Artificial Neural Network (ANN) and Greedy Heuristic (GH) Approach to Transshipment Model in a Bottling Plant

Modestus O. Okwu*, Ikuobase Emovon

Department of Mechanical Engineering, Federal University of Petroleum Resources, Effurun, Nigeria

Email address
mechanicalmodestus@yahoo.com (M. O. Okwu)
*Corresponding author

Citation

Received: January 23, 2018; Accepted: February 18, 2018; Published: March 23, 2018

Abstract: Managing product flow from source to destination in a multi-echelon system is a prime task and often considered more challenging than managing product flow in a single-echelon system. The status quo especially in developing countries is that such decisions are made relying on rule of thumb approaches which has never guaranteed ideal solutions. In this study, an algorithm for effective multi-echelon inventory management system has been proposed and applied in a Nigerian bottling plant. The interactions and product flow in the case company were identified and characterized as a multi-echelon system and it was observed that products were distributed from source to downstream at a higher cost. Results of the current study shows that the developed model eliminated several deficiencies by ensuring that depots got their demanded products at optimal cost. Hence, the model can be applied by decision makers, engineering managers and industry practitioners to effectively minimize distribution expenses for any multi-echelon system while fulfilling demand at all destinations.

Keywords: Single-Echelon, Multi-echelon, Heuristics, Algorithm

1. Introduction

In any manufacturing setting (complex or simple, large or small, inbound or outbound) every producer or manufacturer strive to meet customers demand at lowest possible cost [1]. Supply chain design (SCD) and Inventory management system (IMS) are unified approaches to effective planning and control of design process and inventory management throughout the entire system from the producer to the consumers. Inventory management system and supply chain inventory management goals are tailored towards optimizing cost, improving services of customers and increasing variety of products [2]. Also, response to the demand of products and services are usually very high, values placed on quality of products and services are standard, and products are timely distributed at the right cost, high packaging condition, to the right customer, so as to remain competitive in business. The main focus of an enterprise is to provide high quality products, boost customers’ satisfaction and reduce operational cost [3]. Consequently, this study addresses a distribution problem faced by a bottling company which produces multiple products to be distributed to various locations with the purpose of satisfying retailers’ expectation.

In this case, the demand made by customers at different locations as well as the unit cost of distribution to various locations is known. The quantity of products to be distributed to various locations based on demand cannot exceed the capacity of the vehicle. This is an outbound supply chain network which involves the flow of goods from the point of production to the point of consumption. These challenges as aforementioned are ever posed on producers of manufacturing industries. Apart from the bottling company, the application of this study in real life may arise in different settings like: oil and gas sector; fruit juice companies; food and pharmaceutical industries and so on, where demand of products is required at various distribution centers.
2. Artificial Neural Network

2.1. Training Artificial Neural Network

Artificial neural networks (ANN) are networks of interaction linked with artificial intelligence. ANN is based on the function of human brain and nervous system. Neuron and link are the two basic components of ANN. A link has weight and is basically used to connect one neuron with another while a neuron is a processing element [4]. ANN is a good example of soft computing techniques that has been widely used in diverse field [5]. The architecture of the neuron is designed in such a way that stimulation is received from each neuron which is processed to produce an output. Also, the structure of a neuron is categorized into layers of input, hidden and output. Data is presented to the network via the input layer. This layer receives information and passes it across to the hidden layer. The input layer is not a neural computing layer since it has no input weights and activation functions. The hidden layer is the middle layer which is the processing or control layer with activation function to effectively control information flow. This layer has no connection to the outside world. The solution to the input dataset is presented at the output layer. The stimulation sent to a neuron from other interconnected neurons is increased by the weights of connectivity links which is supplementary to inbound activations. In multi-layer feed-forward propagation of the neural network, there is stimulation or activation spread simply in a forward direction from the input layer via one or more hidden or middle layers to the output layer.

Considering available dataset, a worthy non-linear relationship can be achieved using a multi-layer feed-forward network. In literature of creative algorithms, a feed-forward network even with only one hidden layer can approximate any continuous function. Thus, a feed-forward network is a superior approach. During neural network implementation, it is essential to determine the architecture or structure of a neural network with respect to the number of layers and possible number of neurons in the layers. As the hidden layers and nodes increases the network architecture becomes more complex. A neural network setup with a configuration or structure that is far complex than required usually over fit the training dataset. This implies that the neural network toolbox performs effectively on dataset encompassed in the training process than that of testing procedure.

2.2. Multi-layer Perceptron

According to [6], the Multi-Layer Perceptron (MLP) remains the most commonly applied artificial neural network models. MLP comprises of three layers in a network system (input, output and hidden layers). Neurons are found in all nodes excluding the input layer. The architecture is such that the output of each layer is linked to the input of the next layer. The problem to be solved is proportional the number of nodes available in each layer. MLP functions in a way that during training process, weights are assigned and the best weight is selected based on the anticipated value that gives the desired output result. A good example of the MLP architecture is revealed in Figure 1.

2.3. The Perceptron Learning Rule

The perceptron learning rule in a neural network works by tracking error in the network and making slight modifications optimistically prevent the same errors from reoccurring. This is done by comparing the network actual output with the targeted output in the training set. A situation where there is no match between the actual and target output means there is something wrong in the network architecture. When such signal is noticed, it is very necessary to bring up to date the assigned weights based on error aggregate. Trivial modifications are made to the weights during iteration process accomplished by applying trifling learning rate (r). A situation where the learning rate is too high, the perceptron in most cases miss the solution by navigating too far, but where the learning rate is too low, training process usually takes arbitrarily long time to accomplish navigation process. (www.theprojectspot.com).

2.4. Training Artificial Neural Network

Neural Network training is a complex task that is very significant for supervised learning. Feed-forward artificial neural network like Multi-layer Perceptron (MLP) networks is a well-known model for machine learning extensively used as classifiers in a system [6]. To effectively use MLP to classify data, it is important to go through series of training process. According to [4], the training process of a neural network is achievable by adjusting the weights of network interconnections. This is typically done using Back-Propagation (BP) algorithm. Research has revealed that major disadvantage of using BP algorithm is the hassle of getting trapped in the local minima and slow convergence [7]. Countless efforts have been made by researchers to improve the performance of BP. In other to replace BP, methods like modified meta-heuristic algorithms have been tested at the training stage [8]. Neural networks are computational and
creative analogs inspired by the tractability, dynamism and power of the biological brain. The component parts are: brain-neurons, synapses and dendrite, summing units, weighted connections and neurons that work collectively in parallel and in series. Artificial neural network has the ability to learn relationships between given sets of input and output data by changing the weights. This process is called training the ANN [9]. In literature of creative algorithm and intelligent systems, meta-heuristics have been applied in improving diverse machine learning models such as Bayesian Networks, ANNs, Markov Networks, and others. Good examples of meta-heuristics include: Swarm Intelligence and Stochastic Local Search, Evolutionary algorithms [10]. There are other research areas where ANN has been used to solve problems. For instance, [11] applied the multiple feed-forward neural network in the prediction of students’ final achievements. The result of their study showed that perfect prediction is possible by the application of ANN. Also, [12] used ANN model to address entry level performance of students into the university. Their results showed that ANN model is an efficient predictive tool. The result favoured more than 70% of the prospective students who were in good standing. The presented result greatly demonstrate the expressive power of ANN.

3. Mathematical Modeling

The Neural network model and greedy heuristic techniques have been designed to solve a particular transshipment problem in a bottling plant. In this section, system parameters were identified as well as variables, limitations and criteria so as to effectively offer solution to the distribution problem. This study was conducted using information from Nigerian bottling company (NBC). A transshipment model was developed. The interactions and flow of products in the system was identified and modelled as shown in the frame work in Figure 2. The developed model has a close resemblance of the transshipment model. It is proven in literature that non-deterministic models usually provide better solutions with flexible and expressive power compared to traditional deterministic model input.

$$U_k = \sum_{j=1}^{n} x_j w_{kj}$$  

Introducing an adder, an activation junction and a bias the modified neuron with a strength of connection of the neuron as presented in Figure 3. Introducing a bias ($b_k$) as input to the system, then the combined input ($V_k$) becomes:

$$V_k = U_k + b_k$$  

Substituting the value of $U_k$ from the equation (1) into equation (2), the combined input yield:

$$V_k = \sum_{j=1}^{n} x_j w_{kj} + b_k$$  

Figure 2. Underlying product flow for n-Echelon system.

Figure 3. Modified Neuron Structure of ANN.

Figure 3 shows a clear view of ANN model structure with input values, to arrive at total sum of the neuron $U_k$; it is important to linearly combine all the inputs, multiplied by the appropriate weight or strength of connection so that in mathematical form, a neuron K can be described by equation (1).
Defining the threshold function as a sigmoid function, then the total output can be expressed in equation (4)

\[ Y_k = S_n F_n(V_k) \]  

(4)

\[ Y_k = \varphi(V_k) \]  

(5)

Substitution the value of \( V_k \) in equation (3) transforms the total output as expressed in equation (6).

\[ Y_k = \varphi(\sum_{j=1}^{n} x_j w_{kj} + b_k) \]  

(6)

\[ \sum_{j=1}^{n} x_j w_{kj} + b_k \]  

(7)

\[ Z = \text{Total transshipment in a neuron which equals total output plus sum of linear combiner output.} \]

\[ Z = Y_k + U_k \]  

(7)

\[ Z = \sum_{j=1}^{n} x_j w_{kj} + \varphi(\sum_{j=1}^{n} x_j w_{kj} + b_k) \]  

(8)

\[ \]  

3.1. Summary of the Neural Network Model

\[ U_k = \sum_{j=1}^{n} x_j w_{kj} \]  

(9)

\[ V_k = U_k + b_k \]  

(10)

\[ V_k = \sum_{j=1}^{n} x_j w_{kj} + b_k \]  

(11)

\[ Y_k = S_n F_n(V_k) \]  

(12)

\[ Y_k = \varphi(V_k) \]  

(13)

\[ Y_k = \varphi(\sum_{j=1}^{n} x_j w_{kj} + b_k) \]  

(14)

\[ Z = \sum_{j=1}^{n} x_j w_{kj} + \varphi(\sum_{j=1}^{n} x_j w_{kj} + b_k) \]  

(15)

3.2. Transshipment Model (Problem Constraint)

The problem constraints and equations describing them are:

- **Supply Constraint:**
  In this model, one of the systems of constraints is related to product availability. This explains the fact that everything going into transshipment point is less or equal to the total product \( i \) available at the source \( S_i \). This is shown as system of equation 16.

\[ \text{Supply} \sum_{j=1}^{n} X_{ik} \leq S_i \]  

(16)

- **Policy on transshipment Constraint:**
  Every company has its own policy on distribution. In this research, the policy constraint requires that total product going into the transshipment point is equal to total product going out of transshipment point.

\[ \text{Transhipment} \sum_{j=1}^{m} X_{ik} = \sum_{i=1}^{m} X_{kj} \]  

for each intermediate node \( k \)  

(17)

- **Demand Constraint:**
  This constraint requires that total sum of product going out of transshipment point is less or equal to the demand at that end. This is represented as system of equation 18.

\[ \text{Demand} \sum_{i=1}^{m} X_{kj} \leq D_j \]  

for all \( j = 1, 2 \ldots n \)  

(18)

- **Non-negativity constraint:**
  There explains the fact that there is no negative distribution. This is represented as the system of equation 19.

\[ \text{Non-negativity} X_{ik}, X_{kj} \geq 0 \]  

for all \( i = 1, 2 \ldots m; j = 1, 2 \ldots n \)  

(19)

3.3. Objective Function of the Transshipment Model

The objective function of this transshipment model is to minimize the total distribution cost. This is represented mathematically in system of equation 20.

\[ \text{Minimize} \sum_{i=1}^{n} \sum_{k=1}^{m} C_{ik} X_{ik} + \sum_{k=1}^{m} \sum_{j=1}^{n} \sum_{i=1}^{m} C_{kj} X_{kj} \]  

(20)

3.4. Summary of Generalized Bottling Plant Model

Subject to the constraints:

- **Supply:**
  \[ \text{Supply} \sum_{j=1}^{n} X_{ik} \leq S_i \]  

for all \( i = 1,2,3\ldots\ldots m \)

(16)

- **Demand:**
  \[ \text{Demand} \sum_{i=1}^{m} X_{kj} \leq D_j \]  

for all \( j = 1, 2 \ldots n \)

(17)

- **Policy:**
  \[ \text{Policy} \sum_{j=1}^{m} X_{ik} - \sum_{i=1}^{m} X_{kj} = 0 \]  

for each intermediate node \( k \)

(17)

- **Non-negativity:**
  \[ \text{Non-negativity} X_{ik}, X_{kj} \geq 0 \]  

for all \( i = 1, 2\ldots m; j = 1, 2\ldots n \)

4. Analysis

In this research, attempt was made to identify system parameters, variables, limitations, criteria so as to be able to
define the distribution problem. This study was conducted using information from a leading bottling plant, Nigerian bottling company (NBC). ANN model and transshipment model for product delivery has been developed. The data collected from the company is as shown in table 1.

**Table 1. Quantity of Product Available and Unit Cost of Distribution.**

<table>
<thead>
<tr>
<th>S/No</th>
<th>Source/Sink</th>
<th>Layout</th>
<th>Product Availability / Demand</th>
<th>Unit Cost (₦)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Asejire</td>
<td>Source</td>
<td>135,255</td>
<td>S₁ to D₁ = 0</td>
</tr>
<tr>
<td>2</td>
<td>Ade-Ekiti</td>
<td>Depot 2</td>
<td>11,215</td>
<td>S₁ to D₂ = 62</td>
</tr>
<tr>
<td>3</td>
<td>Ore</td>
<td>Depot 3</td>
<td>20,125</td>
<td>S₁ to D₃ = 60</td>
</tr>
<tr>
<td>4</td>
<td>Akure</td>
<td>Depot 4</td>
<td>18,540</td>
<td>S₁ to D₄ = 55</td>
</tr>
<tr>
<td>5</td>
<td>Abeokuta</td>
<td>Depot 5</td>
<td>21,869</td>
<td>S₁ to D₅ = 50</td>
</tr>
<tr>
<td>6</td>
<td>Ijebu-Ode</td>
<td>Depot 6</td>
<td>11,986</td>
<td>S₁ to D₆ = 45</td>
</tr>
<tr>
<td>7</td>
<td>Ibadan</td>
<td>Depot 7</td>
<td>31,500</td>
<td>S₁ to D₇ = 30</td>
</tr>
<tr>
<td>8</td>
<td>Ondo</td>
<td>Depot 8</td>
<td>8,348</td>
<td>S₁ to D₈ = 48</td>
</tr>
<tr>
<td>9</td>
<td>Ife</td>
<td>Depot 9</td>
<td>7,126</td>
<td>S₁ to D₉ = 35</td>
</tr>
<tr>
<td>10</td>
<td>Ilesha</td>
<td>Depot 10</td>
<td>4,546</td>
<td>S₁ to D₁₀ = 40</td>
</tr>
</tbody>
</table>

**Figure 4.** Direct Shipment to all locations (Route 1).

**Figure 5.** Transhipment at D₇, D₉ and D₁₀.

**4.1. Data Analysis for Route 2**

Transshipment at D₇, D₉ and D₁₀: Here products flows from source to the three transshipment points for onward flow to the final destinations. See figures 6 and 7.

**Figure 6.** Matrix Input For Route two from Source to Transshipment Points.
The solution Min Objective for route two from source to transshipment points using greedy heuristic is ₦2,889,327.

The solution Min Objective for route two from transshipment points to final destinations =₦2,961,885. This is demonstrated in Figure 8. Total Cost of distribution from source to transshipment point and from transshipment point to final destination using greedy heuristic.

\[
\sum_{i=1}^{n} \sum_{k=1}^{m} C_{ik} X_{ik} + \sum_{k=1}^{n} \sum_{j=1}^{m} C_{kj} X_{kj}
\]

\[
= 2,889,327 + 2,961,885 = ₦5,851,212
\]

4.2. Artificial Neural Network Model

ANN application procedure has been explained in details in research methodology. The strength of ANN is demonstrated using the equation developed. By linearly combining all the inputs and the appropriate weight or strength of connection using equations generated in chapter three as well as testing the data set give rise to a perfect solution.

Where the weight of connections is (W) and (X) are the input signals. The weight of connection of neurons will represent the unit cost of distributing products to multiple destinations while the input signal is the product demand at the various depots. Cost is an input signal since the unit cost is a function of product destination.

\[
[Y_{K1} = 65355; Y_{K2} = 35599; Y_{K3} = 34301]
\]

<table>
<thead>
<tr>
<th>S/N</th>
<th>C7</th>
<th>C9</th>
<th>C10</th>
<th>D</th>
<th>Y_{K1}</th>
<th>Y_{K2}</th>
<th>Y_{K3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45</td>
<td>27</td>
<td>20</td>
<td>11215</td>
<td>65355</td>
<td>35599</td>
<td>34301</td>
</tr>
<tr>
<td>2</td>
<td>33</td>
<td>20</td>
<td>25</td>
<td>20125</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>32</td>
<td>22</td>
<td>18</td>
<td>18540</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>29</td>
<td>35</td>
<td>21860</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>32</td>
<td>30</td>
<td>11986</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>26</td>
<td>30</td>
<td>15</td>
<td>8348</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recall equation 1,

\[ U_k = \sum_{j=1}^{n} x_j w_{kj} \]  

\[ U_{7k} = \sum_{j=1}^{6} x_j w_{7kj} \] (21)

\[ U_{9k} = \sum_{j=1}^{6} x_j w_{9kj} \] (22)

\[ U_{10k} = \sum_{j=1}^{6} x_j w_{10kj} \] (23)

\[ V_k = U_{7k} + U_{9k} + U_{10k} + bk \] (24)

\[ V_k = \sum_{j=1}^{6} x_j w_{7kj} + \sum_{j=1}^{6} x_j w_{9kj} + \sum_{j=1}^{6} x_j w_{10kj} \] (25)

\[ Y_{7k} = \varphi(.) \sum_{j=1}^{6} x_j w_{7kj} + b_{7k} \] (26)

\[ Y_{9k} = \varphi(.) \sum_{j=1}^{6} x_j w_{9kj} + b_{9k} \] (27)

\[ Y_{10k} = \varphi(.) \sum_{j=1}^{6} x_j w_{10kj} + b_{10k} \] (28)

**Table 3. Performance Value for ANN Modelling (Rough).**

<table>
<thead>
<tr>
<th>Samples</th>
<th>MSE</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>4.35*10^6</td>
<td>0.9688</td>
</tr>
<tr>
<td>Validation</td>
<td>4.53*10^7</td>
<td>0.5950</td>
</tr>
<tr>
<td>Testing</td>
<td>4.08*10^6</td>
<td>-0.990</td>
</tr>
</tbody>
</table>

MSE- Mean Squared Error, R = Regression Coefficient

From table 3, the performance value for the rough iteration obtained is not acceptable since the MSE value for training, validation and testing are very high. It becomes acceptable when R value is close to one and MSE values are also low for the three data set. That is when the trial data for training, validation and testing are quite low. The performance value for the iteration is also suitable when two of the data sets are reasonably low (training and validation or training and testing).

![Figure 10. Plot of ANN Predicted Output against Actual Value for (A) Training (B) Validation (C) Test (D) All.](image-url)
The straight lines in Figure 10 represent the linear relationships between the output and the target data used in this study. The correlation coefficients (R) between the actual and the predicted values are found to be 0.9688 (training), 0.5950 (validation), -0.9990 (testing) and 0.37153 (performance). Also, the goodness of fit of this network as indicated by the average determination coefficient ($R^2 = 0.138$) implies that 13.8% of the data was used for prediction. The value obtained is quite low and certainly not good for prediction. The low coefficients of correlation (validation and testing) obtained demonstrate poor prediction of this network. It was observed that coefficient of correlation of the entire network decreased to 13.8%. The decrease is a sign of very poor performance of the network.

Table 4 shows the MSE and R value of the data set with six samples of which four samples was selected for training, two samples for validation and two samples for testing. The performance value for the second iteration obtained above is fair and better compared to the result obtained from other iteration process, though not too satisfactory since the MSE value for validation and testing are high. It becomes acceptable when R value is close to unity and the results of the three data sets are low. That is when the trial data for training, validation and testing are quite low. The performance value for the iteration is also suitable when two of the data sets are reasonably low (training and validation or training and testing). With the expected aim of minimizing error and obtaining a better or robust output solution.

<table>
<thead>
<tr>
<th>Samples</th>
<th>MSE</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>$4.35 \times 10^4$</td>
<td>0.999</td>
</tr>
<tr>
<td>Validation</td>
<td>$4.53 \times 10^7$</td>
<td>-0.955</td>
</tr>
<tr>
<td>Testing</td>
<td>$4.08 \times 10^3$</td>
<td>0.771</td>
</tr>
</tbody>
</table>

Figure 11. Plot of ANN Predicted Output against Actual Value for (A) Training (B) Validation (C) Testing (D) Target.
Total cost from TP to final destination= ₦1,856,162
Total cost of distribution to the entire network =
Total Cost of distribution from source to transshipment point + Total cost from transshipment point to final destination

\[ \sum_{i=1}^{n} \sum_{k=1}^{m} C_{ik} X_{ik} + \sum_{k=1}^{m} \sum_{j=1}^{n} C_{kj} X_{kj} \]

= ₦ 2,889,327 + ₦1,856,162 = ₦4,745,489

The straight lines in Figure 11 are the linear relationships between the output and the target data used in this study. The correlation coefficients (R) between the actual and the predicted values are found to be 1.000 (training), -0.95504 (validation), 0.77089 (testing) and 0.80369 (performance). Also, the goodness of fit of this network as indicated by the average determination coefficient \( R^2 = 0.6459 \) implies that 64.6% of the data was used for prediction. The value obtained is better compared to other iteration values. It was observed that coefficient of correlation of the entire network increased to 64.6%. The increase is a sign of better performance of the network.

5. Conclusion

In this article, artificial neural network and greedy heuristic model have been successfully implemented and compared in terms of strength and solution power. An illustration of the performance of the proposed algorithm was demonstrated using a bottling company in a multi-echelon system with the aim of reducing cost. From the general analysis, it is certain that cost has been reduced to the barest minimal using GH and ANN. The bottling company spent approximately ₦6,332,304 supplying products from source to destination points using classical perception. At the end of
the research, the analysis showed that ANN gave a better and robust solution. About ₦4,745,489 was achieved as current optimal operational cost using ANN while ₦5,851,212 was obtained from GH model. A total sum of ₦1,586,157.00 would have been saved using ANN. It is concluded that ANN model produced better solution and much faster response than that obtained from GH. This justifies that non-deterministic models usually provide better solutions compared to the transshipment tableau that uses deterministic model input.

References


