Cluster Based Multiobjective Genetic Programming in Nonlinear Systems Identification

Alina Patelli*, Lavinia Ferariu*

** Department of Automatic Control and Applied Informatics, "Gh. Asachi" Technical University, 27 D. Mangeron St., 700050, Iasi, Romania*

Abstract: Multivariable nonlinear systems identification is addressed, in the following, by means of enhanced multiobjective evolutionary optimisation. The paper suggests a customised genetic programming algorithm that generates nonlinear linear in parameter models, according to a mathematical pattern that has been proven to be a universal approximator. In order to efficiently exploit the parameter wise linearity, the authors propose a symbiosis between the genetic operators and a local optimisation procedure based on QR decomposition. This hybridisation provides simultaneous structure selection and parameter computation, whilst facilitating the unsupervised exploration of the search space. Model assessment is conducted relative to accuracy and parsimony evaluation criteria. The latter has been specifically tailored to encourage the gradual elimination of insignificant model regressors, while preserving the ones which best capture the nonlinear dynamics of the plant, thus rendering the suggested method suitable for identifying multivariable systems, even in the presence of extraneous lags. In order to make the proposed approach compatible with the particular requirements of the identification problem, within the framework of automatic control, the authors have introduced two additional enhancements, namely a dynamic clustering procedure and an adaptive migration mechanism. The performances of the suggested algorithm are revealed by three applications of different complexities: an academic test case featuring an increased number of inputs with time delay and a complex nonlinear industrial plant.

Keywords: genetic programming, multiobjective optimisation, nonlinear systems identification, multivariable systems.

1. INTRODUCTION

Complex nonlinear systems are involved in almost all engineering applications of an industrial nature. Such systems often feature numerous inputs and outputs, complex nonlinearities, and operate within noisy environments, thus, the obtaining of a sound mathematical model is a complicated undertaking, yet necessary for efficient plant exploitation. In this context, the preferred identification method should be capable of dealing with scarce a priori system wise information and of providing effective model structure and parameter selection, at an acceptable computational cost.

A way of complying with all the aforementioned requirements is to consider a general structure template, able to offer the approximation of any bounded nonlinear function, with any desired degree of accuracy. This general template is to be configured for each particular identification problem to solve, the main difficulty consisting in choosing the adequate structure between numerous possible ones. The present paper exploits the nonlinear linear in parameters template, proven to be an universal approximator.

The first attempt in using this mathematical formalism, recommended in the related literature, was based on determining model parameters starting from preset structures, the success of the identification process being entirely dependent on the accuracy of the initial structure choice. To solve this inconvenience, a NARMAX (Nonlinear Auto-Regressive Moving Average for eXogenous inputs) based methodology was perfected, considering a very complex initial model structure, under the assumption that all alien terms would be assigned insignificant coefficients, therefore baring little, if any influence, on the performances of the final model. This tool managed to eliminate the necessity of successive structure swaps, yet the increased size of the considered model structure and the post design term reductions are noticeable drawbacks.

The next step in system identification is represented by GP (Genetic Programming) related techniques (Flemming and Purshouse, 2002). Conceptually, the approach is centred on Koza's idea of encrypting a potential model in the form of a tree. A whole population of such trees is generated and evolved over generations, thus maintaining a variety of simple and flexible possible structures, each with its own set of optimum parameters. From one generation to the next, the trees are evolved according to the Darwinian principle of the survival of the fittest, which, in computational terms, is enforced by one or several objective functions (Coello Coello *et al.*, 2007). Should one tree manage to meet the requested demands, it

will be encouraged to pass on to the reproduction pool and participate in the generation of offspring. In this context, researchers developed the MOO (Multi Objective Optimisation) approach to GP based identification. It involves the use of several assessment criteria exploited from a Pareto optimal, dominance analysis based standpoint (Wey *et al*., 2004). Deb refined the procedure by introducing a highly accurate fitness assignment technique based on static niching (a niche is a reference distance in the objectives space), resulting in a much more effective tree evaluation (Deb, 2001).

The method proposed by the authors incorporates all the advantages of the previous tools developed in the field, while introducing several enhancements of its own. The two objectives employed by the enhanced MOO procedure are accuracy and parsimony. As, within the field of automatic control, the accuracy of a system model is most important, although the complexity criterion is not to be neglected either, the suggested algorithm makes use of two innovative upgrades in order to dynamically increase the priority of the accuracy objective. Firstly, a dynamic clustering technique is employed to divide the population in distinct batches, undergoing a differentiated evaluation process meant to encourage the production of accurate and reasonably parsimonious individuals. Secondly, every few generations, trees migrate in between the complementary batches, in amounts dynamically dictated by their average complexity, which allows the unsupervised balancing of the objectives' priorities.
Additionally, the authors have proposed an adaptation mechanism meant to guarantee a nonlinear linear in parameter compliance of all trees at all times. The suggested upgrade also facilitates the use of a local optimisation procedure based on QR decomposition, aimed at rapidly computing an optimum set of parameters for each tree encrypted model structure. Also, the genetic operators have been specifically tailored to preserve the positive effects of the QR decomposition procedure, as well as effectively improve the model structure relative to the considered objectives.

The paper is organized as follows. Section 2 summarizes the mathematical background of nonlinear models, linear in parameters. The main steps of the genetic loop are reviewed in section 3, while the detailed description of the implemented algorithm enhancements is presented in section 4. Section 5 discusses several experimental results which illustrate the applicability of the proposed design procedure, whereas the conclusions are outlined in section 6.

2. NONLINEAR MODELS

A multivariable nonlinear model, linear in parameters, having *m* inputs and *n* outputs can be described by the following equation:

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

where vector **x** contains lagged inputs and output values:

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

In (1) and (2), $\hat{y}_i$ and y_i specify the i^{th} model output and the i^{th} plant output, respectively, $u_j, j = \overline{1,m}$ denotes the j^{th} plant input, n_u and n_y stand for the maximum allowed lags for the system inputs and outputs and, finally, *k* specifies the current sampling time. It has been proven that the nonlinear linear in parameter models are capable of approximating any nonlinear continuous bounded function, to any desired degree of accuracy (Back *et al.*, 2000).

Functions F_{iq} are nonlinear atoms (regressors), consisting in combinations of terminals, to any exponent and lag, connected by multiplying operators only (Madar *et al.*, 2005). Consequently, the system model is a linear combination of such regressors, more compactly described by:

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

or

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

The model structure is encapsulated by the regressors of all matrices $\mathbf{F}_i$, $i = \overline{1,n}$. Maintaining their diversity falls under the care of genetic operators, customized to better fit the specifics of the linear in parameter formalism. For advantageous parameters calculation, a local optimisation procedure based on QR decomposition is applied, acting as a Lamarckian local optimisation mechanism during the evolutionary loop. Note that, if $n > 1$, the algorithm will run *n* times, generating a separate model for each output of the system, therefore encoding a single matrix $\mathbf{F}_i$ at a time.

3. MULTIOBJECTIVE OPTIMISATION

In order to generate a convenient model, the algorithm requires two sets of experimental data collected during the target system's exploitation, which completely illustrate the process behaviour - one used for the training/evaluation stage and the other used for the model validation stage. Both sets comply with the following pattern:

$$S = \{(\mathbf{u}(k), \mathbf{y}(k))\}, \qquad \mathbf{u} \in \Re^m, \ \mathbf{y} \in \Re^n, k = 1,..,p. \tag{5}$$

Other prerequisites consist in the maximum input and output lags, n_u and n_y. However, these algorithm parameters are application dependent, closely related to the desired model complexity and accuracy. They can be determined as a result of trial and error off-line experiments, at a negligible supplementary cost (Back *et al.*, 2000). Moreover, the algorithm is capable of

preserving the terminals with the most significant contribution to the final model's performances, gradually eliminating the others. Therefore, n_u and n_y are not necessarily required to be minimum.

As models are compliant with (1), GP considers the terminal set **x** indicated in (2) and an operator set containing addition and multiplication, $O = \{+, *\}$, for building the tree-like individuals. Each possible structure results as a recursive combination of terminals connected by operator nodes. To allow the generation of convenient models, sets **x** and O should be compliant with closure and sufficiency requirements (Koza, 1992). As set O is minimally sufficient, the second desiderate is met if enough lagged terms are included in the terminal set **x**. Note that it is not a prerequisite to work with a minimally sufficient **x** set, though any "alien" element could affect the exploration capabilities of the algorithm. The specific choice of the two sets, described above, combined with the flexible way the trees get built, ensures a compact final model, making post design simplifications unnecessary.

The next step consists in building the initial batch of tree encrypted potential models. As there is no way of knowing the particular sub-domain of the problem space where the solution might be situated, the trees are spread evenly across the whole search space, to ensure heightened exploration capabilities. Afterwards, the entire lot of trees is sent to the reproduction pool where the individuals undergo the effects of the genetic operators: crossover and mutation. The result is the generation of new trees, called offspring, some with better performances than those of their parents. The children get reunited with their parents in an intermediary population that constitutes the raw genetic material to be processed, in order to obtain the individuals of the next generation.

It is a known fact that overfitted models which offer a good approximation of the training data set could suffer from poor generalization capabilities (Rodriguez-Vasquez *et al.*, 2004). To diminish the risk of producing too complex structures as an effect of crossover, the problem is formulated as a multiobjective optimisation, addressing both accuracy and parsimony, by means of two objective functions, namely the **S**quared **E**rror **F**unction (*SEF*) and the **C**omplexity **F**unction (*CF*):

$$SEF(M) = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{p}(y_i(k) - \hat{y}_i(k))^2, \quad k = \overline{1,p}, \quad (6)$$

$$CF(M) = z. \quad (7)$$

Here, M denotes a possible model encoded by a tree-based individual and z specifies the number of its regressors.

Because the considered objectives are conflicting (it is highly unlikely to find a solution that is both accurate and simple), the problem admits an infinite set of optimal solutions, called Pareto-optimal set. Therefore, the multiobjective optimisation procedure has to generate, as a final result, a whole set of possible models, each of them representing a possible trade-off between accuracy and parsimony. Moreover, this set must be as close as possible to the Pareto optimal front.

Deb's concept based on dominance analysis may be used to conduct a Pareto optimal wise evaluation of the trees. The idea of this multiobjective approach (MOO) is graphically illustrated in Fig. 1. Inside a finite population P of individuals, a solution is called nondominated if it is better adapted than any other individual of population P, with respect to at least one objective function. All nondominated individuals (denoted 1→4 in Fig. 1) are separated from the current population and included in the first order front. They are assigned the highest fitness values. In order to differentiate between the trees of the same front, Deb suggests a fitness assignment scheme that favours the solitary individuals rather than the ones crowded in clusters. By doing so, trees with few or no neighbours (tree 1 in Fig. 1) are encouraged to survive and to produce offspring in their own vicinity, thus populating the depleted regions of the front. After fitness values are computed for all individuals of the first order front, the procedure is repeated over the remaining population, by separating the second order front and so on. Once every tree in the current population has been assessed, their fitness values are used as selection probabilities at insertion stage.

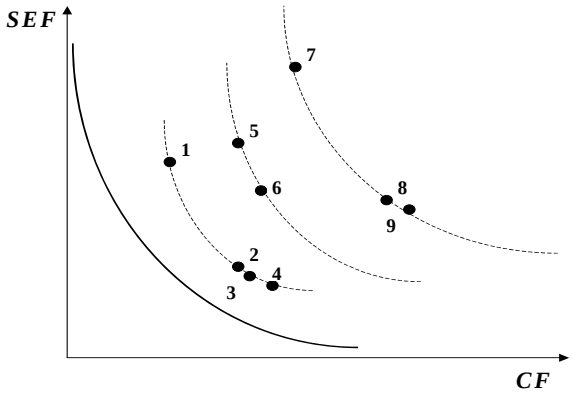

Fig. 1. Fronts of various orders separated within a finite population.

4. ALGORITHM ENHANCEMENTS

The suggested method operates with two categories of enhancements. The first exploits the parameter wise linearity of the generated models and includes customized genetic operators and a raw to regressive adaptation procedure which facilitates the hybridisation with a local optimisation procedure based on QR decomposition. Their goal is to increase the algorithm's convergence speed, as well as to expand its exploration efficiency. The second group of enhancements focuses on upgrading the MOO procedure, in order to enforce a higher priority to the accuracy objective, therefore concentrating the models towards the interest region of the Pareto optimal front (models marked with "o" in Fig. 3).

Given the formalism (1), the models are supposed to encode polynomial regressors. Yet, because the

individuals are built as recursive combinations of terminals from **x** set, a transformation is required in the tree-encoded model, in order to achieve parameter wise linearity. The trees are turned from terminal-based structures into equivalent regressor-based structures, exploiting the equation:

$$a \cdot (b + c) = a \cdot b + a \cdot c \tag{8}$$

After the transformation, "+" nodes will no longer be positioned as successors of "*" nodes. On a particular tree, given its current structure, QR decomposition aims at determining the best achievable parameters, in terms of *SEF* minimization. Evolving the structure into a fitter one, for which the local optimisation procedure would be able to yield better parameter values, falls under the care of genetic operators.

The crossover operator selects one cut point in each of the two selected parents and then swaps the resulting subtrees, thus producing two new individuals, called offspring. Towards the end of the evolutionary process, the trees become extremely well adapted and might encode very similar regressors. In that context, if the cut point selection is conducted in a purely random fashion, there is a good chance that the resulting subtrees might encode the same regressors, with similar coefficients. Such a swap would be completely irrelevant in terms of population diversity, as it would bring neither any fresh genetic material, nor any improvement of accuracy. Therefore, the crossover operator has been upgraded in order to detect such "problem" cut points and to eliminate them from the potential cut point list, prior to the actual subtree swap.

In Fig. 2, nodes 5, 6, and 7 of the first parent encode the same regressor as nodes 4, 5 and 6 of the second parent. The roots of the two similar regressors, along with all the other nodes on the path to the parent root will be eliminated from the potential cut point lists, for both parents.

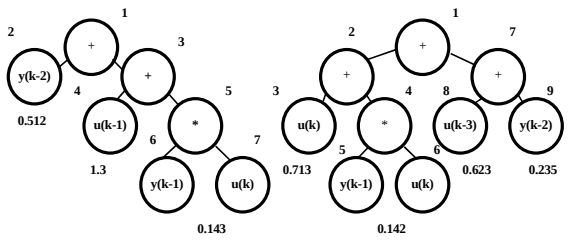

Fig. 2. Parents in the reproduction pool.

This is meant to protect the similar nonlinear atoms from crossover, as it is most likely that they represent well adapted regressors with a positive influence on the model's overall accuracy. Moreover, the performances of QR decomposition are decreased on structures with duplicate regressors. Ergo, the list of potential cut nodes must be reduced as described above, thus protecting the algorithm from redundant or harmful swaps on the one hand, and taking up less computational resources on the other hand, due to the "guided" nature of the cut point search operation.

Enhancing the crossover operator facilitates an efficient cooperation with QR decomposition, yet it does not rule out the compensation phenomenon, also encountered towards the end of the evolutionary loop. This problem occurs when the same accuracy might be achieved by replacing an entire combination of regressors with only one regressor with a different coefficient or exponent. To keep this under control, the mutation operator has been upgraded to allow punctual modifications of terminal exponents, as well as terminal names.

The second category of enhancements complies with the specific context of systems identification, by emphasizing the accuracy criterion over the parsimony one. Based on that remark, several techniques have been implemented to increase the priority of the *SEF* objective function within the context of dominance analysis. Right after generation, the initial population is split into two subpopulations. The first is evolved via a **S**ingle **O**bjective **O**ptimisation (SOO) procedure that only considers the accuracy objective measured by *SEF*. The second subpopulation undergoes a MOO procedure based on a customized dominance analysis. More precisely, the MOO subpopulation is divided into two groups (Fig. 3), depending on the *SEF* and *CF* values of the trees in it. The ones that feature lower objective values than the g_1 and g_2 thresholds, respectively, will be assigned fitness values based solely on *SEF*. The remaining individuals are divided into different order fronts and assigned a fitness value, as described in section 3.

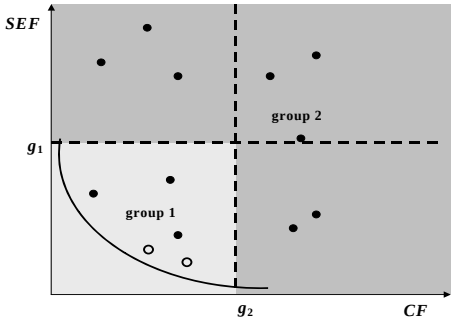

Fig. 3. Grouping within MOO subpopulation.

The two considered clustering parameters, g_1 and g_2 respectively, are computed in a dynamic manner, depending on the diversity of the current population. If the variance of the Euclidian distances between all the trees is low, that is interpreted as a sign that the individuals are close together in the objectives space. Therefore, g_1 and g_2 will be computed as the average *SEF* and *CF* values relative to the entire population. Should the variance of the Euclidian distances be high, then the trees are far apart in the objectives space. In this last case, the two thresholds cannot be defined in the same way as described above, as that would result in a biased clustering. Therefore, g_1 is assigned the mean *SEF* value relative to the two individuals furthest apart, while g_2 is computed in a similar way, considering the *CF* objective.

To increase the selection pressure in favour of the accuracy objective, once every *no_migr* generations a migration occurs, namely the SOO and MOO subpopulations exchange their best individuals. The number of trees that migrate from one subpopulation to the other is determined via a performance dependent adaptive threshold procedure. Three preset migration rates are considered, as follows: rate1 = 10%, rate2 = 20% and rate3 = 25%. If the average complexity of the SOO population is below that of the MOO population, then many of the simple and accurate trees evolved solely via *SEF* have a good chance of being included in the first group (Fig. 3). Therefore, the maximum allowed percentage (25%) of SOO trees will migrate to the MOO population, whilst only 10% will be sent the opposite direction. The second possibility is that of an average SOO complexity comparable to the MOO one, leading to a 20% migration rate in both directions. Finally, if the SOO trees are substantially more complex than the MOO ones, then their chances of being included in group 1 (Fig. 3) and of populating the interest region of the first order front are slim. Only 10% of the SOO trees will migrate towards the MOO population, while 25% simpler, yet less accurate trees will be sent the opposite direction.

5. APPLICATIONS

The algorithm's performances were tested within experimental trials targeting two multivariable systems: a five input one output linear system with dead time and an industrial subsystem from the **E**vaporation **S**tation (ES) of the sugar factory in Lublin, Poland.

Neglecting its dead time (equal to 1 sec), the linear system considered for preliminary verifications can be described by the following input-state-output representation:

$$\mathbf{A}=\begin{bmatrix}-1 & 1 & 1\\ -5 & -1 & 1\\ -2 & 0 & -1\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}1 & 1 & 0 & -1 & 1\\ 1 & 0 & -1 & 1 & 1\\ -1 & 2 & 1 & 1 & 0\end{bmatrix}$$
$$\mathbf{C}=\begin{bmatrix}1 & 2 & 1\end{bmatrix}\quad \mathbf{D}=\begin{bmatrix}0 & 0 & 0 & 0 & 0\end{bmatrix} \tag{9}$$

The training data set is built considering pulses of different magnitudes and widths, applied on both system inputs, subject to $T_S = 0.05$ sampling period. For validation, step response is considered. The discrete time model of the system, with a delay of 1 sec, and the specified sampling period T_s, results:

$$\begin{aligned} y(k) &= 0.080u_1(k-21) - 0.185u_1(k-22) + 0.103u_1(k-23)\\ &+ 0.138u_2(k-21) - 0.278u_2(k-22) + 0.138u_2(k-23)\\ &- 0.046u_3(k-21) + 0.092u_3(k-22) - 0.046u_3(k-23)\\ &+ 0.116u_4(k-21) - 0.184u_4(k-22) + 0.070u_4(k-23)\\ &+ 0.132u_5(k-21) - 0.277u_5(k-22) + 0.144u_5(k-23)\\ &+ 2.837\,y(k-1) - 2.699\,y(k-2) + 0.860\,y(k-3). \end{aligned} \tag{10}$$

The identification procedure ran over 30 generations, considering n_u = 25, n_y = 5 and activating migration once every *no_migr* = 5 generations, and was able to select an accurate and simple final model (**M**ean **R**elative **E**rror (*MRE*) = 0.5% on validation data set, 18 regressors).

$$\begin{aligned} y(k) &= 0.085u_1(k-21) - 0.179u_1(k-22) + 0.099u_1(k-23)\\ &+ 0.138u_2(k-21) - 0.270u_2(k-22) + 0.141u_2(k-23)\\ &- 0.045u_3(k-21) + 0.088u_3(k-22) - 0.043u_3(k-23)\\ &+ 0.121u_4(k-21) - 0.181u_4(k-22) + 0.065u_4(k-23)\\ &+ 0.130u_5(k-21) - 0.278u_5(k-22) + 0.140u_5(k-23)\\ &+ 2.830\,y(k-1) - 2.685\,y(k-2) + 0.859\,y(k-3) \end{aligned} \tag{11}$$

As the identification procedure was not provided with an *a priori* information as to the existence of dead time, the trees in the initial population each contained different combinations of all the terminals in the **x** vector (considering all lags from 1 to 25 for the five inputs, and from 1 to 5 for the output). Yet, as shown by (11), the algorithm was capable of gradually eliminating the unnecessary terminals, in an unsupervised way. Therefore, the method's capacity of automatically tuning model features that are not known pre-design is shown on this simple example.

Another issue that is easier to illustrate on this example, due to its simplicity, is the role of the adaptive threshold migration procedure within the evolutionary loop, outlined in Table 1. Here, *MRE* indicates the model performances on the training/validation data set. Usually, the average accuracy of the SOO subpopulation is, as expected, better than the one of the MOO counterpart. However, the migration from the SOO subgroup towards the MOO one is encouraged only if the average complexity is smaller (generation 5) or almost equal (generation 20). As a result, the final model is a valid trade-off between accuracy and parsimony.

Fig. 4 shows the generalization capabilities of the obtained model, specifically its capacity of accurately approximating the dead – time zone. Without any prior knowledge regarding the dead time and/or the system order, the algorithm was capable to determine the appropriate model structure and parameters, in a completely unsupervised manner.

Table 1. Migration scheme for time-delayed system

gen.	average *CF*		average *SEF*		migration rates	
	SOO	MOO	SOO	MOO	SOO to MOO	MOO to SOO
5	30	35.6	22.8	18.1	0.25	0.1
10	25.5	20.2	10.5	20.1	0.1	0.25
15	19	23.2	7.5	12.5	0.1	0.25
20	19	18.9	5.5	7.3	0.2	0.2
25	18.5	19	3.2	6.5	-	-

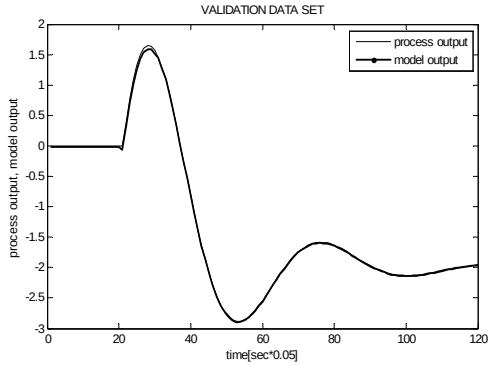

Fig. 4. Multivariable linear system – validation.

The other identification experiment involves a much more complex nonlinear system, with no available mathematical model. The **C**ontrol **V**alve (CV) subsystem of Lublin sugar factory has two inputs, namely the sucrose juice level in the first section of the ES and the pressure at the entry point in the ES. The system output is the juice flow sent back to the entry point in the ES. Training and validation data sets were acquired during different production shifts of plant exploitation (Deb, 2001).

The tests conducted on the control valve subsystem are aimed at assessing the enhanced MOO procedure's capacity of dealing with numerous input terminals. On the one hand, the existence of two inputs instead of just one implies an increased number of terminals to be considered in the tree-encrypted models, in other words, an increased number of regressors. On the other hand, imposing high values for the maximum input and output lags should compel the algorithm to produce even more complex individuals. However, experiments show (Table 2) that as the input and output lag values increase (M1→M3), the final model complexity and accuracy tend to stay approximately the same. This behavior is explained by the dual character of the tree population. Exchanging genetic material between SOO and MOO subpopulations brings slightly more weight to the accuracy objective, whilst keeping the complexity target in focus. Therefore, the exceedingly complex trees are gradually eliminated, leading towards final models which combine the advantages of both SOO and MOO procedures, namely high accuracy, as well as reduced complexity. It is not recommended to increase the maximum lags over a certain limit (M4), as this may increase the risk of producing overfitted models which, in most cases, feature poor generalization capabilities.

A way of providing the enhanced MOO procedure with enough time and genetic material to yield viable models is to increase the maximum number of generations as well as the number of trees per generation. Doing so results in improved model accuracy and parsimony (M5→M7). Though, if these parameters are increased excessively, a saturation phenomenon may emerge, as the supplementary genetic material becomes redundant (M7 and M8). The experimental data to support the conclusions above is included in Table 2. All the results above have been obtained using 290 data points for both training and validation. The accuracy of model M5 on the validation data set is shown in Fig. 5

Table 2**.** Enhanced MOO models for CV subsystem

model ID	individuals/ generations	input lag/ output lag	*MRE* (%)	
			training	validation
M1	30/30	$n_u = 2, n_y = 2$	0.23	1.32
M2	30/30	$n_u = 3, n_y = 2$	0.31	1.20
M3	30/30	$n_u = 5, n_y = 3$	0.27	1.21
M4	30/30	$n_u = 10, n_y = 9$	0.20	5.01
M5	50/50	$n_u = 2, n_y = 2$	0.21	1.30
M6	70/70	$n_u = 2, n_y = 2$	0.15	1.25
M7	70/90	$n_u = 2, n_y = 2$	0.09	1.22
M8	100/150	$n_u = 2, n_y = 2$	0.11	1.21

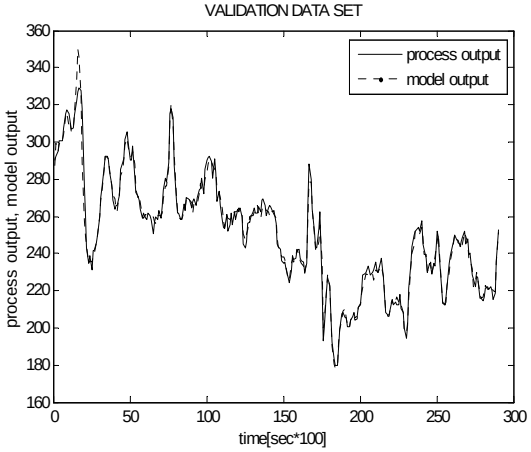

Fig. 5. Control valve model – validation data set.

6. CONCLUSIONS

The evolutionary identification tool proposed by the authors is specifically tailored to provide models for multivariable nonlinear systems. The algorithm performs an unsupervised selection of accurate and simple models, thus being able to reduce the risk of overfitting or compensation phenomena.

The linear in parameter formalism adopted by the generated models facilitates the use of QR decomposition for faster parameter computation. To ensure enhanced exploration capabilities in producing compact model structures, customized crossover and mutation are suggested. Consequently, the simultaneous improvement of the model structure and parameters takes up less computational resources.

Employing a conjoint SOO and MOO evolution scheme fits the specific requirements of systems identification by

increasing the weight of the accuracy objective without completely ignoring the importance of the parsimony one. Specifically, this goal is achieved by employing a dynamic clustering technique based on population diversity assessment via probabilistic indicators, an approach that encourages the production of compact yet flexible models, featuring good performances on both training and validation data sets.

Although time and resource consuming, the algorithm is suitable for complex data driven identification problems that involve heightened accuracy requirements or when no rich *a priori* system information is accessible.

REFERENCES

Back T., D. Fogel, Z. Michalewicz (2000). *Evolutionary Computation 2. Advanced Algorithms and Operators*. US: Institute of Physics Publishing.

Coello Coello C.A., G.B. Lamont, D.A. Van Veldhuizen (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*, second edition, Springer.

Deb K. (2001). *Multi - Objective Optimization using Evolutionary Algorithms*, Wiley, USA.

Flemming P. J., R. C. Purshouse (2002). "Evolutionary Algorithms in Control Systems Engieering: A Survey", *Control Eng. Practice*, 10, 1223 – 1241.

Koza J. R. (1992). *Genetic Programming – On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: MIT Press, 73-190.

Madar J., J. Abonyi, F. Szeifert (2005). *Genetic Programming for System Identification* [Online]. Available: http:// www.fmt.vein.Hu/ softcomp/ isda04_gpolsnew.pdf.

Rodriguez-Vasquez K., C. M. Fonseca, P. J. Flemming (2004). Identifying the Structure of Nonlinear Dynamic Systems Using Multiobjective Genetic Programming. *IEEE Trans. on Syst. Man and Cybernetics, Part A – Systems and Humans*, 34, 531-534.

Wey H., S. A. Billings, J. Lui (2004). Term and Variable Selection for Nonlinear Models, *Int. J. Control*, 77, 86–110.