Process modeling and optimization strategies integrating neural networks and differential evolution,

Nadeem Muhammed Khalfe
This article presents an artificial intelligence-based process modeling and optimization strategy, namely artificial neural networks—differential evolution (ANN-DE) for modeling and optimizing catalytic industrial ethylene oxide (EO) reactors. In the ANN-DE approach, an artificial neural network model is constructed for correlating process data comprising values of operating and performance variables. Next, model inputs describing process operating variables are optimized using DEs with a view to maximizing the process performance. The DE possesses certain unique advantages over the commonly used gradient-based deterministic optimization algorithms. The ANN-DE is a new strategy for chemical process modeling and optimization. The major advantage of the strategy is that modeling and optimization can be conducted exclusively from the historic process data wherein the detailed knowledge of process phenomenology (reaction mechanism, kinetics, etc.) is not required. Using the ANN-DE strategy, a number of sets of optimized operating conditions leading to maximized EO production and catalyst selectivity were obtained. The optimized solutions, when verified in an actual plant, resulted in a significant improvement in the EO production rate and catalyst selectivity.

Introduction. Ethylene oxidation to produce EO is an important reaction in the petrochemical industry for synthesizing glycol. Commercially, EO is produced in shell-and-tube-type EO reactors by reacting oxygen and ethylene at high temperature and pressure in the presence of silver-based catalyst. The ethylene oxidation involves a main reaction producing EO and an undesirable side reaction producing carbon dioxide (CO₂).

Desired main reaction:
Ethylene + Oxygen → Ethylene oxide
ΔHᵣ @ 25°C = −25,550 kcal/kg mol C₂H₄

Undesired side reaction:
Ethylene + Oxygen → Carbon dioxide + Water
ΔHᵣ @ 25°C = −316,200 kcal/kg mol C₂H₄

Reaction performance is measured by selectivity that is calculated by the percentage of ethylene used to produce EO compared to total ethylene used to produce EO and CO₂. Indirectly, selectivity measures the extent of the first reaction compared to the second reaction and it has a profound effect on overall glycol plant economics. The main objective of EO reactor modeling is to increase the selectivity by changing the process parameters.

Inhibitors are added to control the reaction rate and improve the catalyst selectivity. The industrial gas phase inhibitor is usually 1, 2 dichloroethane and its concentration is usually 4–6 ppm. Dichloroethane inhibits the combustion reaction (second reaction) to a greater extent than the epoxidation reaction (first reaction). In this way it promotes the selectivity for EO. An excess quantity of this inhibitor increases selectivity but reduces the production rate by occupying the active catalyst sites, which are otherwise available for the EO reaction. A less than optimum quantity of inhibitor reduces selectivity and produces more CO₂. Thus, the optimum inhibitor concentration value at the reactor inlet is absolutely necessary for maximizing EO production. It is very difficult to adjust the inhibitor flow in the reactor without knowing its concentration and control panel operators usually adjust the inhibitor flow heuristically based on experience and watching other parameters such as steam flow, reactor outlet oxygen concentration, etc., due to absence of a reactor model.

In industrial EO reactors catalyst selectivity is affected by many things such as concentration of catalyst poisons (such as acetylene) in the cycle gas, catalyst age and its sintering effect, the history of catalyst usage and water or other impurity carryover into the reactor (it can permanently damage some of the catalyst active sites) due to maloperation, etc. That’s why it is very difficult for any phenomenological model to succeed in an industrial scenario.

The selectivity increase in operating plants is considered difficult because the quantitative effect of different process parameters on the selectivity is not known. Selectivity also drops with catalyst age due to the sintering effect and permanent deactivation of active catalyst sites.

A rigorous mathematical model for EO reactors is still not available in the literature that can quantify all the process parameters’ effect on selectivity and can be used confidently in industry to optimize the EO reactor. The comprehensive reactor model is expected to take into account the various subjects, such as chemistry, chemical reaction and kinetics, catalysis and physics that consequently become very complex. Industry needs this mathematical model to optimize inhibitor flow and other process parameters so that selectivity is maximized. To develop such a model from basic principles of chemical engineering is very difficult because the reaction kinetics and catalyst mechanism are unknown. Due to
the complex nature of industrial EO reaction processes, the underlying physico-chemical phenomenon is seldom fully understood. Also, collecting the requisite phenomenological information is costly, time-consuming and tedious, and, therefore, development of phenomenological process models poses considerable practical difficulties. Moreover, nonlinear behavior, being common in chemical processes, leads to complex nonlinear models that in most cases are not amenable to analytical solutions; thus, computationally intensive numerical methods must be utilized for obtaining solutions. Difficulties associated with constructing and solving phenomenological models necessitates exploring alternative modeling formalisms.

Modeling using empirical (regression) methods is one such alternative. In conventional empirical modeling, appropriate linear or nonlinear models are constructed exclusively from the process input-output data without invoking the process phenomenology. Despite the availability of efficient numerical methods, the regression approach suffers from a significant drawback in that the form of the data fitting function needs to be prespecified. In the case of reactions exhibiting nonlinear dynamic behavior, guessing an appropriate form of the nonlinear fitting function becomes difficult and, therefore, the functional forms are selected heuristically, which leads to a significant increase in the computational effort. Even after expending such an effort, there is no guarantee that a correct functional form that fits the experimental data with reasonable accuracy can be found.

In the past decade, ANNs have emerged as a useful tool for nonlinear modeling, especially in situations where developing phenomenological or conventional regression models becomes impractical or cumbersome. The most widely utilized ANN paradigm is the multi-layered perceptron (MLP) that approximates nonlinear relationships existing between an input set of data (causal process variables) and the corresponding output (dependent variables) data set. The advantages of an ANN-based model are: it can be constructed solely from the historic process input-output data (example set), detailed process phenomenology knowledge is unnecessary for model development, a properly trained model possesses excellent generalization ability because it can accurately predict outputs for a new input data set and even multiple-input-multiple-output (MIMO) nonlinear relationships can be approximated simultaneously and easily. Because of their several attractive characteristics, ANNs have been widely used in chemical engineering applications such as steady-state and dynamic process modeling, process identification, yield maximization, nonlinear control, and fault detection and diagnosis.1-5 Several algorithms—each possessing certain positive characteristics—can train an MLP network, for example, the most popular are error-back-propagation (EBP),6 quick prop7 and resilient back-propagation (RPROP).8 Training an ANN involves minimizing a nonlinear error function (e.g., root-mean-squared-error, RMSE) that may possess several local minima. Thus, it becomes necessary to use a heuristic procedure involving multiple training runs to obtain an optimal ANN model whose parameters (weights) correspond to the global or the deepest local minimum of the error function.

Once an ANN-based process model is developed, it can be used for process optimization to obtain the optimal values of the process input variables that maximize or minimize a specified objective function. Thus, it is possible to obtain the optimal values of process operating variables that, for instance, maximize reactant conversion and selectivity of the desired products, or minimize reactor temperature and the selectivity of undesired by-products. Conventionally, various deterministic gradient-based methods are used for optimizing a process model.9 Most of these methods, however, require that the objective function should be smooth, continuous and differentiable. The ANN models can not be guaranteed to be smooth, especially in regions wherein the input-output data (training set) used in model building are located sparsely. Hence, gradient-based methods cannot be used efficiently for optimizing the input space of an ANN model. In such situations, an efficient optimization formalism known as differential evolution (DE) that is lenient toward the form of the objective function, can be used. Recently DEs, that are members of the stochastic optimization formalisms, have been used with great success in solving problems involving very large search spaces. DE has been used to design several complex digital filters10 and fuzzy-logic controllers.11 DE can also be used for parameter estimations. Babu and Sastry12 used DE for estimating effective heat transfer parameters in trickle-bed reactors using radial temperature profile measurements. They concluded that DE takes less computational time to converge compared to existing techniques without compromising accuracy of the parameter estimates.

The DEs were originally developed as the genetic engineering models mimicking population evolution in natural systems. Specifically, DE, like genetic algorithm (GA), enforces the “survival-of-the-fittest” and “genetic propagation of characteristics” principles of biological evolution for searching the solution space of an optimization problem. The principal features possessed by the DEs are: they require only scalar values and not the second- and/or first-order derivatives of the objective function, have the capability to handle nonlinear and noisy objective functions, perform global search and thus are more likely to arrive at or near the global optimum and DEs do not impose preconditions, such as smoothness, differentiability and continuity, on the form of the objective function.

In this article, ANN formalism is integrated with DE to arrive at modeling and optimization strategies. The ANN-DE strategy uses an ANN as the nonlinear process modeling paradigm, and the DE for optimizing the input space of the ANN model such that improved process performance is realized. To our knowledge, the hybrid involving an ANN and DE is being used for the first time for chemical process modeling and optimization. In this study, the ANN-DE strategy has been used to model and optimize a commercial plant. The optimized operating conditions maximized yield and selectivity of the desired reaction product (EO). The best sets of operating conditions obtained thereby, when subjected to actual plant validation, resulted in significant enhancements in EO yield and selectivity.

Process modeling and optimization formalisms. The process optimization objective under consideration is expressed as:

Given the catalytic process data comprising values of the multiple process inputs and the corresponding values of the multiple process outputs, find the optimal values of the process inputs such that the prespecified measures of process performance are simultaneously maximized.

The ANN-DE strategy fulfills the stated objective in two steps. In the first step, an ANN-based process model is developed. This model has the inputs describing process operating parameters and variables (reactant concentration, temperature, pressure, etc.) and its outputs represent process output variables (EO production,
selectivity, etc.). In the second step of the ANN-DE procedure, the input space of the ANN model is optimized using a DE algorithm such that the optimized process inputs result in the enhanced values of the output variables.

This optimization problem can be formulated as:

Maximize:

$$\hat{F}_k = \sum_{k=1}^{K} W_k y_k = \sum_{k=1}^{K} W_k f_k(\mathbf{x}, w_k)$$

(1)

where $y_k$ denotes the $k$th output variable; $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is an $N$-dimensional vector of process operating variables, $f_k$ refers to the function correlating the $k$th output variable with the inputs, and $w_k$ represents the parameter vector of function, $f_k$. Eq. 1 describes a multi-objective (MO) optimization problem since it involves simultaneous maximization of $K$ outputs, $(y_k), k = 1, 2, \ldots, K$

Using the aggregation principle (also known as the “weighting objective method”), the MO optimization task can be converted into a single objective (SO) optimization by defining:

Maximize:

$$\hat{F} = \sum_{k=1}^{K} \hat{W}_k y_k = \sum_{k=1}^{K} \hat{W}_k f_k(\mathbf{x}, w_k)$$

(2)

where $\hat{F}$ denotes the single aggregated objective function and $\hat{W}_k$ represents the weighting coefficient. $\hat{W}_k$ signifies the relative importance of the $k$th function in the MO optimization (Eq. 1). The strategy fulfills the SO optimization task in two steps. In the first step, an ANN-based process model, $y_k = f_k(\mathbf{x}, w_k)$, is developed and in the second step, the process model input space, $\mathbf{x}$, is optimized using DEs with a view of maximizing the single aggregated objective function defined in Eq. 2.

ANN-based modeling. Neural networks are computer algorithms inspired by the way information is processed in the nervous system. An ANN is a massively parallel-distributed processor that has a natural propensity for storing experimental knowledge and making it available. An important difference between neural networks and standard regression is their ability to learn. This learning property has yielded a new generation of algorithms. An ANN paradigm is composed of several highly interconnected processing elements, analogous to biological neurons that are tied together with weighted connections analogous to synapses. Learning in biological systems involves adjustments to the synaptic connections between the neurons. This is true for ANNs as well. Learning typically occurs through training or exposure to a true set of input/output data where the training algorithm iteratively adjusts the connection weights. These connection weights represent the knowledge necessary to solve specific problems.

Network architecture. The back-propagation algorithm assumes a feed-forward neural network architecture (Fig. 1) where nodes are partitioned into layers numbered 0 to $L$. The lower most layer is the input layer numbered as layer 0, and the top most layer is the output layer numbered as layer $L$. Back-propagation addresses networks for which $L>=2$, containing “hidden layers” numbered 1 to $L$-1. Hidden nodes do not directly receive inputs from or send outputs to the external environment. Input layer nodes merely transmit input values to the hidden layer nodes and do not perform any computations. The number of input nodes equals the dimensionality of input patterns and the number of nodes in the output layer is dictated by the problem under consideration. Each hidden node and output node applies the activation function to its net input. Normally the following three types of activation functions are reported in literature:

Case 1: sigmoid function:

$$y[i] = \frac{1}{1+exp (-net[i])}$$

Case 2: tan hyperbolic function:

$$y[i] = \tan b (net[i])$$

Case 3: linear function:

$$y[i] = net[i]$$

where $y[i]$ is the output from node $i$ and $net[i]$ is the input to the node $i = \sum w[i]x[i]$.

Training. Training a network consists of an iterative process in which the network is given the desired inputs along with the correct outputs for those inputs. It then seeks to alter its weights to try and produce the correct output (within a reasonable error margin). If it succeeds, it has learned the training set and is ready to perform on previously unseen data. If it fails to produce the correct output it rereads the input and again tries to produce the correct output. The weights are slightly adjusted during each iteration through the training set (known as a training cycle) until the appropriate weights have been established. Depending upon the complexity of the task to be learned, many thousands of training cycles may be needed for the network to correctly identify the training set. Once the output is correct the weights can be used with the same network on unseen data to examine how well it performs.

Back-propagation algorithm (BPA). The BPA modifies network weights to minimize the mean-squared error between the network desired and the actual outputs. Back propagation uses
supervised learning in which the network is trained using data for which input as well as desired outputs are known. Once trained, the network weights are frozen and can be used to compute output values for new input samples.

The feed-forward process involves presenting input data to the input layer neurons that pass the input values onto the first hidden layer. Each hidden layer node computes a weighted sum of its input and passes the sum through its activation function and presents the result to the output layer. The goal is to find a set of weights that minimize mean-squared error. A typical back-propagation algorithm is given in appendix 1.

**Generalizability.** Neural learning is considered successful only if the system can perform well on test data on which the system has not been trained. This capability of a network is called generalizability. Given a large network, it is possible that repeated training iterations successively improve network performance on training data, e.g., by “memorizing” training samples, but the resulting network may perform poorly on test data (unseen data). This phenomenon is called “over training”. The proposed solution is to constantly monitor the network performance network on the test data. Hecht-Nielsen proposes that the weight should be adjusted only on the basis of the training set, but the error should be monitored on the test set. Here we apply the same strategy: training continues as long as the error on the test set continues to decrease and is terminated if the error on the test set increases. Training may thus be halted even if the network performance on the training set continues to improve.

**DE-based optimization of ANN models.** Having developed an ANN-based process model, a DE algorithm is used to optimize the N-dimensional input space, x, of the ANN model. DE, an improved version of GA, is an exceptionally simple evolution strategy that is significantly faster and robust at numerical optimization and is more likely to find a function’s true global optimum. Unlike simple GA that uses a binary coding for representing problem parameters, DE uses real coding of floating-point numbers. The mutation operator here is addition instead of bit-wise flipping used in GA. And DE uses nonuniform crossover and tournament selection operators to create new solution strings. Among the DE advantages are its simple structure, ease of use, speed and robustness. It can be used for optimizing functions with real variables and many local optima.

This article demonstrates the successful application of DE to the commercial EO reactor optimization problem. As already stated, DE in principle is similar to GA. So, as in GA, we use a population of points in our search for the optimum. The population size is denoted by NP. The dimension of each vector is denoted by D. The main operation is the NP number of competitions that are to be carried out to decide the next generation.

To start with, we have a population of NP vectors within the range of the objective function. We select one of these NP vectors as our target vector. We then randomly select two vectors from the population and find the difference between them (vector subtraction). This difference is multiplied by a factor F (specified at the start) and added to a third randomly selected vector. The result is called the noisy random vector. Subsequently, crossover is performed between the target vector and noisy random vector to produce the trial vector. Then, a competition between the trial and target vectors is performed and the winner is replaced into the population. The same procedure is carried out NP times to decide the next generation of vectors. This sequence is continued until some convergence criterion is met. This summarizes the basic procedure carried out in DE. The details of this procedure are described next.

**Steps performed in DE.** Assume that the objective function is of D dimensions and that it has to be optimized. The weighting constants, F, and the crossover constant, CR, are specified.

Step 1—Generate NP random vectors as the initial population: Generate (NP D) random numbers and linearize the range between 0 and 1 to cover the entire range of the function. From these (NP D) numbers, generate NP random vectors, each of dimension D, by mapping the random numbers over the range of the function.

Step 2—Choose a target vector from the population of size NP. First generate a random number between 0 and 1. From the random number value decide which population member is to be selected as the target vector, Xi, (a linear mapping rule can be used).

Step 3—Choose two vectors at random from the population and find the weighted difference: Generate two random numbers. Decide which two population members are to be selected (Xa, Xb). Find the vector difference between the two vectors (Xa – Xb). Multiply this difference by F to obtain the weighted difference. 

Weighted difference = F (Xa – Xb)

Step 4—Find the noisy random vector: Generate a random number. Choose a third random vector from the population, Xc. Add this vector to the weighted difference to obtain the noisy random vector, Xc’.

Step 5—Perform crossover between Xi and Xc’ to find Xt, the trial vector: Generate D random numbers. For each of the D dimensions, if the random number is greater than CR, copy the value from Xi into the trial vector; if the random number is less than CR, copy the value from Xc’ into the trial vector.

Step 6—Calculate the cost of the trial and the target vectors: For a minimization problem, calculate the function value directly and this is the cost. For a maximization problem, transform the objective function, f(x), using the rule, F(x) = 1/(1 + f(x)), and calculate the cost. Alternatively, directly calculate the value of f(x) and this yields the profit. In case cost is calculated, the vector that yields the lower cost replaces the population member in the initial population. In case profit is calculated, the vector with the greater profit replaces the population member in the initial population.

Steps 1–6 are continued until some stopping criterion is met. This may be of two kinds. One may be some convergence criterion that states that the error in the minimum or maximum between two previous generations should be less than some specified value. The other may be an upper bound on the number of generations. The stopping criterion may be a combination of the two. Either way, once the stopping criterion is met, the computations are terminated.

Choosing DE key parameters—NP, F, and CR is seldom difficult and some general guidelines are available. Normally, NP ought to be about 5 to 10 times the number of parameters in a vector. As for F, it lies in the range of 0.4 to 1.0. Initially F = 0.5 can be tried then F and/or NP is increased if the population converges prematurely. A good first choice for CR is 0.1, but in general CR should be as large as possible.

DE has been successfully applied for solving several complex problems and is now being identified as a potential source for accurate and faster optimization.

**Optimization of ANN model.** The optimization objective underlying the DE-based optimization of an ANN model is defined as: Find the N-dimensional optimal decision variable...
vector, $x^* = [x_1^*, x_2^* \ldots x_n^*]^T$ representing optimal process conditions such that it simultaneously maximizes process outputs, $y_k$; $k = 1, 2, \ldots, K$. The corresponding single objective function, $f$, to be maximized by the DE is defined in Eq. 2. In the DE procedure, the search for an optimal solution (decision) vector, $x^*$, begins from a randomly initialized population of probable (candidate) solutions. The solutions are then tested to measure their fitness in fulfilling the optimization objective. Implementing this DE algorithm and looping generates a new population of candidate solutions, which as compared to the previous population, usually fares better at fulfilling the optimization objective. The best vector that evolves after repeating the described loop until convergence forms the solution to the optimization problem. The stepwise procedure for the DE-based optimization of an ANN model is provided in Appendix 2 (also refer to the simplified flowchart in Fig. 2).

**Case study of ethylene oxide reactor.** An EO reactor is like a shell-and-tube heat exchanger where silver catalyst was put as a fixed bed in the tube side. Water is circulated through the shell side to remove the heat of reaction as both the reactions are exothermic. The conversion of ethylene to EO is very low, that’s why ethylene and oxygen are recycled back (Fig. 3). In an industrial setup, ethylene and oxygen are mixed with the cycle gas (a mixture of gas comprising ethylene, oxygen, methane, carbon dioxide, nitrogen, argon, ethylene oxide, etc.) and heated before feeding to the EO reactor. In the EO reactor, some of the ethylene and oxygen react over the catalyst surface and produce EO and undesirable CO$_2$ and water. The reactor outlet gas is fed to the EO absorber to practically absorb all the EO from cycle gas by a liquid absorbent. The EO free-cycle gas is then fed to a CO$_2$ absorber to absorb some of the CO$_2$ from the cycle gas before recycling it back to the reactor with the cycle-gas compressor. Process economics depend on the extent of the first reaction as compared to the second. The main objective of reactor optimization is to suppress the second reaction and promote the first reaction to facilitate valuable ethylene conversion to EO rather than wasting it to CO$_2$ and water.

**EO reaction process modeling and optimization.** The objective of the present case study is to model and optimize the industrial EO reactor to simultaneously maximize EO production (plant capacity) and catalyst selectivity. The main purpose of modeling of the EO reactor is to develop a mathematical equation that can predict the reactor outlet EO production, CO$_2$ production and catalyst selectivity if the reactor inlet feed compositions and process parameters (temperature and pressure) are given. The business objective of EO reactor optimization is to simultaneously maximize EO production rate (in mt/h) and catalyst selectivity. For commercial EO catalyst these two objectives are opposites, i.e., to maximize EO production catalyst selectivity will be minimized. Maintaining feasible highest catalyst selectivity while achieving maximum throughput is the challenge to glycol engineers. This article addresses this issue.
Unlike academic studies, this project was intended to optimize the actual operation of an industrial operating plant. That’s why the considerations and real issues to develop EO reactor models are quite complex and the constraints are unique to actual plant operations.

Developing of the ANN-based correlation. Developing the ANN-based correlation started with collecting a large databank. The next step was to perform a neural regression, and to validate it statistically.

Data collection. The quality and quantity of data are very crucial in ANN modeling since neural learning is primarily based on these data. An hourly average of actual plant operating data at steady state was collected for approximately one year. Data were checked and cleaned for obvious inaccuracies and those data are retained when plant operation was in steady-state and smooth. Finally 6,273 records are qualified for neural regression. This wide range of database includes plant operation data at various capacities starting from 35% capacity to 110% of design capacity. Also, these data capture plant operations at different catalyst ages.

Identifying input and output parameters. Reactor performance was monitored in terms of three reaction output variables, namely EO production (mt/h), CO₂ production (mt/h) and catalyst selectivity (%). Based on the operating experience in the glycol plant, all physical parameters that influence EO and CO₂ production and selectivity are put in a so-called “wish-list.”

Out of the number of inputs in the wish list, we used ANN regression to establish the best set of chosen inputs, which describes reaction behavior. The following criteria guide the choice of the set of inputs:

• The inputs should be as few as possible.
• Each input should be highly cross-correlated to the output parameter.
• These inputs should be weakly cross-correlated to each other.
• The selected input set should give the best output prediction, which is checked by using statistical analysis (e.g., average absolute relative error (AARE), standard deviation and cross-correlation coefficient).
• There should be minimum complexity in the neural network architecture, i.e., a minimum number of hidden layers.

While choosing the most expressive inputs, there is a compromise between the number of inputs and prediction. The cross-correlation analysis that signifies the strength of the linear relationship between input and output is then used to find the dependence between input and output. A number of inputs can be highly cross-correlated to output, but there should not be any strong dependency between these inputs; otherwise, it just adds to the complexity of the structure rather than contributing significantly to improving the quality of the network. One should be careful here: Although the cross-correlation analysis reveals the dependence between inputs and outputs, it also hides nonmonotonic relationships. This can result in losing an important input. Therefore, in this study, several sets of inputs were made and tested via rigorous trial-and-error on the ANN. The mentioned criteria were then used to identify the most pertinent set of input groups. Based on the analysis, the 10 input variables (in Table 1) have been finalized to predict EO and CO₂ production and catalyst selectivity.

Neural regression. For modeling purposes, the reaction operating conditions data (see Table 1) can be viewed as an example input matrix, X, of size (6,273 X 10), and the corresponding reaction output data as the example output matrix, Y, of size (6,273 X 3). For ANN training, each row of X represents a seven-dimensional input vector, \( \mathbf{x} = [x_1, x_2, \ldots, x_{10}] \), and the corresponding row of matrix Y denotes the three-dimensional desired (target) output vector \( \mathbf{y} = [y_1, y_2, y_3] \).

Since the magnitude of inputs and outputs greatly differ from each other, they are normalized in 0–1 scales using following relation:

\[
\mathbf{X}_{\text{normal}} = \frac{\mathbf{X} - \mathbf{X}_{\text{min}}}{\mathbf{X}_{\text{max}} - \mathbf{X}_{\text{min}}}
\]

To avoid the over-training phenomena described earlier, 80% of the total dataset was chosen randomly for training and the remaining 20% was selected for validation and testing.

It has been reported that multilayer ANN models with only one hidden layer are universal approximators. Hence, a three-layer, feed-forward neural network (like Fig. 1) is chosen as a regression model.

Since there is no previous idea about the suitability of the particular activation function, all three activation functions (sigmoid, tan hyperbolic and linear) are chosen in all combinations for both the hidden and output layers. The purpose is to find out which combination gives the lowest error.

The number of nodes in the hidden layer is up to the discretion of the network designer and generally depends on problem complexity. With too few nodes, the network may not be powerful enough for a given learning task. With a large number of nodes (and connections), computation is too expensive and time-consuming. In the present study, the optimum number of nodes is calculated by trial-and-error.

The statistical analysis of network prediction is based on the following performance criteria:

1. The AARE should be minimum:

\[
AARE = \frac{1}{N} \sum_{i=1}^{N} \left[ \frac{y_{\text{predicted}}(i) - y_{\text{experimental}}(i)}{y_{\text{experimental}}(i)} \right]
\]

2. The standard deviation, \( \sigma \), should be minimum:

\[
\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} \left( y_{\text{predicted}}(i) - y_{\text{experimental}}(i) - AARE \right)^2}
\]

<table>
<thead>
<tr>
<th>TABLE 1. Input and output variables for ANN model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input parameters</strong></td>
</tr>
<tr>
<td>Cycle gas flow, mt/h</td>
</tr>
<tr>
<td>Cycle gas pressure, bar g</td>
</tr>
<tr>
<td>Reactor temperature, °C</td>
</tr>
<tr>
<td>Reactor inlet ethylene concentration, mole %</td>
</tr>
<tr>
<td>Reactor inlet CO₂ concentration, mole %</td>
</tr>
<tr>
<td>Reactor inlet oxygen concentration, mole %</td>
</tr>
<tr>
<td>Reactor inlet ethylene concentration, mole %</td>
</tr>
<tr>
<td>Oxygen flow in reactor, mt/h</td>
</tr>
<tr>
<td>Inhibitor flow, kg/h</td>
</tr>
<tr>
<td>Catalyst running hours, h</td>
</tr>
</tbody>
</table>
2. The low and comparable training and test error values of standard deviation of error and the linear activation function at output nodes. The training set. The optimal MLP model that satisfied this criterion has optimal network model, the criterion used was least error (0.4% average error) can be considered as an excellent ANN model. Once developed, this ANN model can be used to quantitatively predict the effects of all input parameters on the EO production and catalyst selectivity.

ANN model DE-based optimization. After developing a successful ANN model of the EO reactor, the next step is to find out the best set of operating conditions that lead to maximum EO production and catalyst selectivity. While performing optimization of the input space of the ANN model, the best values of the following DE-specific parameters were chosen heuristically: population size, \( NP = 100 \), weighting constants, \( F = 0.5 \), the crossover constant, \( CR = 0.2 \) and maximum number of generations, \( Ngen = 300 \). To obtain the best set of operating conditions, DE runs were replicated several times, i.e., 50, using different random-number generator seeds. A different seed value generates a dissimilar population of initial candidate solutions, thus assisting in the exhaustive search of the solution space and thereby locating the globally optimum solution. For computing fitness values of the candidate solutions, EO production and selectivity were given equal weights of 50% each and the following fitness function was employed:

\[
F = \sum_{k=1}^{K} \hat{W}_k y_k = 0.5 y_1 + 0.5 y_2
\]

The three best operating condition sets given by the DE-based optimization of the ANN models are tabulated in Table 3. The three different cases in Table 3 are chosen at different plant capacity (case 1: oxygen load 63.15 mt/h, case 2: oxygen load 53 mt/h and case 3: oxygen load 39 mt/h). It is seen from the tabulated optimized values that the ANN-DE method has yielded an overall optimal solution maximizing both the EO production rate and

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
&m\text{odel} & \text{Training} & \text{Testing} & \text{Training} & \text{Testing} \\
\hline
\text{EO production} & 0.0040 & 0.0041 & 0.0005 & 0.0006 & 0.0048 & 0.0077 \\
\text{Selectivity} & 0.0057 & 0.0058 & 0.0005 & 0.0006 & 0.0057 & 0.0067 \\
\text{CO}_2 \text{ production} & 0.999 & 0.998 & 0.999 & 0.998 & 0.999 & 0.998 \\
\hline
\end{array}
\]
selectivity. From Table 3 some of the following rules-of-thumb are generated for the glycol plant operators and engineers. To maximize plant EO production and catalyst selectivity always run at maximum allowable reactor inlet oxygen concentration (8 mole % usually). Run at maximum possible cycle-gas pressure (21 barg) and run at minimum possible CO$_2$ concentration at the reactor inlet.

The optimum inhibitor flow, reactor temperature and ethylene concentration at the reactor inlet can be calculated on a real-time basis by this DE algorithm and they keep varying with plant capacity. Due to fast execution time of the DE algorithm, this program can be integrated with an online, real-time APC application or the optimum value may be calculated on an offline computer and the results may be transferred to the control panel operators to give them direction for optimizing the EO reactor.

**Actual vs optimum plant.** To evaluate the benefit of running the EO reactor at the optimum state, actual plant data were selected where the oxygen load varied from 61 to 61.5 mt/h. Since the EO reactor model is not available to control panel operators and glycol engineers, they try to optimize the EO reactor heuristically based on their past experience and knowledge. Because of this, the EO reactor is operated at nonoptimum operation and keeps varying from person to person. All the input and output parameter ranges are collected for oxygen loads of 61 to 61.5 mt/h and catalyst running hours 10,000 to 10,491 hr. The same ranges were applied to DE-based algorithm and optimum EO production and selectivity were calculated. Figs. 7 and 8 depict the actual vs optimum EO production and selectivity. Since the control panel operator has no definite knowledge of the optimum input values, there is a variation in EO production rate and selectivity.

**Table 3.** Optimum value of input variables calculated by DE algorithm

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalyst running hours, h</td>
<td>14,600</td>
<td>14,600</td>
<td>14,600</td>
</tr>
<tr>
<td>Cycle-gas flow, mt/h</td>
<td>1,450</td>
<td>1,450</td>
<td>1,450</td>
</tr>
<tr>
<td>Cycle-gas pressure, bar g</td>
<td>19.0</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>Reactor temperature, °C</td>
<td>230.00</td>
<td>233.00</td>
<td>233.00</td>
</tr>
<tr>
<td>Reactor inlet ethylene concentration, mole %</td>
<td>23.00</td>
<td>27.00</td>
<td>27.00</td>
</tr>
<tr>
<td>Reactor inlet CO$_2$ concentration, mole %</td>
<td>2.00</td>
<td>2.00</td>
<td>2.10</td>
</tr>
<tr>
<td>Reactor inlet oxygen concentration, mole %</td>
<td>7.50</td>
<td>8.00</td>
<td>8.00</td>
</tr>
<tr>
<td>Reactor inlet ethane concentration, mole %</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Inhibitor flow, kg/h</td>
<td>0.010</td>
<td>0.062</td>
<td>0.096</td>
</tr>
<tr>
<td>Oxygen flow, kg/h</td>
<td>30</td>
<td>63.15</td>
<td>63.15</td>
</tr>
<tr>
<td>Output</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EO production, mt/h</td>
<td>66.329</td>
<td>60.507</td>
<td>48.69</td>
</tr>
<tr>
<td>Selectivity</td>
<td>79.88</td>
<td>80.996</td>
<td>83.092</td>
</tr>
<tr>
<td>CO$_2$ production, mt/h</td>
<td>33.309</td>
<td>28.356</td>
<td>19.921</td>
</tr>
</tbody>
</table>

**Fig. 7** Actual vs optimum EO production.

**Fig. 8** Actual vs optimum catalyst selectivity.
activity as is clear from Figs. 7 and 8. The gap between optimum and actual value of EO production and selectivity in Figs. 7 and 8 is the benefit one could get by running the plant at optimum all the time. The benefit from delta EO production is found to be quite substantial when converted to monetary terms on an annual basis. After verifying all the calculations, the optimum input parameters were maintained in the actual plant and the benefit was found exactly same as calculated. This ensures the validation and accuracy of this calculation. **HP**

**LITERATURE CITED**

1. Tambe, S. S., B. D. Kulkarni and P. B. Deshpande, “Elements of Artificial Neural Networks with Selected Applications in Chemical Engineering, and Chemical and Biological Sciences,” *Simulations & Advanced Controls*, Louisville, Kentucky, 1996.

**Sandip kumar Lahir** is a production engineer and has 15 years’ technical services and operation experience in the petrochemical industry. He has an M.Tech qualification in chemical engineering and is currently doing research on multiphase flow and CFD. Mr. Lahir’s interests include simulation, optimization, APC and soft-sensor development.

**Nadeem Khalf** has seven years’ experience in petrochemical plants. He has a B.E. qualification in petrochemical engineering and experience in precommissioning, commissioning, troubleshooting and debottlenecking petrochemical plants.

**Musaad Al Garawi** is a senior process engineer and has seven years’ experience in the EOEG plant. He has a bachelor’s degree in chemical engineering and experience in troubleshooting and debottlenecking petrochemical plants.

**Appendix 1:** A typical back-propagation algorithm

While MSE is unsatisfactory and computational bounds are not exceeded, DO

For each input pattern \( x_p, 1 \leq p \leq N \),

Compute hidden node outputs:

\[
(W^{(1)} p, j) = \sum_{i=1}^{NP} w_{i,j} x_{p,i} \]

Compute hidden node outputs:

\[
(x^{(1)} p,j) = \sum_{i=1}^{NP} w^{(1,0)}_{i,j} x_{p,i} \]

Compute inputs to the output nodes:

\[
(W^{(2)} p,k) = \sum_{j=1}^{D} w^{(2,1)}_{j,k} x^{(1)} p,j \]

Compute the network outputs:

\[
(O^* p,k) = \sum_{j=1}^{D} w^{(2,1)}_{j,k} x^{(1)} p,j \]

Modify the weights between hidden and output nodes:

\[
\Delta W^{(2,1)}_{p,k} = \eta (D_{p,k} - O^* p,k) S' (net^{(2)} p,k) \times x^{(1)} p,j \]

End – for

End while

**Appendix 2:** Stepwise procedure for the DE-based optimization of an ANN model

Choose a strategy and a seed for the random number generator.

- Initialize the values of \( D, NP, CR, F \) and \( MAXGEN \).
- Initialize all the vectors of the population randomly. Since the upper bounds are all different for each variable in this problem, the variables are all normalized. Hence, generate a random number between 0 and 1 for all the design variables for initialization.

for \( i = 1 \) to \( NP \)

\{ for \( j = 1 \) to \( D \)

\( x_{i,j} = \text{random number} \}

- Evaluate the profit of each vector. Profit here is the hourly EO production, calculated by the ANN algorithm.

for \( i = 1 \) to \( NP \)

\( C_i = \text{EO production at } i \)

- Calculate by ANN \( C_i/ \)

- Find the vector with the highest profit, i.e., the best vector so far.

\( C_{\text{max}} = C_i \) and best = \( i \)

for \( i = 2 \) to \( NP \)

\{ if \( (C_i > C_{\text{max}}) \)

\( C_{\text{max}} = C_i \) and best = \( i \)

- Perform mutation, crossover, selection and evaluation of the objective function for a specified number of generations.

While \( (\text{gen} < \text{MAXGEN}) \)

\{ for \( i = 1 \) to \( NP \)
For each vector \( X_i \) (target vector), select three distinct vectors \( X_a, X_b \) and \( X_c \) randomly from the current population (primary array) other than the vector \( X_i \).

Do

\[
\begin{align*}
    r_1 &= \text{random number} \times \text{NP} \\
    r_2 &= \text{random number} \times \text{NP} \\
    r_3 &= \text{random number} \times \text{NP}
\end{align*}
\]

while \((r_1 = i) \text{ OR } (r_2 = i) \text{ OR } (r_3 = i) \text{ OR } (r_1 = r_2) \text{ OR } (r_2 = r_3) \text{ OR } (r_1 = r_3)\)

\( /* \ r_1, r_2 \text{ and } r_3 \text{ are randomly selected values from } [1, \ldots, \text{NP}] */ \)

- Perform crossover for each target vector \( X_i \) with its noisy vector \( X_{n,i} \) and create a trial vector, \( X_{t,i} \). The noisy vector is created by performing mutation. If \( CR = 0 \) inherit all the parameters from the target vector \( X_i \), except one which should be from \( X_{n,i} \).

for exponential crossover

\[
\begin{align*}
    &\{ p = \text{random number} \times 1 \\
    &\begin{aligned}
        r &= \text{random number} \times D \\
        n &= 0
    \end{aligned}
\end{align*}
\]

\[
\begin{align*}
    &\text{do} \\
    &\quad \{ X_{n,i} = X_{a,i} + F(X_{b,i} - X_{c,i}) \\
    &\quad /* \text{add two weighted vector differences for two vector perturbation. For best / random vector perturbation the weighted vector difference is added to the best / random vector of the current population. }*/ \\
    &\quad \begin{aligned}
        r &= (r+1) \oplus D \\
        \text{increment } r \text{ by } 1
    \end{aligned}
\end{align*}
\]

\[
\begin{align*}
    &\text{while } \left( (p < CR) \text{ and } (r < D) \right)
\end{align*}
\]

for binomial crossover

\[
\begin{align*}
    &\{ p = \text{random number} \times 1 \\
    &\begin{aligned}
        r &= \text{random number} \times D \\
        \text{for } n = 1 \text{ to } D
    \end{aligned}
\end{align*}
\]

\[
\begin{align*}
    &\left\{ \begin{array}{l}
        \text{if } \left( (p < CR) \text{ or } (p = D-1) \right) /* \text{change at least one parameter if } CR = 0 */ \\
        X_{n,i} = X_{a,i} + F(X_{b,i} - X_{c,i}) \\
        r = (r + 1) \% D
    \end{array} \right. \\
    &\}
\end{align*}
\]

if \( X_{n,i} > 1 \) \( X_{n,i} = 1 \)

/* for discrete function optimization check the values to restrict to the limits */

if \( X_{n,i} < 0 \) \( X_{n,i} = 0 \)

/* 1 - normalized upper bound; 0 - normalized lower bound */

- Perform selection for each target vector, \( X_i \), by comparing its profit with that of the trial vector, \( X_{t,i} \); whichever has the highest profit will survive for the next generation.

\( C_{t,i} = \text{EO production} /* \text{Calculated by ANN} */ \)

if \( C_{t,i} > C_i \) new \( X_i = X_{t,i} \)
else new \( X_i = X_i \) /* for \( i = 1 \) to \( \text{NP} */

- Print the results.