0.5925	0.5925
K_i	0.1112	0.0001	0.1099	0.1098
K_d	0.2256	0.9999	0.2263	0.2267
F	4.1078	8.2212	4.0950	4.0946

Table 1
Convergence results for synchronization of discrete chaotic systems in 30 runs.

Optimization method	Minimum F	Mean F	Maximum F	Standard deviation of F
GA	4.1078	4.1552	4.2170	0.0032
ES	8.2212	12.6284	18.1754	2.7766
ACO	4.0950	4.1304	4.1655	0.0026
MACO	4.0946	4.0949	4.0953	0.0001

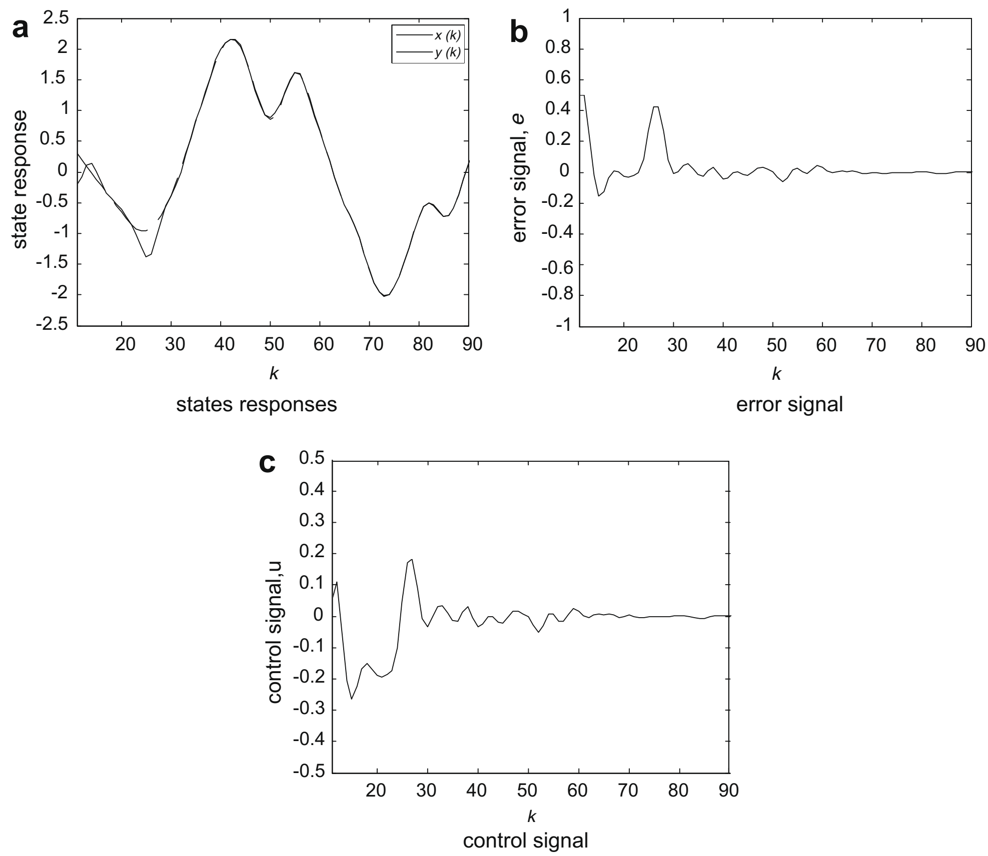

Fig. 1. Best result using MACO.

- ES: uses the sum strategy ES($\mu + \lambda$), where the number of parents and offspring are set to $\mu = 5$ and $\lambda = 25$, respectively.

Convergence results obtained by applying the different optimization approaches for synchronization of identical delayed discrete chaotic systems are summarized in Table 1. It can be seen from Table 1 that with the same preset maximum number of generations, MACO obtained better mean *F* and minimum *F* than GA, ES, and ACO methods.

Table 2 presents the best results of PID controller gains and performance data obtained using optimization approaches. As shown in Table 2, results confirm that ACO and MACO were superior in terms of efficiency than the GA and ES methods for PID's controller tuning.

Fig. 1 shows the state responses of the master and slave systems using the resulting PID controller gains obtained by best result of MACO approach. Fig. 1 includes states, error, and control signals. It can be observed in Fig. 1 that the slave signal tracked the master signal as expected without any problem. In terms of control input effort, note that an acceptable effort is required for the synchronization task.

5. Conclusion

Synchronization in chaotic dynamic systems has been an active research topic over the past decades and has received a great deal of interest among scientists from various research fields. In this paper, an optimization approach to synchronize two identical delayed discrete chaotic systems using PID controller is proposed. The proposed method based on PID control gains tuning using an optimization method synchronizes one slave signal to one master signal.

From the viewpoint of optimization, PID's gains tuning for the discrete chaotic system was formulated as a multidimensional optimization problem in this paper. A novel optimization approach, MACO, was applied to solve a chaotic synchronization problem. Numerical simulation and comparisons based on two discrete chaotic systems demonstrated the effectiveness, efficiency and robustness of MACO.

Finally, we would like to emphasize that the MACO presented here have potential applications to parameter identification and advanced controllers tuning. Future work includes applying an adaptive PID scheme based on MACO to the synchronization of different chaotic systems.

Acknowledgments

This work was supported by the National Council of Scientific and Technologic Development of Brazil – CNPq – under Grant 309646/2006-5/PQ.

References

Aguilar-López, R., & Martinez-Guerra, R. (2007). Partial synchronization of different chaotic oscillators using robust PID feedback. *Chaos, Solitons and Fractals, 33*(2), 572–581.

Alfaro-Cid, E., McGookin, E. W., & Murray-Smith, D. J. (2006). GA-optimised PID and pole placement real and simulated performance when controlling the dynamics of a supply ship. *IEE Proceedings-Control Theory and Applications, 153*(2), 228–236.

Åström, K. J., & Hägglund, T. (1995). *PID controllers: Theory, design and tuning* (2nd ed.). Research Triangle Park, NC, USA: Instrument Society of America.

Åström, K. J., & Hägglund, T. (2001). The future of PID control. *Control Engineering Practice, 9*(11), 1163–1175.

Bäck, T., & Schwefel, H. P. (1993). An overview of evolutionary algorithm for parameter optimization. *Evolutionary Computation, 1*(1), 1–23.

Bagheri, A., & Moghaddam, J. J. (2009). Decoupled adaptive neuro-fuzzy (DANF) sliding mode control system for a Lorenz chaotic problem. *Expert Systems with Applications, 36*(3), 6062–6068.

Bilchev, G., & Parmee, I. C. (1995). *The ant colony metaphor for searching continuous design spaces*. *Lecture Notes on Computer Science*. Germany: Springer (Vol. 993, pp. 25–39).

Blum, C., & Socha, K. (2005). Training feed-forward neural networks with ant colony optimization: An application to pattern classification. In *Proceedings of fifth international conference on Hybrid Intelligent Systems (HIS 2005)* (pp. 233–238). Rio de Janeiro, Brazil.

Chang, J.-F., Yang, Y-. S., Liao, T.-L., & Yan, J.-J. (2008). Parameter identification of chaotic systems using evolutionary programming approach. *Expert Systems with Applications, 35*(4), 2074–2079.

Chang, W. D. (2007). A multi-crossover genetic approach to multivariable PID controllers tuning. *Expert Systems with Applications, 33*(3), 620–626.

Chang, W. D. (2009). PID control for chaotic synchronization using particle swarm optimization. *Chaos, Solitons and Fractals, 39*(2), 910–917.

Chen, F., & Zhang, W. (2007). LMI criteria for robust chaos synchronization of a class of chaotic systems. *Nonlinear Analysis, 67*(12), 3384–3393.

Chen, G., & Dong, X. (1998). *From chaos to order: Methodologies, perspectives and applications*. Singapore: World Scientific.

Cheng, D.-L., Huang, C.-F., Cheng, S.-Y., & Yan, J.-J. (2009). Synchronization of optical chaos in vertical-cavity surface-emitting lasers via optimal PI controller. *Expert Systems with Applications, 36*(3), 6854–6858.

Coelho, L. S., & Mariani, V. C. (2008). Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Systems with Applications, 34*(3), 1905–1913.

Cohen, G. H., & Coon, G. A. (1953). Theoretical consideration of retarded control. *Transactions of ASME, 75*, 827–834.

Colorni A., Dorigo M., & Maniezzo V. (1991) Distributed optimization by ant-colonies. In Varela, F., & Bourgine, P. (Eds.), Proceedings of the European conference on artificial life (ECAL'91) (pp. 134–142). Cambridge, MA, USA: MIT Press.

Cominos, P., & Munro, N. (2002). PID controllers: Recent tuning methods and design to specification. *IEE Proceedings-Control Theory and Applications, 149*(1), 46–53.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Cambridge, MA: The MIT Press.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperation agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, 26*(1), 29–41.

Dorigo, M. (1992). Optimization, learning and natural algorithms (in Italian). PhD Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dreo, J., & Siarry, P. (2002). A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continuous functions. *Lecture Notes on Computer Science, 2463*, 216–221.

Feki, M. (2009). Sliding mode control and synchronization of chaotic systems with parametric uncertainties. *Chaos, Solitons & Fractals, 41*(3), 1390–1400.

Fogel, D. B. (1995). *Evolutionary computation: Toward a new philosophy of machine intelligence*. Piscataway, NJ, USA: IEEE Press.

Goldberg, D. (1989). *Genetic algorithms in search optimization, and machine learning*. Massachusetts: Addison Wesley.

Guan, Z. H., Hill, D. J., & Yao, J. (2006). Hybrid impulsive and switching control strategy for synchronization of nonlinear systems and application to Chua's chaotic circuit. *International Journal of Bifurcation and Chaos, 16*, 229–238.

Hanbay, D., Turkoglu, I., & Demir, Y. (2008). An expert system based on wavelet decomposition and neural network for modeling Chua's circuit. *Expert Systems with Applications, 34*(4), 2278–2283.

Hu, B., Mann, G. K. I., & Gosine, R. G. (1999). New methodology for analytical and optimal design of fuzzy PID controllers. *IEEE Transactions on Fuzzy Systems, 7*(5), 521–539.

Huang, H., Li, H. X., & Zhong, J. (2006). Master-slave synchronization of general Lur'e systems with time-varying delay and parameter uncertainty. *International Journal of Bifurcation and Chaos, 16*, 281–294.

Hung, M. L., Lin, J. S., Yan, J. J., & Liao, T. L. (2008). Optimal PID control design for synchronization of delayed discrete chaotic systems. *Chaos, Solitons and Fractals, 35*(4), 781–785.

Jayaraman, V. K., Kulkarni, B. D., Gupta, K., Rajesh, J., & Kusumaker, H. S. (2001). Dynamic optimization of fed-batch bioreactors using the ant algorithm. *Biotechnology Progress, 17*(1), 81–88.

Krink, T., Filipic, B., Fogel, G., & Thomsen, R. (2004). Noisy optimization problems—A particular challenge for differential evolution. In *Proceedings of the sixth Congress on Evolutionary Computation (CEC-2004)* (Vol. 1, pp. 332–339). Piscataway, NJ, USA.

Lam, H. K., & Seneviratne, L. D. (2007). Synchronization of chaotic systems using neural-network-based controller. *International Journal of Bifurcation and Chaos, 17*, 2117–2125.

Li, C., & Liao, X. (2006). Anti-synchronization of a class of coupled chaotic systems via linear feedback control. *International Journal of Bifurcation and Chaos, 16*, 1041–1047.

Mathur, M., Karale, S. B., Priyee, S., Jayaraman, V. K., & Kulkarni, B. D. (2000). Ant colony approach to continuous function optimization. *Industrial Engineering Chemical Research, 39*, 3814–3822.

Monmarche, N., Venturini, G., & Slimane, M. (2000). On how pachcondyla apicalis ants suggest a new search algorithm. *Future Generation Computer Systems, 16*, 937–946.

Mullin, T. (1993). *The nature of chaos*. Oxford: Claredon Press.

Noman, N., & Iba, H. (2008). Accelerating differential evolution using an adaptive local search. *IEEE Transactions on Evolutionary Computation, 12*(1), 107–125.

Overschee, V., & De Moor, B. (2000). RAPID: The end of heuristic PID tuning. In *Proceedings of the IFAC workshop on digital control* (pp. 595–600). Terrassa, Spain.

Park, J. H. (2006). Chaos synchronization between two different chaotic dynamical systems. *Chaos, Solitons and Fractals, 27*(2), 549–554.

Parker, T. S., & Chua, L. O. (1989). *Practical numerical algorithms for chaotic system*. Heidelberg, Germany: Springer.

Pecora, L. M., & Carroll, T. L. (1990). Synchronization in chaotic systems. *Physical Review Letters, 64*, 821–824.

Peitgen, H. O., Jürgens, H., & Saupe, D. (2004). *Chaos and fractals: New frontiers of science* (2nd ed.). Heidelberg, Germany: Springer.

Peng, M. (2004). Bifurcation and chaotic behavior in the Euler method for a Kaplan–Yorke prototype delay model. *Chaos, Solitons and Fractals, 20*(3), 489–496.

Peng, M., Bai, E. W., & Lonngren, K. E. (2004). On the synchronization of delay discrete models. *Chaos, Solitons and Fractals, 22*(3), 573–576.

Price, K., Storn, R., & Lampinen, J. A. (2005). *Differential evolution – A practical approach to global optimization*. Heidelberg, Germany: Springer-Verlag.

Sakaguchi, A., & Yamamoto, T. (2002). A design of predictive PID control systems using GA and GMDH network. In *Proceedings of IEEE international conference on control applications* (pp. 266–271). Glasgow, Scotland.

Socha, K., & Dorigo, M. (2009). Ant colony optimization for continuous domains. *European Journal of Operational Research, 185*(3), 1155–1173.

Socha, K. (2004). ACO for continuous and mixed-variable optimization. In *Proceedings of fourth international workshop on ant colony optimization and swarm intelligence (ANTS'2004)*. Brussels, Belgium.

Strogatz, S. H. (2000). *Nonlinear dynamics and chaos*. MA: Perseus Publishing.

Swaminathan, S. (2006). Rule induction using ant colony optimization for mixed variable attributes. Thesis of Master in Computer Science, Graduate Faculty, Texas Tech University, Lubbock, TX, USA.

Ursem, R., & Vadstrup, R. (2003). Parameter identification of induction motors using differential evolution. In Proceedings of the fifth Congress on Evolutionary Computation (CEC-2003) (Vol. 1, pp. 790–796). Piscataway, NJ, USA.

Vesterström, J., & Thomsen, R. (2004). A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems. In *Proceedings of the sixth Congress on Evolutionary Computation (CEC-2004)* (Vol. 2, pp. 1980–1987). Piscataway, NJ, USA.

Vlachos, C., Williams, D., & Gomm, J. B. (2002). Solution to the Shell standard control problem using genetically tuned PID controllers. *Control Engineering Practice, 10*(2), 151–163.

Wang, X., & Wang, M. (2007). Adaptive synchronization for a class of high-dimensional autonomous uncertain chaotic systems. *International Journal of Modern Physics C, 18*, 399–406.

Wen, G., Wang, Q. G., Lin, C., Li, G., & Han, X. (2007). Chaos synchronization via multivariable PID control. *International Journal of Bifurcation and Chaos, 17*, 1753–1758.

Wojsznis W. K., & Blevins, T. L. (2002). Evaluating PID adaptive techniques for industrial implementation. In *Proceedings of American control conference* (pp. 1151–1155). Anchorage, AK, USA.

Wu, X., & Wang, M. (2006). Robust synchronization of chaotic Lur'e systems via replacing variables control. *International Journal of Bifurcation and Chaos, 16*, 3421–3433.

Yamamoto, T., Indue, A., & Shah, S. (1999). Generalized minimum variance self-tuning pole-assignment controller with a PID structure. In Proceedings of IEEE international conference on control applications (pp. 125–130). Hawaii, HI, USA.

Yan, J. J., Liao, T. L., Lin, J. S., & Cheng, C. J. (2006). Synchronization control of neural networks subject to time-varying delays and input nonlinearity. *International Journal of Bifurcation and Chaos, 16*, 3643–3654.

Yau, H. T., & Shieh, C. S. (2008). Chaos synchronization using fuzzy logic controller. *Nonlinear Analysis: Real World Applications, 9*(4), 180–1810.

Zhou, J., Chen, T., & Xiang, L. (2006). Adaptive synchronization of coupled chaotic delayed systems based on parameter identification and its applications. *International Journal of Bifurcation and Chaos, 16*, 2923–2933.

Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of ASME, 64*, 759–768.