Preprints of the
9th International Symposium on Advanced Control of Chemical Processes
The International Federation of Automatic Control
June 7-10, 2015, Whistler, British Columbia, Canada

Optimization using ANN Surrogates with Optimal Topology and Sample Size

M Srinivas Soumitri*, Saptarshi Majumdar, Kishalay Mitra*****

**Indian Institute of Technology, Hyderabad, Ordnance Factory Estate, Yeddumailaram, 502205, INDIA (email: ch13m1007@iith.ac.in)*
*** Indian Institute of Technology, Hyderabad, Ordnance Factory Estate, Yeddumailaram, 502205, INDIA (email: saptarshi@iith.ac.in)*
**** Indian Institute of Technology, Hyderabad, Ordnance Factory Estate, Yeddumailaram, 502205, INDIA (email: kishalay@iith.ac.in)*

Abstract: Industrial scale process modelling and optimization of long chain branched polymer reaction network is currently an area of extensive research owing to the advantages and growing popularity of branched polymers. The highly complex nature of these reaction networks requires a large set of stiff ordinary differential equations to model them mathematically with adequate precision and accuracy. In such a scenario, where execution time of model is expensive, the idea of making the online optimization and control of these processes seems to be a near impossible task. Catering to these problems in the ongoing research, the authors presented a novel work where the kinetic model of long chain branched poly vinyl acetate has been utilized to find the optimum processing conditions of operation using Sobol sequence based ANN as meta models in a fast and highly efficient manner. The article presents a novel generic algorithm, which not only disables the heuristic approach of designing the ANN architecture but also allows the computationally expensive first principle model to determine the configuration of the ANN which can emulate it with maximum accuracy along with the size of training samples required. The use of such a fast and efficient Sobol based ANN as surrogate model obtained by the proposed algorithm makes the optimization process 10 times faster as compared to a case where optimization is carried out with the expensive first principle model.

Keywords: Online optimization, Optimum process conditions, Meta models, Sobol, Artificial Neural Networks

1. INTRODUCTION

Continuous endeavour is there to build mathematical models for reaction mechanisms describing the polymer kinetics and long chain branching (LCB) in polymers, so that the operation can be optimized and controlled to harness the maximum benefit out of the system. Optimization of such models requires large simulation time due to the complexity involved in the mathematical models pertaining to the reaction network of branched polymers. Kipparisides et al (1998) have reported a method where, on applying the numerical fractionation technique, the overall molecular weight distribution (MWD) can be obtained using the leading moments derived from moment based modelling. This method, which is currently the most promising and efficient technique to obtain the overall MWD of the live and dead polymer, remains to be computationally exorbitant (Mogalicharla et al, 2014).

On the other hand, the optimization of such models for finding the optimum operating conditions makes space for multi-objective optimization formulations owing to the conflict involved in the objective functions. In spite of robustness and high efficiency, in order to generate a wide spread trade-off in the solution set, called the Pareto optimal (PO) solutions, these evolutionary optimization techniques require a large set of population called the candidate solutions (Nain and Deb 2002). In order to generate these candidate solutions for optimizer, it is required to solve the computationally expensive model repeatedly and thus this approach increases the computational burden rendering the optimization exercise to be an offline one. In such a scenario, online optimization of computationally expensive models might remain a forlorn as the multiple runs of those models necessarily cannot be avoided because any decrease in the population number may not actually lead to a high quality wide spread Pareto optimal front (Nain and Deb, 2002). Meta-models, which can emulate the computationally expensive models accurately using a small and limited set of sampling data originated from the time expensive first principle based model, can be of great assistance to make the optimization process faster. The most widely used meta-models are function approximation models which make use of popular techniques such as response surface methodology (RSM), artificial neural networks (ANN) and Kriging interpolators to emulate the computationally expensive physics driven models . ANNs are mathematical models, which try to mimic the functioning of biological neural network of human brain. They are widely acknowledged for their immense applications in pattern recognition problems and image processing. The parameters, in terms of weights and biases for each neuron in a layer of the architecture, allow a wide range of access to accurately predict the behaviour of highly nonlinear systems.

The process of designing the optimal configuration of the network architecture of ANN really turns out to be

Copyright © 2015 IFAC

computationally laborious and often involves a method of hit and trial. Nuchitprasittichai and Cremaschi (2012) developed a novel algorithm to determine the sample size of a given network architecture using the critically acclaimed statistical model evaluation technique called the K-Fold cross validation (Nuchitprasittichai and Cremaschi, 2012). Their work has delved upon the idea that when a fixed network topology gets trained with different number of sample points, it results in altogether a new set of parameters and thereby a different model. They then applied the K-Fold cross validation technique to evaluate each model in terms of mean of absolute deviations from each fold. The sample size increment was obtained by using a novel approach of incremental Latin Hypercube Sampling (i-LHS). The increment in sample size was terminated when the K-Fold error was minimized to an acceptable tolerance value with respect to sample size. Although this method of determining the sample size for training a network is novel and efficient, the i-LHS way of incrementing the data set demands addition and deletion of sampling points already present in the data set to ensure the newly emerged sampling set conforms the latin hypercube structure. Despite fixing a network topology, this work, therefore, does not reduce the computational burden involved in determining the sample size for training. Many other researchers tried to train the weights by using the genetic algorithm to ensure an optimum hybrid ANN. However, almost no work has been found in the literature that addresses the concern of determining the best ANN architecture along with the sample size required by it which can predict results accurately with less computational burden. Chakraborty et al (2013) have reported a procedure where a multi-objective optimization problem has been formulated while coming up with an ANN topology in course of training the ANN to a given data set. In search of parsimonious models, the complexity of the network evaluated by AIC/BIC criteria is minimized while maximizing the accuracy simultaneously. However, they restricted their study to a single layered fixed network topology and optimal data size required to train a model has also not been considered in this work. Although it has been customary to start with a single layered topology due to the assumption that single layered networks with sufficient nodes can predict almost all the nonlinearity present in a data set, one should not fix the number of layers of the architecture based on the sampling data available (Roy et al,2008).

In this article, the authors have tried to cater the aforementioned needs by proposing a novel, computationally economical algorithm wherein a multi-objective optimization approach has been adopted to (i) optimally design the architecture of a parsimonious ANN by considering multiple layers, and (ii) determine the size of sample points required for training that best ANN model to predict results with maximum accuracy. The network, thus obtained, is then used as surrogate in optimizing the process conditions to obtain desired LCB in polyvinyl acetate (PVAc) polymer using non-dominated sorting genetic algorithm (NSGA II). The data for surrogate building is procured from a first principle based model capable of predicting the overall MWD of PVAc. The algorithm developed is independent of the physics based expensive model and it allows the model to select the network which best emulates it in the most generic way and thereby preventing the over-fitting. The organization of the paper is as follows: the kinetic model of branched PVAc, the optimization problem and the proposed ANN technique for surrogate building are described in section 2. The results of optimization problem with newly developed surrogate are described in section 3. Finally the concluding remarks are provided followed by the references.

2. FORMULATION

2.1 PVAc KINETIC MODEL AND OPTIMIZATION

The reaction network for PVAc model has been obtained from the work of Butte et al (1999) and is presented in Table 1. The net rate of production of live polymer, dead polymer, monomer and initiator can be obtained from the scheme presented in Table 1. To avoid the complication involved in solving a large number of time consuming ordinary differential equation – initial value problems (ODE-IVPs) to describe such a reaction scheme, the approach of moment based modelling was taken up for each class of branched polymer where the 0^{th}, 1^{st} and 2^{nd} ordered moments for live and dead polymers of each class are derived (Mogalicharla et al 2014). To avoid the drawback of method of moments to construct overall MWDs from individual moments, the number of classes for constructing the moments was carefully chosen ensuring the criteria that the summation of first moments of each class remains equivalent to the overall first moment of polymer. This ensured the complete construction of overall MWD from MWD of individual moments, but resulted in a large number of classes since the polymer was highly branched. The resulting model, having large number of highly nonlinear ODE-IVPs (say 285), demanded large time for simulating a single run using the differential algebraic equation (DAE) solver LIMEX. The equations 1 to 4 define the polymer properties such as number average molecular weight (M_n), weight average molecular weight (M_w), Poly dispersity index (PDI) and Branching index (B_n), respectively, in terms of moments obtained by solving the aforementioned set of ODEs. With these equations in place, if the batch monomer concentration (M), amount of initiator (I) and temperature of the isothermal batch polymerization process (T) are given as inputs, the model can be solved to obtain M_n, M_w, B_n and PDI as outputs after a batch polymerization time of t_{poly}.

The current industrial scenario demands the production of high molecular weight, branched and cross linked polymer within the least time of operation to minimize the cost of operations and maximize the profits earned by the advantages of branched polymers. It is, therefore, desired to find the optimum values of M, I, T which results in maximum M_w, B_n, and minimum t_{poly} (Table 2). The upper and lower bounds for the decision variables M, I and T are obtained from the experimental study (Thomas, 1998). The phenomenological constraints are in place for the M_w and PDI. Population based NSGA II framework has been used as optimization algorithm to solve this multi-objective optimization problem (MOOP-1).

Since the reaction network is highly complicated, it required as good as 285 ODE-IVPs to mathematically model it which

Copyright © 2015 IFAC

consumed an overwhelming amount of time. Thus the target was to replace the original first principle model with a fast and accurate Sobol based ANN model in optimization algorithm and study the advantages and disadvantages that surrogate based optimization gives in terms of computational time and accuracy.

Table 1: Kinetic Scheme for PVAc polymerization

Step No	Steps in the Mechanism	Corresponding Reactions
1	Decomposition of Initiator	$I \xrightarrow{k_d} 2R^o$
2	Chain reaction	$R^o + M \xrightarrow{k_I} P_1$
3	Chain Propagation	$P_n + M \xrightarrow{k_P} P_{n+1}$
4	Chain transfer to monomer	$P_n + M \xrightarrow{k_{fm}} D_n + P_1$
5	Chain transfer to solvent	$P_n + S \xrightarrow{k_{fs}} D_n + P_1$
6	Chain transfer to polymer	$P_n + D_m \xrightarrow{k_{fp}} D_n + P_m$
7	Termination by combination	$P_n + P_m \xrightarrow{k_{tc}} D_{n+m}$
8	Termination by disproportionation	$P_n + P_m \xrightarrow{k_{td}} D_n + D_m$

$$M_n = \left(\frac{\mu_1}{\mu_0}\right) MW_{VA} \quad (1)$$

$$M_w = \left(\frac{\mu_2}{\mu_1}\right) MW_{VA} \quad (2)$$

$$PDI = \left(\frac{M_W}{M_n}\right) \quad (3)$$

$$B_n = \sum_{n=1}^{\infty} \sum_{b=0}^{N_c} \left(\frac{bN_{n,b}}{\mu_0}\right) \quad (4)$$

Table 2: Optimization formulation for PVAc model

Objective functions	Constraints	Decision Variables
Maximize M_w	$M_w \geq (M_w)^{min}$	$10 \leq M(mol/lit) \leq 14$
Maximize B_n	$PDI \leq (PDI)^{max}$	$0.00003 \leq I(mol/lit) \leq 0.00015$
Minimize t_{poly}	$C \leq 0.97\ C_{gel}$ (T)	$333K \leq T \leq 352K$

2.2 SOBOL BASED ARTIFICIAL NEURAL NETWORK

2.2.1 Construction of ANN code

The PVAc LCB reaction network model is a 3 input, 4 output system. This first principle model is run to generate the samples required for training and validation of the neural network. A generic ANN code was then developed which takes the architecture, training set and validation set as the input and returns the accuracy of predictions as the output. The code was constructed to serve a multiple input single output (MISO) formulation to capture nonlinearity involved in the functional relationship between input and output data better. The MISO code was then run simultaneously to determine the ANN models to emulate all the four outputs. The ANN was trained by back propagation and Levenberg-Marquardt method was used to estimate the weights and biases. The bipolar sigmoidal (tansig) activation function was implemented in the hidden layers and linear activation function was implemented in the output layer. The sampling technique used is Sobol set which is based on highly converging Sobol sequence. It gives sampling points for any number of dimensions normalised between 0 and 1. To determine the optimal size of data that a particular ANN topology might require, the above ANN model is trained initially with a smaller data set generated by Sobol sequence and the data size is incremented until the change in cross-validation error with respect to the change in data size meets the tolerance limit specified. The advantage with Sobol set is that it always maintains the same sequence of numbers even if the sample size is varied which indeed was not the case with i-LHS suggested by Nuchitprasittichai and Cremaschi (2012).

2.2.2 Formulation of the Algorithm

Looking at a general structure of neural network, it is perceptible that as the number of nodes in the neural network increases, its ability to interpolate with accuracy also increases. This can be primarily speculated to the increase in number of parameters that aid in capturing the nonlinear behaviour of the system. However, this increase in accuracy comes at the cost of obtaining large sampling data and time for training the increased number of parameters in the network. It is also worthwhile to mention that the data greedy neural networks suffer from the problem of over fitting, when provided with large sample set for training. Thus the goal while designing a neural network can be finding of a simple ANN structure that can fit a given set of data with maximum accuracy. This very conflicting nature of objectives leads the authors to formulate a multi-objective optimization problem (MOOP-2), where the aim was to maximize the accuracy of the network while simultaneously minimize the total number of nodes in the architecture. The optimization formulation is presented in Table 3.

Table 3: Optimization Formulation for finding the generic architecture of ANN with minimum complexity and maximum accuracy.

Objective functions	Constraints (optional)	Decision variables
Maximize R2. Minimize N	R2 ≤ 0.98 N ≥ 4	$1 \leq n_1 \leq 16$
		$0 \leq n_2 \leq 15$
		$0 \leq n_3 \leq 15$

Since the accuracy of any surrogate model mainly depends on the sample size used for training, the authors ensured that every network in the population is allowed to attain

Copyright © 2015 IFAC

corresponding maximum accuracy by providing them with sufficient number of training points. This step was implemented in the algorithm using the sample determination technique proposed by Nuchitprasittichai and Cremaschi (2012) where their proposal of using i-LHS as sampling plan has been replaced with more computationally efficient Sobol based sampling plan. Once the network was trained with the sampling data of size estimated in the previous step, the accuracy of its interpolation was evaluated using a completely unknown validation test set of size n_t obtained using Latin Hypercube Sampling (LHS). The original output in the validation set was compared with the corresponding set of predictions made by the surrogate ANN model, $\hat{y}$. The accuracy of predictions was determined in terms of statistical measure R2, correlation coefficient, which is given by the following equation:

$$R2 = \left(\frac{\mathrm{cov}(y,\hat{y})}{\sqrt{\mathrm{var}(y)\mathrm{var}(\hat{y})}}\right)^2 \quad \ldots (5)$$

where the covariance and variance are defined as follows:

$$\mathrm{cov}(y,\hat{y}) = n_t \sum_{i=0}^{n_t} y^{(i)}\hat{y}^{(i)} - \sum_{i=0}^{n_t} \hat{y}^{(i)} \sum_{i=0}^{n_t} y^{(i)}$$

$$\mathrm{var}(y) = n_t \sum_{i=0}^{n_t} y^{(i)2} - \left(\sum_{i=0}^{n_t} y^{(i)}\right)^2$$

The second objective being complexity of the network is evaluated in terms of total number of nodes in the network (N). The conventional assumption of considering only one hidden layer in the network topology was eliminated by considering the number of layers and nodes in each layer (n_i) as the decision variables in the MOOP formulation (see Table 3). The minimum number of hidden layers was kept 1 while the maximum was ensured at 3 due to the amount of complexity involved and time consumed for training a network with much higher number of layers. Although for the current formulation the maximum number of layers considered was fixed at 3, this number is arbitrary and can always be increased as per the requirement of the physics based model. The number of nodes in each of these layers can be varied between the limits as shown in Table 5. Binary coded NSGA II optimizer was run for 200 generations with a population set of 100 to converge at a wide spread Pareto optimal (PO) front. The candidate solutions in the PO front were then evaluated using K-Fold cross validation criteria to obtain the most parsimonious model with least cross validation error, out of all possible solutions. The proposed algorithm and the algorithm for sample size determination are depicted in figures 1 and 2, respectively.

The binary NSGA II framework provides an architecture in terms of number of nodes in each layer. Since the upper and lower bounds are taken as 1 and 16 for first layer and 0 and 15 for the second and third layers, the binary GA framework compels the decision variables to be integer values lying between the stipulated upper and lower bounds. This can be easily implemented by assuming each of these decision variables as a 4 bit long binary strings, which upon decoding leads to decoded values from 0 to 15. If any value of n2 or n3 is assigned to 0, it implies that the architecture ends at one layer ahead and the remaining layers are not considered in the architecture. This ensures the GA to provide with candidate solutions from single layered to multi layered architectures. The size of the training set, which can yield maximum accuracy from the given architecture, is then determined from the sample determination algorithm, where the increment in sample size using Sobol set is allowed until the change in cross validation error with respect to change in sample size is found to be satisfying a tolerance criteria. The ANN network is then trained with that sample set and validated with a completely unknown validation data. The objective functions, R2 value and the total number of nodes, are thus evaluated and returned back to NSGA II. The architecture along with the corresponding values of objective functions are also stored in a database to check redundancy in computation and thereby saving a significant amount of simulation time.

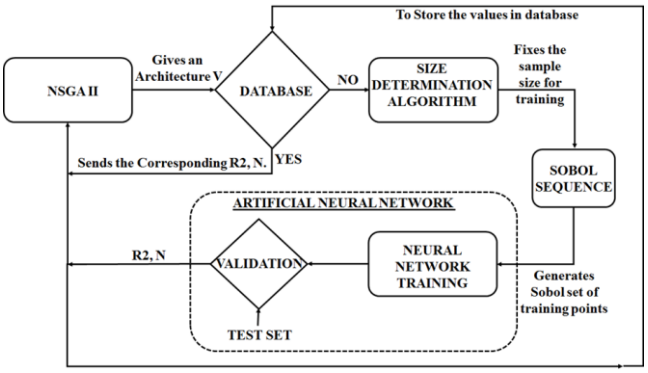

Figure 1: Proposed Algorithm

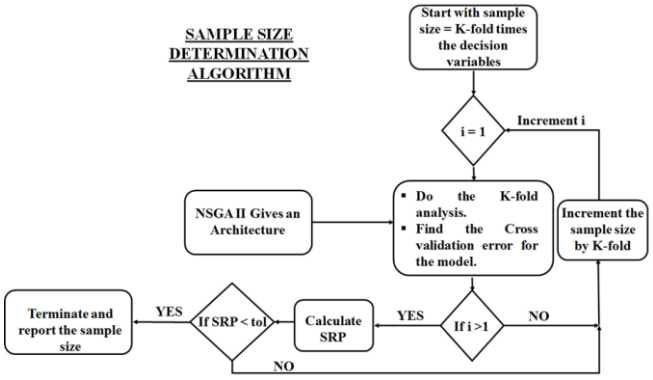

Figure 2: Sample determination algorithm

3. RESULTS AND DISCUSSION

The PVAc LCB model considered in the current work is validated with the experimental data found from Thomas et al. (1998). The architectures of ANN models obtained from the proposed algorithm were used as surrogates in the optimization of the validated PVAc LCB model. The results of the optimization problem and the discussion pertaining to the performance of the proposed algorithm form the rest of the paper.

The PO solutions obtained from the proposed multi-objective optimization formulation (MOOP2) to design the best network topology for the t_{poly} as output is shown in figure 3. A total of 9 architectures and the sample size requirement to attain the corresponding maximum accuracies for each of

Copyright © 2015 IFAC

those candidate solutions in the PO front are listed in Table 4. When it comes to the selection of one architecture out of all these non-dominating solutions, it finally depends on the user to select a model based on their higher order information. Three approaches, which could serve as higher order information to finalize a model out of the PO solutions are reported here. The first approach is to utilize the values of K-Fold cross validation errors of each of the architectures while the second approach is to measure the complexity of the network by using AIC/BIC (Chakraborti et al, 2013) formulation. The third approach could be providing altogether a totally new unseen data set to evaluate all the models in terms of accuracy of their predictions and selecting a model that shows a good balance between two of the objectives i.e. accuracy and complexity. The results of all three approaches to select an architecture out of the PO solutions are reported in Table 5.

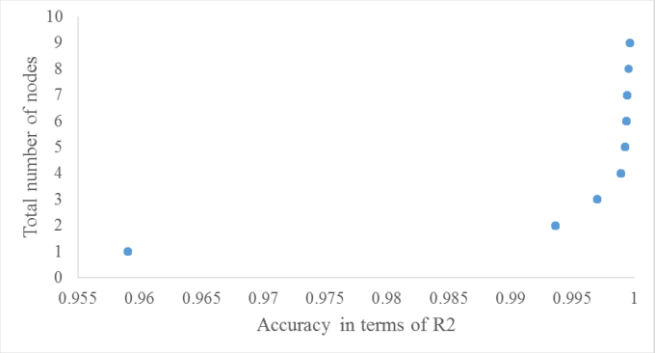

Figure 3: Pareto Optimal Front for output t_{poly}.

Table 4: Candidate solutions in the final PO front

Nodes in Layer 1 (n1)	Nodes in Layer 2 (n2)	Nodes in Layer 3 (n3)	Objective 1 : Accuracy of prediction R2	Objective 2 : Total number of nodes
1	0	0	0.959029	1
5	1	0	0.999392	6
2	0	0	0.993624	2
3	1	0	0.998914	4
3	0	0	0.997036	3
2	2	1	0.999239	5
2	6	1	0.999657	9
2	2	3	0.999434	7
2	4	2	0.999544	8

Although any of these methods can be applied to select the best model out of the set of PO solutions based on the applicability of the end user, the authors, however, have selected a model here by the third approach only for the reason of maintaining an unbiased balance between the two considered objectives. The architecture thus selected, in the case of t_{poly} as output came out to be the network with configuration [3 5 1 1] (read as 3 inputs, 5 nodes in first hidden layer, 1 node in the second hidden layer and 1 node in output layer) with the R2 of 0.9993 and the requirement of training 28 parameters (weights + biases). The sample size determined from the sample determination algorithm for training this configuration is 70 while the capability of this architecture in terms of accuracy can be seen in figure 4. Although the binary NSGA II framework was run for 200 generations with a population size of 100, the emergence of this PO front was very early, at generation number 30. The optimally designed code which included the database for storing the NSGA II runs, helped in checking the redundancy in calculations. With the database in place, it was found that, the number of architectures tested were only 580. This clearly shows the speed of the developed algorithm to converge at the solution, all possibly because of the symphony between multi objective optimization formulation and the efficient usage of the database in the developed code. The stochastic nature of Genetic Algorithm is governed by a random seed, the changes in which would lead to variations in the final results. In order to confirm that the results obtained converge hopefully to the global PO front, several GA simulations were carried out with different random seeds and it was found that the final PO fronts for all runs were similar. A constraint at the level of optimization formulation restricting the total number of nodes to be greater than a certain threshold (say 4) and R2 to be less than a corresponding value (say 0.98) revealed the local PO fronts.

Table 5: Higher order analysis for each of the architectures

n1	n2	n3	Size of Data set	K-fold CV error	AIC measure	R2 for totally unseen data
1	0	0	60	0.454886	-176.744	0.95597
5	1	0	70	1.76467	-384.313	0.999257
2	0	0	60	0.359945	-265.258	0.993765
3	1	0	60	0.298216	-326.996	0.999118
3	0	0	60	0.336309	-291.2	0.997115
2	2	1	60	0.297238	-331.507	0.999283
2	6	1	70	0.329588	-381.889	0.999535
2	2	3	60	0.293778	-317.461	0.999309
2	4	2	60	0.315538	-309.371	0.999467

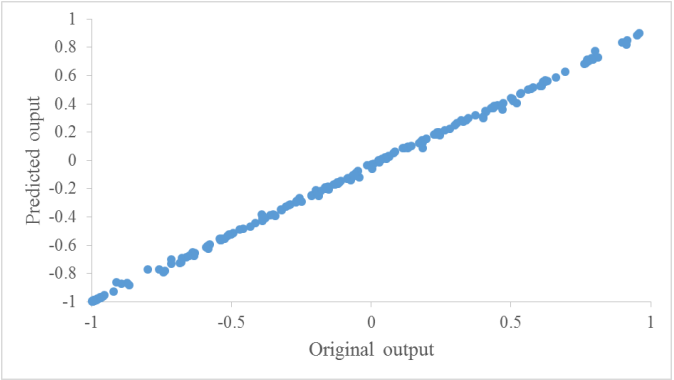

Figure 4: A plot of predicted output versus original output for selected architecture 3-5-1-1

Copyright © 2015 IFAC

Although the maximum number up to which the value of nodes per layer was varied is 16 (and thereby taking the maximum possible number of total nodes to 48), it is evident from the results of final PO front for the case of t_{poly} that the number of nodes per layer required to give accurate results has never crossed 6. These observations certainly put some light over the fact that for the data set from the current PVAc model, higher accuracy can be achieved by providing more number of layers rather than more number of nodes in a single layer. It can also be observed from Table 5 that almost 67 % of the candidate solutions which belong to the global PO front are multi-layered architectures. Thus these results clearly show that if the ANN model was designed with the assumption of considering a single hidden layer, the necessity of considering additional layers probably might not have surfaced out. Amongst one, two and three layers, the emergence of a two layered network as the most parsimonious model with a high accuracy as reported, shows the validity of the point raised earlier in the paper about the requirement of more than one layer which geometrically is a hyper-plane for classifying the sampling data in situations where the data are not linearly separable. The candidate solutions selected from the Pareto plots for finding best architectures to emulate the rest of outputs for PVAc model, their accuracy of predictions and the sample size requirement of each solution are listed in Table 6.

Table 6: Final surrogate models for emulating PVAc model

Output	Architecture	N	R2	Size
t_{poly}	3-5-1-1	6	0.99939	70
M_w	3-3-3-8-1	14	0.99799	50
B_n	3-2-1	2	0.99813	60
PDI	3-2-1	2	0.99791	80

The results, shown in Table 6 were then placed as surrogate models in the optimization algorithm aiming to find the optimum processing conditions for operating the batch PVAc reactor. Figure 5 compares the two PO fronts, one obtained using the ANN surrogate model while the other obtained without using any surrogate model. The quantification of this comparison of the two PO fronts was also obtained by finding the R2 measure when the PO points obtained without surrogate were predicted by the developed ANN model. It was found that the developed ANN surrogate model is predicting the original PO points with an average R2 measure of 0.979. The optimization of PVAc model without surrogate model in place revealed that the optimizer has called the expensive PVAc model approximately 70 (population) × 40 (generation) i.e. 2800 times to converge into the PO solutions while with ANN surrogate model in place, PVAc model was called only for a maximum of 280 times (80 training and 200 validation in case of time as output). These results clearly show the advantage of using the ANN surrogate model in the optimization procedure as it reduced the function calls by 90 %. These results support the fidelity of the proposed algorithm, which allows the sample data to select the configuration of ANN which best emulates them without being specific to the considered model.

4. CONCLUSION

The design of ANN for enabling it to emulate a time expensive model has been revised with proposing a novel, efficient and fast generic algorithm which provides the most parsimonious model along with the sample size required for training the model. The algorithm is tested for determining the best surrogate model for PVAc optimization involving long chain branching and the results indicate that 10 times reduction in function evaluation is possible by adopting this approach.

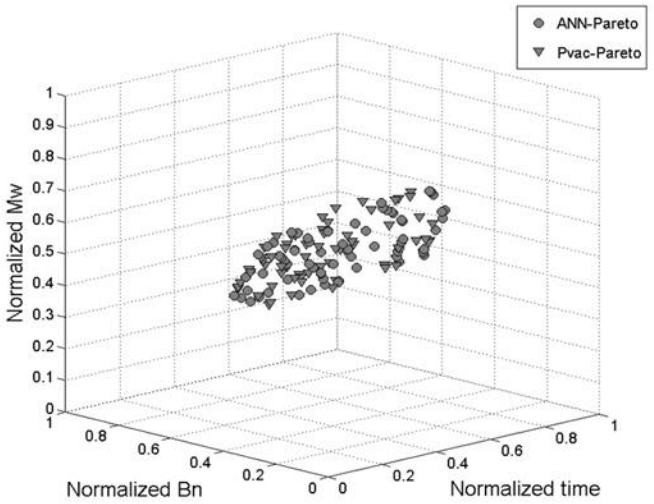

Figure 5: Comparison of PO fronts obtained for optimization of PVAc model with ANN surrogate and original model.

5. REFERENCES

Butte´ A, A. Ghielmi, G. Storti and M. Morbidelli. (1999) Calculation of molecular weight distributions in free-radical polymerization with chain branching.*MTS.*, 8 498-512.

Chakraborti Nirupam Brijesh Kumar Giri , Frank Pettersson & Henrik Saxn (2013) Genetic Programming Evolved through Bi-Objective Genetic Algorithms Applied to a Blast Furnace, *MMP*, 28:7, 776-782,

Kiparissides C and P. Pladis. (1998) A comprehensive model for the calculation of molecular weight-long-chain branching distribution in free-radical polymerizations, *Chem. Eng Sci.,* 53 3315

Mogilicharla, A.; Chugh, T.; Majumdar, S.; Mitra, K. (2014,) Multi-objective Optimization of Bulk Vinyl Acetate Polymerization with Branching. *Materials and Manufacturing Processes* 29, 210-217

Nain, P.; Deb, K. (2002) A computationally effective multi-objective search and optimization techniques using coarse-to-fine grain modeling. *In Proceedings of the PPSN Workshop on Evolutionary Multiobjective Optimization*

Nuchitprasittichai Aroonsri and Selen Cremaschi (2012) An algorithm to determine sample sizes for optimization with artificial neural networks *AIChE* Volume 59 Issue 3 Pages 805 – 812

Roy Nilay K., Walter D. Potter, and David P. Landau,(2008) Polymer property prediction and optimization using neural networks *IEEE Transactions on neural networks*, VOL. 17, NO. 4.

Thomas S(1998) Measurement and modelling of long chain branching of polymers. *PhD Thesis McMaster University Canada.*

Copyright © 2015 IFAC