

# GENETIC PROGRAMMING BASED MODEL STRUCTURE IDENTIFICATION USING ON-LINE SYSTEM DATA

Stephan Winkler\*\*\*, Michael Affenzeller\*, Stefan Wagner\*

\* Institute for Formal Models and Verification

\*\* Institute for Design and Control of Mechatronical Systems

Johannes Kepler University

Altenbergerstrasse 69

A-4040 Linz – Austria

e-Mail: {stephan,michael,stefan}@heuristiclab.com

## KEYWORDS

Genetic Programming, Data Driven Model Identification, Self-Adaption, Machine Learning, On-Line Modeling

## ABSTRACT

Genetic Programming, an heuristic optimization technique based on the theory of Genetic Algorithms, is a method successfully used to identify nonlinear model structures by analyzing a system's measured signals. Mostly, it is used as an offline tool which means that structural analysis is done after collecting all available identification data. In this paper, we propose an enhanced on-line GP approach that is able to adapt its behaviour to new observations while the GP process is executed. As an example we document how a model for a BMW diesel engine's NO<sub>x</sub> exhausts was identified using on-line measurement data provided by a real-time data simulator environment.

## 1 INTRODUCTION

### 1.1 Problem Statement

The problem of finding a model for a system's measured signal, i.e. discovering the mathematical relationship between empirically observed variables measuring a system, is an important problem in technical fields such as mechatronics, but also in economics and other areas of science (Langley et al. 1987). In practice, the observed data may be noisy and there may be no known way to express the relationships involved in a precise way. Problems of this type are of scientific interest in the context of systems control and data mining; usually they are called symbolic system identification problems, black box problems or modeling problems. When the model that is discovered is used in

predicting future values of the state variables of the system, the problem is called a forecasting problem (Koza 1995).

In this paper we present an on-line structure identification method based on a Genetic Programming identification approach. This identification algorithm is able to adapt its behaviour during its execution to new data, i.e. to a changing of the algorithm's environment. By doing so, this method combines the advantages of enhanced, hybrid variants of Genetic Algorithms and heuristic optimization techniques with real-time knowledge discovery and data mining.

Evolutionary programming techniques (especially GAs and GP) have often been and are still often considered not suitable for on-line identification: "For an off-line process, a Genetic Programming method could be utilized to 'evolve' the function that best represents the system dynamics. This is an attractive approach because the actual structure of the dynamic equations would be revealed (and the parameters optimized in the process). Unfortunately, evolutionary programming techniques are ill-suited for on-line learning." (Ellis 1998)

As we demonstrate in Section 4, this widespread opinion has to be reconsidered since the proposed GP-based method is indeed suitable for evolving suitable models (at least for mechatronical systems) on-line. The approach presented here, in fact, is not restricted to any specific problem situation but can be used for any kind of data driven on-line identification process since a wide range of mathematical expressions can be represented and the framework used is very flexible and not tuned to any specific application.

### 1.2 Selected Evolutionary Computing Techniques: Genetic Algorithms and Genetic Programming

Evolutionary computing is the collective name for heuristic problem-solving techniques based on the

principles of biological evolution, which are natural selection and genetic inheritance. One of the greatest advantages of these techniques is that they can be applied to a variety of problems, ranging from leading-edge scientific research to practical applications in industry and commerce; by now, evolutionary algorithms are in use in various disciplines like optimization, artificial intelligence, machine learning, simulation of economic processes, computer games or even sociology.

The forms of evolutionary computation relevant for the work described in this paper are Genetic Algorithms (GA) and Genetic Programming (GP). The fundamental principles of GAs were first presented by Holland (Holland 1975), overviews about GAs and their implementation in various fields were given for instance by Goldberg (Goldberg 1989), Michalewicz (Michalewicz 1996) and Affenzeller (Affenzeller 2003).

A GA works with a set of solution candidates (also known as individuals) called population. During the execution of the algorithm each individual has to be evaluated, which means that a value indicating the quality is returned by a fitness function. New individuals are created on the one hand by combining the genetic make-up of two solution candidates (this procedure is called “crossover”), producing a new “child” out of two “parents”, and on the other hand by mutating some individuals, which means that randomly chosen parts of genetic information are changed (normally a minor ratio of the algorithm's population is mutated in each generation).

Beside crossover and mutation, the third decisive aspect of Genetic Algorithms is selection. In analogy to biology this is a mechanism also called “survival of the fittest”. Each individual is associated with a fitness value, and an individual’s probability to propagate its genetic information to the next generation is proportional to its fitness: The better a solution candidate’s fitness value, the higher the probability, that its genetic information will be included in the next generation's population. This procedure of crossover, mutation and selection is repeated over many generations until some termination criterion is fulfilled.

The basic idea of Genetic Programming, which was first explored in depth by Koza in 1992 (Koza 1992), is that virtually all problems in artificial intelligence, machine learning, adaptive systems, and automated learning can be recast as a search for a computer program, and that GP provides a way to successfully conduct the search for a computer program in the space of computer programs.

Similar to GAs, GP works by imitating aspects of natural evolution to generate a solution that maximizes (or minimizes) some fitness function: A population of solution candidates evolves through many generations towards a solution using certain evolutionary operators and a “survival-of-the-fittest” selection scheme. The main difference is that, whereas GAs are intended to find

an array of characters or integers representing the solution of a given problem, the goal of a GP process is to produce a computer program (or a formula) solving the optimization problem at hand.

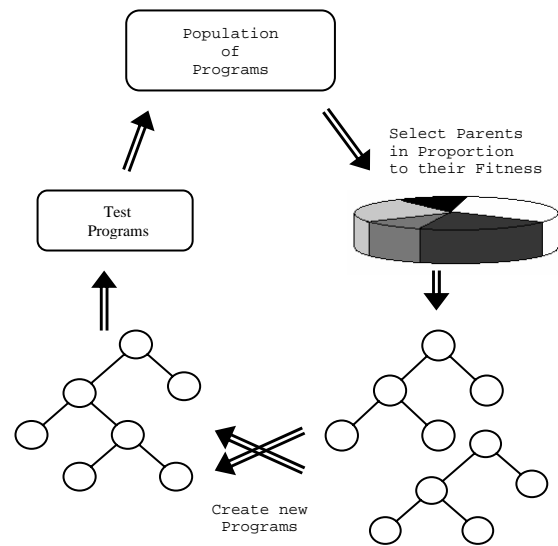


Figure 1: The Genetic Programming Cycle (taken from (Langley 1987))

Typically, the population of a GP algorithm contains a few hundred individuals and evolves through the action of operators known as crossover, mutation and selection. The population of a GP algorithm evolves through the action of operators known as crossover, mutation and selection. Figure 1 visualizes how the GP cycle works: As in every evolutionary process, new individuals are created. They are tested, and the fitter ones in the population succeed in creating children of their own. Unfit ones ‘die’ and are removed from the population (Langdon 2002).

## 2. GP BASED STRUCTURE IDENTIFICATION

Preliminary work for the approach presented in this paper was done for the project “Specification, Design and Implementation of a Genetic Programming Approach for Identifying Nonlinear Models of Mechatronic Systems” in the context of a bigger strategical project at the Johannes Kepler University Linz, Austria. The goal of this project was to find models for mechatronic systems. It was successfully shown (for instance in (Winkler, Affenzeller and Wagner 2004a), (Winkler, Affenzeller and Wagner 2004b) and in further detail in (Winkler 2004)) that methods of GP are suitable for determining an appropriate mathematical representation of a physical system. Furthermore, in (Winkler, Affenzeller and Wagner 2005) we have documented that this approach can also be used for solving classification problems.

We have used the methods implemented for this project for developing a GP-based real time structure identification algorithm. This algorithm operates on a set of training data with measured signals ( $X_1, \dots, X_N$ ). One of these signals ( $X_t$ ) has to represent the system's signal for which a model has to be found for. On the basis of the training data, the algorithm tries to evolve (or, as one could also say, to "learn") a solution, i.e. a formula, that represents the function which models the chosen target channel. In other words, each presented instance of the structure identification problem is interpreted as an instance of an optimization problem; a solution is found by a heuristic optimization algorithm. The goal of this GP process is to produce an algebraic expression

$$\tilde{X}_t = f(X_1, \dots, X_N)$$

approximating  $X_t$  as well as possible (only on the basis of a database containing the measured results of the experiments to be analyzed). Thus, the GP algorithm works with solution candidates that are tree structure representations of symbolic expressions. When the evolutionary algorithm is executed, each individual of the population represents one structure tree.

Details of the basic structure of these formula trees, their implementation and several considerations regarding the function library (since the selection of the library functions is an important part of any GP modeling process because this library should be able to represent a wide range of systems) can be found in (Winkler 2004) and (Winkler, Affenzeller and Wagner 2005).

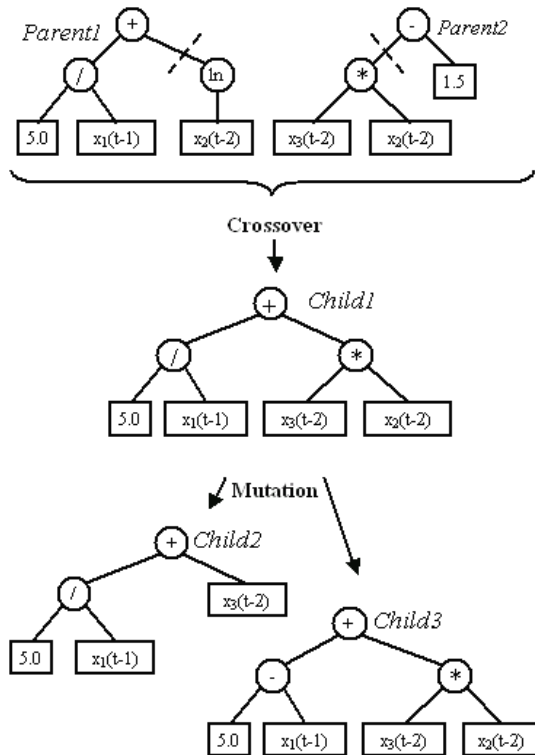


Figure 2: Genetic operations on tree structures.

In general, both crossover and mutation processes are applied to randomly chosen branches (in this context a branch is the part of a structure lying below a given point in the tree). Crossing two trees means randomly choosing a branch in each parent tree and replacing the branch of the tree, that will serve as the root of the new child (randomly chosen, too), by the branch of the other tree. Mutation in the context of Genetic Algorithms means modifying a solution candidate randomly and so creating a new individual. In the case of identifying structures, mutation works by choosing a node and changing it: A function symbol could become another function symbol or be deleted, the value of a constant node could be manipulated or the index or the time-offset of a variable could be modified. This procedure is less likely to improve a specific structure but it can help the optimization algorithm to re-introduce genetic diversity in order to re-stimulate genetic search. Examples of genetic operations on tree structures are shown in Figure 2: The crossover of parent1 and parent2 yields child1, child2 and child3 are possible mutants of child1.

For evaluating solution candidates, the use of various functions is possible. For this project we have decided to use the average squared error function since it is independent of the number of considered samples:

$$E = \frac{1}{N} \sum_{i=1}^N (\tilde{X}_t(i) - X_t(i))^2$$

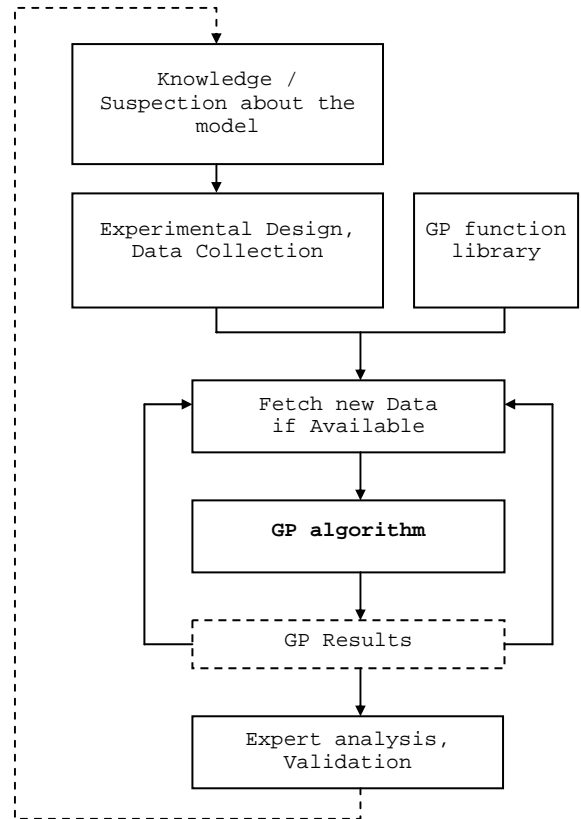


Figure 3: Workflow of the Extended On-Line GP Process.

### 3 ON-LINE GP MODELING

Thanks to the fact that the GP process is executed periodically, the insertion of an additional stage can be designed and implemented quite easily. As is graphically shown in Figure 3, we have added an additional phase to the standard GP cycle: Before the next generation of solution candidates is produced, possibly available new data are collected from a predefined data source (e.g., a file as in the case of our prototypical implementation).

The approach presented here has actually got several advantages:

- One of the major advantages of this approach is that the benefits of Evolutionary Computation (namely the combination of directed and undirected heuristics as well as the use of a certain amount of randomness) are combined with concepts of on-line knowledge discovery and data mining. As described in further detail in the following section, this modeling method can be used as an alternative to existing on-line modeling and identification methods that are for example used in industrial fault detection and identification programs.
- With respect to the measured data, the algorithm is able to adapt its behavior as new identification data are available: Since all individuals of a GP algorithm's population have to be evaluated every generation, the corresponding data set can be modified after every generation step. This of course means a change of the algorithm's environment and is likely to influence the GP process in several (maybe unforeseen) ways.

But since structural identification anyway assumes an underlying concept of the investigated system, this changing of environment is expected to have rather positive than negative effects.

- Last, but surely not least we strongly take advantage of the fact that instead of using standard implementations of the Genetic Algorithm as underlying GP algorithm, a new generic evolutionary algorithm, the SASEGASA, is applied. As presented and explained in further detail in (Affenzeller and Wagner 2004), this hybrid GA uses an enhanced selection model which is designed to directly control genetic drift within the population by advantageous self-adaptive selection pressure steering.

Additionally, this new selection model enables to detect and combat premature convergence which is generally quite a critical issue in GAs. As elaborate test series have shown (Affenzeller and Wagner 2004; Winkler 2004; Winkler, Affenzeller and Wagner 2005), the results obtained for various different optimization problems using the SASEGASA were significantly better than those produced by standard GA implementations.

Figure 4 gives an overview of the taxonomy of optimization techniques (taken from (Affenzeller and Wagner 2004)). As the reader can see, GP belongs to the class of Evolutionary Algorithms as well as GAs (and also Evolutionary Techniques which are not essentially relevant to the work presented here).

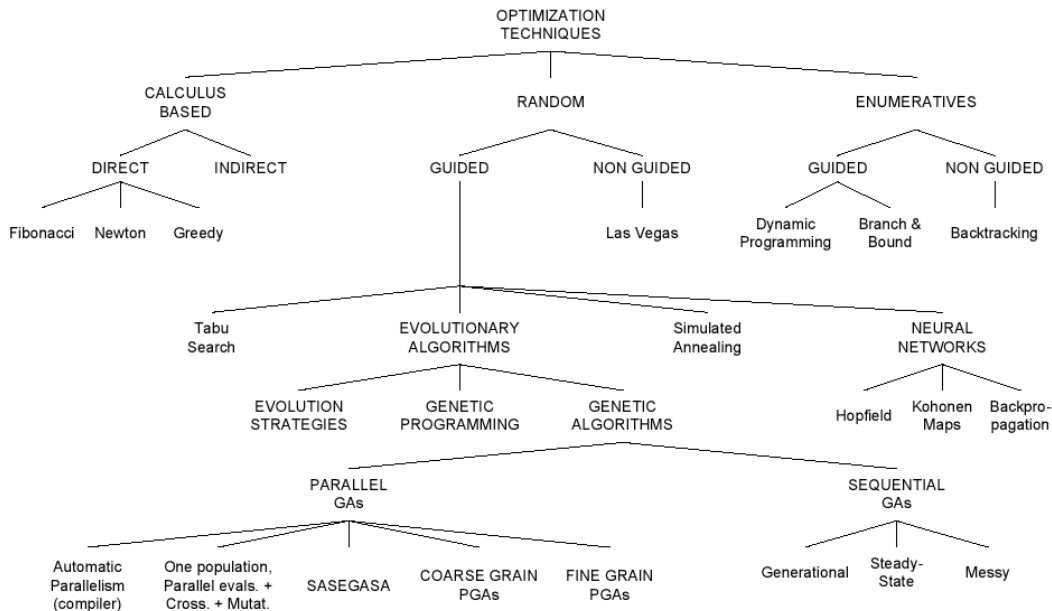


Figure 4: Taxonomy of optimization techniques, taken from (Affenzeller and Wagner 2004).



2004b; Winkler 2004; Del Re et al. 2005; and Winkler, Affenzeller and Wagner 2005)\*.

During all these test series (and also for testing the proposed on-line learning method), the HeuristicLab (Wagner and Affenzeller 2005a), a generic and extensible optimization framework developed by members of the Institute of Systems Theory at the University of Linz, Austria, was used as underlying basic framework.

For testing the presented on-line learning GP algorithm we have analyzed the data representing several signals of a BMW M47D diesel engine (with activated exhaust recirculation). The goal was to identify a model for the engine's NO<sub>x</sub> emissions using the measured values of several other engine parameters (such as temperatures, pressures or the position of the throttle control). Additionally, information about other emissions (mainly CO and CO<sub>2</sub>) and the throttle control should not be incorporated in the model because of redundancies and relatively high costs of exhaust sensors.

A whole FTP 75 cycle was performed within approximately 1,400 seconds; all sensor signals (in total 33) were recorded with 20 Hz resolution, for the GP identification algorithm the data was downsampled to 5 Hz resolution.

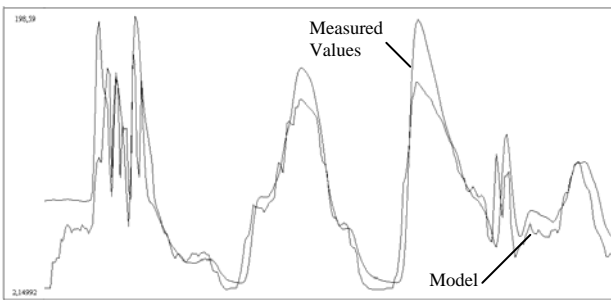


Figure 6: Best result after 30 seconds.  
(Gray line: measured NO<sub>x</sub> exhaust, black line: values calculated by the actual best model.)

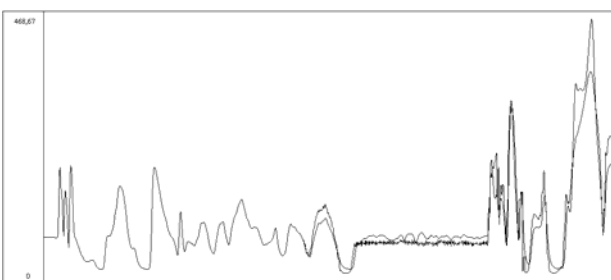


Figure 7: Best result after 3 minutes.  
(Gray line: measured NO<sub>x</sub> exhaust, black line: values calculated by the actual best model.)

For simulating an on-line learning scenario, initially only 50 samples are inserted into the algorithm's data pool adding one more every 0.2 seconds. Since the data basis available to the identification algorithm grows constantly during the simulation causing runtime problems, the identification data was restricted to the most recent 500 samples (representing 100 seconds). As underlying GP algorithm the SASEGASA was applied working with a population size of 300 individuals, 5% mutation rate and a combination of Random Selection and Roulette Selection as selection operator. The average of squared errors was chosen as fitness function.

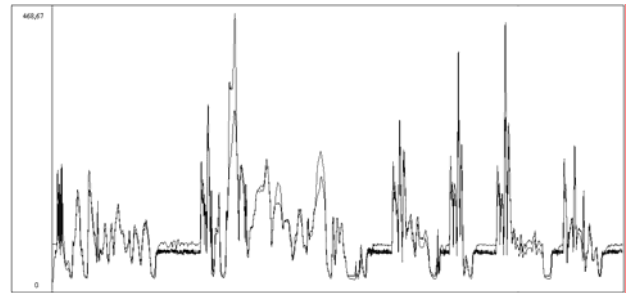


Figure 8: Best Result after 10 Minutes.  
(Gray line: measured NO<sub>x</sub> exhaust, black line: values calculated by the actual best model.)

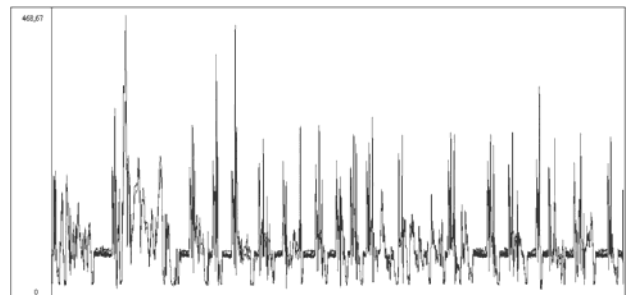


Figure 9: Best result after end of whole FTP cycle.  
(Gray line: measured NO<sub>x</sub> exhaust, black line: values calculated by the actual best model.)

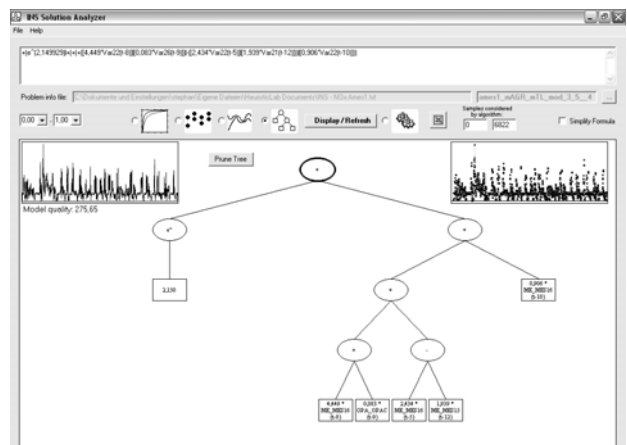


Figure 10: Identified model after end of FTP cycle.

\* All mentioned publications as well as screenshots illustrating recent structure identification results are prepared for download from [www.heuristiclab.com](http://www.heuristiclab.com).

The Figures 6, 7, 8 and 9 illustrate the algorithm's behavior; they graphically show evaluations of the currently best models after 30 seconds (Fig. 6), some minutes (Fig. 7), after 10 minutes (Fig. 8) and finally at the end of the whole simulation (Fig. 9). The model that was returned by the program in the end (after finishing the whole simulation, i.e. after approximately 23 minutes), is shown in Figure 10; it was checked and rated as a very good one by experts in the field of automotive control, namely members of the Institute of Design and Control of Mechatronical Systems at the University of Linz, Austria.

In fact, the presented simulation based GP identification method is able to produce not only better results than the standard GP identification approach; the time needed to produce them is even much lower. Figure 11 shows a graphical representation of the result achieved using standard GP\* for the same NO<sub>x</sub> data set after more than 19 hours: It is far not as good as the result produced by the simulation based procedure after not even half an hour.

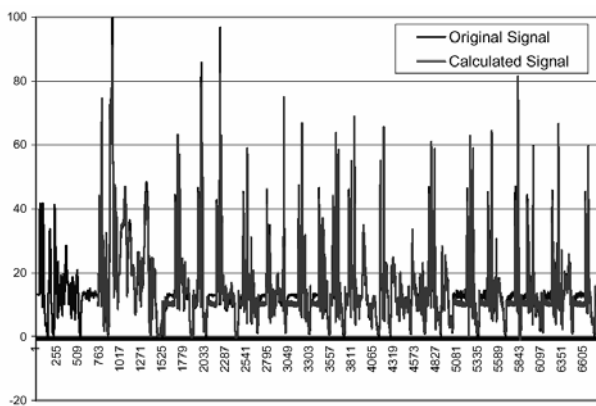


Figure 11: Best result for the same BMW NO<sub>x</sub> data set, produced by GP identification using the SASEGASA on the basis of off-line data after 19 hours.

## CONCLUSION

On the basis of evolution inspired heuristic optimization techniques, an enhanced on-line learning and model structure identification approach based on Genetic Programming has been presented. We have also documented how it was successfully applied to a NO<sub>x</sub> identification problem producing surprisingly good results. Since the results for several problems are very good, even more challenging ones (such as the identification of soot, e.g.) have to be attacked. Furthermore, there are several possibilities how the

\* The SASEGASA was used as underlying genetic algorithm; the population consisted of 4000 individuals, 5% mutation rate and a combination of Random and Roulette selection operators were applied.

results presented in this paper could help develop even more enhanced variants of GP. For example, we are planning to implement a GP method using sliding window effects simulating environmental changes as they occur within on-line learning; this should help avoid overfitting as well as significantly decrease the run-time needed for solving off-line structure identification problems.

## REFERENCES

- Affenzeller, M. 2003. "New Hybrid Variants of Genetic Algorithms - Theoretical and Practical Aspects". Universitätsverlag Rudolf Trauner, Linz, Austria.
- Affenzeller, M. 2005. "Population Genetics and Evolutionary Computation - Theoretical and Practical Aspects". Universitätsverlag Rudolf Trauner, Linz, Austria.
- Affenzeller, M. and Wagner, S. 2004. "SASEGASA: A New Generic Parallel Evolutionary Algorithm for Achieving Highest Quality Results". *Journal of Heuristics - Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems*, vol. 10, pp. 239-263, Kluwer Academic Publishers.
- Affenzeller, M. and Wagner, S. 2005. "Offspring Selection: A New Self-Adaptive Selection Scheme for Genetic Algorithms". *Adaptive and Natural Computing Algorithms*, Springer Computer Science, pp. 218-221.
- Bremer, W. 2000. "On- and Off-board Diagnostics: The Role of Legislation and Standardization". *On- and Off-board Diagnostics*, R. K. Jurgen (ed.), SAE Inc, Warrendale, PA, pp. 557-566.
- Del Re, L.; Langthaler, P.; Furtmüller, C.; Winkler, S.; and Affenzeller, M. 2005. "NO<sub>x</sub> Virtual Sensor Based on Structure Identification and Global Optimization". *Proceedings of the SAE World Congress 2005*, paper number: 2005-01-0050.
- Ellis, J.B. 1998. "An Investigation of Predictive and Adaptive Model-Based Methods for Direct Ground-to-Space Teleoperation with Time Delay". Master Thesis, Wright State University.
- Goldberg, D.E. 1989. "Genetic Algorithms in Search, Optimization and Machine Learning". Addison Wesley Longman.
- Holland, J.H. 1975. "Adaption in Natural and Artificial Systems". MIT Press, Cambridge, Mass.

Koza, J. 1992. "Genetic Programming: On the Programming of Computers by means of Natural Selection". MIT Press, Cambridge, Mass.

Koza, J. 1995. "Genetic Programming for Econometric Modeling". *Intelligent Systems for Finance and Business*, 1<sup>st</sup> edn. Wiley & Sons, pp. 251-269.

Langdon, W. and Poli, R. 2002. "Foundations of Genetic Programming". Springer Verlag, Berlin Heidelberg New York.

Langley, P. et al. 1987. "Scientific Discovery: Computational Explorations of the Creative Process". MIT Press, Cambridge, Mass.

Lazarus, C. and Huosheng, H. 2001. "Using Genetic Programming to Evolve Robot Behaviours". *Proceedings of the 3<sup>rd</sup> British Conference on Autonomous Mobile Robotics & Autonomous Systems*.

Martin, M. 2002. "Genetic Programming for Real World Robot Vision". *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 67-72.

Michalewicz, Z. 1996. "Genetic Algorithms + Data Structures = Evolution Programs". 3<sup>rd</sup> edn. Springer-Verlag, Berlin Heidelberg New York.

Wagner, S. and Affenzeller, M. 2005. "HeuristicLab: A Generic and Extensible Optimization Environment". *Adaptive and Natural Computing Algorithms*, Springer Computer Science, pp. 538-541.

Wagner, S. and Affenzeller, M. 2005. "SexualGA: Gender-Specific Selection for Genetic Algorithms". *Proceedings of The 9<sup>th</sup> World Multi-Conference on Systemics, Cybernetics and Informatics*.

Werner, J.C. and Fogarty, T.C. 2001. "Genetic Programming Applied to Gene Function Identification". *Proceedings of The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Winkler, S.; Affenzeller, M.; and Wagner, S. 2004. "Identifying Nonlinear Model Structures Using Genetic Programming Techniques". *Cybernetics and Systems 2004*, pp. 689-694, Austrian Society for Cybernetic Studies.

Winkler, S.; Affenzeller, M.; and Wagner, S. 2004. "New Methods for the Identification of Nonlinear Model Structures Based Upon Genetic Programming Techniques". *Proceedings of the 15<sup>th</sup> International Conference on Systems Science*, vol. 1, pp. 386-393, Oficyna Wydawnicza Politechniki Wroclawskiej.

Winkler, S. 2004. "Identification of Nonlinear Model Structures by Genetic Programming Techniques". Diploma Thesis, Institut für Systemtheorie und Simulation, Technisch-Naturwissenschaftliche Fakultät der Johannes Kepler Universität, Linz, Austria.

Winkler, S.; Affenzeller, M.; and Wagner, S. 2005. "Solving Multiclass Classification Problems by Genetic Programming". *Proceedings of The 9<sup>th</sup> World Multi-Conference on Systemics, Cybernetics and Informatics*.

## BIOGRAPHY

Stephan Winkler received his MSc in Computer Science from Johannes Kepler University Linz, Austria in 2004 (title of his diploma thesis: "Identification of Nonlinear Model Structures By Genetic Programming"). Currently he is a research associate at the Institute of Design and Control of Mechatronical Systems as well as the Institute of Fromal Models and Verification, both at Johannes Kepler University Linz, Austria.

His research interests include Genetic Programming, Nonlinear Model Identification, Fault Detection and Machine Learning.

Michael Affenzeller has published several papers and journal articles dealing with theoretical aspects of Genetic Algorithms and Evolutionary Computation in general. In 1997 he received his MSc in Industrial Mathematics and in 2001 his PhD in Computer Science, both from the Johannes Kepler University Linz, Austria.

He is a member of the Institute of Fromal Models and Verification at Johannes Kepler University Linz, Austria, where he currently holds the position of an Associate Professor since his habilitation in "Applied Systems Sciences with particular regard to Heuristic Optimization" in 2004.

Stefan Wagner also received his MSc in Computer Science from Johannes Kepler University Linz, Austria in 2004 (title of his diploma thesis: "Looking Inside Genetic Algorithms"). He is now a research associate at the Institute of Fromal Models and Verification at Johannes Kepler University Linz, Austria.

His research interests include Evolutionary Computation and Heuristic Optimization, Theory and Application of Genetic Algorithms, Machine Learning and Software Development.

More detailed information about the authors, their ongoing research activities and publications can be found at [www.heuristiclab.com](http://www.heuristiclab.com). Information about the HeuristicLab as well as the program itself (including selected plug-ins, example problems, importable problems and much more) are also ready for download from this site.