A NEW METHODOLOGY FOR EMERGENT SYSTEM IDENTIFICATION USING PARTICLE SWARM OPTIMIZATION (PSO) AND THE GROUP METHOD OF DATA HANDLING (GMDH)

Mark S. Voss
Dept. of Civil and Environmental Engineering
Marquette University
Milwaukee, WI
Voss@Rocketmail.com

Xin Feng
Dept. of Electrical and Computer Engineering
Marquette University
Milwaukee, WI
Xin.Feng@Marquette.edu

Abstract

A new methodology for Emergent System Identification is proposed in this paper. The new method applies the self-organizing Group Method of Data Handling (GMDH) functional networks, Particle Swarm Optimization (PSO), and Genetic Programming (GP) that is effective in identifying complex dynamic systems. The focus of the paper will be on how Particle Swarm Optimization (PSO) is applied within Group Method of Data Handling (GMDH) which is used as the modeling framework.

1 INTRODUCTION

The methodology of System Identification was developed for the extraction of mathematical models from system data. Evolutionary System Identification has been used to designate the use of evolutionary computation for the determination of the approximate mathematical model from experimental data. Evolutionary systems rely on a notion of competition among a population of individuals that compete to reproduce to form future generations. Emergence is used to describe the self-organization (order for free) exhibited by Complex Dynamic Systems. Emergent systems rely on the self-organization properties of the underlying system (information) (Holland, 1998). These emergent systems use iterative stochastic methodologies to discover the underlying connections implied by the system data. From this perspective, evolutionary methodologies are also emergent, but the opposite is not always true. The methodology of Emergent System Identification that is proposed here is concerned with extending the concept of Evolutionary System Identification by combining the methodologies of System Identification (Pandit, 1984) self-organizing functional networks (GMDH) (Ivakhnenko, 1968a, 1971b; Madala and Ivakhnenko, 1994), Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 2001a) and Genetic Programming (GP) (Iba and Kurita, 1994).

The rest of this paper is organized as follows. Section 2 describes traditional System Identification and introduces the use of Particle Swarm Optimization (PSO) for determining the coefficients of a simple autoregressive moving average model (SwARMA). Section 3 explains Particle Swarm Optimization. Section 4 describes the results of using PSO for determining the ARMA model parameter (SwARMA) for an example problem. Section 5 introduces and explains the Group Method of Data Handling (GMDH) and the extension of the GMDH algorithms using PSO. Section 6 describes the results of using the GMDH combined with PSO for two example problems and an additional example problem that illustrates nodal selection criterion. The paper ends with the primary conclusions we draw from the results.

2 SYSTEM IDENTIFICATION

Generally speaking, the discipline of system Identification is concerned with the derivation of mathematical models from experimental data. When given a data set one typically applies a set of candidate models and chooses one of the models based on a set of rules by which the models can be assessed. One of the simplest System Identification models is the Autoregressive Moving Average (ARMA) model as shown in equation 1.

$$\begin{aligned} & y_t + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_n y_{t-n} \\ & = \theta_0 + \theta_1 a_{t-1} + \theta_2 a_{t-2} + \cdots + \theta_m a_{t-m} \end{aligned} \tag{1}$$

where ϕ_i, $i = 1,n$ and θ_j, $j = 0, m$ are the parameters for an ARMA(n,m) model. For a given ARMA(n,m) model the model parameters ϕ_i and θ_j are selected such that equation (2) is minimized,

$$\sum_{t=1}^{N} \left(y_{t\_data} - y_{t\_ARMA} \right)^2 \tag{2}$$

where,

$$N = \text{ number of data points.} \tag{3}$$

ARMA models are numerically efficient due to their ability to utilize traditional parameter estimation methods and typically employ non-linear least squares for the determination of their parameters. In this paper it is shown that Particle Swarm Optimization (PSO) can be used to determine the parameters for ARMA models. The use of PSO to determine the constants of the ARMA model is denoted by what the authors are calling SwARMA. It will be shown that SwARMA was able to determine a better parameterization for the ARMA model than the IMSL (International Mathematical and Statistical Libraries) routines.

3 PARTICLE SWARM OPTIMIZATION

The Particle Swarm Algorithm is an adaptive algorithm based on a social-psychological metaphor (Kennedy and Eberhart, 2001a). A population of individuals adapt by returning stochastically towards previously successful regions in the search space, and are influenced by the successes of their topological neighbors. Most particle swarms are based on two sociometric principles. Particles fly through the solution space and are influenced by both the best particle in the particle population and the best solution that a current particle has discovered so far. The best particle in the population is typically denoted by (global best), while the best position that has been visited by the current particle is denoted by (local best). The (global best) individual conceptually connects all members of the population to one another. That is, each particle is influenced by the very best performance of any member in the entire population. The (local best) individual is conceptually seen as the ability for particles to remember past personal successes.

Particle Swarm Optimization is a relatively new addition to the evolutionary computation methodology, but the performance of PSO has been shown to be competitive with more mature methodologies (Eberhart and Shi, 1998a; Kennedy and Spears, 1998). Since it is relatively straightforward to extend PSO by attaching mechanisms employed by other evolutionary computation methods that increase their performance; PSO has the potential to become an excellent framework for building custom high-performance stochastic optimizers (Løvbjerg, *et al.*, 2001). It is interesting to note that PSO can be considered as a form of continuous valued Cellular Automata. This allows its hybridizations to extend into areas other than computational intelligence (Kennedy and Eberhart, 2001).

3.1 PSO EQUATIONS

The ith particle is represented as,

$$X_I = (x_{i1}, x_{i2}, ..., x_{iD}) \tag{4}$$

where D is the dimensionality of the problem. The rate of the position change (velocity) of the i^{th} particle is represented by,

$$V_I = (v_{i1}, v_{i2}, ..., v_{iD}) \tag{5}$$

where v_{ik} is the velocity for dimension "k" for particle "i". The best previous position (the position giving the best fitness value) of the i^{th} particle is represented as,

$$P_I = (p_{i1}, p_{i2}, ..., p_{iD}) \tag{6}$$

The best previous position so far achieved by any of the particles (the position giving the best fitness value) of the i^{th} particle is recorded and represented as,

$$P_G = (p_{g1}, p_{g2}, ..., p_{gD}) \tag{7}$$

On each iteration the velocity for each dimension of each particle is updated by,

$$v_{ik} = w_k v_{ik} + c_1 \varphi_1 p_{ik} + c_2 \varphi_2 p_{gk} \quad \{k, g\} \in \{1, 2, ..., D\} \tag{8}$$

where w_k is the inertia weight that typically ranges from 0.9 to 1.2. c_1 and c_2 are constant values typically in the range of 2 to 4. These constants are multiplied by φ (a uniform random number between 0 and 1) and a measure of how far the particle is from its personal best and the best particle so far. From a social point of view, the particle moves based on its current direction (w_k), its memory of where it found its personal best (p_{ik}), and a desire to be like the best particle in the population (p_{gk}).

3.2 PSO – POSITION UPDATE RULE

After a new velocity for each particle is calculated, each particle's position is updated according to:

$$x_{ik} = x_{ik} + v_{ik} \tag{9}$$

It typically takes a particle swarm a few hundred to a few thousand updates for convergence depending on the parameter selections within the PSO algorithm (Eberhart and Shi, 1998b).

3.3 RESULTS: PSO+ARMA

Particle Swarm Optimization was used to determine the ARMA parameters for the Wolfer Sunspot Data (1770-1869). The results from the study are shown in Fig. 1. This model was chosen to demonstrate the use of PSO on a well understood System Identification problem. The authors were somewhat surprised at the results. The particle swarm converged after a few thousand iterations in less that a minute on a 200 Mhz Pentium PC. In all cases the solution found was substantially better than those found using the IMSL Libraries.

For both the ARMA(2,1) and ARMA(4,2) models the PSO solution was better than the IMSL solution. In light of these results the authors chose to call the combination of PSO with ARMA: SwARMA (Voss and Feng, 2001). These results suggest some interesting future research with regards to traditional System Identification.

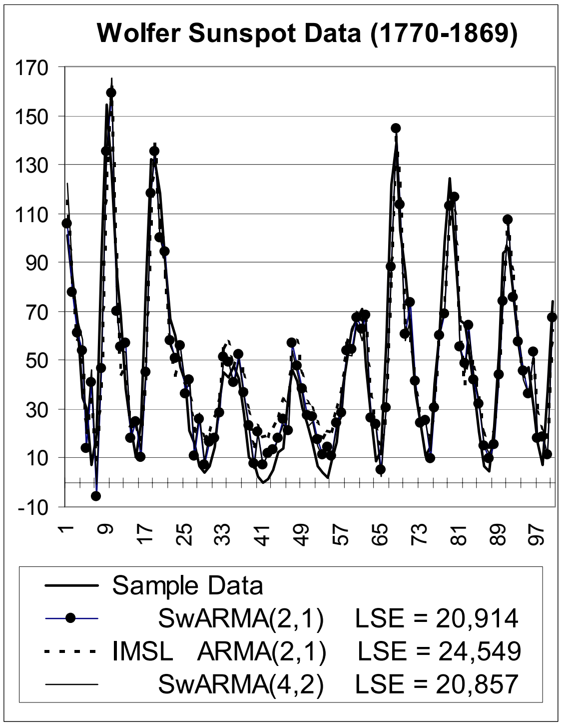

Fig. 1. Wolfer Sunspot Data.
Static SwARMA models. Weight = 0.7.

4 THE GROUP METHOD OF DATA HANDELING

The group method of data handling (GMDH) was first proposed by Alexy G. Ivakhnenko (1968). The traditional GMDH method is based on an underlying assumption that the data can be modeled by using an approximation of the Voltera Series or Kolmorgorov-Gabor polynomial as shown in the following equation,

$$y = a_0 + \sum_{i=1}^{m} a_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} a_i x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} a_i x_i x_j x_k \cdots \qquad (10)$$

Ivakhnenko accomplished this by using a feed-forward self-organizing polynomial functional network shown in Fig 2.

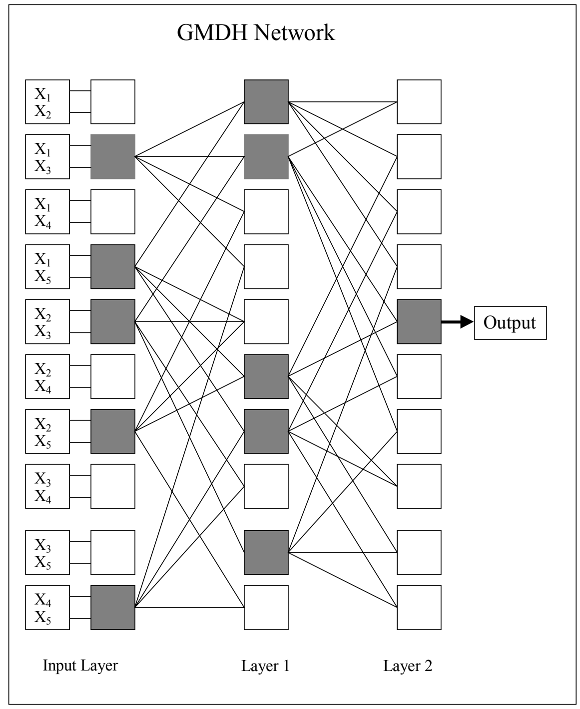

Fig. 2. GHDH forward feed functional network

The inputs to the input layer are determined by taking all combinations (taken two at a time) of the input vector "x_i". Each combination of inputs forms an input node that tries to model the corresponding system output using a second order polynomial surface specified by the polynomial in equation 11.

$$y = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2 + c_4 x_1^2 + c_5 x_2^2 \qquad (11)$$

The nodes in the input layer that do the "best job" (shaded nodes) at modeling the system output are retained and form the input to the next layer. The inputs for layer 1 are formed by taking all combinations of the surviving output approximations from the input layer nodes. It is seen that at each layer the order of the approximation is increased by two. The layer 2 nodes that do the "best job" at approximating the system output are retained and form the layer 3 inputs. This process is repeated until the current layer's best approximation is inferior to the previous layer's best approximation. The previous layer's best approximation is then used as the final solution. Determining the method for ranking the nodes at a given layer is somewhat problem dependant. Typically the data is spit into two groups. One data group is used to train the network and the other data group is used to rank the nodes to determine which nodes survive to form the input to the next layer. The GMDH can thus be seen as a methodology for distributed self-organizing computation.

4.1 EXTENDING THE GMDH

The GMDH can be used as an embryo for more complex methodologies for distributed self-organizing computation (Nikolaev,N. and Iba,H., 2001). The methodology is modified here by substituting equation 12 in place of the traditional six term linear polynomial approximation,

$$y = c_0 + c_1 x_1 + c_2 x_2 + sign(x_1)\,|\,x_1\,|^{c_3} + sign(x_2)\,|\,x_2\,|^{c_4} \tag{12}$$

This non-linear equation was designed to test the application of Particle Swarm Optimization for nodal optimization within a GMDH network. The non-linear equation has one less parameter than the traditional polynomial approximation and does not admit the application of simple gradient based optimization methods due to the incorporation of the absolute value function. The GMDH methodology has also been used as a starting point for many new approaches to the System Identification problem (Iba and Kurita, 1994). Here it is demonstrated that it is practical to allow for low-level non-linear emergent nodal representations embedded in a higher-level self-organizing network. This is the type hierarchical model that is necessary for the methodology of Emergent System Identification.

5 RESULTS: PSO + GMDH

Three example problems were considered. The first problem was a test of the applicability of substituting equation 12 in place of equation 11 in a GMDH node. The second problem was a comparison of using equations 11 and 12 for the System Identification of a simple string vibrating in a non-linear fluid, where the damping force was set proportional to the square of the velocity of the string movement. The third problem was the prediction of natural gas flow for a location in the Midwest United States.

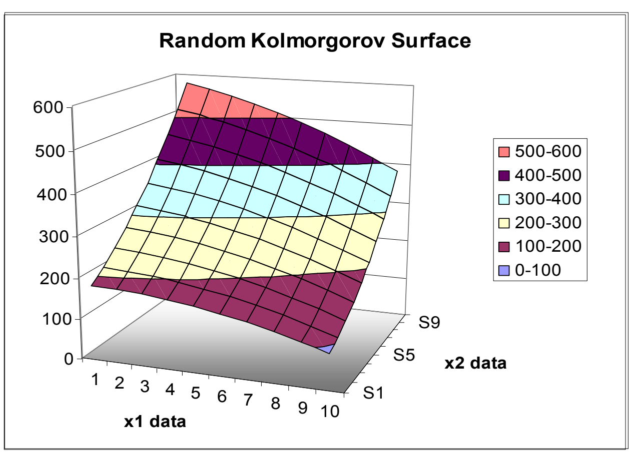

Fig. 3. Random Kolmorgorov Surface

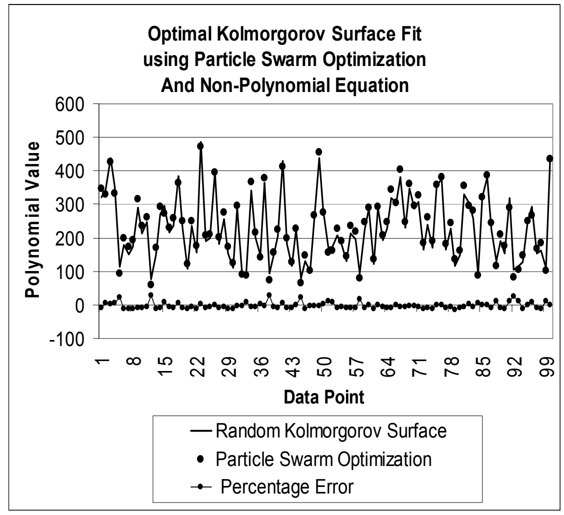

Fig. 4. PSO - non-linear surface fit

5.1 GMDH AND NON-LINEAR NODAL FUNCTIONAL REPRESENTATION

For the purposes of determining the applicability of equation 6 a surface as shown in Fig. 3 was generated using 100 random values for x_1 and x_2 between 0 and 1 using the following form of equation 11.

$$y = 150 - 175x_1 + 500x_2 - 200x_1x_2 + 100{x_1}^2 - 175{x_2}^2 \tag{13}$$

The results for using Particle Swarm Optimization are shown in Fig. 4.

The results were very good considering that the Particle Swarm Algorithm parameters where not optimized for solving this problem. These results lend support for the use of Particle Swarm Optimization in combination with non-linear formulations for the GMDH nodes (such as equation 12) within the GMDH methodology.

5.2 VIBRATING STRING – GMDH AND NON-LINEAR NODAL FUNCTIONAL REPRESENTATION

Since we are investigating the application of Particle Swarm Optimization for its utility in optimizing non-linear models within the GMDH nodes, the specifics of the discrete string model are not given here. The inputs to the GMDH were the previous four amplitudes calculated at the center of a string vibrating in a non-linear fluid. The string was given an initial displacement of 1.0 and then released. The previous four amplitudes were then used to predict the future position of the string at its central location. The System Identification results for equation 11 and 12 are shown in Fig. 5. The particle swarm quickly converged for the two models with the results for the two equations almost equal.

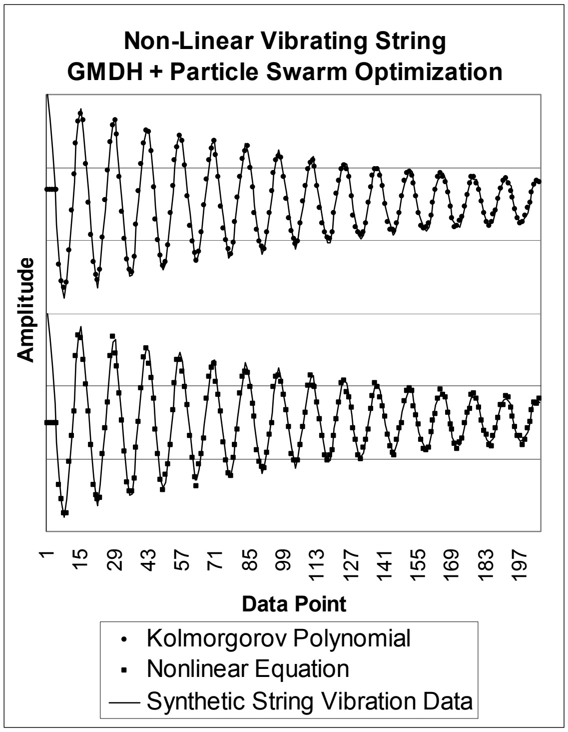

Fig. 5. GMDH - Non-linear vibrating string.

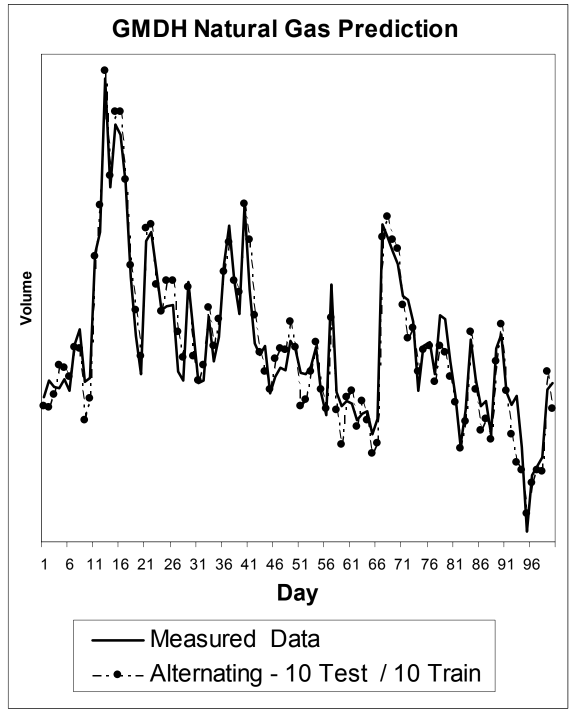

Fig. 6. GMDH - Natural Gas: Alt. Test/Train.

This lends support for GMDH nodal equations of the form given in equation 12 since it requires one less degree of freedom than equation 11. For models where training time is not critical these results support the use of PSO and non-linear functional representations within the GMDH nodes.

5.3 NATURAL GAS PREDICTION - GMDH

In the last example we investigated the applicability of a traditional GMDH for predicting natural gas consumption. In Fig. 6 the GMDH network was trained on all 100 days in the data set. Fig. 7 was trained and tested on alternating 10 day periods.

For both gas consumption studies the temperature, wind and volume for the previous two days were used to predict today's required volume. No solid conclusions can be drawn from this study, but it does illustrate the trade-off that is made with respect to choosing a criterion for selecting the surviving nodes at a given layer in a traditional GMDH network. The GMDH network, shown in Fig. 6., that was trained on 10 days with the next 10 days used for selecting the surviving nodes within a layer does not do as good a job on average but never over or under predicts as much as the GMDH network trained on all the data.

The GMDH network trained on all the data shown in Fig. 7 does a good job on almost all of the days except for day 70 where the prediction is noticeably high.

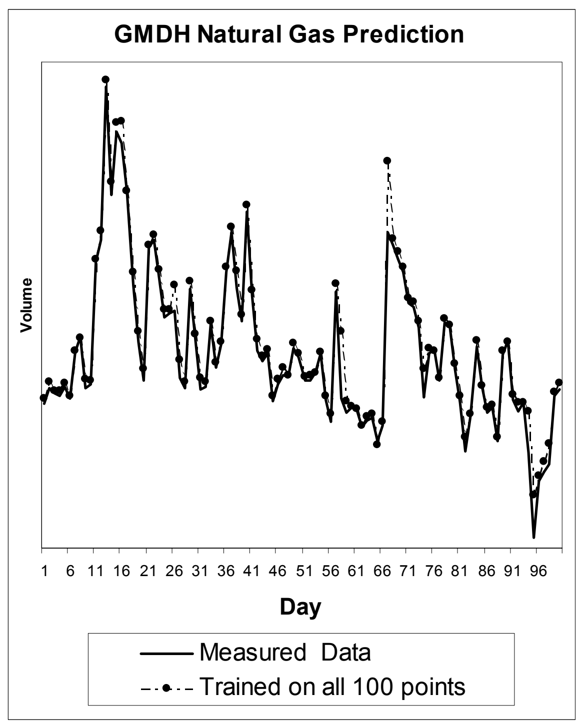

Fig. 7. GMDH - Natural Gas: Trained on all data.

A more in depth study would have to be undertaken to determine the quality of these results as compared to traditional neural networks, but the results are promising when one takes into account that these networks were trained in a few seconds.

For both gas consumption studies the temperature, wind and volume for the previous two days were used to predict today's required volume. No solid conclusions can be drawn from this study, but it does illustrate the trade-off that is made with respect to choosing a criterion for selecting the surviving nodes at a given layer in a traditional GMDH network. The GMDH network, shown in Fig. 6., that was trained on 10 days with the next 10 days used for selecting the surviving nodes within a layer does not do as good a job on average but never over or under predicts as much as the GMDH network trained on all the data. The GMDH network trained on all the data shown in Fig. 7 does a good job on almost all of the days except for day 70 where the prediction is noticeably high. A more in depth study would have to undertaken to determine the quality of these results as compared to traditional neural networks, but the results are promising when one takes into account that these networks were trained in a few seconds.

6 CONCLUSION

Preliminary studies indicate that Particle Swarm Optimization can be used to develop superior estimates for the ARMA model parameters for noisy (real world) data. Since the run time for these studies was only a few minutes (at most) it can be inferred that Particle Swarm Optimization is competitive with traditional non-linear least squares algorithms for determining the parameters for many traditional System Identification tasks. Additionally, Particle Swarm Optimization does not need to exploit any mathematical properties that are specific to a particular system model.

The practical use of Particle Swarm Optimization for training non-linear nodes within a GMDH network was demonstrated. This was illustrated using a non-linear equation that has one less parameter than the traditional polynomial approximation while producing competitive training results. Since the non-linear nodal equation that was demonstrated is only one of many that can be used, this implies that families of non-linear functions could be trained for each node. This would allow for GMDH networks that are self-organizing at multiple levels. The examples studied provide experimental support for the practical use of low-level non-linear emergent nodal representations embedded in a higher-level self-organizing network. This hierarchical network (self-organizing at many levels) forms the basis for the methodology that we are calling Emergent System Identification.

References

Iba, H., Kurita, T., (1993), System Identification using Structured Genetic Algorithms, *Proceedings Fifth International Conference on Genetic Algorithms*, 279-286, Morgan Kaufmann.

Eberhart, R.C., Shi, Y., (1998a). Comparison between Genetic Algorithms and Particle Swarm Optimization, *Evolutionary Programming VII*, Porto, V.W., Saravanan, N. Waagen, D., Eiben, A.E., 1447, 611-618, Springer-Verlag.

Eberhart, R.C., Shi, Y., (1998b). Parameter Selection in Particle Swarm Optimization, *Evolutionary Programming VII*, Porto, V.W., Saravanan, N. Waagen, D., Eiben, A.E., 1447, 591-600, Springer-Verlag.

Holland, J. H., (1998). *Emergence, From Chaos to Order*, Adison-Wesley.

Ivakhnenko A.G., (1968a). The Group Method of Data Handling – A rival of the Method of Stochastic Approximation, *Soviet Automatic Control*, Vol. 13, No. 3.

Ivakhnenko A.G., (1971b). Polynomial theory of complex systems. *IEEE Trans. on Systems, Man, and Cybernetics*, 364--378.

Kennedy, J., Eberhart, R.C., (Contributor), Shi, Y., (2001a). *Swarm Intelligence*, Morgan Kaufmann, San Francisco, CA.

Kennedy, J., and Spears, W.M., (1998b). Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator, *Proceedings of the 1998 International Conference on Evolutionary Computation*.

Løvbjerg, M., Rasmussen, T.K., Krink, T., (1998). Hybrid Particle Swarm Optimizer with Breeding and Subpopulations, *Proceedings Genetic and Evolutionary Computation Conference*, GECCO-2001, Morgan Kaufmann, San Francisco, CA.

Madala, H.R., Ivakhnenko, A.G., (1994). *Inductive Learning Algorithm for Complex Systems Modeling*, CRC Press.

Nikolaev,N. and Iba,H., (2001) Accelerated Genetic Programming of Polynomials, *Genetic Programmimg and Evolvable Machines*, vol.2, N: 3.

Pandit, S.M., Wu, S., (1983) *Time Series and System Analysis with Applications*, John Wiley & Sons.

Voss, M.S., Feng, X., (2001). Emergent System Identification using Particle Swarm Optimization, *Proceedings Complex Adaptive Structures Conference*, SPIE, Bellingham, WA.

Voss, M.S., Feng, X., (2002). ARMA Model Selection using Particle Swarm Optimization and AIC Criteria, *Accepted for Procedings 15th International Federation of Automatic Control World Congress*, IFAC, Barcelona, Spain.