An Experimental Application of Learnable Evolution Model and Genetic Algorithms to Parameter Estimation in Digital Signal Filters Design

M. Coletti, T. Lash, C. Mandsager,
R. S. Michalski, and R. Moustafa

P99- 9
MLI 99-5

May 1999

An Experimental Application of Learnable Evolution Model and Genetic Algorithms to Parameter Estimation in Digital Signal Filter Design

**Mark Coletti,Tom Lash, Craig Mandsager,
Ryszard S. Michalski*, and Rida Moustafa**

**Machine Learning and Inference Laboratory
George Mason University**

***Also with Institute of Computer Science
Polish Academy of Sciences**

ABSTRACT

This paper describes an application of LEM1, a preliminary implementation of Learnable Evolution Model (LEM), and two canonical genetic algorithms, GA1 and GA2, to parameter estimation in digital signal filter design. LEM1 alternates between two modes of operation: Machine Learning mode, which employs AQ-18 rule learning system, and Darwinian Evolution mode, which employs genetic algorithm GA2. Machine Learning mode generates hypotheses as to what type of individuals in a population represent high fitness solutions. These hypotheses, expressed in the form of attributional rules, are used to generate new populations of solutions. When the top fitness of a population has not improved sufficiently during one mode, LEM1 switches to another mode. LEM1 alternates between the two modes until a global termination condition is satisfied. In the experiments, LEM-1 significantly outperformed genetic algorithms GA1 and GA2.

Keywords: Evolutionary computation, Learnable Evolution Model, Genetic Algorithms, Function Optimization, Symbolic Learning, AQ18, Digital Filters.

1 INTRODUCTION

Parameter estimation problems solved by steepest descent methods require a computation of the gradient of the error surface. Computation of the gradient of a multi-modal error surface can, however, be extremely computationally intensive for problems with many variables. For some spaces derivatives are not defined and gradients cannot be computed at all. To solve such complex parameter estimation problems genetic algorithms can be used (Yao, Sethares, 1994).

Unlike the steepest-descent approach, genetic algorithms do not require to compute the gradient of the error surface. In addition, due to their inherent stochastic nature, genetic algorithms are less susceptible to settling on a local minimum.

A major difficulty with genetic algorithms is that they are computationally inefficient because they conduct a stochastic trial and error process. A new approach is offered by the Learnable Evolution Model (LEM), which was recently introduced by Michalski (1998, 1999). LEM combines machine learning with evolutionary computation in a novel way, and appears to produce a significant speed-up of evolutionary processes over traditional evolutionary computation algorithms.

In Machine Learning mode, LEM employs a learning system that generates descriptions that differentiate high performing and low performing individuals in the population. Machine Learning mode can use different machine learning methods. The first system implementing the LEM approach, LEM1, applied the AQ15c symbolic learning method (Michalski, 1973; Michalski et al., 86; Wnek et al, 1996). LEM1 and two evolutionary computation algorithms, GA1 and GA2, were applied to the De Jong's problem set (De Jong, 1975) and their inverse (Michalski, 1998, Michalski and Zhang, 1999; Michalski, 1999). In these experiments, LEM1 significantly outperformed GA1 and GA2.

This paper describes an improved version of LEM1 (which uses AQ18 rather than AQ15c learning program; see Michalski and Kaufman, 1999), and applies it to a different class problems, specifically, to parameter estimation in digital filter design. In the experiments, LEM1 strongly outperformed the genetic algorithms GA1 and GA2.

2 BRIEF DESCRIPTION OF LEM1, GA1 AND GA2

The LEM method works by alternating between Darwinian Learning mode and Machine Learning mode (the duoLEM version), or works solely in Machine Learning mode (the uniLEM version). In this study we experimented with duoLEM version. In this version, a switch from one mode to another is made when the fitness of the population is not improving sufficiently for a certain number of generations.

The LEM method is defined as follows (Michalski, 1998, 1999):

1. Randomly, or according to certain prior rules reflecting domain knowledge, generate the starting population of solutions (in the case of parameter estimation problems, this population is a starting set of parameter values).
2. Execute *Darwinian Evolution mode* (using some form of selection, crossover and mutation operators), as long as the best solution in a sequence of *dar-probe* iterations is

better by the *dar-threshold* than the best solution found in previous generations.

3. Execute *Machine Learning mode* (using a machine learning method)
   (a) Determine HIGH (high-performance) and LOW (low-performance) solutions in the current population, according to the fitness function defined for a given task or problem.
   (b) Apply machine learning method for characterizing differences between HIGH and LOW solutions.
   (c) Generate a new population of solutions by replacing not-HIGH individuals by those satisfying the learned description of HIGH solutions; the selection of new solutions among those satisfying the description is random or according to the predefined selection rules (a simple variant of the LEM method).
   (d) Go to step (a), and then continue Machine Learning mode as long as the best solution in a sequence of *learn-length* iterations is better by the *learn-threshold* than the previously found best solution.
4. Switch to (2), and repeat the process. Continue switching between mode (2) and (3) until the *termination condition* is met (the generated solution satisfactory, or the allocated computation resources are exhausted).

In the LEM1 system, the step 3(b) is executed by supplying HIGH and LOW fitness individuals to an AQ-type learning system. In our experiments, we used AQ18 system (Kaufman and Michalski, 1999), which hypothesized attributional descriptions of high performing individuals. These descriptions were used to generate a new population of individuals for the next evolutionary computation step. GA1 and GA2, obtained from Ken De Jong, employ mutation and standard selection operators to alter populations. GA2 also has a uniform crossover operator, so a child has an equal probability of receiving a gene value from each parent. LEM1 uses GA2 during its genetic algorithm mode, employing mutation, standard selection, and crossover. These algorithms do not use elitism.

3 PROBLEM STATEMENT

We applied the LEM1 system to problems of identifying parameters in digital filters. Fitness functions were defined on the basis of equations specifying linear and non-linear filters. These equations are described in the paper "Nonlinear Parameter Estimation via the Genetic Algorithm" (Yao and Sethares, 1994):

Linear filter:

$$y(k) = -0.3y(k - 1) + 0.4y(k - 2) + 1.25u(k - 1) - 2.5u(k - 2) + n(k)$$

Non-linear fitler:

$$y(k) = \left[\frac{3 - 0.3y(k-1)u(k-2)}{5 + 0.4y(k-2)u^2(k-1)}\right]^2 + (1.25u^2(k-1) - 2.5u^2(k)))\ln(\left|1.25u^2(k-2) - 2.5u^2(k)\right|) + n(k)$$

where:
k – is the sample index (or time)
$n()$ – is a noise component ranging from -.25 to .25
$u()$ – represents an inserted function (sin, step, etc.)

In the experiments, we assumed that the coefficients of these equations are unknown, that is, that they are variables. Each individual in a population is thus a vector of four variable values. The problem to be solved was to determine coefficients of these equations, given a list of *y(k)* values for a number of *k* values.

To apply the evolutionary computation approach, coefficients -0.3, 0.4, 1.25, and -2.5 in both equations were replaced by four real-valued variables. Values of these variables constitute "genes," and vectors of these values comprises "chromosomes," or individuals of a population. When substituted in the equation, the individual's chromosome yields a result that is compared with the correct value computed from the equation. The fitness of an individual is inversely proportional to the difference between the obtained result and correct value. Thus, the individual whose chromosome gives the lowest error has the highest fitness.

The parameters of the genetic algorithms used were as follows, for both the linear and non-linear equations:

Gene representation	Real
Number of genes per individual	4
Gene landscape (constraint on range)	-30 to 30
Number of individuals per generation	60
Mutation Rate	25%
Maximum number of births	100,000
Maximum number of generations	1500

4 EXPERIMENTS

We ran LEM1, GA1, and GA2 for the linear and nonlinear filters with three different input data. Yao and Sethares used a uniformly distributed random input over the interval (-2.5, 2.5). In addition to this input, we used a unit step function 2.5u(k) and a sine wave 2.5sin(□/10) for comparison. The landscape function generated an output array based on a 200 sample input sequence and stored it for comparison against the populations.

Populations were generated, and the fitness of each individual was calculated by computing the mean-squared error between the known values and the output generated by the individual's genes. Yao and Sethares defined the fitness of the individual as the reciprocal of the mean-square error over the 200 sample window:

$$Fitness = \frac{1}{MeanSquareError} = \frac{200\,samples}{\sum_{200\,samplewindw}(individual - known)^2}$$

We ran LEM1, GA1, GA2 ten times for the linear and nonlinear filters using the sine, step, and uniform random inputs. Since the initial populations were generated randomly, the convergence rate varied greatly between populations and generations. We averaged the ten runs of each type together to try to get an aggregate performance comparison, but we found that the performances varied greatly due to the random initial conditions of the system.

It was difficult to attain an average performance because a few runs would dominate the average. So instead of an average performance for each method, the following figures show the learning curves which converged the fastest from the three systems with different input functions.

Figures 1 to 9 show the results. Figures 1-3 show results from GA1, GA2 and LEM1 for nonlinear filter with a sine wave input, Figures 4-6 show results from GA1, GA2 and LEM1 for nonlinear filter with unit step input, Figures 7-9 show results from GA1, GA2 and LEM1 for nonlinear filter with uniform noise input.

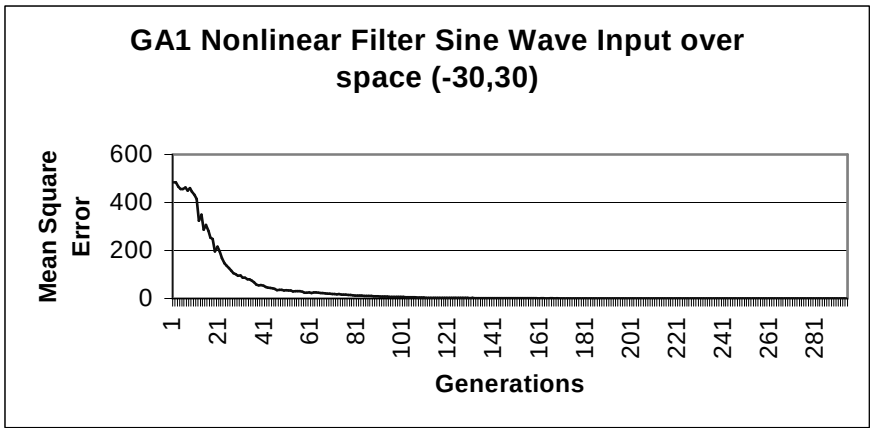

Figure 1: GA1 learning curve, nonlinear filter, sine wave input.

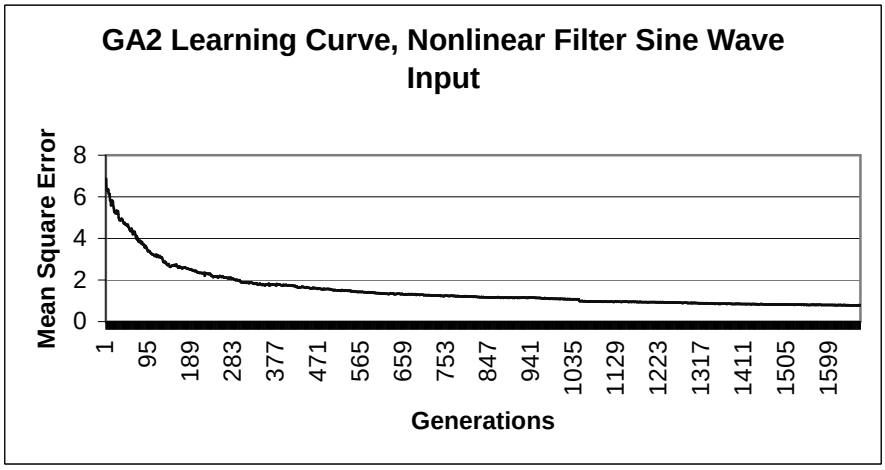

Figure 2: GA2 learning curve, nonlinear filter, sine wave input.

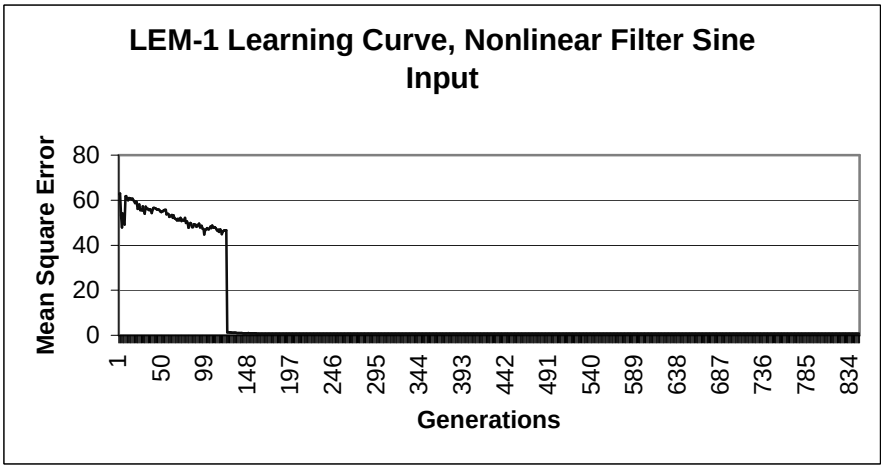

Figure 3: LEM-1 learning curve, nonlinear filter, sine input.

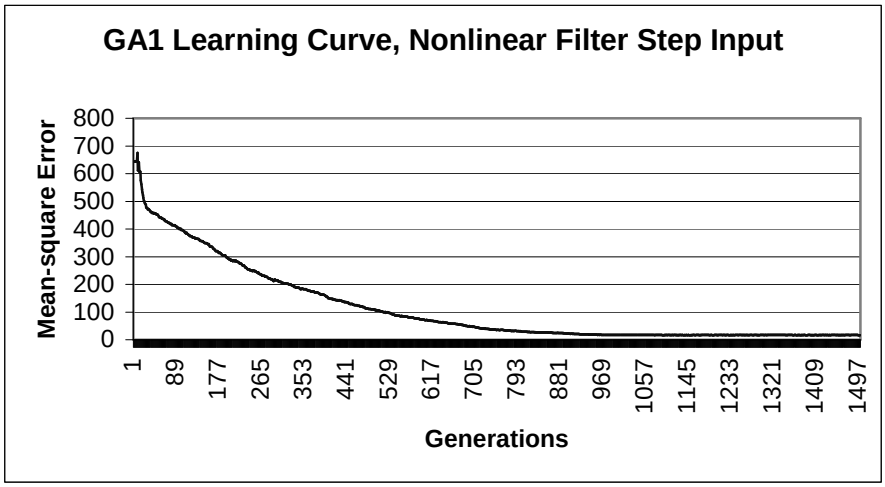

Figure 4: GA1 learning curve, nonlinear filter, unit step input.

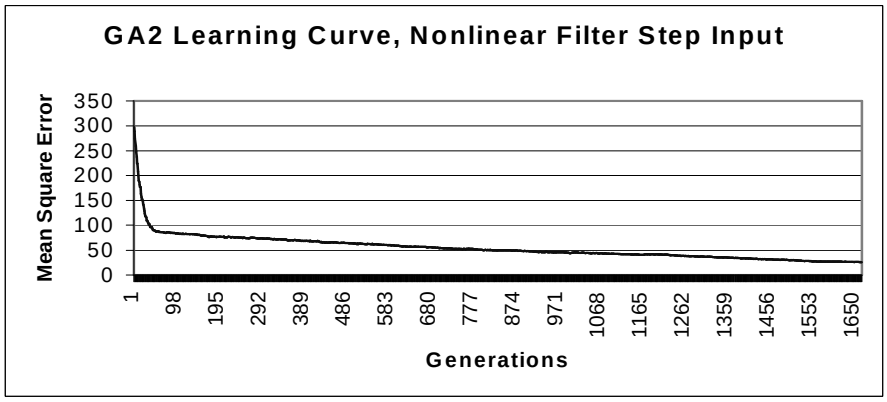

Figure 5: GA2 learning curve, nonlinear filter, unit step input.

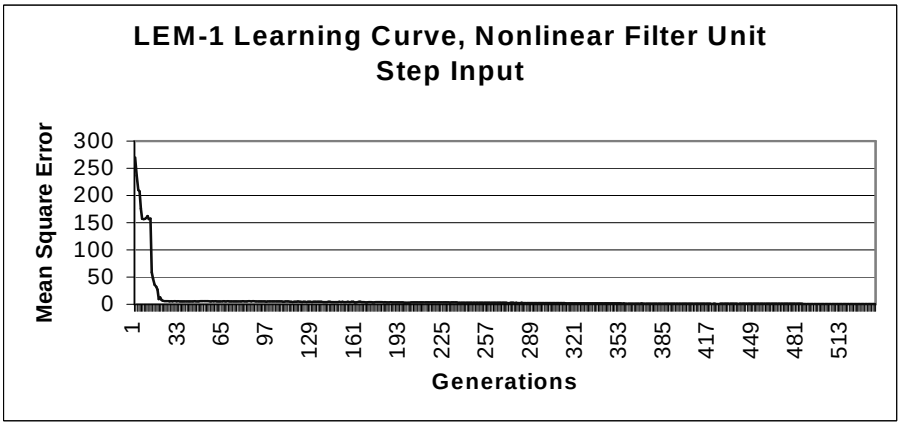

Figure 6: LEM-1 learning curve, nonlinear filter, unit step input

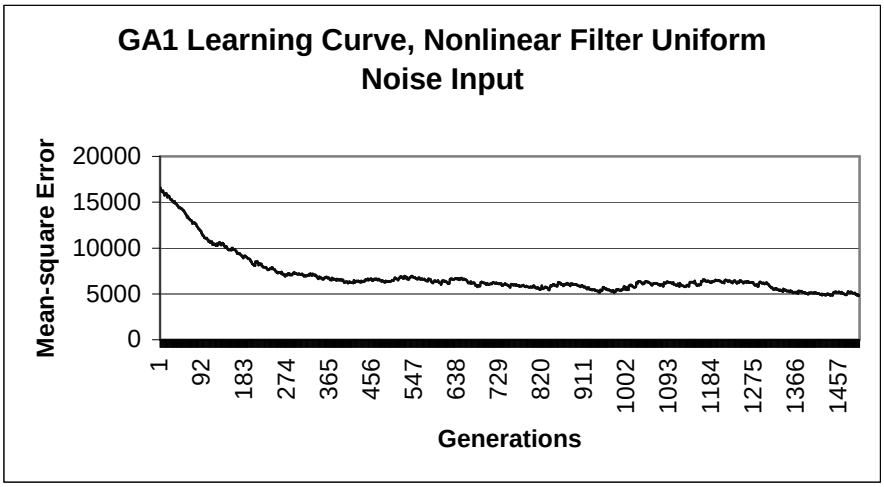

Figure 7: GA1 learning curve, nonlinear filter, uniform noise input.

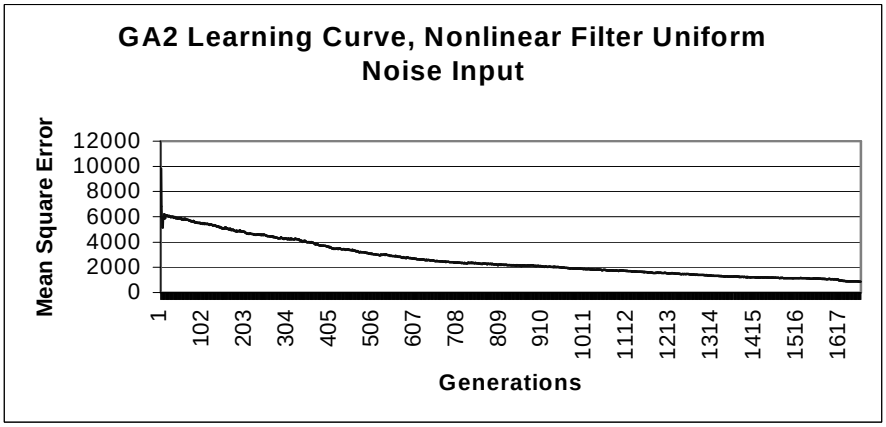

Figure 8: GA2 learning curve, nonlinear filter, uniform random noise input.

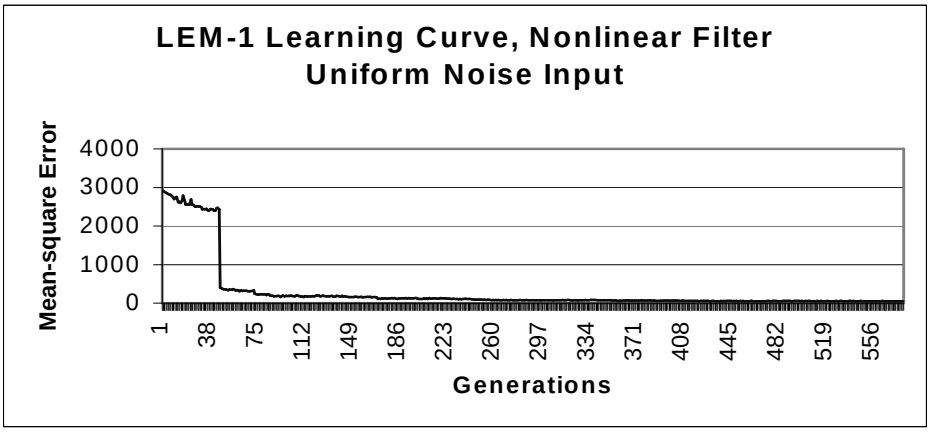

Figure 9: LEM-1 learning curve, nonlinear filter, uniform random noise input.

In the figures, the horizontal axis represents number of generations and the vertical axis represents the mean square error. As shown in the figures, in these experiments, LEM1 strongly outperformed GA1 and GA2 in terms of the speed of convergence to the zero error. The effect of Machine Learning mode is demonstrated by a dramatic drop in the mean-square error when the system learned useful rules for generating the next generation of individuals. LEM1 toggled into Machine Learning mode roughly 10-800 times through out course of a typical experiment. A dramatic drop in mean-square error usually occurs within the first 100 generations.

Since we used four variables to represent the four weights of the filter, the error surface generated by the mean-square error is four-dimensional. Such an error surface creates difficulties for traditional search techniques. Such techniques, for example, the gradient descent and LMS, are prone to finding local minima. To achieve the robustness of the evolutionary computation approach, they would have to explore solutions in parallel. LEM1 improves over genetic algorithms by accelerating the evolutionary process through a series of machine learning steps.

5 SUMMARY

This experimental study demonstrated that LEM1, a simple implementation of the LEM methodology, can produce significant speedups in the evolutionary process in comparison to conventional genetic algorithms. This speedup comes from Machine Learning mode that hypothesize rules characterizing differences between high and low performing individuals in consecutive populations. New populations are generated based on these rules. Thus, LEM generates new solutions using discovered patterns rather than stochastic operators employed in genetic algorithms. As described in (Michalski, 1999) these patterns can be viewed as qualitative differentials of the fitness landscape.

In our experiments, GA1, GA2, and LEM1 were applied to parameter estimation for non-linear filters for unit step, sine wave, and noisy input. In each case, LEM1 strongly outperformed GA1 and GA2 in terms of the number of generations needed to reach the optimal solution.

In conclusion, the LEM1 system has performed surprisingly well on the problem of parameter estimation for digital filters. It appears that the LEM approach is suitable for tackling complex search problems. Another practical problem areas where we expect the LEM approach to work well are function optimization problems and determining complex planning strategies.

ACKNOWLEDGEMENTS

Authors thank Dr. Ken Kaufman and Qi Zhang for providing the original LEM-1 source code and assisting us in using the system. Thanks go also to Dr. Kenneth De Jong for the source code of GA1 and GA2. We are also grateful to Guido Cervone for his technical assistance. LEM1 and AQ18 were developed at the GMU Machine Learning and Inference Laboratory.

This research was partially supported by a National Science Foundation under Grant No. IIS-9904078. The development of AQ18 has been supported in part by the National Science Foundation under grants IRI-9510644 and DMI-9496192, in part by the Office of Naval Research under grant N00014-91-J-1351, and in part by the Advanced Research Projects Agency under grant No. N00014-91-J-1854, administered by the Office of Naval Research.

REFERENCES

De Jong, K.A., An Analysis of the Behavior of a Class of Genetic Adaptive Systems, *Ph.D. Thesis*, Department of Computer and Communication Sciences, University of Michigan, An Arbor, 1975.

Kaufman, K. and Michalski, R. S., The AQ18 System for Natural Induction: A User's Guide, *Reports of Machine Learning and Inference Laboratory*, George Mason University, P98-11, October, 1998.

Michalski, R. S., "AQVAL/1 -- Computer Implementation of a Variable-Value3d Logic System VL and Examples of its Application to Pattern Recognition," *Proceedings of the First International Joint Conference on Pattern Recognition,* Washington DC, pp. 3-17, October 30 - November 1, 1973.

Michalski, R.S., "Learnable Evolution: Combining Symbolic and Evolutionary Learning," *Proceedings of the Fourth International Workshop on Multistrategy Learning* (MSL '98), Desenzano del Garda, Italy, pp. 14-20, June 11-13, 1998.

Michalski, R.S., "LEARNABLE EVOLUTION: Evolutionary Processes Guided by Machine Learning," *Reports of Machine Learning and Inference Laboratory*, MLI 99-3, 1999; to appear in *Machine Learning Journal*, 1999.

Michalski, R.S., Mozetic, I., Hong, J. Lavrac, N., "The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains," *Proceedings of the AAAI*, Philadelphia, August 11-15, 1986.

Michalski, R.S. and Zhang, Q., Initial Experiments with Learnable Evolution Model: An Application of LEM1 to Function Optimization and Evolvable Hardware, *Machine Learning and Inference Reports*, MLI 99-4, George Mason University, 1999.

Wnek, J., Kaufman, K., Bloedorn, E. and Michalski, R.S., "Inductive Learning System AQ15c: The Method and User's Guide," *Reports of the Machine Learning and Inference Laboratory*, MLI 95-4, George Mason University, Fairfax, VA, March 1995.

Yao, Leehter and Sethares, William, "Nonlinear Parameter Estimation via the Genetic Algorithm," *IEEE Transactions on Signal Processing*, Vol. 42 No.4 p.927-935. 1994.