AMEBA-Evolutionary Computation Method: Comparison and Toolbox Development

Marko Corn*, Maja Atanasijević-Kunc

Faculty of Electrical Engineering, University of Ljubljana, Tržaška cesta 25, 1000 Ljubljana, Slovenia; marko.corn@ameba.si*

Simulation Notes Europe SNE 26(4), 2016, 229 – 236
DOI: 10.11128/sne.26.tn10354
Received: November 15, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. Evolution algorithms are optimization methods that mimic a process of the natural evolution. Their stochastic properties result in a huge advantage over other optimization methods, especially regarding solving complex optimization problems. In this paper, several types of evolutionary algorithms are tested regarding a dynamic nonlinear multivariable system modelling and control design. We have defined three problems: the first one is the so-called grey box identification problem where the characteristic of the system's valve is under investigation, the second one is a black box identification where the goal is a dynamic system's model development using system's measurements data, while the third one is a system's controller design. The efficacy of solving presented problems was compared to the usage of the following optimization methods: genetic algorithms, differential evolution, evolutionary strategies, genetic programming, and a developed approach called AMEBA algorithm. All methods have proven to be very useful for grey box identification and design of a system's controller, but AMEBA algorithm has also been successfully used in a black box identification, where it generated a corresponding dynamic mathematical model.

Introduction

In general, the evolutionary algorithms can be divided into two major groups: parametrical and structural algorithms. Parametrical algorithms evolve parameters, while structural algorithms evolve structures or mapping functions. For example, if we would have to design a controller for a dynamic system, parametric algorithm would demand to define parameters of the chosen controller structure (very frequently a PID controller is used).

In contrast to parametrical algorithms, structural algorithms do not require predefined form of the controller, as they can evolve the whole controller through their evolutionary process.

The most popular parametrical algorithms are genetic algorithms (GA) [1][2], evolutionary strategies (ES) [3], differential evolution (DE) [4] and others [5].

Most established structural algorithm is genetic programing (GP) that has multiple implementations from the three-based implementation [6] to the grammatically based implementation [7] and the evolutionary programming that is directed into the evolvement of finite state machines [8].

Evolutionary algorithms can be used also in the complex field of the design of controllers of dynamic systems, e.g. multivariable, non-linear, time-variant [9].

In this paper, the evolution of different models and control strategies are designed and compared with the usage of different evolutionary algorithms. From the parametrical group the efficacy of GA, ES and DE is illustrated, while from the structural group an algorithm of tree based genetic programming and the Agent Modelled Evolutionary Based Algorithm (AMEBA) are used [10],[11]. Relative advantages and disadvantages have been estimated regarding modelling and control design of non-linear multivariable dynamic system of three coupled thanks.

The paper is organized in the following way. In the first section a short description of the three coupled tanks system is given. In the second section a structure of the system's model and the corresponding controller are specified. In the third and fourth sections the modelling and the control design results which were generated using different evolutionary algorithms are presented and compared. The result section is followed by the description of AMEBA system toolbox that was used to generate the results of AMEBA method [12]. At the end, the conclusions and some ideas for the future work are given.

1 Three Coupled Tanks System

System of three coupled thanks is illustrated in Figure **1**. It consists of three identical cylindrical assembled water tanks with cross area S, which are interconnected with the pipes and two valves V_1 and V_2, while the valve at the output pipe is V_3. Actuators of this system are two water pumps that supply the first and the third tank with water flows $\Phi_{vh1}(t)$ and $\Phi_{vh2}(t)$. Water levels in each tank $h_1(t)$, $h_2(t)$, and $h_3(t)$ are measured with the corresponding sensors. Level difference between the first and the second tank generates water flow $\Phi_3(t)$ through the valve V_1 and level difference between the second and the third tank generates flow $\Phi_4(t)$ through the valve V_2. The output flow $\Phi_{izh}(t)$ depends only on the water level $h_3(t)$ and valve V_3 properties.

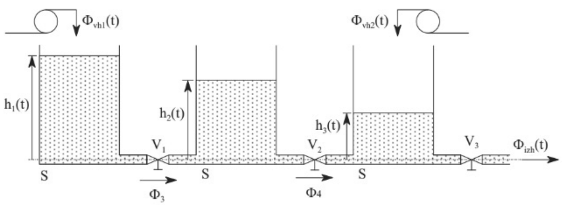

Figure 1: System of three coupled tanks.

System of three coupled thanks represents a laboratory device but for the testing we have used its model [13].

1.1 Model structure

During the phase of designing a model of certain dynamic system it is usually desired to include as much knowledge of the system as possible. In such a way, we have more chances of building a suitable model. Theoretical modelling approach enables model building on the basis of the equilibrium equations which determine system's basic behaviour. For further model improvement, additional nonlinear functions are needed which describe different specific parts of the system. In the first phase the system's model can be presented with three equilibrium equations which are described with equations (1).

$$\begin{aligned} \Phi_{vh\,1}(t) - \Phi_3(t) &= S \cdot \dot{h}_1(t) \\ \Phi_3(t) - \Phi_4(t) &= S \cdot \dot{h}_2(t) \\ \Phi_4(t) + \Phi_{vh\,2}(t) - \Phi_{izh}(t) &= S \cdot \dot{h}_3(t) \end{aligned} \tag{1}$$

Input flow rates are determined by the water pumps which are controlled with the voltage signals u_1 and u_2. Water flows from the first to the second tank and from the second to the third tank are given with the equations (2).

$$\begin{aligned} \Phi_3(t) &= k_1\sqrt{h_1(t) - h_2(t)} \\ \Phi_4(t) &= k_2\sqrt{h_2(t) - h_3(t)} \end{aligned} \tag{2}$$

These water flows depend on the water levels in the tanks and the characteristics of the valves. These characteristics are expected to be of the square root type. From the experimental data it was established that static characteristic of the valve V_3 is not square root function and so we have tried to estimate corresponding description by the so-called indirect identification method or 'grey box identification' [14]. Grey box identification is a process in which we firstly gather measurements of the system's behaviour, secondly we build a mathematical model and include all the data that we have into it. Thirdly we try to estimate the missing parameters or functions to the constructed model. Block diagram of the chosen structure is illustrated in Figure 2.

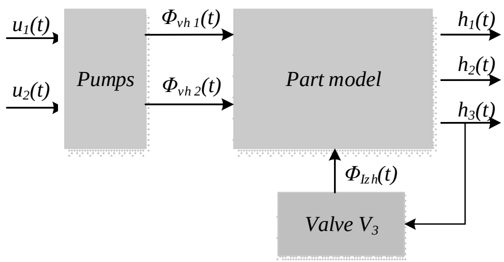

Figure 2: Block diagram of the three coupled tanks system structure.

Estimation of the characteristic of the valve V_3 is defined optimization problem as the rest of the model had been constructed using the equilibrium equations and measured characteristics of the other parts. Optimization process was minimizing the difference between responses of the model and measurements of the system by adapting valve's characteristic. The fitness function used in this optimization process is presented by equation (3).

$$\mathrm{J} = \sum_{i=1}^{3} \int |h_i(t) - h_i^*(t)| dt \tag{3}$$

Fitness function is equal to the absolute sum of difference between responses of the model and corresponding measurements.

Measurements obtained for the identification process consist of eight responses to the different input or excitation signals. Six of them were used in the identification process and two for the validation of the model. One pair of the excitation signals and corresponding responses is illustrated in Figure 3 and Figure 4.

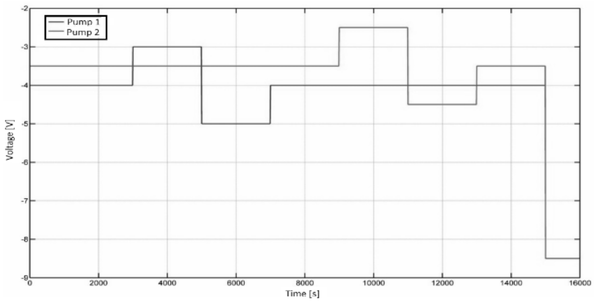

Figure 3: Input signals $u_1(t)$ and $u_2(t)$.

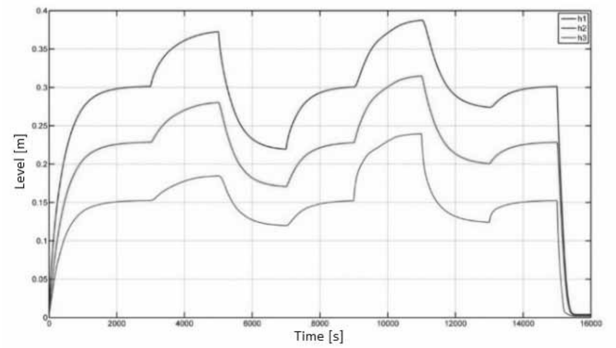

Figure 4: Responses of the system to chosen input signals.

From the presented responses, the cross couplings are visible (each input influences both systems' outputs $h_1(t)$ and $h_2(t)$). These cross couplings also prove that the system is a multivariable one.

1.2 Controller design

Block diagram of system's close loop operation is presented in Figure 5.

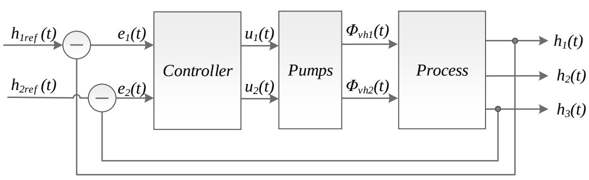

Figure 5: Closed-loop system operation

Close loop system should maintain water levels in the first and in the third tank at the corresponding reference values h_{ref1} and h_{ref2}. Fitness function that is used in the optimization process of the controller design is presented in equation (4).

$$J = w_{opt} \int |e_1(t)| + |e_2(t)| dt + (1 - w_{opt}) \int |u_1(t)| + |u_2(t)| dt \quad (4)$$

Fitness function represents a sum of the integrals of errors e_1 and e_2 (that represents difference between actual water levels h_1 and h_3 and referenced values h_{ref1} and h_{ref2}) and integrals of the pumps activity u_1 and u_2. Both contributions are weighted with the weight w_{opt}. The control system was tested with the usage of the reference signals that are presented in Figure 6.

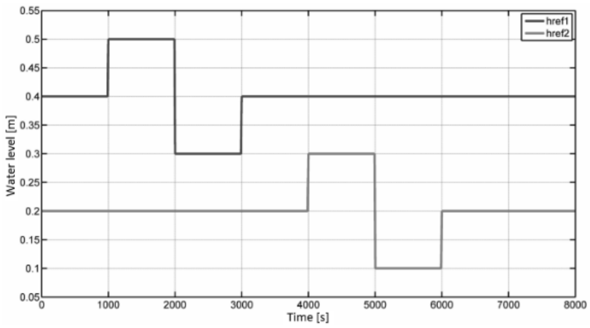

Figure 6: Reference signals.

Controller must be able to control the systems water levels in a way that is demanded by the step shaped changes of the reference signals.

2 Modelling Results

Modelling results are divided into two groups. The first group consists of the results obtained by the parametrical evolutionary algorithms and the second group by the structural evolutionary algorithms.

2.1 Parametrical evolutionary algorithms

Parametrical evolutionary algorithms can optimize only parameters, so we have constructed a polynomial mathematical function with four parameters a_1, a_2, a_3, and a_4 which should describe as good as possible the relation between the water level $h_3(t)$ and output water flow $\Phi_{izh}(t)$.

$$\Phi_{izh}(t) = a_1 h_3^3(t) + a_2 h_3^2(t) + a_3 h_3(t) + a_4 \quad (5)$$

We have tested and compared three parametrical methods GA, ES, DE. Optimization process was defined for all methods identically in order to get comparable results. Solutions have been evolved during 1000 generations and with the generation size of 30 individuals. Results are presented in two ways. The first way is the comparison of the quality of the model that was generated by each method and the second is the comparison of the convergence of the used methods. Quality of generated solutions is presented in Table 1.

Met.	Error identification [%]	Error validation [%]
DE	1.77	3.27
ES	1.79	3.58
GA	1.88	4.57

Table 1: Evaluation of modelling results of parametrical algorithms.

Error column represents a relative average deviation from the identification signals of the system and validation column represents relative average deviation from the validation signals. All results are quite similar, which means that there is high probability that we have found a global minimum of the proposed valve function. Best algorithms are DE and ES that have managed to generate 1% better result. Example of the system's responses of the best model generated by the DE method is presented in Figure 7.

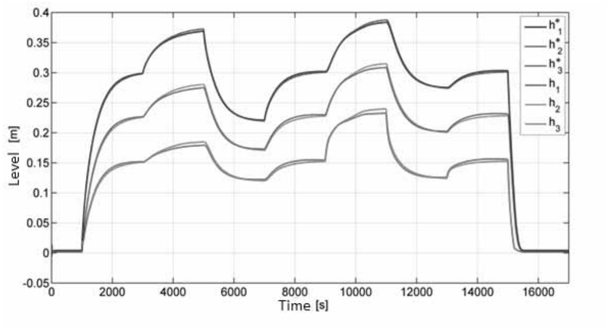

Figure 7: Comparison of measurements with the response of the model generated by the DE method.

We have compared also the convergence of the algorithms and the results which represent the average convergence of 10 optimization runs for each method are presented in Figure 8.

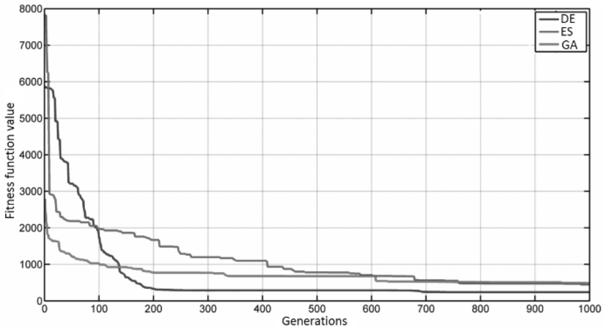

Figure 8: Average convergence of parametrical methods

Statistical analysis of the methods' convergences shows efficiency of each algorithm during the search of optimal solution. DE has the fastest convergence and it generates the best results.

2.2 Structural evolutionary algorithms

In addition to parametrical optimization also two structural algorithms, namely GP method based on trees and AMEBA were tested. For the AMEBA algorithm additional test has been conducted. Test, where the model of the whole system has been built (not just model of the valve V_3) with the black box identification method as the AMEBA algorithm can be used also for multi-input multi-output systems.

Structural algorithms are capable of building system's structure automatically. Settings of the evolution were the same for both methods which enable the comparison of the results. For the GP, we have used addition, subtraction, multiplication, division, power and constant types of nodes and for the AMEBA algorithm we have used the same nodes' types as for the GP with the use of additional dynamic nodes like delay, integral, derivative, low pass filter and high pass filter. Results are evaluated in Table 2.

Algorithm	Error ident. [%]	Error valid. [%]
GP	1.62	3.12
AMEBA valve	3.57	4.65
AMEBA full model	5.63	7.23

Table 2: Evaluation of modelling results when using structural algorithms.

GP algorithm has generated the best solution and its tree representation is presented in Figure 9.

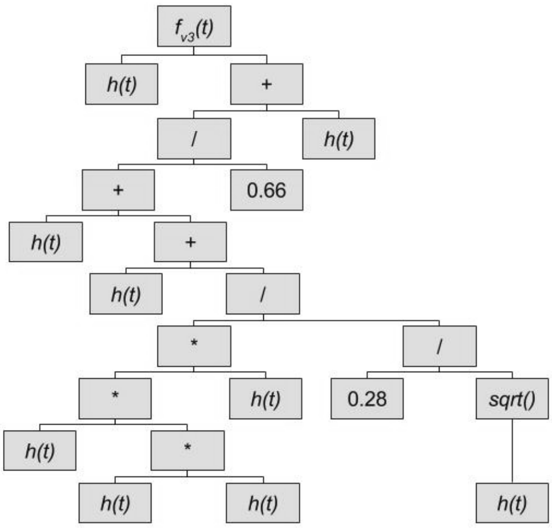

Figure 9: Solution generated with the GP method.

Simplified solution of GP is presented in equation (6). This is a polynomial function with two parts, the first has rational number in the exponent and the other is a linear one.

$$f_{V3}(t) = 2.086\, h(t)^{\frac{9}{2}} + 5.023\, h(t) \tag{6}$$

Result generated by the AMEBA algorithm is not as good as the result obtained by GP and it is presented in Figure 10.

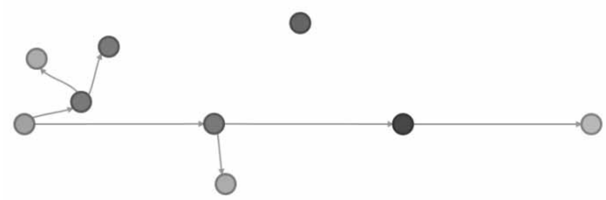

Figure 10: Graph representation of model of the valve generated with AMEBA algorithm.

In Table 3 a legend is presented that shows colours of different types of nodes assembling AMEBA algorithm solutions.

Color	Node	Color	Node
	Input		Amplification
	Output		Exponent
	Low pass filter		Delay
	High pass filter		Derivative
	Multiply		Integral
	Divide		Add

Table 3: Color-legend of different types of nodes.

Valve function that was generated by the AMEBA algorithm is presented in equation (7).

$$\Phi_{izh}(t) = -0.5 \cdot 0.54(-0.8(h_3(t)))^{0.68} \tag{7}$$

The result of the valve function generated with the AMEBA algorithm is a nonlinear function. AMEBA algorithm has successfully generated also a model of the whole system with the process of black box identification. We have used the same measurements for generating this model that were in use for the identification of the valve. Model is represented in Figure 11. Model generated with AMEBA algorithm is complex, full of nodes of all types and feedback loops that represent system dynamic properties.

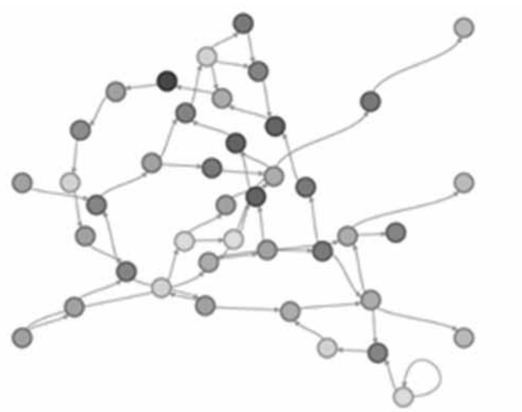

Figure 11: Graph representation of system's model generated with the use of AMEBA algorithm.

3 Results of the Controller Design

Results of designing control algorithm are also divided into two groups: into a parametrical and a structural group.

3.1 Parametrical evolutionary algorithms

Parametric methods usage demands a parametrically defined problem so we constructed a controller that is assembled with four proportional-integral (PI) controllers with 8 parameters to be optimized.

The proposed controller is a multivariable one with two inputs (differences between desired and actual water levels) and two outputs to drive water pumps. Controller's parameters to be optimized are described with equations (8).

$$\vec{u}(t) = \boldsymbol{K_p}\vec{e}(t) + \boldsymbol{K_i}\int \vec{e}(t)dt$$

$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} K_{p11} & K_{p12} \\ K_{p21} & K_{p22} \end{bmatrix}\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} + \begin{bmatrix} K_{i11} & K_{i12} \\ K_{i21} & K_{i22} \end{bmatrix}\int \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} dt \quad (8)$$

$$\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} = \begin{bmatrix} h_{1\,ref}(t) - h_1(t) \\ h_{3\,ref}(t) - h_3(t) \end{bmatrix}$$

All 8 parameters are represented in two matrices $\boldsymbol{K_p}$ and $\boldsymbol{K_i}$. Results calculated with the parametrical methods are presented in Table 4.

Algorithm	Error	Energy used
DE	2.04 %	35.9%
GA	2.04 %	36.5%
ES	2.48 %	35.3%

Table 4: Evaluation of controller optimization results calculated with parametrical methods.

Results of all algorithms are very similar but the DE method has ones again proven to be the best as it calculated the controller with the lowest error and minimum estimated usage of energy.

3.2 Structural evolutionary algorithm

Structural evolutionary algorithms don't need the controller's structure to be defined in advance in contrast to parametrical methods. This group is capable to evolve the structure as well as all the parameters automatically. Results of two methods, GP an AMEBA, are presented in Table 5.

Algorithm	Error	Energy used
AMEBA	1.5 %	34.1 %
GP	9.3 %	35.5 %

Table 5: Results of controllers generated by structural evolutionary methods.

The solution which was generated by the GP method is presented by equation (9).

$$u_1(t) = e_1(t)^{2.2\, e_2(t)-1} + e_1(t) \qquad u_2(t) = u_1(t) \quad (9)$$

GP method didn't generate a suitable solution as the controller is not capable to follow corresponding reference signals. The solution generated by the AMEBA algorithm is presented in Figure 12.

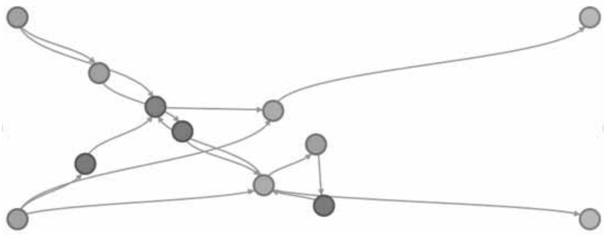

Figure 12: Graph representation of controller generated by the AMEBA algorithm.

Controller that was generated by AMEBA algorithm is illustrated by equation (10). AMEBA algorithm generated a controller with the best performance.

$$\begin{gathered} u_1(k) = X_2(k) \\ u_2(k) = 0,74(+e_1 + 0,11(X_0(k) * X_2(k) * 0,95(-0,87)(e_2))) \\ X_0(k) = e_2 \\ X_1(k) = 0,34 fltLP(X_0(k)) \\ X_2(k) = -0,2(-e_1 + 0,40 fltLP(X_2(k-1)) + 0,08X_1(k-1)) \end{gathered} \quad (10)$$

4 Toolbox development

AMEBA algorithm is being developed also as a software package with user friendly graphical interface. The core development is being built in Java programming environment that can be used also with Matlab, which allows a very efficient support in simulation of dynamic systems via Simulink. Graphical interface is also developed in Matlab due to its good graphical support.

Toolbox enables settings of the simulation environment with the inclusion of Simulink model as it is shown in Figure 13.

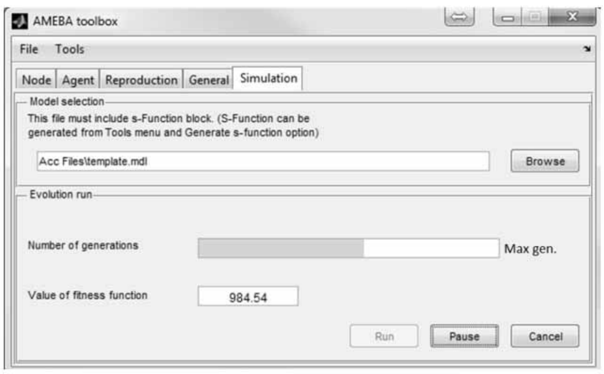

Figure 13: Settings of simulation environment.

The agent of AMEBA algorithm is implemented as S-function so it can be included into the model as a standard block. Toolbox enables control and monitoring of the optimization process where it displays current generation number and the value of the fitness function of the best agent.

Toolbox enables settings of population properties like size of population, size of reproductive population that determines how many best agents will be given opportunity to reproduce, number of elite agents, and other settings that determine the end of optimization process like maximum number of generations and minimum change in fitness function value (Figure 14).

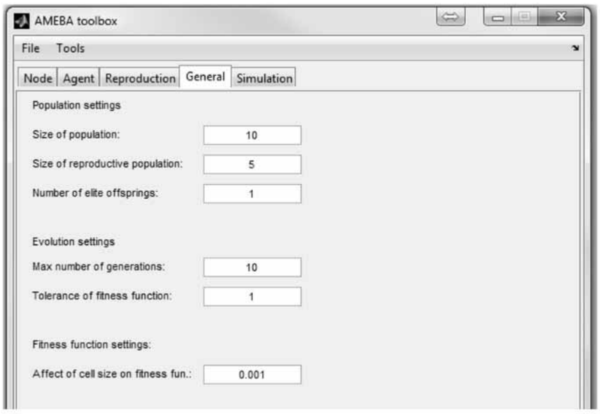

Figure 14: General setting.

The number of inputs and outputs of an agent can be defined together with the maximum number of nodes that can be generated at the agent's creation (Figure 15).

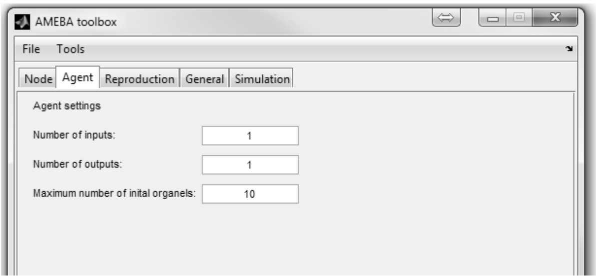

Figure 15: Agent settings.

Different types of nodes can be selected from which the algorithm will chose and build agents. Each node has its own settings that determine initial value of the nodes parameter and steepness of change in case node mutates (Figure 16).

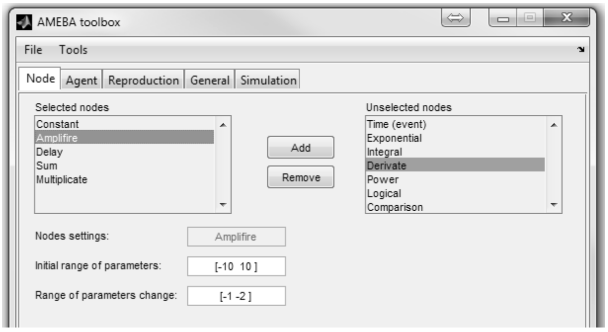

Figure 16: Node settings.

Reproduction mechanisms can be set with their parameter of probability. As agents are evaluated and selected for reproduction the reproduction mechanism is randomly selected and the probability parameter determines their possibility of being selected (Figure 17).

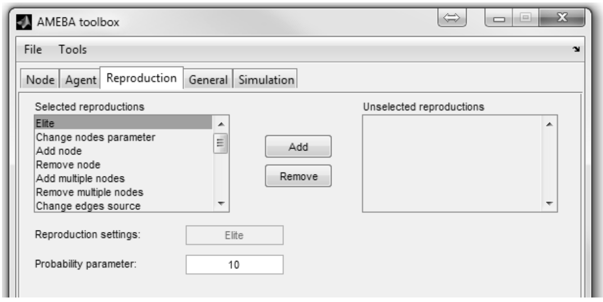

Figure 17: Reproduction settings.

Additional functionalities enable better usability of the method such as saving and importing of all setting into file for later use. With this option, also the initial population can be imported which enables the inclusion of certain knowledge of the solution into the optimization problem. It is also possible to convert agent into mathematical equation to observe its structure. It can also generate Matlab S-function file for the easier implementation in Simulink (Figure 18).

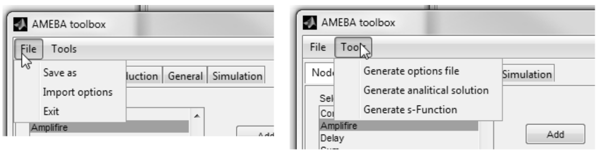

Figure 18: Additions functionalities of Toolbox.

5 Conclusions

The system of three coupled thanks was selected to present the efficacy of three different approaches of the usage of the evolutionary algorithms methods: the grey box identification, the black box identification and the controller design.

Parametrical evolutionary algorithms generated good results for both modelling and control of the system. Also, structural methods manage to generate good solutions for both types of problems. In general, the most important advantage of the structural algorithms in comparison to the parametrical methods is the absence of the need to define a suitable structure. This property is especially important when dealing with more complex systems with multiple inputs and outputs. With the usage of AMEBA algorithm, we have managed to generate also a complete model of the system and we generated a system controller with the best performance.

Future work on AMEBA algorithm development will be focused on optimization process as we are going to explore the impact of various effects on the quality of the solution and on the convergence rate of optimization process like the effect of size of the population size, suppression of the agents with large number of nodes, using multiple environments at once and similar, of course in comparison with other optimization approaches. Special attention will be devoted to the so called smart optimization where additional knowledge from chosen area can be taken into account to improve searching efficacy. The AMEBA method is a work in progress and the method will be available as an open source project.

References

[1] Atanasijević-Kunc M, Belič A, Karba R. *Optimal multivariable control design using genetic algorithms.* Vienna University of Technology; 5th Vienna Symposium on Mathematical Modeling. 2006.

[2] Goldberg D. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley; 1st1989.

[3] Beyer HG. *The Theory of Evolution Strategies (Natural Computing Series)*. Springer; 2010. 400 p.

[4] Storn R, Price K. *Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*. Journal of Global Optimization. 1997.

[5] Brownlee J. *Clever Algorithms: Nature-Inspired Programming Recipes*. Swinburne University in Melbourne, Australia; Search. 2011.

[6] Koza JR. *Genetic Programming On the Programming of Computers by Means of Natural Selection*. MIT Press; 6th1992.

[7] Whigham PA. *Grammatically-based Genetic Programming*. Departent of Computer Science, University College, University of New South Vales; Workshop on Genetic Programming: From Theory to Real-World Applications. 1992.

[8] Fogel LJ, Owens AJ, Walsh MJ. *Artificial Intelligence through Simulated Evolution*. John Wiley; 1966.

[9] Tomažič S, Logar V, Kristl Ž, Krainer A, Škrjanc I, Košir M. *Indoor-environment simulator for control design purposes*. Building and Environment. 2013.

[10] Corn M, Atanasijević-Kunc M. *Cell based Genetic Programming Toolbox (CGP - Toolbox)*. 20th ERK Conference. 2011.

[11] Corn M, Černe G. *A Graph-Based Evolutionary Algorithm : Cell Based Genetic Programming*. Proceedings of the 5th International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2012. 2012.

[12] Corn M. *AMEBA – Structural evolutionary optimization: method and toolbox development.* 27th European Modeling and Simulation Symposium. 2015.

[13] Atanasijević-Kunc M. *Multivariabilni sistemi, Zbirke kompleksnejših problemov*. Založba FE in FRI; 42005. 265 p.

[14] Tan KC, Li Y. *Grey-box model identification via evolutionary computing*. Control Engineering Practice. 2002.