Modeling and optimization of Fischer–Tropsch synthesis in the presence of Co (III)/Al$_2$O$_3$ catalyst using artificial neural networks and genetic algorithm

Hooman Adib$^a$, Reza Haghbakhsh$^a$, Majid Saidi$^a$, Mohammad Ali Takassi$^b$, Fatemeh Sharifi$^c$, Mehdi Koolivand$^d$, Mohammad Reza Rahimpour$^{a,e,*}$, Simin Keshtkari$^a$

$^a$School of Chemical and Petroleum Engineering, Department of Chemical Engineering, Shiraz University, Shiraz 71345, Iran
$^b$Department of Petroleum Engineering, Petroleum University of Technology, Ahwaz, Iran
$^c$Computer Engineering Department, Isfahan University of Technology, Isfahan, Iran
$^d$Department of Petroleum, National Iranian South Oil Field Company, Ahwaz, Iran
$^e$Gas Center of Excellence, Shiraz University, Shiraz 71345, Iran

A R T I C L E   I N F O

Article history:
Received 12 August 2012
Received in revised form 9 September 2012
Accepted 17 September 2012
Available online

Keywords:
Artificial neural network
Genetic algorithm
Fischer–Tropsch synthesis
Catalyst

A B S T R A C T

Fischer–Tropsch synthesis is a collection of chemical reactions that converts a mixture of carbon monoxide and hydrogen into hydrocarbons. In this study, application of FTS is studied in a wide range of synthesis gas conversions. Artificial neural networks (ANN) were used to predict the molar percentage of CH$_4$, CO$_2$ and CO in the Fischer–Tropsch process of natural gas and also genetic algorithm (GA) was applied to obtain the optimum values of operational parameters. The input parameters consist of a 3-dimensions vector which includes the reaction time, operating pressure and temperature and also the output was molar percentage of CH$_4$, CO$_2$ and CO. Topology and decision parameters have been calculated by trial and error and acceptable correlation coefficients ($R^2 = 0.94$ for CH$_4$, $R^2 = 0.95$ for CO$_2$ and $R^2 = 0.96$ for CO) were obtained. Also the results obtained by sensitivity analysis represent that operation time has significant influence on molar percentage of CH$_4$ as desired product with respect to other operational parameters. Finally the results justify that GA-ANN could be effectively used for FTS as a powerful estimation technique.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Fischer–Tropsch synthesis

The Fischer–Tropsch synthesis (FTS) is a reaction for converting a mixture of hydrogen and carbon monoxide which derived from coal, methane or biomass to liquid fuels (Malek et al., 2011; Cano et al., 2011).

$$(2n + 1)H_2 + nCO \rightarrow C_nH_{2n+2} + nH_2O \quad (1)$$

The Fischer–Tropsch technology is applied for converting synthesis gas to long-chain hydrocarbons such as C$_3$ (Borg et al., 2008). Generally, the Fischer–Tropsch process is operated in the temperature range of 150–300 °C. This reaction is very sensitive respect to temperature, for example, increasing temperatures lead to faster reactions and higher conversion rates, but also tend to favor methane production. For this reason, the temperature is usually maintained at the low to middle part of that range. The common pressures range for this process is from one to several tens of atmospheres. Increasing the pressure leads to higher conversion rates and also favors formation of long-chained alkanes. Even higher pressures would be favorable, but higher pressures can lead to catalyst deactivation via coke formation and the benefits may not justify the additional costs of high-pressure equipment. This process occurs at the presence of cobalt catalyst which supported by refractory oxides such as alumina, silica, titanium, and etc (Karaca et al., 2011). According to the FTS feedstock and the type of desired product, some other catalysts, for example cobalt or iron catalysts are used in industrial processes (Smit et al., 2009). The activity and selectivity of catalyst which is used in Fischer–Tropsch reactions are dependent to some parameters such as the nature and the structure of catalyst support, the nature of active metal, metal dispersion, metal loading and the catalyst preparation methods (Malek et al., 2011).

Different metals are used as a catalyst in the FTS process, but nowadays, due to commercial consideration, Co and Fe catalyst are so common for this process (Cano et al., 2011). In the low hydrogen content synthesis gases, for example, the gas which earn from coal, iron is more useful respect to other catalyst, because it promotes the water–gas-shift reaction. Also cobalt–based catalysts have large

* Corresponding author. Tel.: +98 711 2303071; fax: +98 711 6287294.
E-mail address: rahimpour@shirazu.ac.ir (M.R. Rahimpour).
number of active site which is so effective on reactions rate. One of the metals which mostly was used as a support in FTS is alumina, since it has favorable mechanical properties, but an alumina-supported catalyst has a limited reducibility due to a strong interaction between the support and the cobalt oxides (Iglesia et al., 1993; Schanke et al., 1995).

Due to producing different types of gaseous, liquid and solid phase products, phenomenological modeling of FTS is very complicated (Sharma et al., 2011). In recent years, new modeling methods such as artificial neural network (ANN) based modeling approach, has opened a wide view for developing empirical models. The relations between catalytic performances (such as the selectivity of catalyst and the conversion of reactants) and the components of catalyst could be expressed effectively on the basis of properties of artificial neural network. Artificial neural network could be used as a predictor for any process and multi objective function systems. Especially for non-linear systems, application of ANNs is very effective respect to other typical modeling methods. In addition to ANN, genetic algorithm is another helpful optimizing method for finding the best inputs and outputs of the system.

1.2. Artificial neural network approach

In recent years, there has been an increasing interest in studying the mechanisms and structure of the brain. This fact led to development of artificial neural network (ANN) computational models for solving complex problems (Jamialahmadi and Javadpour, 2000). Numerous benefits have been made in developing artificial intelligent systems which is inspired by biological neural networks, fuzzy systems and combination of them. ANN is used to solve a variety of problems in the fields of optimization, perception, prediction and so on. Due to better performance, tolerance fantastic relationships in data and learns through experience, not from programming (Myshkin et al., 1997; Schooling et al., 1999). The ANN, the meaning of training is continuous adjusting of the connection weight functions until they reach distinctive values that help the network to produce sufficiently appropriate outputs which are close to the actual desired outputs. The accuracy of a developed model depends on its weight functions values. When optimum weights are achieved, the weights and biased values encode the network’s state of knowledge (Genel et al., 2003).

An ANN collects its knowledge by detecting the patterns and relationships in data and learns through experience, not from programming (Myshkin et al., 1997; Schooling et al., 1999). The major characteristic of ANNs is the simulation of complex and non-linear problems by employing a different number of non-linear processing elements, such as the nodes or neurons (Yuhong and Wenzin, 2009). The modeling process is similar to a “black-box” operation without any external information, therefore it is very complicated to recognize any logical relationships within the dataset used in ANN (Gyurova and Friedrich, 2011; Koolivand et al., 2011). In the overall structure of an ANN (Fig. 1), there are several layers of units (neurons), namely the input layer, hidden layer(s), and output layer. The number of neurons in the input and output layers are equal to the number of input and output variables. The numbers of layers in hidden section are different and it depends to the type of problem and can contain more than one layer (Myshkin et al., 1997; Haykin, 1999; Faussett, 1994; Tchaban et al., 1998). As shown in Fig. 2, basically the artificial neurons are a simple calculator that works in the following way. At the first, the inputs of the each neuron \((a_1, a_2, a_3)\) are multiplied by the corresponding weight functions that assigned to them \((w_{ij}, w_{i2}, w_{ij})\). For each model, the weights represent the model fitting parameters. Then the combined input \(c_j\) forms as a summation of products of neurons and weight functions.

\[
c_j = a_1w_{ij} + a_2w_{i2} + a_3w_{ij} \tag{2}
\]

There are lower and upper bounds for neuron’s input \(a_i\) and to ensure that neuron’s input does not exceed its maximum or minimum activation values, a neuron’s combined input must be put through an activation function (commonly a sigmoid function) that “compressed” it into the require deactivation value range (between 0 and 1). Sigmoid function is an exponential function that is mostly used as an activation function for hidden layer in neural network models. Then the inputs \(a_i\) transfer to the neurons in the above layers which are connected to that neuron. In summary, an artificial neuron is a non-linear function of its inputs and an ANN is a super position of simple non-linear functions (Pai et al., 2008; Zeng, 1998; Genel, 2004).

In the ANN, the meaning of training is continuous adjusting of the connection weight functions until they reach distinctive values that help the network to produce sufficiently appropriate outputs which are close to the actual desired outputs. The accuracy of a developed model depends on its weight functions values. When optimum weights are achieved, the weights and biased values encode the network’s state of knowledge (Genel et al., 2003).

According to the methods of data processing, artificial neural networks are divided to two classes, feed-forward and recurrent. Feed-forward neural network is an artificial neural network where connections between the units do not form a directed cycle while a recurrent neural network is a class of neural network where connections between units form a directed cycle. Also based on the methods of learning, some of ANNs employ supervised training, while others used unsupervised or self-organizing methods. From a theoretical point of view, supervised and unsupervised learning differ only in the causal structure of the model. In supervised learning, the model defines the effect of one set of observations which called inputs, on another set of observations, called outputs. In unsupervised learning, all the observations are assumed to be caused by latent variables, that is, the observations are assumed to be at the end of the causal chain (Poggio and Girosi, 1990a, 1990b).
1.2.1. Pseudo ANN code

The main steps of ANN can be described in the following pseudo code:

(1) Forward propagation:
   (a) For each input node i, given input $x_i$;
   (b) For each input node i:
       (c) Output$_i = x_i$;

(2) Hidden layers nodes j:
   (a) For each hidden neuron j:
       (b) Output$_j = \sum w_{kj} \cdot $output$_j$;

(3) Output layer neurons k:
   (a) For each output neuron k:
       (b) Output$_k = \sum w_{kj} \cdot $output$_j$;

1.3. Genetic algorithm

Genetic algorithm (GA) is an optimization method with stochastic global search that imitate natural biological evolution. GA is a global heuristic, optimization technique based on genetic principles which is developed by Holland. Genetic algorithm uses a different search space and solution space. The search space is the space of actual solutions. In GA application, any genotype must be transformed into the corresponding phenotype before its fitness be evaluated (Holland, 1975; Chen et al., 1996).

When GA is applied in a problem, a proper fitness measurement and representation must be designed. Many representations are possible, but some of them are more practical. Determining the termination criterion is the next step in GA. Predefined number of iterations and verifying whether an acceptable solution has been found are common termination criterion in GA. GA utilizes the rules of Darwinian evolution, which is the idea of the survival of the fittest, to optimize the goal functions. GA is useful for complex and non-linear optimization problems as they do not require any derivative information (Karr et al., 1995). In GA, Parameter sets that define during the model’s evaluations are converted to binary form which simulates chromosomes. These chromosomes are adapted by mutation, crossover of genetic material and differential reproductive success to search through parameter space for the optimum value of the chosen objective function. The fittest individuals are then used to generate the next generation of strings (Whitcombe et al., 2006). There are three operations in GA, such as selection, crossover and mutation. But, it is necessary to say that the ability of local searching in GA is weak. So it would lead to slow convergence or divergence (Huang et al., 2003). Indeed, in some problems that there are several local optimum points, GA algorithm could not reach the global optimum point and evaluation stops at local optimum points.

For choosing two parents offspring, a reproduce selection process is used by genetic algorithm operator which is called crossover. Crossover operator is the basic operator for producing new chromosomes in the GA. Crossover produces new individuals that have some parts of both parent’s genetic material. As shown in Fig. 3, the crossover separate the “parent” individuals and cut the chromosomes at a certain point, then changes the genes of the two parents from split point.

Crossover can generate a very large amount of different strings. However, depending on the initial population (initial guess) chosen, there may not be enough variety in the strings to ensure that the GA covers the entire problem space. Also due to an unsuitable initial population guess, the GA may find itself converging on strings that are not quite close to the optimum point. So mutation has a secondary role in the simple GA operation and some of those problems are overcome by introducing a mutation operator into the GA. Mutation is randomly applied with low probability, typically in the range 0.001 and 0.01, and modifies elements in the chromosomes. The role of mutation is often seen as providing a guarantee that the probability of selecting any given string will never be zero. Also mutation acts as a safety net to recover good genetic material that may be lost through the action of selection and crossover. Mutation probability dictates the occurrence of mutation frequency. It is much lower than crossover probability. In simple GA mutation, a random bit is selected and the gene’s value will change to a new one (1s is changed to 0s and 0s is changed to 1s). After mutation, the fitness will be evaluated and the old generation will be replaced completely or partially. This process is repeated until the loop termination condition is satisfied. Loop termination condition is the predetermined number of generations. In some cases, mutation will be repeated until the standard deviation among the fitness levels converges toward zero. All of these steps are shown in Fig. 4.

1.3.1. Pseudo GA code

The main steps of GA can be described in the following pseudo code:

(1) Encoding decision variables in a string called chromosomes;
(2) Creating an initial random population;
(3) Calculation of each individual’s fitness in the current population;
(4) Selecting high fitness value solutions (better chromosomes) for mating to enhance the chance of passing their genes to the next generation;
(5) Running crossover operators (generate new strings from the best chromosomes);
(6) Running mutation operators (randomly alter one or more genes of a selected string to help protect the loss of some of the good solutions);
(7) If the stopping criteria are not met, go to step (3);
(8) The best solution is found

1.4. Genetically optimization of ANN

Many investigators indicated the requirements of applying GAs in optimizing ANNs’ parameters (Hegazy et al., 1994). Especially, when the exploration space of the ANN is so extensive, GA is found to be quite useful and efficient in these cases. Also, those who...
supported this proposal were in favor of optimizing the connection weights and the architecture of ANNs using GA (Van Rooij et al., 1996; Vonk et al., 1997; Miller et al., 1989; Marshall and Harrison, 1991). In the present work, GA is used for optimizing the parameters of ANNs (the number of neurons in the hidden layer, the coefficient of the learning rate and the momentum). More details are shown in Fig. 5 (Saemi et al., 2007). Because the fitness value must be evaluated for every chromosome, therefore in each generation, chromosomes population size and number of generations have effects on training time. There are no general rules for determining the population size, but generally population sizes of 50–100 are used in GA research. Once the population size is chosen, the initial population is randomly generated. Hence in this study, the initial population size is set to 60 chromosomes and every chromosome in a population developed into new chromosomes for 100 generations.

In this paper, a chromosome is represented by a binary string as in feature selection. Genetic operators modify individuals within a population to produce a new individual for testing, evaluation and achieving optimum point. The crossover operator produce two new chromosomes by swapping some bits takes from parent’s chromosomes, therefore the search space become larger and the process of reaching the optimal solution will be accelerated. After the crossover and mutation operations, a new population is generated and fitness function will be evaluated (Salehi et al., 2011).

2. Experiments

2.1. Catalyst preparation

One-tenth mole of ammonium heptamolybdate (Aldrich) was dissolved in 1 L of distilled water. The color of the solution was white. Then the aluminum oxide (Merck135mesh) catalyst support was added to the solution as a proportion of 11 g of alumina per 3 g of the complex. The solution was mixed by a high speed mechanical agitator for 10 h until heptamolybdate anion was chemisorbed on the surface of aluminum oxide particles. Deposition of complex anion on the catalyst support is a very important step in precatalyst formation. At this time, 0.2 mol of hexaammoniumcobalt (III) chloride (Aldrich) \([\text{NH}_3]^6\text{Co}^3+\text{Cl}_3\) complex (burnt orange to red crystals) was dissolved in sufficient distilled water. Next, as the solution was stirring, the cobalt complex was added drop-wise to the solution. The stirring continued for more than 5 h. Formation of an evenly pale pink color on the catalyst support and a colorless solution indicated that the following reaction had occurred.

\[
2\left(\text{NH}_3\right)_6\text{Co}^3+\text{Cl}_3 + \left(\text{NH}_4\right)_6\text{Mo}_7\text{O}_{24}\text{Cl}_3 \rightarrow \text{Al}_2\text{O}_3 + 6\text{NH}_4\text{Cl} + \left(\text{NH}_3\right)_6\text{Co}^2+2\left[\text{Mo}_7\text{O}_{24}\text{Cl}_3\right]^\text{3+} \geq \text{Al}_2\text{O}_3
\]  

(3)

Then the pre-catalyst filtration was done. The colorless products indicated that no cobalt complex remained in the solution and all of them had reacted with the molybdate complex on the catalyst support. The pre-catalyst was washed with distilled water to remove all ionic co-products. The pre-catalyst was insoluble and inactive in water at 273–373 K. The pre-catalyst was quietly dried in an oven. The formation of the pink color \([\text{NH}_3\text{Co}]_6\text{Mo}_7\text{O}_{24}\text{Al}_2\text{O}_3\) pre-catalyst was demonstrated by two indicators. First is the developed color of the alumina using potassium bromide disc infrared spectroscopy, which indicated the presence of \((\text{NH}_3\text{Co})^3+\) ion by NH3 spreading modes and the second is the Mo=O units of Mo=O32+ ion by strong Mo=O stretching absorption. The partial reduction of cobalt/molybdate pre-catalyst was carried...
out using hydrogen at a pressure of 40 bars and at a temperature of 700°K for 5 h in a batch reactor. During reduction, the color of the pre-catalyst changed from pale pink to dark black.

The following procedure for carbon monoxide hydrogenation reaction was followed: 100 g of the partially reduced catalyst was placed in the reactor. Then the reactor was allowed to equilibrate at desired temperature. After injection of carbon monoxide and hydrogen to the reactor with certain composition and pressure, the stirring motor was turned on. After the selected time, the product was passed through a condenser to condense out the water vapor. The condenser was cooled by a salt and ice mixture. Then the product was stored in the sample collector for gas chromatography analysis. The chromatograph was used to analyze the products. A Varian Aerograph Model 90 P with a carbosieve B 60/80 mesh column, thermal conductivity detector (TCD) and helium as carrier gas were used for CO, CO₂, and CH₄ analysis. At the end, the water was weighed and there was no oily material above the condensed water.

2.2. Reactor

The catalyst analysis was carried out in 1 L volume stainless steel autoclave reactor. The autoclave reactor is convenient to use at medium to high pressure 150 bars and at temperature up to 350 °C. The input and output gas lines were made using 316 stainless steel tubing. This reactor was equipped with electrical heater, magnetic stirring motor, and magnetic stirrer. The magnetic stirring motor was driven by air flow. Also the temperature of the reactor was controlled by a F2M (Scientific 240 temperature programmer) thermocouple model (Hewlett Packard).

3. Results and discussion

In present study the entire data have been obtained in the chemistry laboratory. Carbon monoxide and hydrogen were injected in the autoclave reactor with certain ratio. In this reactor, Co(III)/Al₂O₃ is used as a catalyst. The effects of resident time, pressure and temperature on the formation of product were investigated. 6 dataset including 45 data in each set have been used for input/output of ANN.

For developing ANN model, model input data were resident time, pressure and temperature of the reactor. Also the molar percentages of CH₄, CO₂ and CO have been used as network outputs. Table 1 represents the range of the input/output data which are used for developing ANN model. Also Figs. 6–11 present the three-dimensional plots of some of experimental data which is used for developing ANN model.

3.1. Trial and error method

Trial and error is a basic method for training a neural network. It means that in ANN, if the accuracy of a network decreases, then another network with different specification will be substituted and tested. In this study, a 5-8-1 architecture network provided the best model fitting for molar percentage of CH₄ prediction in terms of MSE. So it means that in 5-8-1 architecture, there are 5, 8, and 1 neuron in input, hidden and output layers, respectively. Using trial and error in the evaluations, the least cross validation error and correlation coefficient between the predicted and actual data points of %CH₄ were 0.052 and 0.94 respectively. It must be mentioned that the ANN predictions are in optimum point if the values of $R^2$ and MSE are found to be close to 1 and 0, respectively.

By using trial and error method and changing the initial parameters, a 4-7-1 ANN model architecture provides the best prediction of the molar percent of CO₂ in terms of MSE. Since the least cross validation error and the best correlation coefficient were 0.035 and 0.93 respectively, this model can be considered as the appropriate network. Also in prediction of molar percent of CO, the best ANN model structure consisted of a 4-9-1 architecture which provided the least cross-validation error of 0.023. The value of correlation coefficient of this network was 0.96 which deserved as an acceptable value for an ANN-based model. In order to build the ANN model, coding was developed using MATLAB software and the standard functions of neural network toolbox were used.

For optimizing the neural network architecture, the computations started using one neuron in the hidden layer as the initial guess. Then by increasing the number of neurons, the performance function MSE was calculated as shown in Figs. 12–14.

<table>
<thead>
<tr>
<th>Measured parameters</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>9.1</td>
<td>48.6</td>
</tr>
<tr>
<td>T</td>
<td>150</td>
<td>350</td>
</tr>
<tr>
<td>t</td>
<td>0.5</td>
<td>36</td>
</tr>
<tr>
<td>% CH₄</td>
<td>24</td>
<td>94</td>
</tr>
<tr>
<td>% CO₂</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>% CO</td>
<td>6</td>
<td>100</td>
</tr>
</tbody>
</table>
The aim of using ANN as a practical approach model was to consider the ability of model to predict the molar percentage of CH₄ (Parlak et al., 2006). As shown in Figs. 15–17, predicted molar percentages of CH₄, CO₂ and CO using artificial neural networks are in appropriate agreement with experimental results.

The coefficient of determination ($R^2$), mean squared error (MSE) and mean absolute error (MAE) were applied to choose the best number of layers and neurons in ANN. These statistical expressions calculations which defined by Shoo et al. are shown in equations (5)–(7) (Bera et al., 2005; Sahoo and Ray, 2006). The ANN predictions are in optimum point if the values of $R^2$, MAE and MSE are found to be close to 1, 0 and 0, respectively.

\[
R^2 = 1 - \frac{\sum_{i=1}^{N}(O_i - T_i)^2}{\sum_{i=1}^{N}(O_i - T_m)^2}
\]

(4)

\[
\text{MAE} = \frac{\sum_{i=1}^{N}(O_i - T_i)}{N}
\]

(5)

\[
\text{MSE} = \frac{\sum_{i=1}^{N}(O_i - T_i)^2}{N}
\]

(6)

where $i$ and $m$ are the actual desired output and the network output and $N$ is the number of data used for the network.

Table 2 shows the comparison between ANN performance and experimental results of CH₄, CO₂ and CO based on MSE, MAE and $R^2$.

### 3.2. Genetic algorithm optimization of ANN (GA-ANN) method

The structure of ANNs has significant impact on their efficiency. If the number of neurons is too little, they cannot learn the input and output variables in training. If the number of nodes is so high, network performance possibly improved, but structural complexities enhanced the training time and increases computation burden. Thus we need to create networks, which are as simple as possible, provided their errors are within the tolerance limits.

In this study, the logistic sigmoid function, which is an appropriate choice for many non-linear functions, was employed as the activation function, while the delta rule was applied as the error correcting rule. The delta rule is a gradient descent learning rule for

![Fig. 8. Output composition of CO₂ versus input residence time and pressure.](image)

![Fig. 9. Output composition of CO₂ versus input residence time and temperature.](image)

![Fig. 10. Output composition of CH₄ versus input residence time and temperature.](image)

![Fig. 11. Output composition of CO versus residence time and pressure.](image)
updating the weights of the artificial neurons in a single-layer experience. Also the back propagation algorithm was constructed as the learning algorithm to adapt the weight functions (Callan, 1999). For achieving best results of algorithms, the input data were normalized to the range $[-1, +1]$ by the 'minmax' Matlab function before the training. For input-hidden layers, the transfer function is sigmoid 'tansig' which is a hyperbolic tangent sigmoid transfer function, and also for output layer, linear 'purelin' is applied as a transfer functions. The transfer function of input-hidden layers updates weight and bias values according to the 'resilient back propagation algorithm'. Wang et al. (1996) showed that a feed-forward neural network with one hidden layer which employed sigmoidal basis functions is capable for approximating many non-linear functions. The sigmoidal basis function and input transformation are defined as equations (7) and (8) respectively.

$$
\varphi(z) = \frac{1}{1 + \exp(-az)}
$$

(7)
\[ z = \sum_{j=1}^{P} \beta_j x_j + \beta_0 \]  

(8)

For different experimental data, initial weight functions generated using random function in the interval of \(-0.1\) to \(+0.1\). Each network topology which was trained at an average of three times, gave similar results. In this study, the best simulating and generalizing experimental model results was searched by starting from small networks (small number of weights) and then enlarging the networks until the best model without over-fitting was achieved. Enlarging a network was done by adding more neurons to the existing two hidden layers of the network.

Process of phenotypic fitness measurement, selection, crossover recombination and mutation was iterated through 100 generations and the network with the lowest error was designated as the optimal evolved network. Figs. 18 and 19 illustrate the best fitness and the average fitness values for all the data. According to Table 3, MSE for the best and average fitness is 0.002 and 0.003 respectively.

The entire dataset (6 dataset, including 45 data in each set) were divided into three sets with different length: the first set (60% of dataset) was used for model building or “training”, the second set (20% of dataset) for cross validation and the third set (20% of dataset) for test the model. The target was to build a model that can predict the result of this experiment for different conditions. GA-MLP which is a combination of genetic algorithm and multi-layer perceptron is used to select the optimal subset of variables. A multi-layer perceptron (MLP) is a feed-forward artificial neural network model that maps sets of input data into a set of appropriate output. Parameter values which are employed are summarized in Table 4. The execution time was less than 15 min for a Pentium IV 1.86 MHz processor. It shows that this method is not time consuming. The optimum point is achievable mostly by running the algorithm just one time.

The results of optimization are illustrated in Figs. 20–22. As it is shown in these figures the GA-ANN results are in perfect match with experimental data. Comparison of the prediction of the GA-ANN model with the trial and error approach model showed that GA-ANN model significantly outperforms to trial-and-error ANN approach. On the other words, GA found to be a good alternative over the trial-and-error approach to determine the optimal ANN architecture and internal parameters quickly and efficiently. The performance of GA-ANN model on the test sets which is reported in Table 5 shows that the neural network model incorporating a GA is

<table>
<thead>
<tr>
<th>Table 2</th>
<th>ANN model performance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN performance</td>
<td>% molar of CH4</td>
</tr>
<tr>
<td>MSE</td>
<td>0.052</td>
</tr>
<tr>
<td>MAE</td>
<td>0.199</td>
</tr>
<tr>
<td>(R^2)</td>
<td>0.94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>The best and average fitness of GA.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization summary</td>
<td>Best fitness</td>
</tr>
<tr>
<td>Generation #</td>
<td>21</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.002</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>GA parameters for the GA-MLP model.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>Maximum number of generations</td>
<td>100</td>
</tr>
<tr>
<td>Maximum number of consecutive generations for which no improvement is observed</td>
<td>10</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.9</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.01</td>
</tr>
<tr>
<td>Cluster centers</td>
<td>15</td>
</tr>
</tbody>
</table>

Fig. 18. Average fitness (MSE) versus generation.

Fig. 19. Best fitness (MSE) versus generation of GA-ANN.

Fig. 20. Comparison of GA-ANN prediction and actual molar percentage of CH4.
able to sufficiently estimate the output variables of the Fischer–Tropsch synthesis with high accuracy.

3.3. Sensitivity analysis

Through sensitivity analysis, dependency of model outputs to model inputs verified and also information about the importance of the input variables and model parameters obtained. Sensitivity analysis information is useful for interpretation and assessing the modeling results. The input variables and model parameters include a lot of uncertainty arising from different sources. This uncertainty influence through the model output variables.

In neural networks, sensitivity analysis could be used for determination of the relative significance of model input

<table>
<thead>
<tr>
<th>GA-ANN performance</th>
<th>% molar of CH4</th>
<th>% molar of CO2</th>
<th>% molar of CO</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>0.002</td>
<td>0.005</td>
<td>0.003</td>
</tr>
<tr>
<td>MAE</td>
<td>0.091</td>
<td>0.019</td>
<td>0.016</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Fig. 21. Comparison of GA-ANN prediction and actual molar percentage of CO2.

Fig. 22. Comparison of GA-ANN prediction and actual molar percentage of CO.

Fig. 23. Sensitivity analysis about the mean percent of CH4 output.

Fig. 24. (a) Effect of temperature on conversion of CH4, (b) effect of temperature on conversion of CO2, (c) effect of temperature on conversion of CO.
parameters and ranking them in order of importance. As a result, some variables which could be ignored respect to significant parameters during the analysis will be identified.

Fig. 23 shows the sensitivity analysis about the molar percent of CH$_4$ output with respect to input parameters of the network such as temperature, pressure and time. Sensitivity analysis is done for pressure range of [22–41] bar, temperature range of [180–300] °C and between [0–17] h time periods. The results indicate that time is the most influential (sensitive) parameter affecting the values of the conversion of CH$_4$ in the train and full datasets of experiments.

Figs. 24–26 show the effect of these parameters on molar percent of CH$_4$, CO$_2$ and CO where in each case the other two parameters are kept constant. As shown in these figures, by increasing pressure, temperature and time, the rate of production of CH$_4$ increased, but the effect of time is more sensible than other variables.

Fig. 25. (a) Effect of pressure on conversion of CH$_4$, (b) effect of pressure on conversion of CO$_2$, (c) effect of pressure on conversion of CO.

Fig. 26. (a) Effect of time on conversion of CH$_4$, (b) effect of time on conversion of CO$_2$, (c) effect of time on conversion of CO.

4. Conclusion

This study demonstrates the applicability and feasibility of GA-ANN models to prediction and optimization of Fischer–Tropsch synthesis of natural gas in the presence of Co (III)/Al$_2$O$_3$ catalyst. GA was used to determine the optimal architecture and parameters of the ANN models.

Comparison of the prediction performance efficiency of the GA-ANN model with that of trial and error approach represents that neural network model incorporating genetic algorithm is able to sufficiently estimate the output composition of Fischer–Tropsch synthesis in presence of Co (III)/Al$_2$O$_3$ catalyst with the least values of error and higher correlation coefficient. Also in this study, sensitivity analysis showed that among input parameters, operation time has significant effect on production of CH$_4$ as desired product with respect to other operational parameters. Finally, it can be concluded that such GA-ANN model can be used to predict the
output composition of Fischer–Tropsch synthesis as a reliable estimation method with high accuracy.

**Nomenclature**

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>linear correlation coefficient</td>
</tr>
<tr>
<td>$RMSE$</td>
<td>root mean square error</td>
</tr>
<tr>
<td>$NMSE$</td>
<td>normalized mean-squared error</td>
</tr>
<tr>
<td>$mu_{ma}$</td>
<td>maximum of $mu$</td>
</tr>
<tr>
<td>$mu_{dec}$</td>
<td>$mu$ decrease factor</td>
</tr>
<tr>
<td>$mu_{inc}$</td>
<td>$mu$ increase factor</td>
</tr>
<tr>
<td>$mu_{ma}$</td>
<td>maximum of $mu$</td>
</tr>
<tr>
<td>$Max_{fail}$</td>
<td>maximum validation failures</td>
</tr>
<tr>
<td>$mem_{reduce}$</td>
<td>speed tradeoff factor</td>
</tr>
<tr>
<td>$MLP$</td>
<td>multilayer perception</td>
</tr>
<tr>
<td>$MSE$</td>
<td>mean square error</td>
</tr>
<tr>
<td>$Mu$</td>
<td>momentum</td>
</tr>
<tr>
<td>$mu_{dec}$</td>
<td>$mu$ decrease factor</td>
</tr>
<tr>
<td>$mu_{inc}$</td>
<td>$mu$ increase factor</td>
</tr>
<tr>
<td>$NMSE$</td>
<td>normalized mean-squared error</td>
</tr>
<tr>
<td>$RMSE$</td>
<td>root mean square error</td>
</tr>
<tr>
<td>$R^2$</td>
<td>coefficient of determination</td>
</tr>
</tbody>
</table>

**Greek letters**

| $A$ | linear parameter |
| $\alpha^*$ | best-fit of linear parameter |
| $B$ | non-linear parameter |
| $\delta^*$ | best-fit of non-linear parameter |
| $H$ | learning rate |
| $\Theta$ | threshold (bias) |
| $\sigma^2$ | variance of experimental data |
| $\Phi$ | activation function (basis function) |

**Abbreviations**

| AI | artificial intelligence |
| ANN | artificial neural networks |
| FTS | Fischer–Tropsch synthesis |
| GA | genetic algorithm |
| MAE | mean absolute error |
| $Max_{fail}$ | maximum validation failures |
| $mem_{reduce}$ | speed tradeoff factor |
| MLP | multilayer perception |
| MSE | mean square error |
| $mu$ | momentum |
| $mu_{dec}$ | $mu$ decrease factor |
| $mu_{inc}$ | $mu$ increase factor |
| $mu_{ma}$ | maximum of $mu$ |
| NMSE | normalized mean-squared error |
| RMSE | root mean square error |
| $R$ | linear correlation coefficient |
| $R^2$ | coefficient of determination |

**References**


