

Box and Jenkins non linear system Modelization using RBF Neural Network designed by NSGAI

K. LAMAMRA¹, K. BELARBI², S. BOUKHTINI²

¹ University of Oum El Bouaghi, Algeria

² University of Constantine1 - Algeria

l_kheir@yahoo.fr; kbelarbi@yahoo.com; sou_boukh@yahoo.fr

Abstract. The modeling process is to find a parametric model whose dynamic behavior close to that process. This model will be used to make predictions of the process output, or to simulate the process in a control system...etc. In this work we used RBF neural networks for modeling nonlinear systems. Generally the problem in neural networks is often to find a better structure. We propose in this work a method based on No Sorting Genetic Algorithm II (NSGA II) to determine the best parameters of a RBF NN. The NSGAI should provide the best connection weights between the hidden layer and output layer, find the parameters of the radial function of neurons in the hidden layer and the optimal number of neurons in the hidden layers and thus ensure learning necessary. Two functions are optimized by NSGAI: the number of neuron in the hidden layer of the RBF NN, and the error which is the difference between desired input and the output of the RBF NN. This method is applied to modeling Box and Jenkins system that is widely used. The obtained results are very satisfactory.

Keywords: NSGA II, RBF Neural Networks, modelization.

1 Introduction

Among the advantages of a NN is its ability to adapt to the conditions imposed by any environment, and ease to change its parameters (weight, number of neurons, etc ...) depending on the behavior of its environment. The NN are used to model and control dynamic systems linear and nonlinear where conventional methods fail. Generally learning neural networks can be made in three ways: supervised learning, we have a set of examples (input-output pairs) and we must learn to give the correct output of new inputs. The reinforcement learning: we have inputs describing a situation and we receive a punishment (or error) if we

give out is not adequate. [1]. The identification of the parameters of NN is often performed by the algorithms of back-propagation, based on minimizing the training error and the chaining rule. This algorithm is a gradient descent on a differentiable error function calculated on the output nodes to approximate a mathematical function on the inputs. This algorithm showed several disadvantages such as slow convergence, sensitivity to local minima and the difficulty to adjust the learning parameters (the number of neurons in the hidden layers, learning step etc). In some networks using Hebbian learning where the synaptic weights can be adjusted during a learning phase patterns through Hebb's formula leads to a formula expressing the weights based on grounds recognized [2].

Several approaches have been proposed to improve the method of back-propagation as the modification of learning step, decentralization of learning step algorithms using quasi-Newton [3], genetic algorithms [4] ... etc..

In this work we propose the use of NSGA II to construct a model using a RBF NN with optimal structure. In this approach the NSGA II is used to optimize the number of neurons in the hidden layer of NN, find the best connection weights between the hidden layer and output layer, find the parameters of the radial function of neurons in the hidden layer and ensure learning of NN. This paper is organized as follows: in the second section we briefly recall the basic principles of NN, in the third section we present the NSGA II algorithm, its operating principle and its application in our method, and the fourth section we present the learning of RBF NN by the NSGAI, and finally we present the simulation results of the developed method in the fifth section.

2 Neural Networks "NN"

A NN is a computational model whose design is inspired schematically the functioning of real neurons. Artificial neural networks are increasingly used and applied in various fields [5-7]. This concept, as well as genetic algorithms, is linked to the notion of learning that allow computers to learn by example, experience or analogy, forming the foundation for adaptive systems. An artificial NN is generally composed of a succession of layers, each of which takes its inputs to the outputs of the previous one. Each layer is composed of neurons, and each synapse is associated a synaptic weight. A NN has the ability to adapt to the conditions imposed by any environment, and easy replacement when a there are a change of the parameters of this environment, which allows him to solve problems previously qualified as difficult. [8-10].

2.1. Radial basis function RBF NN

Radial basis functions are powerful techniques for interpolation in multidimensional space. Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. A RBF is a function which has built into a distance criterion with respect to a center. Radial basis functions have been applied in the area of neural networks where they may be used as a replacement for the sigmoidal hidden layer transfer characteristic in multi-layer Perceptron. RBF networks have two layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. The RBF chosen is usually a Gaussian. In regression problems the output layer is then a linear combination of hidden layer values representing mean predicted output. In classification problems the output layer is typically a sigmoid function of a linear combination of hidden layer values, representing a posterior probability.

RBF networks have the advantage of not suffering from local minima in the same way as Multi-Layer Perceptron. Generally this is because the only parameters that are adjusted in the learning process are the linear mapping from hidden layer to output layer. RBF networks have the disadvantage of requiring good coverage of the input space by radial basis functions. RBF centers are determined with reference to the distribution of the input data, but without reference to the prediction task. As a result, representational resources may be wasted on areas of the input space that are irrelevant to the learning task. In this work, it is to the NSGAI algorithm to find the best RBF centers.

Currently, the RBF-NN are used in many works for different tasks, for example in network traffic identification [11], in a hybrid multi-logistic methodology, called logistic regression [12], in classification problems [13] as a nonlinear controller [14], hyper-chaotic systems control [15], in energy-based controller [16] in Adaptive Control [17] in identification of non-linear systems [18,19] in optimization [20] in modeling for nonlinear and non-stationary systems [21].

2.2. Learning of a NN:

Learning is to determine the weights for the output of the NN to be as close as possible to the target. The main problem is how to build an NN, how many layers of covers and the number of units (neurons) in hidden layer required to achieve a good approximation. Since a wrong choice can lead to poor network performance matching [22]. The first attempts to solve the problem of determining the architecture have been to test several networks with different architectures to achieve the desired performance [23].

In recent years, many studies have been devoted to developing methods for optimizing the architecture of the NN. The main algorithms have been proposed can be classified into three families:

1. Pruning algorithms: detect and remove the weights or units that contribute little to the network performance [24].
2. Ascending or constructive algorithms: start from an approximate solution to the problem with a simple network and add if necessary unit or hidden layers to improve network performance [25].
3. The direct algorithms: define a suitable architecture and perform learning or perform both operations simultaneously, such as genetic algorithms [26].

3. Non-dominated Sorting Genetic Algorithm II: NSGA II

The Multi Objective Genetic Algorithm we used in this work is the NSGAII (Non-dominated Sorting Genetic Algorithm) introduced and enhanced by Deb et al. [27, 28]. It is one of the most used and most cited in the literature algorithms. It is widely used by many authors, not only in the context of multi-objective optimization, but also for comparison with other algorithms, it is considered as a benchmark by several researchers [29-34].

NSGAII is an algorithm establishing the dominance relationships between individuals and providing a fast sorting method of chromosomes. This algorithm uses a measure of crowding around individuals to ensure diversity in the population. The principle of this algorithm is shown in Figure 1.

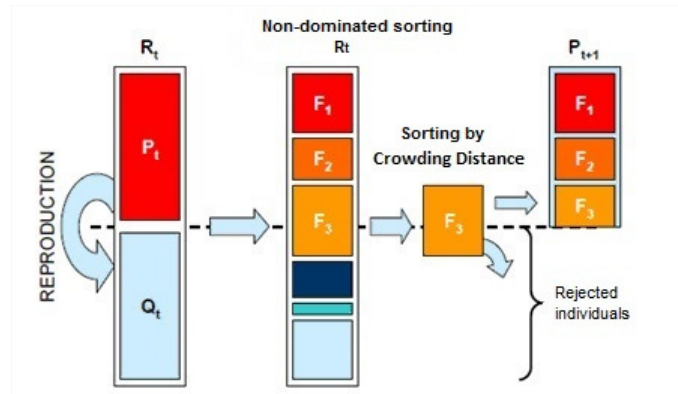


Fig 1. Operating principle of NSGAII

At the beginning, an initial population is randomly generated, and then it undergoes a sorting using the concept of non-domination. Each solution is assigned a strength or rank equal to the level of non-dominance (1 for best, 2 for the next level, etc ...). The reproduction step consists of a tournament for the selection of parents. When two individuals of the population are chosen randomly in the popu-

lation, the tournament is based on a comparison of the domination with constraints of the two individuals. For a given generation t , we create $R_t = P_t \cup Q_t$, Q_t is children population of the previous population P_t (generated from the parents through the operators of crossover and mutation), R_t includes individuals of P_t , which ensures the elite nature of the algorithm NSGAI. Population R_t contains $2N$ individuals (it is composed of N parents and N children). Then R_t undergoes a sorting using the concept of non-dominance of Pareto. Individuals are grouped into non-dominated fronts such as F_1 represents individuals of rank 1, F_2 individuals of rank 2, etc ... The next objective is to reduce the number of individuals in the $2N$ population R_t for a population P_{t+1} of size N . If the size of F_1 is less than N , then all F_1 individuals are retained. It is the same for the other fronts as long as the number of individuals retained does not exceed the size N . If we take the example of Figure 1, the fronts F_1 and F_2 are fully retained but the conservation front F_3 will result in exceeding the size N of the population P_{t+1} . It must then make a selection of F_3 individuals to keep. It is then necessary to make a selection of F_3 individuals to maintain. In this case, NSGAI involves a mechanism for preserving the diversity of the population based on the evaluation of the density of individuals around each solution through a procedure for calculating the "distance proximity". A low value of the proximity distance for an individual is an individual "well surrounded". It then proceeds to a descending sorting according to this distance proximity to retain individuals F_3 front and eliminate individuals from the densest areas. This way we complete the population P_{t+1} . Individuals with extreme values for the criteria are preserved by this mechanism, thereby maintaining the external terminals of the Pareto front.

At the end of this phase, the population P_{t+1} is created. Then a new population Q_{t+1} is generated by reproduction from P_{t+1} . We continue iteratively the procedure described above to the satisfaction of stop criteria set by the user. Generally, the NSGAI keeps elitism and diversity without adding additional parameters, while using an algorithm attractive in its simplicity with a minimum of parameters.

4. Learning of RBF NN by the NSGAI:

The NSGA II is used to optimize the structure and parameters of the RBF NN. The following two objective functions are chosen:

- The first function to be optimized (f_1) is the number of neuron of the hidden layer of the RBF NN.
- The second function (f_2) is the quadratic error which is the difference between the desired input of the RBF NN and its output.

For that NSGAI must find the best number of neurons in the hidden layer (N_n) and provide the best connection weights between neurons in the hidden layer and the output layer, and also find the parameters of the radial function of neurons hidden layer. In this work we used the radial functions of Gaussian form, and NSGA II must find the best centers (C_i) and the best widths sigma (Sig_i) for these

functions. The chromosome then contains the number of neurons in the hidden layer, Gaussian functions centers and widths of the hidden layer neurons, and the weights of connections between the hidden layer and the output layer.

5. Results of simulation

This method is applied to modeling the BOX and JENKINS system. This process considered is a gas-fired boiler with the input the gas at the inlet and the output the concentration of released CO₂. The data of BOX and JENKINS consist of 296 measurements of input/output [35].

The RBF NN has two inputs, one output and a hidden layer. The number of neurons in the hidden layer N_n is determined by the NSGA II as well as the centers and the widths of the Gaussian functions, and the weights of connection between the hidden layer and the output layer.

The NSGA II optimizes simultaneously N_n and the quadratic cumulated error e_c given by:

$$e_c = \sum_{i=1}^N e^2(i); \text{ with } e(i) = \sum_{i=1}^N (Y_d(i) - Y_r(i))$$

Where:

e_c : cumulative error. $e(i)$: instantaneous error. N : length of the simulation sequence (the number of data $N=296$). Y_d : the desired output, Y_r : reel output (output of the RBF NN model).

The neural network learning is performed on 100 data of model of Box and Jenkins and validation is performed on the remaining data.

The results obtained which represent the first Pareto front (non dominated individuals) of the last generation is given in table 1.

Table 1. Pareto front

N° of Individual	Number of neurons in the hidden layer N_n	Training error (100 data) e_t	Validation error (196 data) e_v	Global error (296 data) e_g
1	2	5,4241	11,0611	16,4852
2	3	4,7137	8,7448	13,4585
3	4	3,0042	6,1185	9,1227
4	6	1,0218	3,5033	4,5251
5	7	0.2043	0.3216	0,5259

By analyzing this table, we can see that there is a important difference between the global error of the individual $N^\circ 7$ and the other results, for this we preferred this

individual whose model of RBF NN is 7 neurons in the hidden layer and with an overall error of 0.5259.

The figure 2 shows the concentration of CO2 released from the output of the boiler (desired output y_d) and the output of the RBF neural model (y_r), figure 3 shows the global error, figure 4 represents the training error, and the validation error is shown in Figure 5, and the Pareto front "global cumulated error function of the number of neurons" is shown in figure 6.

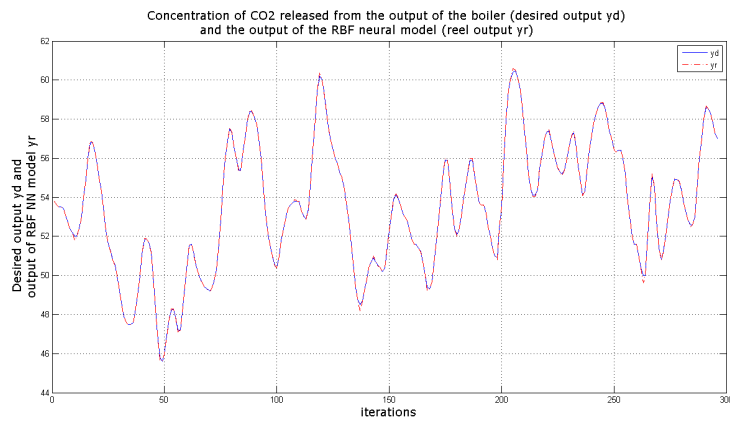


Fig. 2. The concentration of CO2 released from the output of the boiler (desired output y_d) and the output of the RBF neural model (y_r)

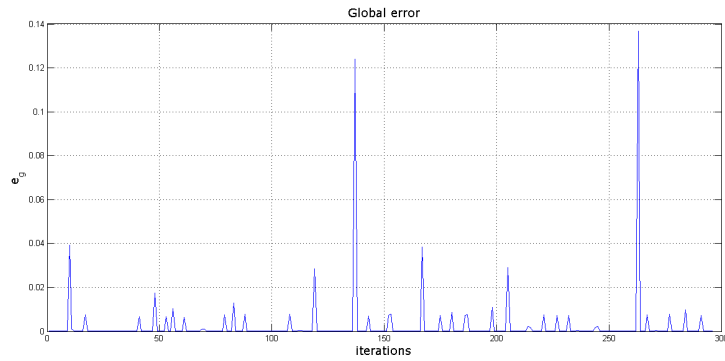


Fig. 3. The global error



Fig. 4. The training error

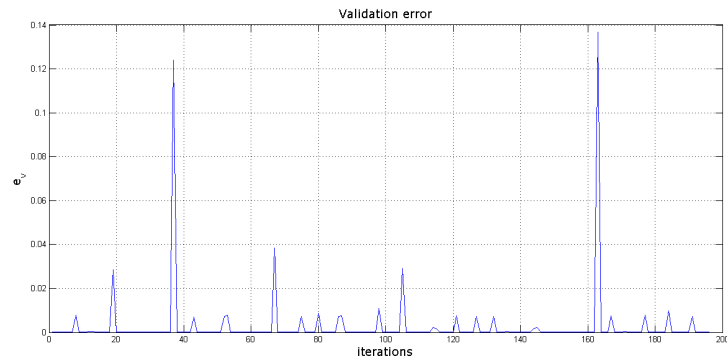


Fig. 5. The validation error

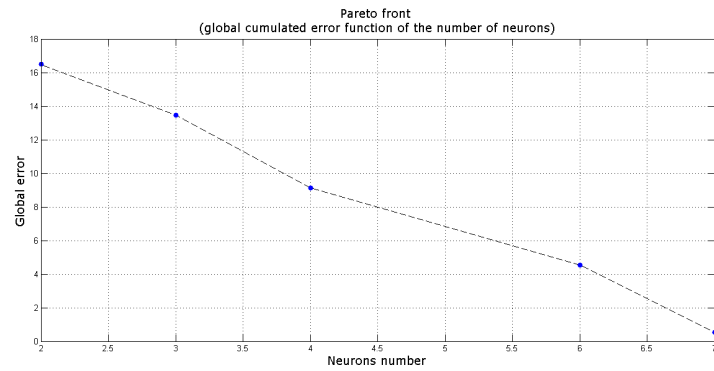


Fig. 6. Pareto front "global cumulated error function of the number of neurons"

Based on these results, we can conclude that the multi-objective genetic algorithm NSGAIII gave a good structure of RBF neural network model, with the best number of neurons in the hidden layer and the best connection weights Between the

hidden layer and the output layer, and it also found the best parameters of the radial function of hidden layer neurons, because we see that the RBF neural network model output is very close to that of the desired output, with a global error of 0.5259.

6. Conclusion

In this work we considered the designed of RBF neural network using the multi-objective genetic algorithms type NSGAII, by optimizing simultaneously two objective functions: the first function is the quadratic cumulatively error, which is the difference between the desired signal and the RBF NN model output signal, and the second is the number of neurons in the hidden layer, thereby the NSGA II chromosome contains the number of neurons in the hidden layer, Gaussian functions centers and widths of the hidden layer neurons, and the weights of connections between the hidden layer and the output layer. At the end of the evolution of this algorithm off-line, we have a set of RBF models which forming the final Pareto front, and includes all allowed results, and ensuring the predefined criteria. This optimization technique is applied to modeling a nonlinear system which is the BOX and JENKINS process. It's a gas-fired boiler with the input the gas at the inlet and the output the concentration of released CO₂. The results show that using NSGAII to optimize the RBF NN provides a good model, these results are very satisfying.

References

- [1] Mirta B. Gordon, "The dynamics of complex systems and applications to SHS: models, concepts, methods Leibniz-IMAG", Grenoble, 2004.
- [2] Matthew Gilson, "Training recurrent neural networks to pulses applied to the modeling of a biological neural tissue," University of Montreal, 2003.
- [3] J. M. Zurada, "Introduction to artificial neural systems", West Publishing, 1992.
- [4] N. Baba, "utilisation of stochastic automata and genetic algorithms for neural network learning", In parallel Problem Solving from Nature 2. Elsevier, 1992.
- [5] Xiao, Z. ; Liang, S. ; Wang, J. ; Chen, P. ; Yin, X. ; Zhang, L. ; Song, J, "Use of General Regression Neural Networks for Generating the GLASS Leaf Area Index Product From Time-Series MODIS Surface Reflectance", Geoscience and Remote Sensing, IEEE Transactions on, Volume: 52 , Issue: 1 , Part: 1, Page(s): 209 – 223, Publication Year: 2014.
- [6] Nikdel, N. ; Nikdel, P. ; Badamchizadeh, M.A. ; Hassanzadeh, I, "Using Neural Network Model Predictive Control for Controlling Shape Memory Alloy-Based Manipulator", Industrial Electronics, IEEE Transactions on, Volume: 61 , Issue: 3 , Page(s): 1394 – 1401, Publication Year: 2014.

- [7] Yuan, Jiawei ; Yu, Shucheng, "Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing", *Parallel and Distributed Systems, IEEE Transactions on*, Volume: 25 , Issue: 1, Page(s): 212 – 221, Publication Year: 2014.
- [8] H. Yu; T. Xie; S. Paszczynski; B.M. Wilamowski, "Advantages of Radial Basis Function Networks for Dynamic System Design", *IEEE Transactions on Industrial Electronics*, Vol 58, Issue 12 , pp 5438 – 5450, 2011.
- [9] R. Furtuna, S. Curteanu, F. Leon, "Multi-objective optimization of a stacked neural network using an evolutionary hyper-heuristic", Elsevier B.V. *Applied Soft Computing*, doi:10.1016, Available online 24 September 2011.
- [10] K. Lamamra and K. Belarbi, J. Bosche, A. EL Hajjaji, "A neural network controller optimised with multi objective genetic algorithms for a laboratory Anti-lock Braking System", *Revue Sciences et Technologie B*, university of Constantine 1, N° 35, June 2012.
- [11] Yabin Xu and Jingang Zheng, "Identification of Network Traffic Based on Radial Basis Function Neural Network", *Intelligent Computing and Information Science, Communications in Computer and Information Science*, Volume 134, 173-179, 2011.
- [12] Pedro Antonio Gutiérrez, , César Hervás-Martínez, ,and Francisco J. Martínez-Estudillo, "Logistic Regression by Means of Evolutionary Radial Basis Function Neural Networks", *IEEE Transactions On Neural Networks*, VOL. 22, NO. 2, February 2011.
- [13] Parag C. Pendharkar, "A hybrid radial basis function and data envelopment analysis neural network for classification" , *Computers & Operations Research*, science direct, Volume 38, Issue 1, Pages 256-266, January 2011.
- [14] Mansour Sheikhan, Reza Shahnazi, Ehsan Hemmati, "Adaptive active queue management controller for TCP communication networks using PSO-RBF models", *Neural Computing and Applications*, Volume 22, Issue 5, pp 933-945, April 2013.
- [15] Mansour Sheikhan, Reza Shahnazi, Sahar Garoucy, "Hyperchaos synchronization using PSO-optimized RBF-based controllers to improve security of communication systems", *Neural Computing and Applications*, Volume 22, Issue 5, pp 835-846, April 2013.
- [16] Deyin Xia; Liangyong Wang ; Tianyou Chai, "Neural-Network-Friction Compensation-Based Energy Swing-Up Control of Pendubot", *Industrial Electronics, IEEE Transactions on*, Volume:61 , Issue: 3, Page(s): 1411 – 1423, Date of Publication: March 2014.
- [17] Jafarnejadsani, H., Pieper, J. ; Ehlers, J, "Adaptive Control of a Variable-Speed Variable-Pitch Wind Turbine Using Radial-Basis Function Neural Network", *Control Systems Technology, IEEE Transactions on*, Volume:21 , Issue: 6, Page(s): 2264 – 2272, Date of Publication: Nov. 2013.
- [18] Wei Chai, Junfei Qiao, "Non-linear system identification and fault detection method using RBF neural networks with set membership estimation", *International Journal of Modelling, Identification and Control*, Volume 20, Number 2, 114-120, 2013.
- [19] Dos Santos Coelho, L. ; Ferreira da Cruz, L. ; Zanetti Freire, R, "Swim velocity profile identification by using a modified differential evolution method associated with RBF Neural Network", *Innovative Computing Technology (INTECH)*, 2013 Third International Conference on. 2013, Page(s): 389 – 395, Date of Conference: 29-31 Aug.

- [20] Mukhopadhyay, S. ; Panigrahi, P.K. ; Mitra, A. ; Bhattacharya, P. ; Sarkar, M. ; Das, P. "Optimized DHT-RBF model as replacement of ARMA-RBF model for wind power forecasting", Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013 International Conference on, Page(s):415 – 419, Date of Conference: 25-26 March 2013.
- [21] Hao Chen ; Yu Gong ; Xia Hong, "Online Modeling With Tunable RBF Network, Cybernetics", IEEE Transactions on, Volume:43 , Issue: 3, Page(s): 935 – 947, Date of Publication: June 2013.
- [22] D. Urbani, "Méthodes Statistiques de Sélection d'Architectures Neuronales: Application à la Conception de Modèles de Processus Dynamiques", Thèse de Doctorat de l'Université Pierre et Marie Curie, Paris VI, 1995.
- [23] C.L. Giles, C.B. Miller, D. Chen, H. H. Chen, G.Z. Sun, Y. C. Lee, "Learning and extracting finite state automata with second order recurrent neural networks", Neural Comput, vol 4, pp. 393-405, 1992.
- [24] J.N. Morse, "Reducing the size of the nondominated set, Pruning by clustering", Comput. Oper. Res. 7 (1-2), pp. 55-66, 1980.
- [25] C.L. Giles, D. Chen, G.Z. Sun, H. Chen, Y. C. Lee, M.W. Goudreau, "Constructive learning of recurrent neural networks: problems with recurrent cascade correlation and a simple solution", IEEE Trans. Neural Networks 6 (4), pp. 829, 1995.
- [26] P.J. Angeline, G.M. Saunders, J.B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks", IEEE Trans. Neural Networks 5 (1), pp. 54-65, 1994.
- [27] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan , "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II". Parallel Problem Solving from Nature (PPSN VI), Berlin, pp. 849-858, 2000.
- [28] K. Deb, T. Goel, "Controlled elitist non-dominated sorting genetic algorithms for better convergence", Proceedings of the first International Conference on Evolutionary Multi-criterion Optimization, Berlin, pp. 67-81, 2001.
- [29] Hashmi Khayyam, Alhosban Amal ; Najmi Erfan ; Malik Zaki, "Automated Web service quality component negotiation using NSGA-2", IEEE, ACS International Conference on Computer Systems and Applications (AICCSA), pp. 1-6, May 2013.
- [30] B.H. Min, C. Park, I.S. Jang, H.Y. Lee, S.H. Chung and J.M. Kang, "Multi-Objective History Matching Allowing for Scale-Difference and the Interwell Complication", 75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013, Session: Reservoir Simulation III, June 2013.
- [31] D. Gossard, B. Lartigue, F. Thellier, "Multi-objective optimization of a building envelope for thermal performance using genetic algorithms and artificial neural network", Science Direct, Energy and Buildings, Volume 67, Pages 253–260, 2013.
- [32] Jean Dipama, "Optimisation multi-objectif des systèmes énergétiques", thèse PHD, université de Montréal, avril 2010.
- [33] Hashmi Khayyam, Alhosban Amal ; Najmi Erfan ; Malik Zaki, "Automated Web service quality component negotiation using NSGA-2", IEEE, ACS International Conference on Computer Systems and Applications (AICCSA), pp. 1-6, May 2013.
- [34] Jean Dipama, "Optimisation multi-objectif des systèmes énergétiques", thèse PHD, université de Montréal, avril 2010.
- [35] G.E.P BOX and G.M JENKINS, "Time series analysis: forecasting and control", CA: Holden-Day, San Francisco, pp 575, 1976.