Elite Based Multiobjective Genetic Programming in Nonlinear Systems Identification

Alina PATELLI[1], Lavinia FERARIU[2]
"Gh. Asachi" Technical University of Iasi, Romania
Bd. D. Mangeron 53 A, RO-700050 Iasi
[1]*apatelli@ac.tuiasi.ro,* [2]*lferaru@ac.tuiasi.ro*

***Abstract*—The nonlinear systems identification method described in the paper is based on genetic programming, a robust tool, able to ensure the simultaneous selection of model structure and parameters. The assessment of potential solutions is done via a multiobjective approach, making use of both accuracy and parsimony criteria, in order to encourage the selection of accurate and compact models, characterized by expected good generalization capabilities. The evolutionary process is implemented from an elitist standpoint, and upgraded by means of two original contributions, namely an adaptive niching mechanism and an elite clustering procedure. The authors have also suggested a set of enhancements to aid the genetic operators in effectively exploring the space of possible model structures. In symbiosis with the customized genetic operators, a QR local optimization procedure was integrated within the algorithm. It exploits the nonlinear, linear in parameter form that the working models are generated in, for providing a faster parameter computation. The performances of the proposed methodology were revealed on two applications, of different complexity levels: the identification of a simulated nonlinear system and the identification of an industrial plant.**

***Index Terms*—evolutionary algorithms, genetic programming, multiobjective optimization, nonlinear system identification**

I. Introduction

In modern automatic control, nonlinear systems identification is a frequent issue of major importance [1, 2, 3]. Data driven methodologies are extensively used, especially because the analytical approaches impose unrealistic assumptions, given the nature of the industrial environment, and offer only a limited level of approximation of the complex phenomena that govern the inner behavior of nonlinear plants [3].

Most identification methods select the optimal models using a set of predefined structures, chosen according to available expertise. A viable approach is to generate the model structures based on a general template, one proven to be a universal approximator of nonlinear continuous bounded dynamic functions (e.g. the nonlinear linear in parameters formalism). Even if such a template is considered, a very large number of possible architectures have to be investigated for each particular identification problem to solve. Hence, in order to efficiently select the proper model, the identification method has to provide an automatic, unsupervised search of the best structure configuration [3]. Even after the model structure has been selected, determining the corresponding optimal set of model parameters represents a challenging task, as it involves exploration within large, multidimensional spaces, while confronted with nonlinearities, multimodalities and discontinuities.

The use of nonlinear, linear in parameters formalism may bring some benefits to the identification methodology. On the one side, it guarantees the existence of an appropriate model, at any desired degree of accuracy. On the other side, the design procedure may exploit the linearity of the output in terms of model parameters. The main difficulty remains the determination of the proper regressors to be included in the model structure, especially in the case of systems having many inputs/states/outputs.

The paper layout follows here: section II browses through the essential results obtained by researchers in the field of nonlinear identification, and outlines the authors' original contributions. Section III briefly introduces nonlinear models, linear in parameters. Section IV describes initial population generation, as well as the reconfiguration mechanism meant to ensure parameter wise linearity of all potential models. Tree evaluation is the topic of section V, whereas section VI deals with two elite related upgrades. Next, the genetic operators are outlined in section VII, whilst section VIII is dedicated to applications, and section IX to conclusions.

II. Related Work and Original Contributions

The research conducted in this field records numerous attempts to address the issue of determining the ideal regressor combination [1, 2, 3]. Therefore, a brief description of the most important achievements and other significant related work is considered next. One of the first major breakthroughs in systems identification consisted in perfecting a customized tool based on the NARMAX (**N**onlinear **A**uto-**R**egressive **M**oving **A**verage for e**X**ogenous inputs) framework [1]. It considers an initial model structure of maximum complexity, thus eliminating the need for sequential preset structure investigations. The complexity of the considered structure and the post design reduction of terms are serious drawbacks. As an alternative, Genetic Programming (GP) offers the possibility of working concomitantly on the structure and the parameters of the model, in a flexible manner, without demanding rich a priori information about the neighborhood of the solution [2], [3]. GP identification methods evolve populations of potential models, each encrypted, as Koza suggested [4, 5, 6], by a hierarchical tree structure. An appropriate set of parameters

Digital Object Identifier 10.4316/AECE.2010.01017

is computed for each tree, resulting in models of various degrees of adaptation, out of which the best individuals are encouraged to reproduce. The generated offspring compete with their parents, according to the Darwinian principle of the survival of the fittest [2, 6].

Most attempts to identify nonlinear systems via evolutionary techniques appeal to a single objective evaluation [4], [5]. In order to control model complexity, it has been suggested to hybridize the GP algorithm with an **O**rthogonally **L**east **S**quare (OLS) procedure aimed at locating and eliminating the least significant model terms [5]. Though easy to implement, the approach is greedy in nature, as an insignificant regressor may become important, at a future generation, if targeted by one of the implemented genetic operators. In this context, it is preferable to consider the minimization of model complexity as a separate optimization criterion by making use of the **M**ulti **O**bjective **O**ptimization (MOO) paradigm [3, 7]. By controlling both the accuracy and the parsimony of the evolved models, the approach permits the selection of a solution with good expected generalization capabilities [6]. MOO is efficiently addressed by means of Pareto-based approaches [6, 7], which employ dominance analysis in terms of objectives with equal or different priorities [7]. This method was later on refined by Deb, who introduced a fitness assignment scheme based on the static niching of individuals in the objective space, as a strategy to provide a well spread distribution of the solutions along the Pareto optimal front. [8, 9].

When dealing with dynamic search spaces, the individuals can be aided to adapt to the abrupt and unexpected variations of the objective functions, by means of variable relocation, which consists in translating the decision variables with an amount called relocation radius. [10]. This technique may prove efficient if the decision variable space is of fixed size, yet, in the case of system modeling, the parameter space is of variable dimension, therefore it is recommended to use the information available in the objectives space rater then the decision variables one.

The method proposed by the authors continues in the same trend as the related work described above, yet adds new enhancements to better cope with nonlinear system identification problems. Firstly, the authors introduce several enhancements which target the multiobjective and elitist aspects of the evolutionary identification paradigm. To begin with, the complexity assessment criterion has been specifically tailored to consider not only the compactness of a certain model, but also the relevance of the terms it contains. A second upgrade targets the elitist nature of the evolution procedure by introducing an original dynamic niching strategy, for elite fitness computation, designed to maintain the diversity of the models situated on the Pareto optimal front. To drive the models towards a certain interest region of the Pareto front, a third innovative enhancement has been deployed, namely an elite clustering technique aimed at increasing selection pressure.

Another category of enhancements refers to customization of GP which exploit the specific type of evolved models. The suggested initialization procedure assures a rich batch of initial genetic material, evenly distributed across the problem search space [11]. Secondly, a transformation mechanism is provided to ensure parameter wise linearity of all potential models, at all algorithm stages, thus facilitating the hybridization with the QR local optimization routine [11]. Thirdly, the authors introduce custom made genetic operators, optimized to work in an effective symbiosis with the aforementioned parameter computation procedure [11].

III. Nonlinear Models Linear in Parameter

Nonlinear models, linear in parameters are used, as they are proven to approximate the behavior of any nonlinear system, to any desired degree of accuracy [1]. Whilst it supports the universality of the approach, this mathematical formalism also brings some important advantages in terms of parameters' computation, as described below. Hence, the suggested identification method handles models of the following form:

$$\hat{y}_i(k) = \sum_{q=1}^{r} c_{iq} F_{iq}(\mathbf{x}(k)), \quad c_{iq} \in \Re, i = \overline{1,n}, \tag{1}$$

where

$$\begin{aligned}\mathbf{x}(k) = [&u_1(k),..,u_1(k-n_u),..,u_m(k),..,u_m(k-n_u),\\ &\cdots y_1(k-1),..,y_1(k-n_y),....,y_n(k-1),y_n(k-n_y)]\end{aligned} \tag{2}$$

is the terminal vector containing the current and the lagged values of plant inputs and outputs, k stands for the current time instant, the system inputs and outputs are denoted with u_i, $i = \overline{1,m}$ and y_j, $j = \overline{1,n}$ respectively, $\hat{y}_i$ indicates the estimation of y_j which is provided by the designed model, n_u and n_y are the maximum permitted input and output lags [1].

Nonlinear functions F_{iq} are called regressors and represent atomic combinations (products) of terminals, namely $\mathbf{x}$ vector elements, considered to any exponent. The polynomial model can be rewritten in matrix-based formalism:

$$\begin{bmatrix} F_{i1}(\mathbf{x}(1)) & .. & \cdots & F_{ir}(\mathbf{x}(1)) \\ F_{i1}(\mathbf{x}(2)) & .. & \cdots & F_{ir}(\mathbf{x}(2)) \\ \vdots & \vdots & & \vdots \\ F_{i1}(\mathbf{x}(p)) & .. & \cdots & F_{ir}(\mathbf{x}(p)) \end{bmatrix} \begin{bmatrix} c_{i1} \\ c_{i2} \\ \vdots \\ c_{ir} \end{bmatrix} = \begin{bmatrix} \hat{y}_i(1) \\ \hat{y}_i(2) \\ \vdots \\ \hat{y}_i(p) \end{bmatrix} \tag{3}$$

or:

$$\mathbf{F}_i \cdot \mathbf{c}_i = \hat{\mathbf{y}}_i, \quad \mathbf{F}_i \in \Re^{p \times r}; \mathbf{c}_i \in \Re^r; \hat{\mathbf{y}}_i \in \Re^p, i = \overline{1,n} \tag{4}$$

which better outlines its two main components: model structure encrypted by the regressors matrix $\mathbf{F}_i$, and model parameters encapsulated by the coefficients vector $\mathbf{c}_i$.

The goal of the GP procedure is to obtain the optimal model structure, a task accomplished by specifically enhanced genetic operators, via a multiobjective optimization. To compute an appropriate set of corresponding model coefficients, an QR local optimization procedure is called before the evaluation stage. QR decomposition exploits the linearity of the model subject to $\mathbf{c}_i$, performing a faster, one-step parameter computation. Consequently, the generated models stand a good chance of featuring a simple structure, with expected good accuracy on validation data sets.

IV. Internal Model Encryption

GP continues in the trend set by genetic algorithms relative to problem representation, by increasing the complexity of the structures undergoing adaptation. In the case of GP, the chromosomes are hierarchical structures of varying size and shape, generated by means of recursive combinations between the elements of a set of functions/operators and a set of terminals [4].

Given the mathematical form of the working models (1), one can define the operator set $O = \{+,*\}$ and the terminal set **x** (2). This choice of terminals and operators satisfies the closure requirement, meaning that any possible combination leads to a valid model. Moreover, *O* is minimally sufficient, as it contains only the elements necessary to generate a possible desired model (1). The sufficiency of the terminal vector **x** is satisfied if it includes an appropriate number of lagged values. n_u and n_y could be set by "trial and error" without rich expertise and/or extensive tuning effort. Note also that the design procedure may cope with extraneous terminals, as it flexibly builds the needed regressors F_{iq}.

As each tree-based chromosome may have a single root node, and therefore a single output, for the case of multi outputs systems ($n > 1$), one has to configure *n* different models, during *n* different evolutionary loops, each model estimating one output of the system, therefore encoding a single matrix $\mathbf{F}_i$ at a time.

Trees are initially generated by recursively inserting randomly selected elements from **x** and *O*, resulting in representations like the one indicated in Fig. 1a. To start the algorithm off with a rich batch of relevant genetic material, every tree in the initial population is a distinct combination of all the terminals in vector **x**. Given the random nature of the generation process, individuals are most likely not compliant with the formalism considered in (1) [11]. To fix that problem, a transformation mechanism has been introduced by the authors, in order to transform the trees from a raw, terminal based form to a linear in parameter, regressor based one. The mathematical basis of this procedure is straightforward:

$$a \cdot (b + c) = a \cdot b + a \cdot c \tag{5}$$

In computational terms, all the "+" nodes that are situated below "*" nodes within the tree, are lifted one level at a time, towards the tree root, until a linear in parameter compliant form is achieved (Fig. 1b).

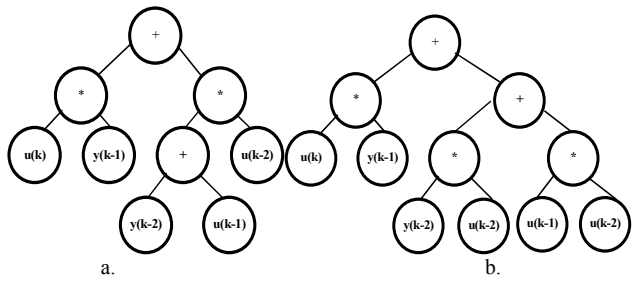

Figure 1. Trees in the initial population: a. terminal-based form; b. regressor-based form $F_1 = u(k)y(k\text{-}1)$, $F_2 = y(k\text{-}2)u(k\text{-}2)$, $F_3 = u(k\text{-}1)u(k\text{-}2)$.

To compute the optimal set of coefficients for each of the population chromosome, a representative set of input - output data pairs, $S = \{(\mathbf{u}(k), \mathbf{y}(k)), k = \overline{1, p}\}$, is needed [12]. *S* can be collected during plant exploitation and it has to completely describe the behavior of the system to be identified. According to (3), for each $\mathbf{F}_i$ matrix, encoded by an individual, QR procedure based on Householder reflectors is separately applied. Each resulted $\mathbf{c}_i$ set is afterwards inserted within the corresponding individual, achieving full tree-like representations of both structure and parameters ready for evaluation.

Evaluation Criteria

Tree assessment is performed relative to accuracy and parsimony criteria, as these are the main requirements for a good quality model, within the framework of identification problems. From a mathematical point of view, the accuracy of a model *M* is evaluated via *SEF* (**S**quared **E**rror **F**unction):

$$SEF(M) = \frac{1}{2} \sum_{k=1}^{p} (y_i(k) - \hat{y}_i(k))^2 \tag{6}$$

To avoid the production of over fitted models, with small *SEF* values, but improper generalization capability, *CF* (**C**omplexity **F**unction) is used for evaluating the complexity of tree-based individuals.

$$CF\ (M) = r + \frac{t}{n_u + n_y + 1} - \sum_{q=1}^{r} \lg c_{iq} \tag{7}$$

Here, *r* and *t* represent the number of regressors and terminals used in the encrypted model, respectively. *CF* penalizes the individuals with many regressors and/or with regressors encapsulating numerous terminals. The third term in (7) is aimed at evaluating the relevance of the regressor coefficients (c_{iq}). A model that features few regressors, each made up of a reasonable number of terminals, yet factored by very small parameters, is of no practical value, even if it complies with the accuracy demands. As there is no limitation as to the range of values of the data in the training set, all members of set *S* are scaled within the interval [-1 1], in order to make the last component of the complexity criterion representative.

The two objectives described above are conflicting, meaning that a very simple model is probably not able to achieve high accuracy and vice versa. In order to compare one tree against another under these circumstances, the non-dominance concept is considered. An individual in the current population is said to be non-dominated if it is better than all the other trees in the same population with respect to at least one of the considered objectives. The whole set of the non-dominated individuals at a certain generation constitutes the first order Pareto front. It is expected that in the early stages of the algorithm, the first order Pareto front will be quite far from the optimal one, and it is the evolutionary algorithm's purpose to bring it as close as possible, whilst preserving the diversity of solutions, for the *a posteriori* selection of the desired one.

V. Elitist Fitness Assignment

After objective value computation, the next step is the assignment of fitness values, also called selection probabilities, representing the chances that each individual stands of being chosen for reproduction/insertion. The scheme deployed to fit that end consists in gradually

building a set of elites out of the individuals in the current population.

An individual from the regular population is considered to be elite material if it is not dominated by any of the other elites, previously inserted in the set [9]. Once such an individual is identified, a copy is created and included in the elite set. All the pre-existent elites, which are dominated by the new comer, are excluded from the set. The same fitness is assigned to both the regular tree and its copy in the elite set [9]. To suit that purpose, the authors have proposed a novel dynamic niching technique, based on Deb's idea of dominance analysis [9], yet upgraded with an adaptive proximity range computation mechanism. More precisely, the proximity range σ is computed in the objective's space as the average Euclidian distance between the members of the elite set up to the current moment, and not preset according to a grid of an offline determined step, like the original alternative. The result is a more flexible evaluation of elite proximity, needed for spotting crowded elites, which will be assigned with slightly diminished fitness values as compared to the solitary ones, in order to preserve elite diversity.

$$Fit(e) = \frac{F_M}{\sum_{z \in E} \begin{bmatrix} 1 - \dfrac{d(e,z)}{\sigma}, d(e,z) \le \sigma \\ 0 \qquad\quad , d(e,z) > \sigma \end{bmatrix}}. \tag{8}$$

In specific terms, the elite fitness computation starts off with a reference value, FM, equal to the highest elite fitness from the previous generation. That reference value will be diminished or not, depending on the proximity of the other elites in the set, denoted symbolically by z. Euclidian distance *d(e, z)* is used to measure proximity, E marks the elite set and σ stands for the proximity range.

The trees in the regular population which were not duplicated in the elite set (trees 5 through 9 in Fig. 2), will be assigned a fitness value equal to the fitness of the closest elite minus the distance in between the two, computed in the objective space. According to their newly assigned fitness values, the trees in the regular population will be selected for reproduction via a roulette based procedure. Note that the elites do not participate in the reproduction process; their purpose is that of dynamically handling selection pressure aimed at pulling the regular population as close to the optimal Pareto front as possible.

In the field of engineering, a model's accuracy is more important than its simplicity. Of course, the parsimony objective is not to be neglected, yet it should be met to such an extent as not to interfere with the overall model accuracy. Consequently, there is a specific interest region within the elite set, that as many individuals as possible should be pulled towards (elites marked with "o" in Fig. 2). In order to integrate this desiderate in the algorithm, a second original upgrade has been implemented, namely an elite clustering technique. The average performances of the elite set are computed, resulting in a point within the objectives space, which could be assimilated with a "fake" tree (marked with a black square in Fig. 2). The mean Euclidian distance between all elites and the substitute individual is determined, and the fitness values of the elites situated in the lower left quadrant (relative to the square dot) are increased with the resulted amount. In the following generation, the trees neighboring the elites in the interest region will receive higher fitness values, thus encouraging the production of offspring in the vicinity of the targeted zone of the Pareto front.

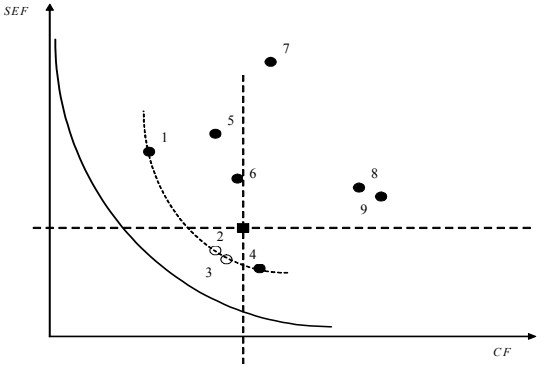

Figure 2. Elite set and Pareto optimal set.

VI. Genetic Operators

Once in the reproduction pool, the selected parents are subject to the action of genetic operators namely crossover and mutation. Crossover selects one cut point in each of the two parents that it handles at a time, and swaps the resulting sub-trees, thus generating a pair of offspring. Mutation, on the other hand, is responsible with small modifications, consisting in replacing tree nodes with one of their alleles. Its usefulness is revealed especially towards the end of the evolutionary loop, when the population is saturated with quite similar well adapted trees that crossover cannot improve significantly.

To optimize the hybridization between the crossover operator (which handles trees structure) and the QR decomposition based optimization procedure (which deals with parameters' computation), the cut point search process has been upgraded to spot the sub-trees which would bring the highest gain in terms of accuracy, if swapped. Especially towards the end of the evolutionary loop, when the elite set comes close to the Pareto optimal front, well adapted identical nonlinear atoms start to appear in many trees (nodes 5, 6, 7 in the first parent and nodes 4, 5, 6 in the second parent in Fig. 3). When two such trees become parents, the similar regressors that they contain should not be considered for swapping, as the increase in children accuracy would be null. Fig. 3 illustrates such a case, where all the nodes within the identical sub-trees, along with the nodes on the path to the parents' roots are eliminated from the cut point list, in order to protect the well adapted regressors. After the potential cut point list simplification,

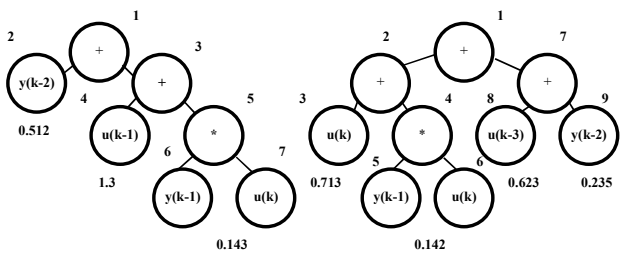

Figure 3. Parents with similar regressors in the reproduction pool.

the only valid remaining options are node 4 in the first parent and node 3 and 8 in the second. Thus, the cut point search that blind crossover would have performed is significantly simplified.

As far as the second genetic operator is concerned, mutation has been trained to modify not only the terminal names, but also their exponents. This behavior is targeted at dealing with an unwanted phenomenon called compensation that occurs when a model features an increased number of regressors, while the same accuracy could be achieved by a solution with fewer nonlinear atoms, yet to higher exponents.

VII. Applications

Firstly, the proposed design methodology was tested by simulation, on a nonlinear system described by the following model:

$$y(k) = 0.2y(k-1) + 1.5u^2(k-1) + u^2(k-2) + u^3(k-3) \quad (9)$$

A training data set of 100 points was generated, normally distributed in between –100 and 100, and used for both a basic version of the multiobjective evolutionary algorithm (without the customized genetic operators, dynamic niching and elite clustering) and for the algorithm presented above, featuring all described enhancements. Both models achieved a global *SEF* of 0.0000, yet the one produced by the basic identification procedure is not compact and shows regressors compensation (Fig. 4). It features several "alien" terms, with low coefficients, which do not influence significantly the overall performances and must be eliminated post design. In addition, the contribution of regressor $u^2(k\text{-}1)$ has been "compensated" by including two different terms, $u^2(k\text{-}2)$ and $y(k\text{-}2)$, whose cumulated effect on the model accuracy is quite similar, in terms of *S*, to the one of the substituted regressor.

The model obtained via the enhanced procedure does not feature any terms with insignificant coefficients, due to the definition of the complexity criterion (7), which considers coefficient relevance as well as regressor count. Furthermore, the fact that upgraded mutation also alters the terminal exponents, encourages the inclusion of compact regressors in the model, thus preventing compensation. A positive effect of the guided nature of the cut point search, as well as of the two elite related enhancements, is a speedy convergence of the upgraded algorithm (the final model was generated in 15 generations), relative to the basic alternative (the selected model was found in 50 generations), given that both versions evolved a population of 20 individuals.

$$\begin{aligned} y(k) &= 0.2 * y(k-1) + 0 * u^2(k-1) + \\ &+ u^2(k-2) + \ 0.7 * u^2(k-2) + \\ &+ u^3(k-3) + 1.2 * y(k-2) \\ &+ 0.000013 * u(k-3) \end{aligned}$$

a

$$\begin{aligned} y(k) &= 0.2 * y(k-1) + 1.5 * u^2(k-1) \\ &+ u^2(k-2) + u^3(k-3) \end{aligned}$$

b

Figure.4. Obtained model: a. basic algorithm; b. enhanced algorithm.

A more realistic analysis of the benefits that the algorithm customizations bring to the quality of the generated models can be conducted on a real, complex, industrial system. The steam subsection of the evaporation station within the sugar factor of Lublin, Poland has been chosen for that purpose. It is a one input (vapor temperature), one output (vapor pressure) nonlinear system, which was monitored for an entire production shift in order to collect a 3000 row data set that illustrates the maximum possible excitation of the process. Each 10^{th} sampled value was included in the learning data set used to train the enhanced identification algorithm. The validation of the generated model was performed with respect to a data set collected in the previous month of plant exploitation [13]. Some analytical models are available yet they provide only a limited level of approximation.

A series of comparative experiments was performed on the fully enhanced evolutionary algorithm (EMOO), as well as on basic versions of the method, in order to outline the contribution of each proposed upgrade to the overall model performances and algorithm convergence speed and robustness. A separate experiment was run for each set of algorithm parameters indicated in a row of Table 1. The results discussed below refer to the best individual, in terms of accuracy, from the final elite set generated by each version of the identification method.

Firstly, the fully enhanced evolutionary algorithm is compared against a version that features a coarse definition of the complexity criterion, that only counts regressors, without considering how many terminals are involved, nor coefficient relevance. As the maximum allowed lags increase, both versions of the algorithm manage to maintain satisfactory model accuracy and control the regressor count. However, the coarse *CF* generates more complex models, due to the regressors containing an increased number of terminals, a tendency successfully eliminated by the refined *CF* algorithm, which preserves the terminal count at a reasonable value.

Table I. Algorithm Parameters

Model	Lags	Population size [number of individuals]
M1	$n_y = 2$ $n_u = 2$	75
M2	$n_y = 2$ $n_u = 2$	150
M3	$n_y = 2$ $n_u = 2$	500
M4	$n_y = 4$ $n_u = 5$	75
M5	$n_y = 4$ $n_u = 5$	150
M6	$n_y = 4$ $n_u = 5$	500
M7	$n_y = 10$ $n_u = 11$	75
M8	$n_y = 10$ $n_u = 11$	150
M9	$n_y = 10$ $n_u = 11$	500

When the enhanced genetic operators are replaced with their classical versions, two main setbacks become noticeable. The most disturbing one has to do with convergence speed. As the maximum allowed lags increase, larger trees enter the reproduction pool, where they are processed by a "blind" version of the crossover operator. The unguided cut point search usually leads to children with poorer performances than those of their parents, delaying the generation of well adapted individuals. The raw genetic operator does not manage to produce as good a model as its enhanced counterpart, within the maximum allowed number of generations (M7→M9 in Table II). Secondly, because of the fact that classic mutation cannot alter terminal exponents, the evolutionary algorithm tries to compensate by including a larger number of regressors in the model, when the same accuracy could be achieved by only one nonlinear atom, to a higher exponent. This compensation phenomenon, characterized by artificially increased model

complexity, is not encountered with the models generated by the fully upgraded EMOO procedure.

TABLE II. EFFECT OF REFINED COMPLEXITY FUNCTION DEFINITION

Model	Coarse *CF* (EMOO)			Refined *CF* (EMOO)		
	MSEF	*CF*	*t*	*MSEF*	*CF*	*t*
M1	0.753	4	6	0.634	5	7
M2	0.621	5	7	0.734	4	6
M3	0.745	4	6	0.612	5	8
M4	0.832	6	12	0.734	6	7
M5	0.723	5	14	0.712	5	6
M6	0.656	5	17	0.453	4	8
M7	0.783	5	23	0.545	5	6
M8	0.812	6	19	0.567	5	5
M9	0.757	6	24	0.678	5	7

MSEF – mean squared error function (*SEF*/*p*)
CF – complexity function
t – number of terminals

TABLE III. EFFECT OF ENHANCED GENETIC OPERATORS

Model	Classic GO (EMOO)			Enhanced GO (EMOO)		
	MSEF	*CF*	*Gen*	*MSEF*	*CF*	*Gen*
M1	0.783	5	76	0.624	5	50
M2	0.725	7	75	0.690	6	45
M3	0.832	12	80	0.656	4	55
M4	0.841	3	95	0.645	7	60
M5	0.912	7	100	0.745	4	56
M6	0.656	13	97	0.734	6	46
M7	1.435	9	100	0.634	5	57
M8	1.234	11	100	0.712	7	62
M9	0.957	10	100	0.632	5	49

MSEF – mean squared error function (*SEF*/*p*)
CF – complexity function
Gen – number of generation at which the solution was found
GO – genetic operators

The final comparative study is aimed at highlighting the influence of the dynamic niching and elite clustering upgrades. The most obvious gain has to do with convergence speed. Increasing the selection pressure by awarding high fitness values to the elite individuals close to the interest region of the optimal Pareto front results in obtaining a valid model in less than half the number of generations that the raw algorithm version requires. Furthermore, regardless of how much the maximum allowed number of generations is increased, the raw EMOO gets saturated, and does not register an improvement in accuracy, even when a rich batch of genetic material is provided (M3, M6, M9 in Table 4).

For more detailed evaluation, the performances of the most accurate elite generated by the fully enhanced EMOO in the third comparative run for M1 configuration are shown in Fig. 5. This model features 5 regressors, with a mean *SEF* of 0.653 over the validation data set, and was obtained after 50 generations.

VIII. CONCLUSION

The enhanced elite based multiobjective optimization algorithm described in this paper is a powerful identification tool that draws part of its advantages from the classic features of genetic programming related approaches. Therefore, the procedure is capable of performing the model search by means of unsupervised generation of new solutions, simultaneously working on the structure and parameters of the nonlinear models. This recommends it for complex nonlinear system identification issues, with poor *a priori* information available. Moreover, the compact, tree-based model encryption, combined with the capacity of handling discontinuities and nonlinearities, makes the method suitable for engineering applications, provided that sufficient computational resources are available.

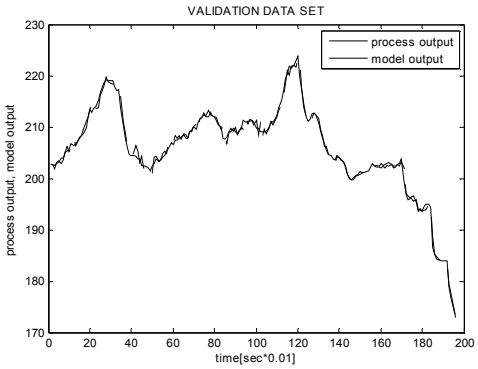

Figure 5. Model validation.

The first set of specific enhancements that the authors have implemented target the core of the evolutionary algorithm, namely the genetic operators, upgraded to increase both convergence speed and exploration capabilities of the algorithm. Parameter computation is achieved via QR decomposition, by exploiting the linear in parameter model compatibility. The procedure also features two elite related customizations, aimed at dynamically increasing selection pressure to the desired extent and preserving individuals' variety.

All integrated upgrades make the suggested approach a valid alternative for solving identification problems, as illustrated on the achieved experimental results.

REFERENCES

[1] H. Wey, S. A. Billings, J. Lui, "Term and Variable Selection for Nonlinear Models", *Int. J. Control* 77, 2004, pp. 86-110
[2] N. Nedjah, A. Abraham, L. de Macedo Mourelle (Eds.) Genetic Systems Programming - Theory and Experiences, Springer, Netherlands, 2006.
[3] P. J. Flemming, R. C. Purshouse "Evolutionary Algorithms in Control Systems Engineering: A Survey", *Control Engineering Practice* 10, 2002, pp. 1223-1241.
[4] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA: MIT Press, 1992, pp. 73-190
[5] J. Madar, J. Abonyi, F. Szeifert Genetic Programming for System Identification [Online]. Available: http://www.fmt.vein.Hu/softcomp/isda04_gpolsnew.pdf, 2005.
[6] R. Riolo, T. Soule, B. Worzel (Eds), *Genetic Programming Theory and Practice IV*, Springer, USA, 2007.
[7] K. Rodriguez-Vasquez, C. M. Fonseca, P. J. Flemming, "Identifying the Structure of Nonlinear Dynamic Systems Using Multiobjective Genetic Programming" in *IEEE Transactions on Systems Man and Cybernetics, Part A – Systems and Humans,* 34, 2004, pp. 531-534
[8] J. Knowles ,· D. Corne, K. Deb (Eds.) Multiobjective Problem Solving from Nature - From Concepts to Applications, natural Computing Series, Springer, USA, 2008.
[9] K. Deb, Multiobjective Optimization using Evolutionary Algorithms. John Wiley and Sons, USA, 2001.
[10] Y. G. Woldesenbet, G. C. Yen Dynamic Evolutionary Algorithm with Variable Relocation. IEEE Transactions on Evolutionary Computation, vol.13, no.3, 500-513, 2009
[11] L. Ferariu, A. Patelli, "Multiobjective Genetic Programming for Nonlinear System Identification", *Proc. of ICANNGA09*, Kuopio, Finland, 2009.
[12] T. Back, D. Fogel, Z. Michalewicz, *Evolutionary Computation 2.Advanced Algorithms and Operators.* US: Institute of Physics Publishing, 2000.
[13] L. Ferariu, M. Voicu, "Nonlinear System Identification Based on Evolutionary Dynamic Neural Networks wih Hybrid Structure" in *Proc. of IFAC Congress,* Prague, Czech Republic, 2005.