

Modelado de procesos caóticos mediante inducción genética

José R. Villar, Luciano Sánchez

Departamento de Informática
Universidad de Oviedo
Campus de Viesques s/n
33203 Gijón

Resumen

Con frecuencia, los datos de variables pertenecientes a bio-sistemas son de comportamiento caótico. El modelado de sistemas caóticos suele implementarse mediante modelos que consigan obtener unos valores aceptables de comportamiento respecto al sistema que modela. La interpretabilidad del modelo obtenido no es un parámetro que suele ser tomado en cuenta. De igual manera, la decisión sobre cuando un modelo cumple o no las características deseadas se suele basar en valores derivados del error generado. En el modelado de sistemas caóticos, tan importante es disponer de un bajo error como que los parámetros de caracterización caótica del modelo obtenido sean lo más próximos a los del sistema a modelar. Para poder obtener un modelo válido según un número variable de criterios tiene sentido utilizar algoritmos multiobjetivo. En este trabajo se describe una propuesta evolutiva multicriterio en desarrollo para el modelado de sistemas caóticos representados mediante series temporales, con el objetivo de obtener un modelo analítico del mismo basado en sus ecuaciones en diferencias.

1. Introducción

Con frecuencia, los datos de variables pertenecientes a bio-sistemas son de comportamiento caótico. Es usual que los algoritmos de identificación de sistemas caóticos se basan en el empleo de modelos no interpretables ([6], [16], [21], [32]). Aunque con buenas propiedades numéricas ([13], [22]), estos modelos no proporcionan al ingeniero de control una visión acerca de las características del proceso más profunda que la que, por otra parte, el mismo

puede deducir a partir de estadísticos –como la dimensión de correlación y el exponente de Lyapunov ([10], [15], [25], [36], [38]), entre otros– que son directamente aplicables a un muestreo de las señales del proceso.

Por lo tanto es necesario obtener una descripción de las ecuaciones que rigen el comportamiento de un sistema si se desea analizar el sistema para su comprensión ([12], [26], [31]). Las técnicas evolutivas son una importante herramienta para obtener tales modelos, tal como reflejan varios autores ([5], [37]). En particular, en algunos trabajos recientes ([2], [3], [8], [11], [31]) se ha propuesto el uso de algoritmos genéticos para obtener modelos analíticos de sistemas caóticos como de sistemas lineales no caóticos, que en principio proporcionan la visión de la estructura del modelo, si bien no se resuelve en ellas el comportamiento caótico.

Por las razones mencionadas, en este trabajo se propone el uso de algoritmos genéticos multicriterio para optimizar un modelo basado en ecuaciones diferenciales de un proceso caótico donde, teniendo en cuenta tanto la diferencia numérica entre la predicción del modelo y la salida deseada en un número fiable de periodos, como que el comportamiento sea lo más próximo posible.

La estructura de este trabajo es la siguiente: en primer lugar, se hace un estudio comparado de los trabajos más recientes en identificación genética de modelos caóticos. En la siguiente sección, se explica con detalle las características del modelo que se propone. Por último, se ha realizado un análisis numérico comparado de las propiedades del método propuesto con las de otros enfoques, y se concluye el estudio con los comentarios finales y las líneas futuras de trabajo.

2. Sistemas genéticos en la identificación de sistemas caóticos

La aplicación de algoritmos genéticos a la identificación y modelado de sistemas caóticos puede diferenciarse en dos grandes bloques por ser los más extendidos, los de caja negra y los de caja blanca. Es posible estudiar el resultado de la evaluación de los algoritmos genéticos frente a otras técnicas de identificación en ([20], [33]).

Entre los trabajos documentados que utilizan modelos de caja negra se encuentran ([4], [14], [34], [27]). En el primero de ellos se utiliza evolución genética para calcular los umbrales de una transformación wavelet. En estos trabajos, se hace uso de algoritmos genéticos para entrenar sistemas neuro-borrosos o redes neuronales con los que modelar el sistema. En todos ellos, la evaluación de los individuos se basa en la medida de parámetros dependientes del error instantáneo cometido, con lo que la capacidad de predicción de estos no cabe esperar sea alta. También es usual el modelo polinómico, como en ([6], [30]), trabajos que se comentan mas adelante.

Entre los trabajos documentados que utilizan un modelo de caja blanca se puede nombrar los presentados en ([11], [5], [1]), en los que el interés radica en obtener el modelo estructural del sistema. Nuevamente, el error instantáneo es utilizado para la evaluación de los individuos, excepto en ([1]), donde utilizan el error en la predicción como medida de fitness. En ([8]) se utilizó programación evolutiva para obtener los parámetros de un sistema caótico dada su estructura.

La inclusión de un método de búsqueda local esta también documentado ([2], [7], [17]). De igual manera, el coste computacional de este tipo de modelado está documentado en ([5], [28]). Es por ello que se incluyen gramáticas de operadores y operandos, así como restricciones del tipo dimensional a los procesos de evolución, con el objetivo de acotar en cierto grado dicho coste computacional ([7], [17], [28]).

La división del modelado en modelado de la parte estructural (conjunto de ecuaciones que rigen la dinámica de un sistema) mas estimación de los parámetros y constantes del mismo suele abordarse mediante programación genética la primera, y algoritmo genético la segunda, generando los algoritmos denominados GAP ([5],

[7], [17]), si bien éste último es utilizado para modelar sistemas no lineales.

Tanto en ([5]) como en ([7]) se utiliza un algoritmo GAP hibridado con Nelder-Mead ([23]) para modelado de sistemas caóticos. En este caso, se valoran diferentes parámetros aparte de los derivados del error instantáneo, como pueden ser la dimensión de correlación y el mayor exponente de Lyapunov. La función de fitness es una combinación lineal de los errores debidos a todos los parámetros considerados.

También se ha utilizado evolución genética multiobjetivo, como se presenta en ([6], [30]). Ambos trabajos utilizan la aproximación de optimalidad de Pareto. Si bien existen otras interpretaciones para la evaluación multicriterio según la optimalidad de Pareto, como puede ser la aproximación estricta de Pareto (Strength Pareto Evolutionary Algorithm, SPEA) y la extensión por ordenación de no dominados (extensión of the nondominated sorting, NSGA), y hasta nuestro conocimiento, éstas no han sido utilizadas en modelado de sistemas caóticos, y por lo tanto, forma parte de trabajos futuros como ampliación del presente trabajo.

Como se mostrará seguidamente, los trabajos ([6], [30], [5], [7], [17], [35]) forman parte de la base conceptual de partida para la propuesta a describir en la presente contribución.

3. Prototipo para modelado de sistemas de dinámica caótica

El objetivo de este trabajo es estudiar la generación de modelos analíticos de sistemas cuyas dinámicas tengan características caóticas mediante la aplicación de un algoritmo genético multicriterio hibridado con programación genética –en adelante GAP–, el primero de ellos para la caracterización de las constantes del modelo, el segundo para la obtención de la estructura del modelo. La propuesta incluye el uso de unas etapas de búsqueda local que ajuste las constantes. El sistema se tiene representado en ecuaciones en diferencias dado que es más sencillo de simular y computacionalmente es menos costoso ([11]).

La propuesta se basa en los trabajos relativos a análisis de dinámicas caóticas y dinámicas no lineales ([7], [17]), además de otros trabajos que se irán detallando a medida que surja la referencia a la aportación. La principal aportación de este

trabajo reside en la evaluación multicriterio del algoritmo GA-P, utilizando una función de fitness que estudia las similitudes entre las dinámicas de la serie a modelar y las de los diferentes individuos de la población. La evaluación multicriterio se basa en las ideas planteadas en los trabajos ([6], [30], [32]), siendo algunas de las medidas las indicadas en ([1], [35]). Los métodos de cálculo de los invariantes descriptivos de la dinámica caótica son los indicados en ([25], [36]).

3.1. Descripción del problema

El problema viene definido por un sistema de n variables observables, y una de ellas denominada como objetivo. No se conoce si entre las variables observables existe o no relación.

Por lo tanto, el sistema está representado por series de datos de las variables conocidas. Para estas variables se desea obtener un modelo analítico de la forma indicada en la ecuación (1).

$$X_{k+1}^i = f_i(X_k^1, \dots, X_k^m) \quad (1)$$

$$\forall i \in \{1, \dots, m\} \wedge m \leq n$$

donde m es el número de variables dependientes, n es el número de variables que observables del sistema.

3.2. Representación estructural de los individuos de la población

Cada individuo dispone de una representación estructural, la cuál intervendrá en el proceso de evolución según programación genética. Se utiliza notación prefija, con una jerarquía de nodos acorde para soportar una gramática. La representación soporta la gramática indicada en el algoritmo (2).

Un modelo dispone de una o mas ecuaciones, cada ecuación dispone de una variable de salida asociada a una de las variables del sistema. De igual manera, cada ecuación contiene uno o mas nodos, siendo los nodos de tipo funcional o terminal.

S → Exp
Exp → Función | Terminal
Función → Ope. Aritmético |
 Ope. No lineal | Ope. Temporal
Terminal → Variable |
 Constante No Paramétrica

Variable → variable del proceso
Constante → Constante Paramétrica
 | Constante No Paramétrica
Constante No Paramétrica → valor
 real entre [-100, 100]
Constante Paramétrica → valor
 real entre [-100, 100]
Ope. Aritmético → + Exp Exp |
 -Exp Exp | * Exp Exp |
 / Exp Exp
Ope. No Lineal → Sat CI CS CV Exp
 | Zm CI CS CV Exp
Ope. Temporal → Ret Variable
CI, CS → Constante Paramétrica
 $100 \leq CI \leq CS \leq 100$
CV → Constante Paramétrica
 $CV \in [-100, 100]$

Algoritmo 1. Gramática utilizada

De esta forma, cada ecuación es un vector de nodos ordenados, y un individuo es un conjunto de vectores encadenados. Como consecuencia, una hipótesis de modelo está representada por una lista de nodos que forman el sistema de ecuaciones.

3.3. Representación paramétrica de los individuos de la población

Cada individuo, además de poseer una representación estructural para la evolución según programación genética, dispone de una representación paramétrica para la evolución según algoritmos genéticos.

Cada ecuación es responsable de las constantes que almacena, disponiéndolas ordenadas. Como a su vez, las ecuaciones están ordenadas dentro del modelo, es posible obtener un vector de constantes. Este vector de constantes es la representación paramétrica del individuo. Se indicará un límite al número de constantes utilizadas por cada individuo.

3.4. Estructura del algoritmo de modelado

Como ya se ha indicado, el algoritmo propuesto es un MOGAP (multi objective genetic algorithm-programming) con búsqueda local implementada como algoritmo anytime, ejecutando solo unos pocos pasos de dicha búsqueda. Con probabilidad p se ejecutará el algoritmo genético, con

probabilidad $q - q = (1-p)$ – se ejecutará programación genética.

Tras la evolución se propone la búsqueda local. El algoritmo de búsqueda local propuesto es el de Nelder & Mead ([17], [23], [29]) dado su carácter multidimensional, si bien, al momento de la escritura de este trabajo dicho algoritmo no ha sido todavía puesto en funcionamiento. El algoritmo propuesto es el indicado a continuación.

```

t = 0;
P0 ← Generar población inicial
Evaluar la población P0
Mientras no se cumplan los
criterios marcados y no haya
generaciones suficientes
  Op ← Selección de Operadores
  Pt+1 ← Selección de individuos
  Pt+1 ← Cruce( Pt+1, Op)
  Pt+1 ← Mutación( Pt+1, Op)
  t++
  Evaluar Pt
  Selección mejores individuos

```

Algoritmo 2. Algoritmo MOGAP propuesto

3.5. Generación de la población inicial de individuos

En los trabajos publicados ([18], [19]) se estudian y comparan diversos algoritmos para la generación de la población inicial en programación genética. Se eligió el algoritmo PTC2 ([19]) para especificar tanto el tamaño en número de nodos, como la profundidad máxima, las probabilidades de aparición de cada tipo de nodo, y la distribución de probabilidad de las profundidades de las ecuaciones. Para las Constantes paramétricas se generará un valor aleatorio.

3.6. Objetivos y criterios a cumplir por los individuos

Un criterio está formado por el cumplimiento de todos los objetivos asociados. La evaluación de los individuos utiliza la evaluación multicriterio basada en la no dominancia o suboptimalidad de Pareto, tal como se presenta en ([9], [30], [32]).

Los objetivos planteados son minimizar el error cuadrático (ec) de la salida instantánea respecto a la salida del sistema, menor

discrepancia (en signo y valor) en el mayor exponente de Lyapunov (λ) entre la serie de datos salida del sistema y la obtenida como evolución del modelo, y menor discrepancia en dimensión de correlación (dc) entre la serie de datos salida del sistema y la obtenida como evolución del modelo – la cuál debe ser mayor que la unidad–.

Partiendo de estos objetivos, los criterios a utilizar son los siguientes: Criterio 1: (ec, dc, λ), Criterio 2: (dc, λ), Criterio 3: dc, Criterio 4: λ , Criterio 5: ec.

3.7. Operador genético de selección

Como se ha indicado, cada individuo se evalúa respecto a cada criterio, estableciendo cada criterio una ordenación parcial. Si se ordenan los criterios (no los objetivos, que se evalúan sin orden según Pareto), entonces se puede obtener una ordenación completa de los individuos.

Esta ordenación es utilizada para seleccionar los k mejores individuos, siendo k un parámetro del algoritmo. Está previsto implementar K-Random Opponents ([24]).

3.8. Operador genético de cruce estructural

Para realizar el cruce estructural entre dos individuos, y dado que se dispone de un vector de nodos por cada individuo, se seleccionarán un nodo aleatoriamente por individuo, y se intercambiarán los subárboles a partir de dichos nodos. Se generarán dos nuevos individuos con diferentes estructuras si se cumplen las restricciones en número de nodos y profundidad.

3.9. Operador genético de mutación estructural

La mutación estructural se realizará siguiendo las reglas indicadas mediante la semántica presentada en la sección *Representación de las estructuras*.

3.10. Operador genético de cruce paramétrico

Para realizar el cruce paramétrico se tendrá en cuenta solo y exclusivamente a las Constantes no paramétricas del individuo. Cada individuo dispone de un vector de Constantes. Para realizar el cruce de dos individuos, se seleccionarán un

valor aleatorio entre los posibles índices del vector de constantes y se generarán dos nuevos individuos intercambiando los vectores sobre dicho valor aleatorio

3.11. Operador genético de mutación paramétrica

La mutación paramétrica generará un nuevo individuo. Para ello se realizará seleccionando aleatoriamente una constante del individuo, incluyendo tanto las constantes no paramétricas como las paramétricas, y se le asignará un valor aleatorio. Para las constantes paramétricas se supervisará que cumplen las restricciones asociadas.

4. Ensayos y resultados

Para evaluar el prototipo hasta ahora implementado se ha vuelto al objetivo inicial: que el sistema disponga de un bajo error cuadrático y que mantenga las características caóticas de la serie a modelar. Por ello, las pruebas realizadas se basan en comparar un algoritmo GAP con el MOGAP propuesto. El algoritmo GAP utilizado dispone de una función fitness basada en el error cuadrático, la dimensión de correlación y el mayor exponente de Lyapunov.

La serie considerada es el Mapa de Henon, la cuál es mostrada en la ecuación (2).

$$\begin{cases} x_{n+1} = y_n - 1.4 \cdot x_n^2 + 1 \\ y_{n+1} = 0.3 \cdot x_n \end{cases} \quad (2)$$

Todo entrenamiento se ha realizado con 1000 generaciones con población 100 individuos, 30 de los cuales sobrevivirán. La probabilidad de operadores genéticos es de 0.5, y de 0.5 de programación genética. Los operadores utilizados son el conjunto {+, -, *, /, Sat, Ret, Zm}, con probabilidades {0.21, 0.21, 0.21, 0.21, 0.02, 0.12, 0.02}. La profundidad máxima es de 10 operandos, el número máximo de nodos 20. Las probabilidades de profundidad y de distribución de número de nodos, a efectos de generación de individuos, se mantuvieron constantes en todos los ensayos.

Al momento de escribir este trabajo se detectó un error en las rutinas de caracterización caótica,

lo que ha producido que los modelos resultantes no fueren correctamente evaluados en el fitness, induciendo errores considerables.

Sin embargo, dado que lo que se pretende demostrar es la utilidad de sistemas multicriterio para obtener mejores modelos basados en el comportamiento, asumimos que dichas evaluaciones eran correctas y nos centramos en la evolución de las poblaciones. Si las evoluciones son adecuadas el modelo puede ser válido. Para este análisis se han revisado los archivos logs de los análisis, donde se almacenaban todos los sucesos de evaluación de cada individuo en cada generación.

El resultado observado en las evoluciones del GAP marcan una relación importante del error cuadrático como medida que facilita la aceptación de individuos en la evolución. El error cuadrático puede ser muy bajo, y sin embargo, no disponer de características caóticas.

En el caso de MOGAP, se observa que esta dependencia no existe, pues se busca la minimización global de los parámetros. Se observa la necesidad de las restricciones lexicográficas para las características caóticas. Pero a diferencia del caso GAP, aparecen modelos con características caóticas similares a las de la serie y con peor error cuadrático. Esto es el comportamiento buscado en el algoritmo, y con todos los matices indicados en esta sección, apunta a una correcta formulación.

5. Conclusiones y trabajos futuros

Tras los resultados obtenidos se puede concluir que el prototipo necesita aún mucha mejora para conseguir resultados tangibles. Sin embargo, parece apuntar que una vez se implementen las mejoras el funcionamiento del prototipo será el planteado como hipótesis de trabajo. Además, la implementación de todos los aspectos de la presente propuesta influirán, a nuestro entender, en un mejor modelado final.

Es importante la pérdida de competitividad en resultados ocasionado por la ausencia de subpoblaciones y de nichos, elementos que se incorporarán en breve en el sistema. De igual manera, es notable la importancia de las restricciones lexicográficas para podar los elementos que aún teniendo ciertas características deseables tienen otras con valores no aceptables.

Todos estas ideas forman parte de trabajos futuros de esta línea de investigación.

Otros trabajos futuros una vez finalizado el prototipo será estudiar las diferentes técnicas existentes para evolución multiobjetivo, y probar el prototipo con señales biológicas con el fin de obtener modelos analíticos de procesos biológicos complejos. Dado que este sistema admite cualquier combinación de variables de entrada y propone relaciones existentes entre subconjuntos de las mismas pensamos que será un diseño apropiado para afrontar con cierto grado de éxito el modelado analítico de tales sistemas, en concreto, los bio-sistemas.

Agradecimientos

El trabajo aquí presentado ha sido desarrollado en el ámbito del proyecto Keel, financiado por el Ministerio de Ciencia y Tecnología con referencia TIC-04036-C05-05.

Referencias

- [1] Alvarez, A., Orfila, A., Tintore, J.; DARWIN: An evolutionary program for nonlinear modeling of chaotic time series; *Computer Physics Communications* 136 (2001) 334–349
- [2] Cao, H., Guo, L., Chen, Y., Guo, T.; The Dynamic Evolutionary Modeling of HODEs for Time Series Prediction; *Computers and Mathematics with Applications* 46 (2003), 1397-1411
- [3] Chang, Y., Park, K. S., Kim, B. Y.; Nonlinear model for ECG R–R interval variation using genetic programming approach; *Future Generation Computer Systems* (in press 2004)
- [4] Chung, I-F., Lin, C-J., Lin, C-T.; A GA-based fuzzy adaptive learning control network; *Fuzzy Sets and Systems* 112 (2000), 65-84
- [5] Downing, K.; Using evolutionary computational techniques in environmental modelling; *Environmental Modelling & Software* 13 (1998) 519–528
- [6] Evans, C., Fleming, P.J., Hill, D.C., Norton, J.P., Pratt, I., Rees, D., Rodríguez-Vázquez, K.; Application of system identification techniques to aircraft gas turbine engines; *Control Engineering Practice* 9 (2001) 135-148
- [7] Fernandez, A. I.; Identificación de sistemas con dinámica caótica a partir de series con ruido mediante algoritmos GA-P; Tesis Doctoral, Universidad de Oviedo (1999)
- [8] Fogel, D. B., Fogel, L. J.; Preliminary Experiments on Discriminating between Chaotic Signals and Noise using Evolutionary Programming; *Genetic Programming 96*, eds. J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, The MIT Press, Cambridge, MA, 1996
- [9] Fonseca, C.S., Fleming, P.J.; Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms II : Application Example; *IEEE Transactions on Systems, Man and Cybernetics*, vol 28, 1 (1998) 26-37
- [10] Grassberger, P., Procaccia, I.; Measuring the strangeness of strange attractors; *Physica D*, Vol 9, 1983, 189-208
- [11] Gray, G. J., Murray-Smith, D. J., Li, Y., Sharman, K. C., Weinbrenner, T.; Nonlinear model structure identification using genetic programming; *Control Engineering Practice* 6 (1998) 1341-1352
- [12] Izquierdo, J., Perez, P., Iglesias, P. L.; Mathematical Models and Methods in the Water Industry; *Mathematical and Computer Modelling* 39 (2004) 1353-1374
- [13] Jiang, W., Guo-Dong, Q., Bin, D.; Observer-based robust adaptive variable universe fuzzy control for chaotic system; *Chaos, Solutions and Fractals* 23, 2005, 1013–1032
- [14] Kim, D.; Improving the fuzzy system performance by fuzzy system ensemble; *Fuzzy Sets and Systems* 98 (1998) 43-56
- [15] Kugiumtzis, D., Lillekjendlie, B., Christophersen, N.; Chaotic time series. Part I: Estimation of some invariant properties in state space; *Modeling, Identification and Control*, vol 4, 15 (1995) 205-224
- [16] Lillekjendlie, B., Kugiumtzis, D., Christophersen, N.; Chaotic time series part II: System identification and prediction; *Modeling, Identification and Control*, vol 4, 15 (1995) : 225-243.
- [17] López, A.; Modelado de Sistemas Dinámicos Mediante Algoritmos GP con Estructuras Basadas en Diagramas de Bloques; Tesis Doctoral, Universidad de Oviedo (2002)
- [18] Luke, S., Panait, L.; A Survey and Comparison of Tree Generation Algorithms;

- In GECCO-2001: Proceedings of the Genetic and Evolutionary Computation Conference. Lee Spector et al, eds. Morgan Kaufmann, (2001) 81-88.
- [19] Luke, S.; Two Fast Tree-Creation Algorithms for Genetic Programming; IEEE Transactions on Evolutionary Computation vol 4 n° 3 September (2000), 274-283
- [20] Mak, M.W., Ku, K.W., Lu, Y.L.; On the improvement of the real time recurrent learning algorithm for recurrent neural networks; Neurocomputing 24 (1999) 13-36
- [21] Mei-Ying, Y., Xiao-Dong, W.; Chaotic time series prediction using least squares support vector machines; Chinese Physics 13 (2004) 454-458
- [22] Mukherjee, S., Osuna, E., Girosi, F.; Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines; Proceedings of IEEE NNSP'97, Amelia Island, FL, Sep. 1997, 24-26
- [23] Nelder, J. A., Mead, R.; A simplex method for function minimization; The Computer Journal, Volume 7, Issue 4, 308-313
- [24] Panait, L., Luke, S.; A Comparison of Two Competitive Fitness Functions; In GECCO-2002: Proceedings of the Genetic and Evolutionary Computation Conference. W. B. Langdon et al editor(s), Morgan Kaufmann Publishers (2002) pag 503-511
- [25] Parker, T. S., Chua, L. O.; Chaos: A Tutorial for Engineers; Proceedings of IEEE, Vol 75, n° 8, 1987, 982-1008
- [26] Petrill, S. A., Pike, A., Price, T., Plomin, R.; Chaos in the home and socioeconomic status are associated cognitive development in early childhood: Environmental mediators identified in a genetic design; Intelligence 32 (2004) 445-460
- [27] Potocnik, P., Grabec, I.; Nonlinear model predictive control of a cutting process; Neurocomputing 43 (2002), 107-126
- [28] Ratle, A., Sebag, M.; Grammar-guided genetic programming and dimensional consistency: application to non-parametric identification in mechanics; Applied Soft Computing 1 (2001), 105-118
- [29] Renders, J.-M., Bersini, H.; Hybridizing Genetic Algorithms with Hill-Climbing Methods for Global Optimization: Two Possible Ways; In Proceedings of the First IEEE Conf. On Evolutionary Programming (1994)
- [30] Rodríguez-Vázquez, K., Fonseca, C. M., Fleming, P. J.; Identifying the Structure of NonLinear Dynamic Systems Using Multiobjective Genetic Programming; IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans vol 34, 4, (2004), 531-545
- [31] Rowland, J.J.; Model selection methodology in supervised learning with evolutionary computation; BioSystems 72 (2003) 187-196
- [32] Ruano, A. E., Fleming, P.J., Teixeira, C., Rodriguez-Vazquez, K., Fonseca, C.M.; Nonlinear identification of aircraft gas-turbine dynamics; Neurocomputing 55, 2003, 551-579
- [33] Sexton, R. S., Gupta, J. N. D.; Comparative evaluation of genetic algorithm and backpropagation for training neural networks; Information Sciences 129 (2000), 45-59
- [34] Shim, T., Han, I.; Optimal signal multi-resolution by genetic algorithms to support artificial neural networks for exchange-rate forecasting; Expert Systems with Applications 18 (2000), 257-269
- [35] Wei, Z., Zhi-ming, W., Gen-ke, Y.; Genetic programming-based chaotic time series modeling; Journal of Zhejiang University SCIENCE vol 5, 11, (2004), 1432-1439
- [36] Wolf, A., Swift, J. B., Swinney, H. L., Vastano, J. A.; Determining Lyapunov Exponents from a Time Series; Physica D 16, 1985, 285-317
- [37] Woodward, A. M., Gilbert, R. J., Kell, D. B.; Genetic programming as an analytical tool for non-linear dielectric spectroscopy; Bioelectrochemistry and Bioenergetics 48 (1999) 389-396
- [38] Yang, G., Lu, J., Luo, A. C.J.; On the computation of Lyapunov exponents for forced vibration of a Lennard-Jones oscillator; Chaos, Solutions and Fractals 23 (2005) 833-841