

A comparative study of artificial neural network (ANN) and adaptive neuro-fuzzy inference system (ANFIS) models in distribution system with nondeterministic inputs

Modestus O Okwu¹ and Olufemi Adetunji²

Abstract

Most deterministic optimization models use average values of nondeterministic variables as their inputs. It is, therefore, expected that a model that can accept the distribution of a random variable, while this may involve some more computational complexity, would likely produce better results than the model using the average value. Artificial neural network (ANN) is a standard technique for solving complex stochastic problems. In this research, ANN and adaptive neuro-fuzzy inference system (ANFIS) have been implemented for modeling and optimizing product distribution in a multi-echelon transshipment system. Two input parameters, product demand and unit cost of shipment, are considered nondeterministic in this problem. The solutions of ANFIS and ANN were compared to that of the classical transshipment model. The optimal total cost of distribution using the classical model within the period of investigation was 6,332,304.00. In the search for a better solution, an ANN model was trained, tested, and validated. This approach reduced the cost to 4,170,500.00. ANFIS approach reduced the cost to 4,053,661. This implies that 34% of the current operational cost was saved using the ANN model, while 36% was saved using the ANFIS model. This suggests that the result obtained from the ANFIS model also seems marginally better than that of the ANN. Also, the ANFIS model is capable of adjusting the values of input and output variables and parameters to obtain a more robust solution.

Keywords

Meta-heuristics, nondeterministic input, artificial neural network, ANFIS, transshipment, fuzzy

Date received: 22 December 2017; accepted: 1 March 2018

Background

Transshipment problem allows the movement of goods from some source points to some destination points, with the possibility of transshipping through some other source or destination points if it would provide cost benefit. This model is a further generalization of the transportation model because it relaxes the assumption that movement can only be made to a destination point from one or more supply point/s without passing through an intermediate point. This model should be capable of producing a better solution than that of typical transportation problem, if such exists, but it is also known to be more difficult to solve.

The transshipment problem itself is deterministic, and being non-deterministic polynomial-time (NP) hard,

¹ Department of Mechanical Engineering, Federal University of Petroleum Resources, Warri, Delta State, Nigeria

² Department of Industrial and Systems Engineering, University of Pretoria, Pretoria, South Africa

Corresponding author:

Modestus O Okwu, Department of Mechanical Engineering, College of Technology, Federal University of Petroleum Resources, Effurun (FUPRE), P.M.B. 1221, Effurun, Delta State, Nigeria.
Email: mechanicalmodestus@yahoo.com



solving it is a challenge on its own. Making some of the input parameters nondeterministic makes its solution even more arduous. However, the deterministic transshipment model only uses approximate values of the model parameters, and the extension of the problem to that with some nondeterministic variables is expected to produce a better solution, if tractable. This is what has been done in many stochastic models, and the marginal difference is usually referred to as the value of the stochastic solution.¹

Representing nondeterminism in input parameters is another interesting issue. While traditionally the nondeterministic variables of most models have been represented in stochastic forms, recent advances in artificial intelligence have brought the use of fuzzy models to prominence. Solutions to stochastic models of linear programming problems (LPPs), for instance, use various techniques that exploit the block separability of the variables, while fuzzy representation makes use of fuzzy operations and fuzzy arithmetic theories. A popular approach of fuzzy logic is based on the fuzzy decomposition theorem and utilizes the properties of alpha cuts along various sections of the fuzzy set. Birge and Louveaux¹ and Klir and Yuan² are good readings in order to compare the stochastic LPP and fuzzy modeling and solution approaches to problems.

Finding solution to NP hard problems in itself is difficult; hence, finding solutions to transshipment problems with nondeterministic input parameters could be even more challenging. A number of solutions have been proposed in each case. The main difference in their representation is that while a stochastic model uses an indicator function to allocate a variable to the most likely class in its set based on the estimate of the likelihood of its membership of each set, fuzzy logic assumes that the variable is a member of all of the sets, but by different likelihood of membership. It uses a function ranging from zero to unity to represent the membership across its different sets and uses a different type of set operations from those of probability models to find the variables' joint membership values. Gaines³ tried to explain the difference between fuzzy logic and stochastic process in relationship with probability logic, where stochastic processes assume statistical independence and fuzzy logic is purely logical inference.

This article presents solutions to a transshipment problem with nondeterministic input variables in order to estimate the value of the nondeterministic solutions. The demand and unit cost of product distribution were considered to be nondeterministic and solved through adaptive models. Artificial neural network (ANN) model was used to solve the model with stochastic demand and cost variables, while adaptive neuro-fuzzy inference system (ANFIS) was used to solve the same problem with both variables considered fuzzy. The solutions obtained in both cases were then compared. The main difference in these model solutions is in the training paradigm. The data in both cases were separated into training and test samples, and the performance of the test samples in both cases was

compared to that of the traditional solution approach. While the manufacturing area seems replete with comparison of these different paradigms of solution in diverse contexts, no author seems to be known in logistics functions, and in transshipment of goods in particular, that has compared the quality of solutions obtained from these two paradigms.

This article is further sectioned as follows: the next section is a presentation of the articles comparing the performance of ANN models to those of ANFIS models in various contexts. The third section is a presentation of the ANN and ANFIS models of the transshipment problem with the two nondeterministic input variables (demand and cost). In the fourth section, the ANN and the ANFIS models were trained, and the levels of fitness achieved were presented. The final section discusses the result of the test sample on the two trained models.

Comparative performance of ANN and ANFIS

ANN and ANFIS performances have been compared in many instances, but mostly in mechanical systems. He et al.⁴ used both ANN and ANFIS to predict the flow of river in a small river basin in a semiarid mountain region, where the actual mathematical relationships were not fully understood, but the results were observable under diverse conditions. In the study, support vector machine (SVM), ANFIS, and ANN methods were compared. The values of coefficient of correlation (R), root mean squared error, Nash–Sutcliffe efficiency coefficient, and mean absolute relative error were used in assessing the strengths of the three modeling paradigms. The outcome suggests that the three techniques were good enough, while SVM outperforms both ANN and ANFIS. Anari et al.⁵ did a study on infiltration rate using ANN, ANFIS, local linear regression (LLR), and dynamic LLR and observed that ANFIS appeared to have performed better than all other three techniques.

Do and Chen⁶ used ANN and ANFIS to forecast post-admission evaluation of students using some preadmission parameters as inputs. They claimed that ANFIS produced better results than ANN. Kamali and Binesh⁷ used ANN and ANFIS to study the diffusion of water through nanotubes using the molecular dynamics data. They also experimented with data set of different sizes and concluded that ANFIS outperformed ANN. Sredanovic and Dica⁸ used both ANN and ANFIS to predict the strength of minimal quantity lubrication and high-pressure cooling in the turning task. Both were said to have performed well. Azeez et al.⁹ compared the performance of ANN and ANFIS in the triage of emergency patients using various vital signs of patients as input parameters. Data were collected from a Malaysian University Hospital. They used ANN backpropagation technique for training, testing, and validation of data set. Fuzzy rules were developed with alpha cuts for

ANFIS model, and fuzzy subtractive clustering was used to find the fuzzy rules for the ANFIS model. ANN model gave a better result.

Esen¹⁰ conducted experiment on heat pump system by comparing the performance of ANFIS and ANN in the upright ground structure. The author stated preference for ANFIS over ANN. Vieira¹¹ modeled the temperature evolution of a kiln using both the feed-forward ANN and ANFIS system and concluded that ANFIS produced results with less error than the ANN model. Kisi and Ay¹² compared the performance of radial basis ANN with ANFIS in the study of dissolved oxygen in evaluating water quality. It was established that the ANN model outpaced the ANFIS model. Farahmand et al.¹³ analyzed the temporal distribution of groundwater contaminants in Iran. They found that ANFIS model outperformed the ANN models, having higher efficiency of estimation. Research areas in other fields of comparative study of ANN and ANFIS are available in the literature but it cannot be said emphatically that either technique always performs better than the other. The performance depends on the context.

There have also been cases where the performance of ANN or ANFIS (or both) has been compared to those of other techniques (like meta-heuristics); or other situations where they have been complementarily applied with other techniques. Li and Chen¹⁴ used both the least square method (LSM) and the ANFIS to train the artificial systems used in modeling traffic flow in Beijing ring roads and connected lines in order to develop and evaluate the intelligent transport systems. The results obtained from the two approaches were validated against real-life data, and they concluded that model trained with ANFIS produced better results than that trained with LSM. Azadeh and Zarrin¹⁵ used data envelopment analysis, and meta-heuristics trained ANN and ANFIS logics to evaluate the efficiency of staff along three parameters. It was concluded that ANFIS models trained with genetic algorithm (GA) and particle swarm optimization fared best. Barak and Sadeh¹⁶ used a combination of autoregressive integrated moving average (ARIMA) and ANFIS to model energy demand. The linear time series parameters were modeled with the ARIMA logic, and the nonlinear residual was later captured using the ANFIS logic in building an expanded model. They concluded that this model performed better than some others. Mansouri et al.¹⁷ compared the performance of M5Tree, ANN, ANFIS, and multivariate adaptive regression splines (MARS) in analyzing the axial compression of fiber-reinforced polymer and found that ANN predicted an accurate result. Mostafa and El-Masry¹⁸ used gene expression programming (GEP), ANN, and ARIMA to forecast the evolution of oil prices between January 1986 and June 2016. They reported that both ANN and GEP outperformed ARIMA, and that GEP produced the best result of the three using R^2 as the measure of performance. Oduro et al.¹⁹ did a comparative analysis of the prediction of vehicular emission using boosting MARS (BMARS), classification and

regression trees (CART), ANN, and a hybrid of CART–BMARS. They reported that the combined system proved superior to the models of ANN and those of each of the techniques when not hybridized.

While the transshipment problem has been solved using a number of heuristics and meta-heuristics, it is hard to come about those solved using artificial intelligence. Most transshipment models presented recently have been done under the theme of cross-docking. Augustina et al.²⁰ presented a summary of recent works done on transshipment. They classified the literature into three main categories based on strategic intent and planning horizon: operational, tactical, and strategic. They reviewed some 50 papers. At operational level, the two main issues addressed are about scheduling the cross-docking of trucks and dock-door assignment problems. The most common solution approaches are the different variants of modified Johnson, modified linear programming solution techniques, and meta-heuristics. Tactical-level decision focuses mainly on facility layout decisions, and a number of these problems are solved using meta-heuristics. The long-term decision relates to network design: cross-dock area location and sizes and choice of vehicle sizes and routes. Some other recent papers in the area include a multi-objective GA solution for transportation, assignment and transshipment problems,²¹ and a Just-in-time (JIT) multi-objective truck forecast problem with door assignment using differential evolution in the cross-dock system.^{22,23}

Model notations and formulation

The following notations are adopted for the modeling purpose:

- I is the set of all source points, indexed by i and including transshipment points (TPs) as necessary
- J is the set of all demand points, indexed by j and including TPs as necessary
- K denotes set of TPs, indexed by k
- P denotes set of product types, indexed by p
- X_{ijp} is the amount of product p transported from production point i to depots
- X_{ikp} is the amount of product p shipped from production point i to TP k
- X_{kjp} is the amount of product p shipped from TP k to depot j
- Z_{pi} is the capacity of source point i , including the TP, to produce or store product p , whichever is smaller
- d_{pj} is the demand for product p at the destination point j , including the TP
- c_{pij} is the unit amount of transporting product p from production point i to depot j
- c_{pik} is the unit amount of transporting product p from production point i to TP k
- c_{pkj} is the unit amount of transporting product p from TP k to depot j

ξ is a time indexed random vector on which the outcome of the demand and cost depends, which helps to partition decisions into periods before and after the evolution of ω , a random event that affects demand and shipment cost

ω is a random event, $\omega \in \Omega$, that influences demand and cost of product, p

q is the second-stage objective vector of a stochastic programming model, $q \in Q$

x is the first-stage feasible decision vector, $x \in X$

y is the second-stage feasible decision vector, $y \in Y$

$E(\cdot)$ is the expectation operator

$M(\cdot)$ is the membership function operator

A is the first-stage constraint matrix in a two-stage stochastic model with recourse

h is the RHS vector of the second stage (recourse action) of the stochastic program;

W is the Recourse matrix, based on the values from the evolution of ξ

T is called the Technology matrix such that $Wy = h - Tx$

a is the lower bound value of a triangular distribution

b is the upper bound value of a triangular distribution

m is the modal value of a triangular distribution

The deterministic model of the transshipment problem can be represented as:

Minimize

$$\sum_i \sum_j \sum_p c_{pij} X_{pij} + \sum_i \sum_k \sum_p c_{pik} X_{pik} + \sum_j \sum_k \sum_p c_{pjk} X_{pjk} \quad (1)$$

subject to

$$\sum_j X_{pij} + \sum_k X_{pik} \leq z_{p,i} \quad \forall p, i \quad (2)$$

$$\sum_i X_{pij} + \sum_k X_{pkj} \geq d_{pj} \quad \forall j, p \quad (3)$$

$$\sum_i X_{pij} - \sum_j X_{pjk} \geq d_{pk} \quad \forall k, p \quad (4)$$

$$X_{pij}, X_{pik}, X_{pkj} \geq 0 \quad (5)$$

Equation (1) is the objective function and is the total cost of shipment of product p , consisting of three cost terms: the first term being the cost of shipment from source i directly to destination j , the second term being that from source i to TP, k , and the third term being that from TP, k , to destination j . Equation (2) is the set of supply constraints, ensuring that the quantity of items delivered to TPs, k , and directly to final destination points, j , do not exceed the capacity of the supply point, i . Equation (3) is the set of demand constraints, ensuring that the quantity of items

delivered from TPs, k , and directly from supply points, i , do not fall below the demand at the destination point, j . Equation (4) is the set of balancing equation for TPs to ensure that the difference between the sum of item p , shipped to the TP k , and the sum of items shipped away from the point is the demand at the TP if the TP also serves as a destination point (or zero otherwise). The last set of equations is the non-negativity constraints of the variables.

Allowing for nondeterministic variables (demand and cost) in the model, the two-stage stochastic model with recourse is presented next. The general equivalent of the deterministic objective function is minimize $c^T X$, with c and X being the cost and demand vectors, respectively. To modify equations (1) to (5) as a stochastic programming model, according to Birge and Louveaux,¹ the model equations (6) to (9) become:

Minimize

$$c^T X + E_\xi[\min q(\omega)^T y(\omega)] \quad (6)$$

subject to

$$AX = b \quad (7)$$

$$T(\omega)x + Wy(\omega) = h(\omega) \quad (8)$$

$$x, y(\omega) \geq 0 \quad (9)$$

$c^T X$ is the objective function presented in equation (1), which is the deterministic component of equation (6). $E_\xi[\min q(\omega)^T y(\omega)]$ is the expected value of the stochastic component of the objective function and is added to the deterministic function to be minimized together for the optimal values of x and y , where the X vector is the set of the foremost decision variables, and the Y vector is the set of the subsequent decision variables, which are made after the evolution of the random event, ξ , which affects the result observed after the first decisions, and thus, guides the optimal decision taken in the second stage. Equations (2) to (4) from the first stage (before ξ evolves) are rolled into the first-stage constraint matrix, A , as shown in equation (7) in the stochastic model. Equation (8) is the set of matrices and vectors defining the set of constraints that the decision vector, $y(\omega)$, must satisfy at the optimal point, when choosing recourse values based on the time-dependent random variable, ξ , which have been realized based on the evolution of the random event, ω . Equation (9) is the extended non-negativity constraint for the decision variables, x and $y(\omega)$.

The model in equations (6) to (9) representing the two-stage stochastic programming model with recourse has been solved using various solution approaches to large matrix linear programs (LP) such as the column generation technique, the Lagrange decomposition technique, and various meta-heuristics. In this article, two artificial intelligence approaches, ANN and ANFIS, are utilized, and the results of the two are compared to that of the traditional

Table 1. Quantity of product available and unit cost of distribution.

S/No.	Source/sink	Layout	Product availability; demand unit (weekly)	Unit cost
1	Asejire	Source/depot 1	135,255	S_1 to $D_1 = 0$
2	Ado-Ekiti	Depot 2	11,215	S_1 to $D_2 = 62$
3	Ore	Depot 3	20,125	S_1 to $D_3 = 60$
4	Akure	Depot 4	18,540	S_1 to $D_4 = 55$
5	Abeokuta	Depot 5	21,869	S_1 to $D_5 = 50$
6	Ijebu-Ode	Depot 6	11,986	S_1 to $D_6 = 45$
7	Ibadan	Depot 7	31,500	S_1 to $D_7 = 30$
8	Ondo	Depot 8	8348	S_1 to $D_8 = 48$
9	Ife	Depot 9	7126	S_1 to $D_9 = 35$
10	Ilesha	Depot 10	4546	S_1 to $D_{10} = 40$

transportation tableau. The modeling approach is presented next.

The second term in equation (6) could be expanded in different ways depending on whether it is being considered to be stochastic or fuzzy, and on whether it is discrete or continuous. For the purpose of this work, a triangular distribution is assumed for both the cost and demand variables, howbeit with different parameter values for the demand and cost variables involved. The function can be written out as either a stochastic component or a fuzzy component for the second term of equation (6), respectively, in the following equations^{1,2}

$$f(x) = \begin{cases} 0 & \text{for } x < a \\ \frac{2(x-a)}{(b-a)(m-a)} & \text{for } a \leq x \leq m \\ \frac{2(b-x)}{(b-a)(b-m)} & \text{for } m \leq x \leq b \\ 0 & \text{if } x > b \end{cases} \quad (10)$$

$$M(x) = \begin{cases} 0 & \text{for } x \leq a \\ \frac{x-a}{m-a} & \text{for } a < x \leq m \\ \frac{b-x}{b-m} & \text{for } m \leq x < b \\ 0 & \text{for } x \geq b \end{cases} \quad (11)$$

The deterministic component (i.e. the first term of equation (6)) still remains as defined in equation (1).

Company of case study

The company used as a case study is a multinational beverage bottling company with a single production plant, $n = 1$, a range of products represented by the single product family, $i = 1$, servicing the south western part of Nigeria and some neighboring cities via 10 depots, $j = 10$, and three TPs, $k = 3$, which also serve as depots. Data collected from the bottling plant are used for analysis. Matrices are developed for the distances, and the associated cost is as shown in Table 1. The current practice in the company is to ship directly to all depots from the plants, although they could

have three TPs, D7, D9, and D10 selected through greedy heuristics, through which they could ship to the other depots as shown in Figure 2.

Model implementation

The triangular functions were approximated by the deterministic equivalents (quantile values for the stochastic model and alpha cuts for the fuzzy model) to derive the training rules during implementation. The stochastic model is divided into quantiles of demand and cost value ranges from which the class values are used for determining the values required for ANN training process. The associated linguistic values for ANFIS model were defined, and different alpha cut values were defined using the fuzzy rules developed and the corresponding values of linguistic variables determined in consistency with the fuzzy decomposition theorems. ANN and ANFIS training of models were done using the derived values from the available data set.

In solving the transshipment problem where product flows from source through TPs for onward distribution to the final destination, the deterministic model was solved using Tora software 2.0. For the nondeterministic input variables problem, popular solvers were not useful. The ANN technique, using neural weight update algorithm, and the ANFIS technique, which works based on the principle of fuzzy inference rules, were used to determine the optimal weight vectors that minimize the objective function of the LP problem presented.

The ANFIS rule viewer is a simplified toolbox with inbuilt algorithms for both the fuzzification of input parameters and the optimization of neural weights inside the same toolbox. It presents the analyst with the input interface to specify the necessary fuzzy parameters for each variable involved and presents the output in a defuzzified manner once the appropriate de-fuzzification algorithm of choice has been specified.

The creative model

In implementing the ANN model, a feed-forward model with backpropagation, having 4-neuron input

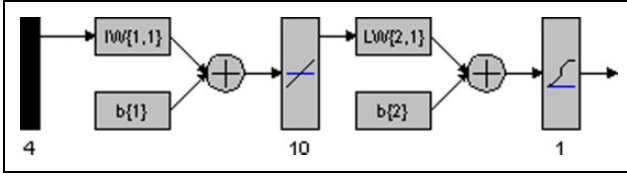


Figure 1. Structure of the MATLAB ANN model. ANN: artificial neural network.

layer with linear combiner, 10-neuron middle layer, and a single-neuron output layer was designed using the sigmoid output transfer function. This is shown in Figure 1. The model is trained with the MATLAB ANN toolbox using the gradient descent algorithm. Mean squared error (MSE) was selected as the performance function, and R^2 values were examined throughout the training period.

The following notations are defined for exclusive use within the ANN and ANFIS models.

- U_k is the linear combiner output due to input signal
- W_{kv} the connection weight from neuron v to neuron k
- X_j the signal flow from source/input through synaptic weights to summing junction, that is, $X_1, X_2, X_3, \dots, X_j$
- j the index identifying the number of inputs
- n the index identifying destinations
- k the index identifying the processing elements
- V_k the linear combiner with bias to support input signal
- Y_k the total weighted sum of input to the k th processing element
- p the index representing TPs ($p = 1$ for depot 7, $p = 2$ for depot 9, and $p = 3$ for depot 10)
- W_{ki} the weight or strength of connection from neuron, i to k
- X_{ki} the demand input signal from depots i to TPs k
- η the learning rate of the neural network (NN) in the backpropagation algorithm implementation

A typical standard neuron consists of set of synapses, each characterized by weight or strength of connections. It also comprises the bias or summing junction and the activation junction. The total sum of the neuron U_k is obtained by linearly combining all the inputs, multiplied by the appropriate weight or strength of connection so that in mathematical form, a neuron K is expressed as shown in the below equation

$$U_k = \sum_{j=1}^n x_j w_{kj} \quad (12)$$

Rewriting equation (12) for each transshipment location (i.e. at each TP)

$$U_{pk} = \sum_{j=1}^6 x_j w_{pkj} \quad \forall p \quad (13)$$

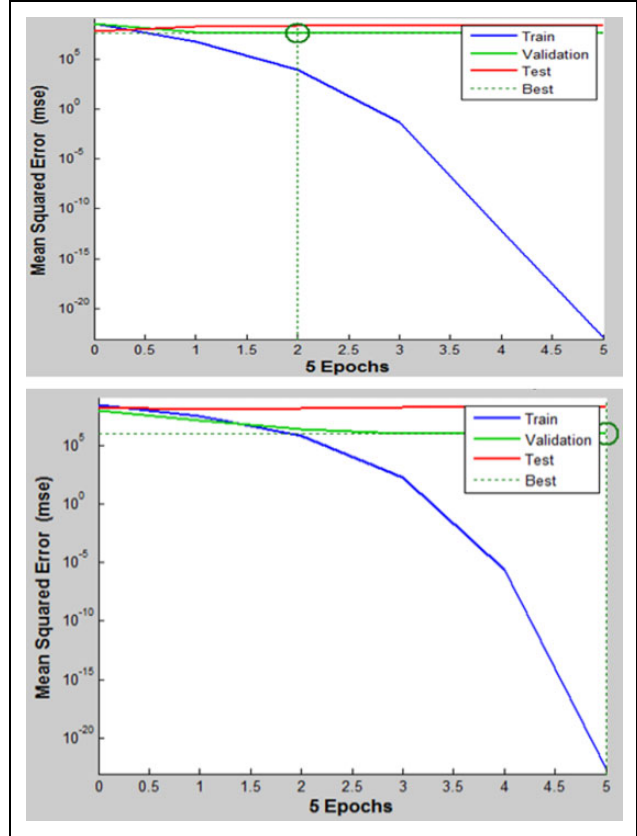


Figure 2. Rough and final MSE variations for training, testing, and validation data sets against epoch during the training. MSE: mean squared error.

Introducing a bias (b_k), then the combined input (V_k) can be expressed as

$$V_k = \sum_{p=1}^3 U_{pk} + b_k \quad (14)$$

Substituting the value of U_k for each p in equation (13) into equation (14), the combined input can be computed as

$$V_k = \sum_{p=1}^3 \sum_{j=1}^6 x_j w_{pkj} \quad \forall p \quad (15)$$

Defining the output threshold function as a sigmoid function $\varphi(\cdot) = \frac{1}{1+e^{-\alpha x}}$, then the total output can be expressed for each p , and with W_{kn} from middle layer neuron k to output layer neuron n , equation (12) can be written as

$$Y_{pk} = \varphi(w_{pkn} x_n + b_{pk}) = \frac{1}{1 + e^{-\sum w_{pkn} x_n + b_{pk}}} \quad \forall p \quad (16)$$

Batch update was done to train the neurons, and the general weight update rule for the NN is

$$w_{\text{new}} = w_{\text{old}} - \eta \sum \delta x_j \quad (17)$$

Table 2. ANFIS training data.

S/No.	Cost at TP7	Cost at TP9	Cost at TP10	D	Y_{K1}	Y_{K2}	Y_{K3}
1	45	27	20	11,215	65,355	35,599	34,301
2	33	20	25	20,125			
3	32	22	18	18,540			
4	20	29	35	21,869			
5	25	32	30	11,986			
6	26	30	15	8348			

ANFIS: adaptive neuro-fuzzy inference system; D: product demand; TP: transshipment point. Note: Y_{K1} = product available at transshipment point 1; Y_{K2} = product available at transshipment point 2; Y_{K3} = product available at transshipment point 3.

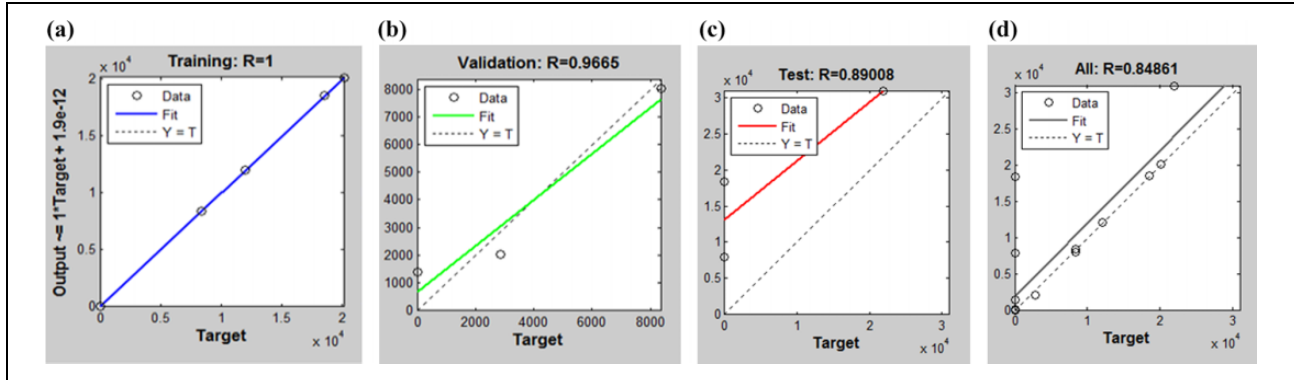


Figure 3. Plot of ANN-predicted output against actual value for (a) training, (b) validation, (c) testing, and (d) target. ANN: artificial neural network.

where $\delta = (Y_{exp.m} - Y_{pred})$ when the transfer function is linear, $\delta = Y_{pred}(1 - Y_{pred})(Y_{exp} - Y_{pred})$ when the transfer function is sigmoid, Y_{pred} represents the predicted ANN value, and Y_{exp} denotes the expected or targeted output value. According to Lotfan et al.,²⁴ equations (18) and (19) are used to test the fitness during training

$$MSE = \frac{1}{n} \sum_{m=1}^n (Y_{exp.m} - Y_{pred.m})^2 \quad (18)$$

$$R = \left(\frac{\sum_{m=1}^n (Y_{pred.m} - y_{pred})(Y_{exp.m} - y_{exp})}{\sqrt{\sum_{m=1}^n (Y_{pred.m} - y_{pred})^2 \sum_{m=1}^n (Y_{exp.m} - y_{exp})^2}} \right) \quad (19)$$

Data analysis, results, and discussion

Approach 1 (Using ANN)

Using a three-layer (input, hidden, and output) NN model as a result of the quantitative and architectural characteristics of the NN which has a close resemblance of the outbound manufacturing system where the input denotes product available at the source, and the route to different destinations represents weighted strength of the network. Table 2 shows the available

data set for network training, with cost and demand set as input variables of W and X and the output which is the product availability, Y , at different TPs. D is the demand of products at various locations. In NN programming, it is necessary to train the network for accurate value. The accuracy of the value is a function of MSE and R values. Our goal is to minimize these errors. This was achieved by running series of iterations and training continuously with many weighted input parameters till a fairly minimal error is obtained.

Results and discussion. The goal of this section is to obtain the optimum cost of product distribution in a bottling system. The ANN was trained with four inputs, 10 hidden, and an output layer (4–10–1). The network was trained many times for effective evaluation of the accuracy of the model using MSE and R value to determine the degree of association between the predicted and expected values. The system of equation for MSE and R value is established in equations (18) and (19). Y_{pred} and Y_{exp} denote the predicted and experimental values obtained for the NN model, respectively. y_{exp} and y_{pred} also signify the average experimental and predicted values, correspondingly. The performance and best possible iteration result of the ANNs are presented in Table 1. Figure 4 presents testing performance with worthy MSE value of 1.57×10^{-5} . Bold items in Table 3 also reveal good performance solution. Training and performance values of this network were obtained at 1.000 and 0.80369, respectively.

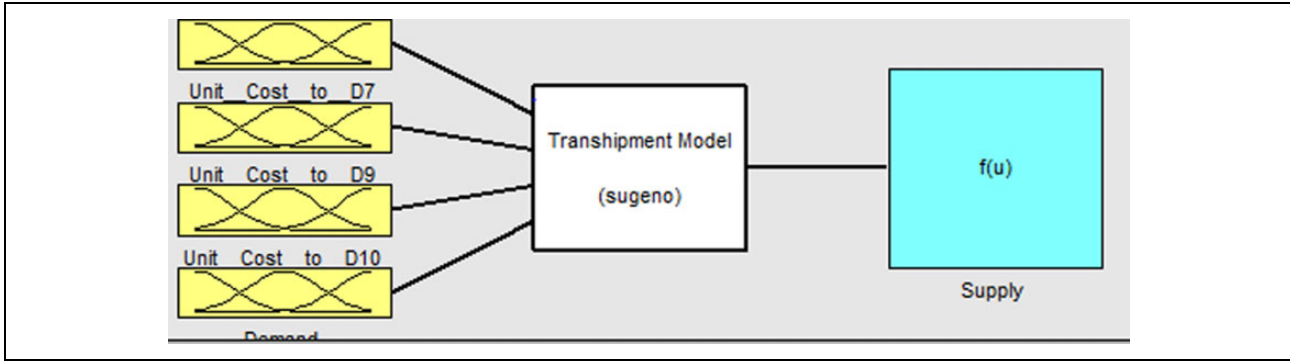


Figure 4. High-level fuzzy architecture for transshipment model.

Table 3. Performance results of multilayer perceptron (MLP) network for different numbers of neurons in the hidden layer for testing data set.

No.	MSE	R value			
		Training	Testing	Validation	Performance
1 (Rough)	6,514,260	0.9688	-0.9990	0.5950	0.37153
2	2,940,735	0.4768	-1.0000	1.0000	0.41513
3	3505	0.9994	-0.5558	-0.19681	0.63866
4	2800	1.0000	0.1035	-0.5057	0.69111
5	2851	1.0000	0.11657	0.80388	0.68865
6	3586	1.0000	0.10346	-0.5057	0.69111
7	3357	1.0000	-0.55358	0.71808	0.56428
8	2676	0.6810	0.4638	0.81260	0.65247
9	2280	0.9821	0.7218	0.56580	0.73260
10	1950	0.9784	0.7136	0.6842	0.71216
11 (Final)	1.57 × 10⁻⁵	1.0000	0.77089	0.95504	0.80369
12	78	0.9999	0.4447	0.67610	0.69830
13	1900	0.9998	0.6669	0.85110	0.78962
14	3839	0.9999	0.7134	0.82341	0.79990
15	2348	0.9999	0.7614	0.71320	0.68530

MSE: mean squared error, R: regression coefficient; R²: average determination coefficient.

Figure 2(a) is the MSE value for rough iteration in which the best validation result was obtained at epoch 2 with the minimum MSE value of 65×10^{-5} (see Table 3, no. 1 in bold items).

In Figure 2(b), the training process was repeated for MSE and the best validation result was obtained at epoch 5 with minimum MSE of 1.57×10^{-5} . This performance is meritorious compared to the MSE result obtained for rough iteration (see Table 3, no. 11 in bold items). Figure 5 shows the plots of the ANN-predicted outputs generated by the ANN model for training, validation, testing, and target. The performance value obtained is demonstrated in Table 4. A satisfactory iteration, though not hundred percent, is obtained in Table 4.

Interpretation of the correlation coefficient (R). The correlation coefficient specifies the degree of association or relationship among some variables of interest. Generally speaking, a correlation value of 0 is believed to be the absence of linear relationship, while 1 implies perfect relationship between variables. There are rules for interpreting the R² value, but

a value greater than 0.7 and less than 1 can be regarded as substantial to meritorious for a reasonably sized data set.

Continuous iteration (or refining) of weight parameters was done to obtain a model with the best possible fit. The iteration was performed several times to obtain the best value as shown in Table 3 and Figure 3. Table 3 is a summary table showing the MSE and R values for the iterations. Apart from the 11th value in Table 3, other values have high MSE value for training, validation, and testing. The best solution to the programmed ANN system is obtained at the 11th run with the minimal MSE value. The ANN model at the first trial stage was trained with 50 data points of which the best 15 are shown in Table 3. This was done to test the effectiveness of the tool.

As shown in Figure 3(a) to (d), the R values for training, validation, and testing are: 1.000, 0.95504, and 0.77089, respectively, with performance value of 0.80369. Therefore, the ANN prediction for training, validation, and testing are substantial and meritorious in terms of correlation. The performance value for the

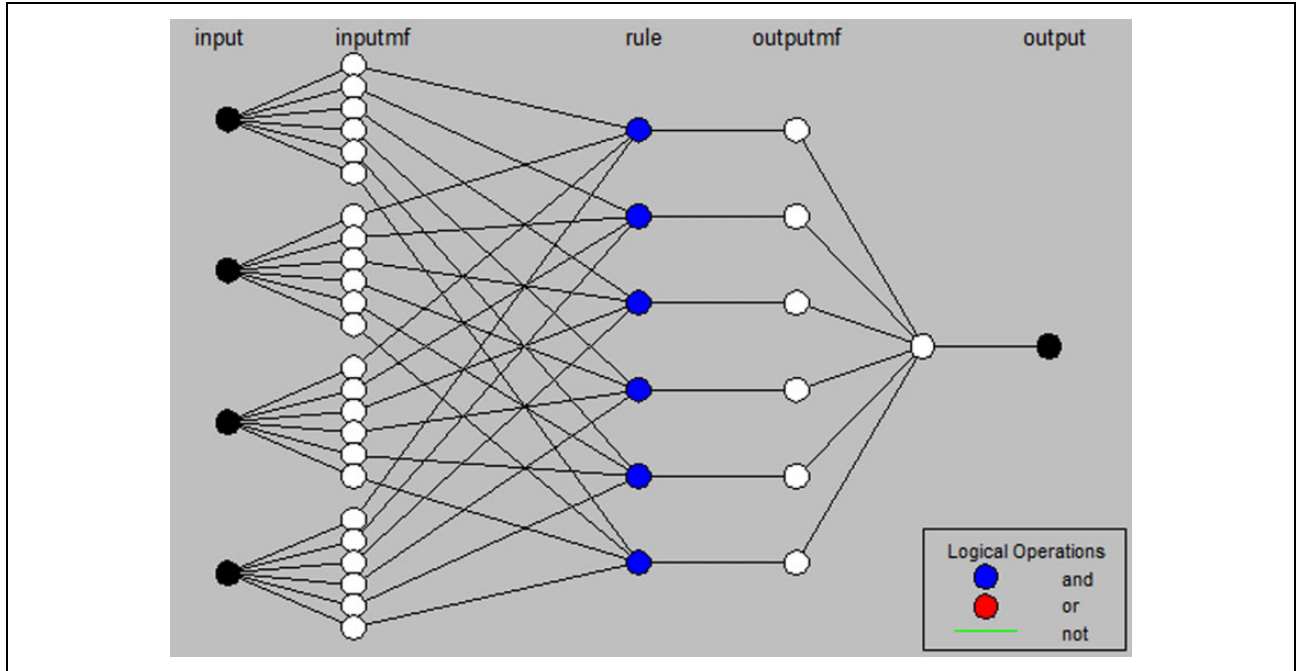


Figure 5. ANFIS architecture with input–output membership functions. ANFIS: adaptive neuro-fuzzy inference system.

Table 4. Performance value for ANN modeling (final iteration).

Results	Samples (%)	MSE	R
Training	68	1.56999×10^{-5}	1.000000
Validation	16	—	0.955041
Testing	16	—	0.770892

ANN: artificial neural network; MSE: mean squared error; R: regression coefficient.

final iteration gave a satisfactory result as shown clearly in Figure 9, where SD7 represents the Supply signal at D7; SD9 is the Supply signal at D9; SD10 is the Supply signal at D10; TP is the transshipment point; D is the product demand; UTP7 is the Unit cost from TP7 to depots; UTP9 is the Unit cost from TP9 to depots; UTP10 is the Unit cost from TP10 to depots; and D represents the Depots which are 10 in all.

Total cost of distribution at D7 = $(45 \times 0) + (33 \times 0) + (32 \times 0) + (20 \times 21,869) + (25 \times 11,986) + (26 \times 0) = 737,030$

Total cost of distribution at D9 = $(27 \times 8348) + (20 \times 20,125) = 225,396 + 402,500 = 627,896$

Total cost of distribution at D10 = $(20 \times 2867) + (18 \times 18,540) + (15 \times 8348) = 516,280$

Total cost of transshipment = Supply cost at D7 + Supply cost at D9 + Supply cost at D10 = $737,030 + 627,896 + 516,280 = 1,881,206$

Total cost of distribution in the entire system = Cost of distribution from source to TPs plus cost of distribution from TPs to final destination

$$= \sum_{i=1}^n \sum_{k=1}^m C_{ik} X_{ik} + \sum_{k=1}^n \sum_{j=1}^m C_{kj} X_{kj} = 2,289,327 + 1,881,206 = 4,170,533$$

Therefore, total cost of distribution using the developed ANN model = **4,170,533.00**

Approach 2 (Using ANFIS model)

In this section, fuzzy logic was used to find a delivery pattern that would meet customer demand and minimize delivery cost. Since there exists a source and 10 destinations with three possible TPs and there are two imprecise inputs, finding optimal solution requires continuous iterations. Using ANFIS, transshipment routes were created and analysis was carried out. Using the graphical user interface of Simulink to generate input, an initial fuzzy model was derived to trigger the modeling process. Membership functions were established for the input signals.

The high-level fuzzy model logic is shown in Figure 4. The input variables were passed into the fuzzy toolbox that utilizes the Sugeno class function for fuzzification. The membership functions (mf) at the input and output ends of the ANFIS logic were then defined so that the appropriate rules can be developed for training as shown in Figure 5. The triangular membership function (mf) was selected for the inputmf variables and the Gaussian for the outputmf supply signal. Examples of these can be seen in Figure 6(a) to (d). Figure 8 is an example of the three-dimensional (surface) views generated using the ANFIS rule viewer. Example of the application of

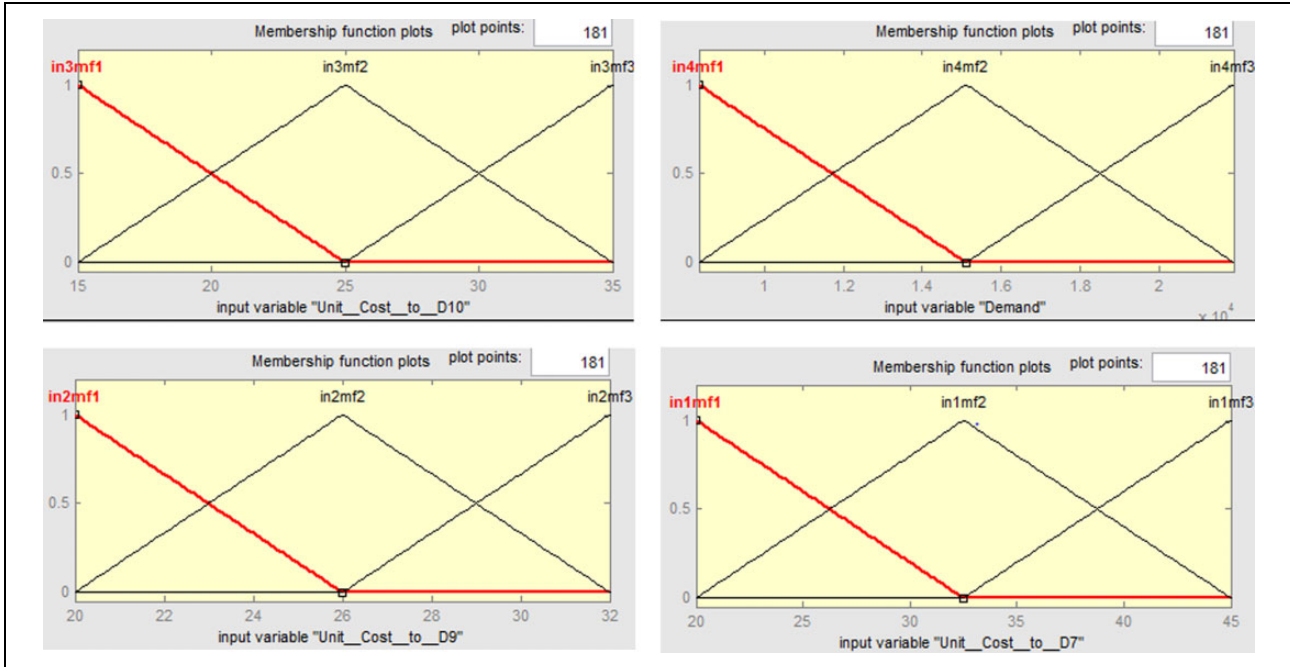


Figure 6. Fuzzy Inference System (FIS) membership function plot for cost of distribution from TP7, TP9, and TP10 and demand at various destinations. TP: transshipment point.

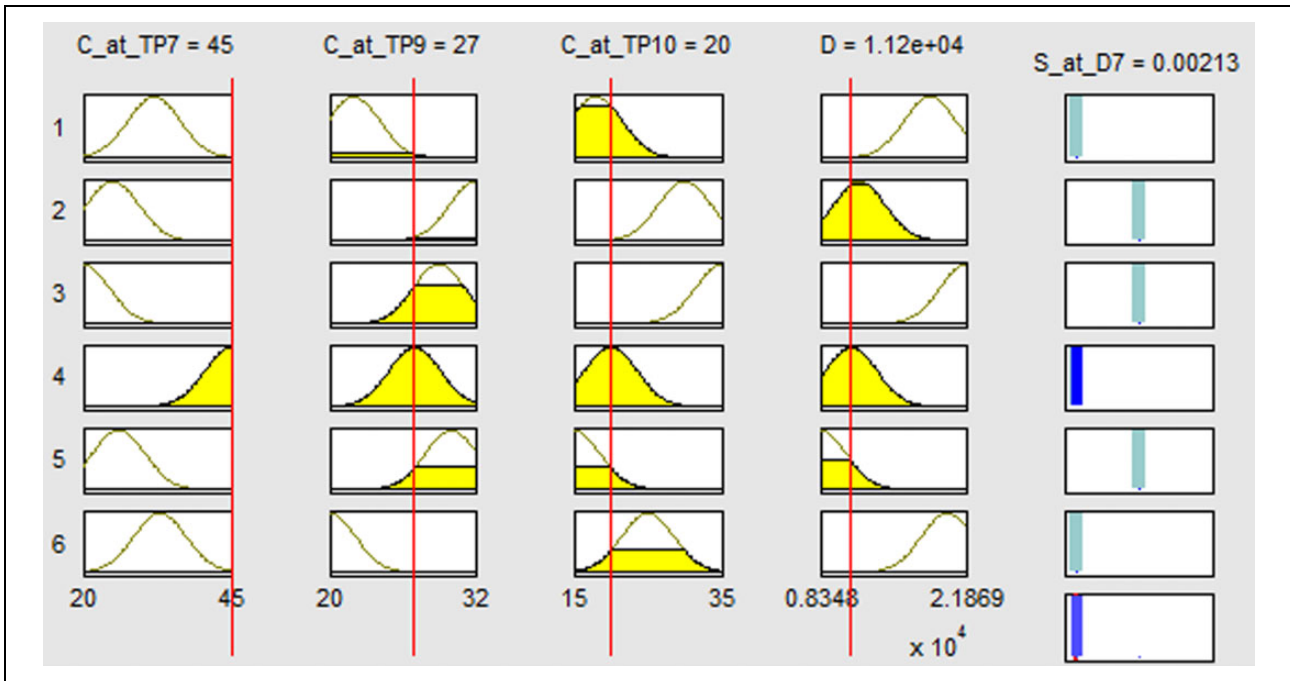


Figure 7. Rule viewer for input–output supply signals.

the rule viewer using discretization of the continuous triangular inputs to take the relevant alpha cuts in order to generate the fuzzy inputs is shown in Figure 7. Eighty one training rules were developed. These rules are shown in the Appendix.

Table 2 is the data set representing product demand at source and the available products at the TPs 7, 9, and 10 for

onward distribution to the final destinations. Figure 8 shows the membership function plot for input variables in terms of demand.

The final allocation is presented in Figures 9 and 10, showing the quantity to be supplied from each TP to each of the destinations using ANN and ANFIS, respectively. It should be noted that each of the TPs supplied its own

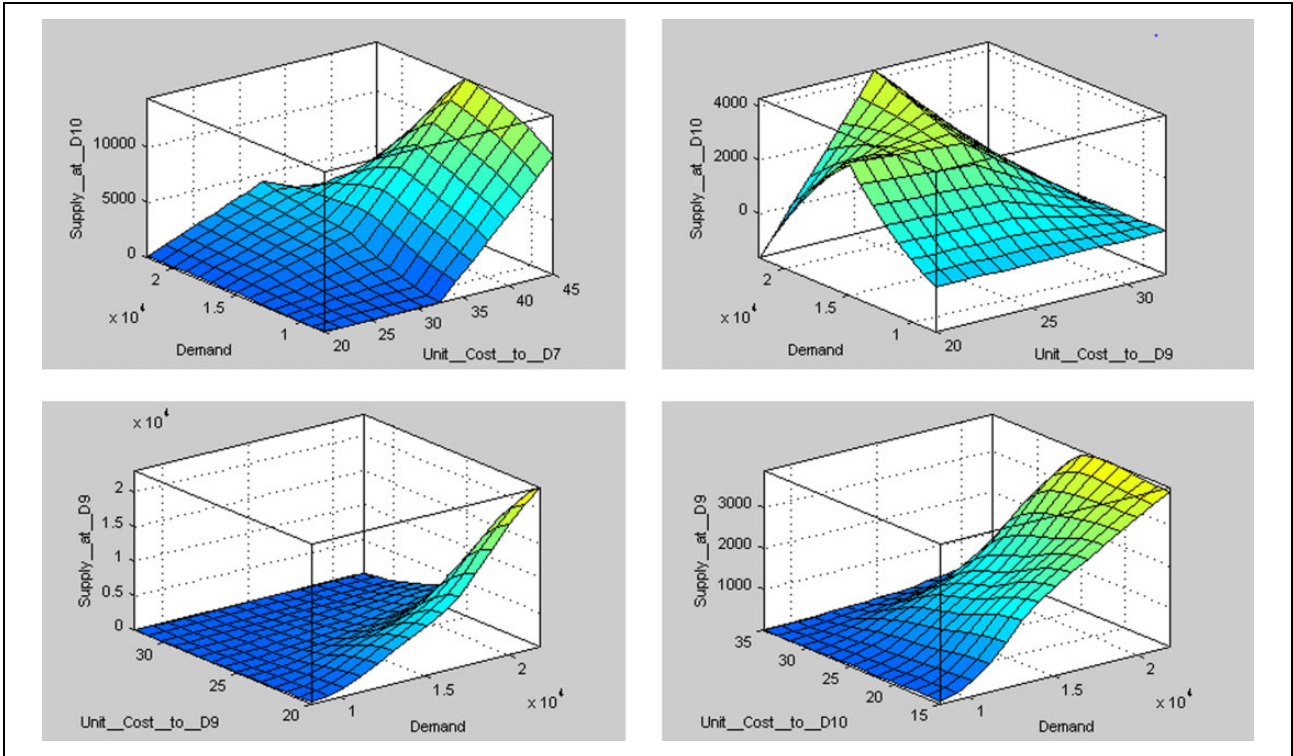


Figure 8. Three-dimensional input–output surface plots.

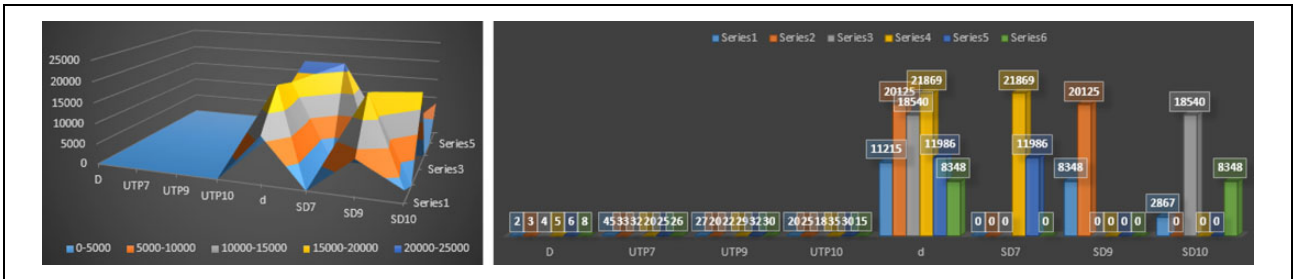


Figure 9. ANN solution figures for supply signal at TPs 7, 9, and 10. ANN: artificial neural network; TP: transshipment point.

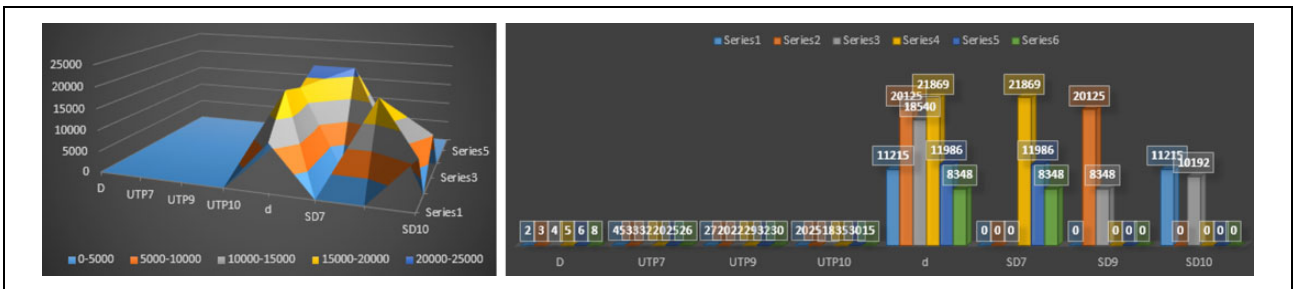


Figure 10. ANFIS optimal solution figures for supply signal at TPs 7, 9, and 10. ANFIS: adaptive neuro-fuzzy inference system; TP: transshipment point.

demands. The total distribution cost per week obtained (in Nigerian Naira, NGN) is shown in Figures 9 and 10. The solution produced using ANFIS is marginally cheaper than the one produced using ANN and the company’s traditional

transshipment model. While the company’s traditional solution produced NGN6,332,304 minimum cost per week, ANN generated a minimum of NGN4,170,533. ANFIS, however, yielded a minimum cost of 4,053,661, which is

better than both the traditional technique of the company and the ANN model solution.

Total cost from TP to final destination = **1,764,334**

Total cost of distribution from source to TPs = **2,289,327**

Total cost of distribution in the entire system

$$= \sum_{i=1}^n \sum_{k=1}^m C_{ik} X_{ik} + \sum_{k=1}^n \sum_{j=1}^m C_{kj} X_{kj} = 2,289,327$$

$$+ 1,764,334 = 4,053,661$$

Conclusion

In this article, ANN and ANFIS models have been applied successfully to solve a transshipment problem with nondeterministic demand and cost input variables. With data set obtained from a bottling company, training was performed using ANN fitting tool where the best iteration was considered based on MSE and R values obtained. The model based on the ANFIS logic was also implemented using the rule viewer interface. This model is capable of adjusting parameters based on series of alpha cuts taken on the data set. Our results seem to have followed the pattern reported in other papers that have compared the performance of models with nondeterministic inputs to those that used the mean values to approximate such variables. The results suggest that the ANFIS logic seems to have produced a slightly better solution than that obtained using the ANN, and both are much better than that obtained using the company's traditional distribution algorithm. ANFIS is, however, only marginally better in this context, and it is generally inconclusive in the literature if one technique is better than the other.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

- Birge JR and Louveaux F. *Introduction to stochastic programming*. Booth school of business, University of Chicago: USA: Springer Series in operations Research, Springer, 1997.
- Klir GJ and Yuan B. *Fuzzy sets and fuzzy logic, theory and applications*. New Jersey: Prentice Hall, 1995.
- Gaines BR. Stochastic and fuzzy logics. *Elect Lett* 1975; 11: 188–189.
- He Z, Wen X, Liu H, et al. A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semi-arid mountain region. *J Hydrol* 2014; 509: 379–386.
- Anari PL, Darani HS, and Nafarzadegan AR. Application of ANN and ANFIS models for estimating total infiltration rate in an arid rangeland ecosystem. *Res J Environ Sci* 2011; 5(3): 236–247.
- Do GH and Chen JF. A comparative study of hierarchical ANFIS and ANN in predicting student academic performance. *WSEAS Trans Inf Sci Appl* 2013; 10(12): 396–405.
- Kamali R and Binesh AR. A comparison of neural networks and adaptive neuro-fuzzy inference systems for the prediction of water diffusion through carbon nanotubes. *Micro Nano* 2013; 14: 575–581.
- Sredanovic B and Cica D. Comparative study of ANN and ANFIS prediction models for turning process in different cooling and lubricating conditions. *SAE Int J Mater Manuf* 2015; 5: 49–64.
- Azeez D, Ali MAM, Gan KB, et al. Comparison of adaptive neuro-fuzzy inference system and artificial neural networks model to categorize patients in the emergency department. *SpringerPlus* 2013; 2: 416.
- Esen H. ANN and ANFIS models for performance evaluation of a vertical ground source heat pump system. *Expert Syst Appl* 2010; 37(12): 8134–8147.
- Vieira JAB, Dias FM, and Mota MM. Comparison between artificial neural networks and neurofuzzy systems in modeling and control: a case study. *IFAC J* 2003: 265–273.
- Kisi A and Ay M. Comparison of ANN and ANFIS techniques in modelling dissolved oxygen. In: *Sixteenth international water technology conference, IWTC-16*, Istanbul, Turkey, 2012; 1–10.
- Farahmand AR, Manshouri M, Liaghat A, et al. Comparison of kriging, ANN and ANFIS models for spatial and temporal distribution modelling of groundwater contaminants. *J Food Agri Environ* 2010; 8(3–4): 1146–1155.
- Li Y and Chen D. A learning-based comprehensive evaluation model for traffic data quality in intelligent transportation systems. *Multi Tools Appl* 2016; 75: 11683–11698.
- Azadeh A and Zarrin M. An intelligent framework for productivity assessment and analysis of human resource from resilience engineering, motivational factors, HSE and ergonomics perspectives. *Saf Sci* 2016; 89: 55–71.
- Barak S and Sadegh SS. Forecasting energy consumption using ensemble ARIMA–ANFIS hybrid algorithm. *Elect Power Energy Syst* 2016; 82: 92–104.
- Mansouri I, Ozbakkaloglu T, Kisi O, et al. Predicting behavior of FRP-confined concrete using neuro fuzzy, neural network, multivariate adaptive regression splines and M5 model tree techniques. *Mater Struct* 2016; 49: 4319–4334.
- Mostafa MM and El-Masry AA. Oil price forecasting using gene expression programming and artificial neural networks. *Econ Model* 2016; 54: 40–53.
- Oduro SD, Ha QP, and Duc H. Vehicular emissions prediction with CART-BMARS hybrid models. *Trans Res D* 2016; 49: 188–202.
- Augustina D, Lee CKM, and Piplani R. A review: mathematical models for cross docking planning. *Int J Eng Bus Manag* 2010; 2(2): 47–54.
- Zaki SA, Mousa AA, Geneedi HM, et al. Efficient multi-objective genetic algorithm for solving transportation, assignment, and transshipment problems. *Appl Math* 2012; 3: 92–99.

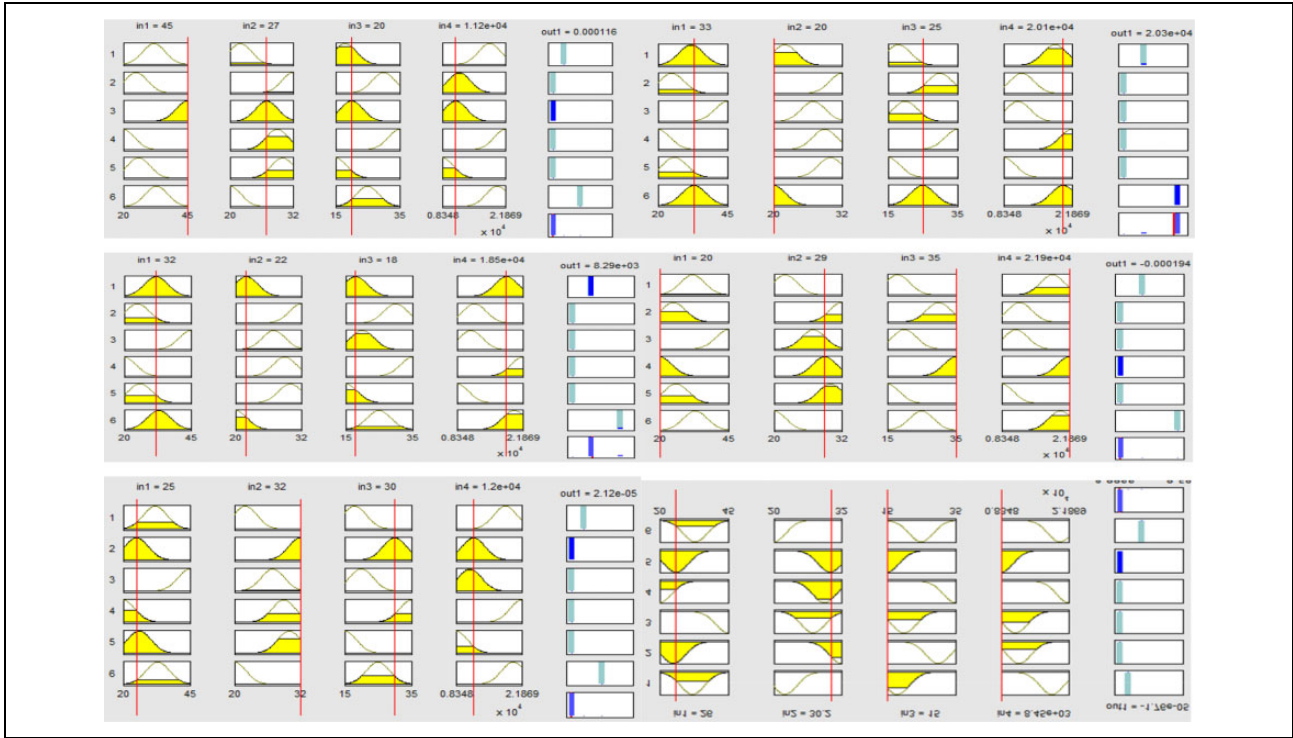
- 22. Wisittipanich W and Hengmeechai P. A multi-objective differential evolution for just-in-time door assignment and truck scheduling in multi-door cross docking problems. *Ind Eng Manag Syst* 2015; 14(3): 299–311.
- 23. Cota PM, Gimenez BMR, Araújo DPM, et al. Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre. *Comput Ind Eng* 2016; 95: 135–143.
- 24. Lotfan SR, Ghiasi A, Fallah M, et al. ANN-based modeling and reducing dual-fuel engine’s challenging emissions by multi-objective evolutionary algorithm NSGA-II. *J Appl Energy* 2016; 175: 91–99.

APPENDIX I – FURTHER SAMPLES OF FUZZY VARIABLES

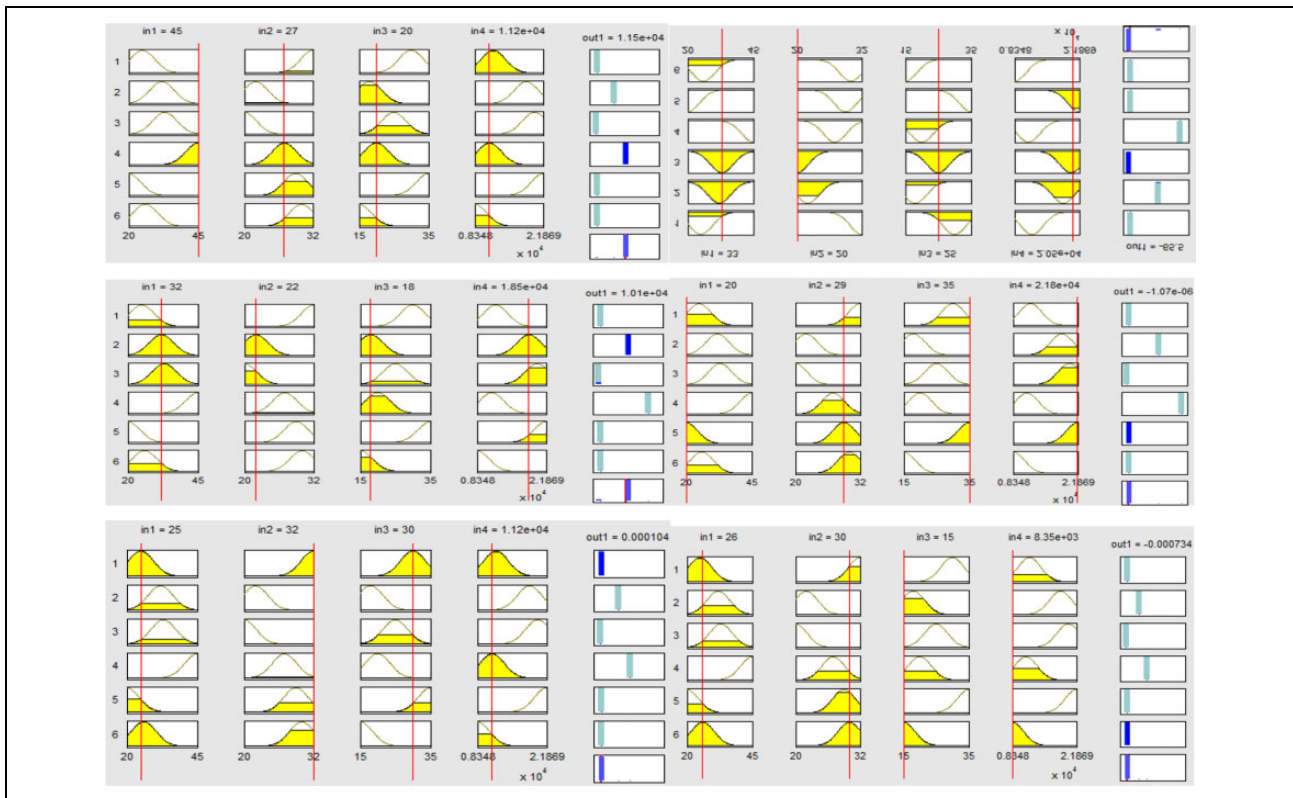
OUTPUT RESULT AT Transhipment Point 7



OUTPUT VALUE AT Transhipment Point 9



OUTPUT VALUE AT Transhipment Point 10



(73) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf1 and demand is in 4mf1 then supply at D7 is out 1mf73.

(74) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf1 and demand is in 4mf2 then supply at D9 is out 1mf74.

(75) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf1 and demand is in 4mf3 then supply at D10 is out 1mf75.

(76) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf2 and demand is in 4mf1 then supply at D7 is out 1mf76.

(77) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf2 and demand is in 4mf2 then supply at D9 is out 1mf77.

(78) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf2 and demand is in 4mf3 then supply at D10 is out 1mf78.

(79) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf3 and demand is in 4mf1 then supply at D7 is out 1mf79.

(80) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf3 and demand is in 4mf2 then supply at D9 is out 1mf80.

(81) If the unit cost to D7 is in 1mf3 and unit cost to D9 is in 2mf3 and unit cost to D10 is in 3mf3 and demand is in 4mf3 then supply at D10 is out 1mf81.