Kenan Özden, Atakan Tahsin, "Line Balancing Based on Error Rate Estimation with Artificial Neural Networks in Assembly Line Operations", **Istanbul Gelisim University Journal of Social Sciences,** 10 (1), April 2023, pp. 16-32.

Line Balancing Based on Error Rate Estimation with Artificial Neural Networks in Assembly Line Operations

Kenan ÖZDEN*, Atakan TAHSİN**

Abstract

In this study, in the assembly line systems consisting of the operations in interaction with each other; To reduce the number of faulty products, to prevent poor quality and to reduce the production time, Error Ratio Estimation with Artificial Neural Networks and probabilistic Line Balancing method have been performed. The error rate estimation provides information on which jeans models should be applied in the improvement work to eliminate existing errors in place. In the study, using the Levenberg - Marquardt Learning Algorithm, machine learning was determined by the experimental design method. At the same time, it has been used as an artificial intelligence algorithm in the multi-directional decision making stages, estimation and line balancing parts. In Assembly Line Equilibration, it has been aimed to re-stabilize the unbalanced line with the influence of post-forecasting process recovery. The Probabilistic Line Balancing method has been used because the processing times are stochastic (variable) and statistical data and mathematical algorithms (digital algorithms can be created). When the results are examined, a successful forecasting process has been carried out for two different five-pocket jeans models which has been selected and it has been seen that the work components of the probabilistic line balancing method enable it to be precisely assigned to work stations. And it has given reliable results.

Keywords: Assembly Line Balancing, Artificial Neural Networks, Multilayer Perception Model Probability, Levenberg-Marquardt Learning Algorithm, Probabilistic Line Balancing Method, Apparel Department in Textile, Artificial Intelligence application example to increase Efficiency and Quality.

Montaj Hattı Sistemleri İşlemlerinde Yapay Sinir Ağları ile Hata Tahminine Dayalı Hat Dengelemesi

Öz

Bu çalışmada birbiri ile etkileşim halinde olan operasyonlardan oluşan montaj hattı sistemlerinde; Hatalı ürün sayısını azaltmak, kalitesizliği önlemek ve üretim süresini azaltmak için Yapay Sinir Ağları ile Hata Oranı Tahmini ve olasılıksal Hat Dengeleme yöntemi yapılmıştır. Hata oranı tahmini, mevcut hataları yerinde gidermek için iyileştirme çalışmasında hangi kot modellerinin uygulanması gerektiği hakkında bilgi

Özgün Araştırma Makalesi (Original Research Article)
Geliş/Received: 26.10.2020
Kabul/Accepted: 20.03.2023
DOI: https://dx.doi.org/10.17336/igusbd.812239

* Prof., PhD, Istanbul Gelisim University, Department of Industrial Engineering, Istanbul, Turkey, E-mail: kozden@gelisim.edu.tr ORCID https://orcid.org/0000-0002-3379-2951
** M.Sc. Graduated, Istanbul Arel University, Department of Industrial Engineering, Istanbul, Turkey, E-mail: atakantahsin@gmail.com ORCID https://orcid.org/0000-0001-6045-9874

verir. Levenberg–Marquardt Öğrenme Algoritması kullanılarak yapılan çalışmada deneysel tasarım yöntemiyle makine öğrenmesi belirlenmiştir. Aynı zamanda çok yönlü karar verme aşamalarında, tahmin ve hat dengeleme kısımlarında yapay zekâ algoritması olarak kullanılmıştır. Montaj Hattı Dengeleme ‘de, tahmin sonrası süreç iyileştirme etkisi ile dengesiz hattın yeniden dengelenmesi amaçlanmıştır. İşlem süreleri stokastik (değişken) ve istatistiksel veriler ve matematiksel algoritmalar (dijital algoritmalar oluşturulabilir) olduğu için Probabilistic Hat dengeleme yöntemi kullanılmıştır. Sonuçlar incelendiğinde seçilen iki farklı beş cepli kot pantolon modeli için başarılı bir tahmin süreci gerçekleştirilmiş ve olasılıksal hat dengeleme yönteminin iş bileşenlerinin iş istasyonlarına tam olarak atanmasını sağladığı ve güvenilir sonuçlar görülmüştür.

Anahtar Kelimeler: Montaj Hattı Dengeleme, Yapay Sinir Ağları, Çok Katmanlı Algı Modeli Olasılık, Levenberg-Marquardt Öğrenme Algoritması, Olasılık Hattı Dengeleme Yöntemi, Tekstilde Konfeksiyon Bölümü, Üretim Verimliliğini ve Kaliteyi Artırmak için Yapay Zeka uygulama örneği.

Introduction

Assembly line systems are the production systems that are developed to meet the increasing needs of the humans. These systems are aimed at ensuring that products with high demand for production are manufactured in the shortest time, in the most efficient manner with a low cost and in desired quality. In the process of industrialization, the idea that by dividing the aggregate work into its items (parts, operations); faster, bulk (series) and low-cost production can be conducted by separate workers. As a result, production is carried out through the transfer of materials via a specific line on which different workstations are located. The system is an assembly line that is formed by arranging the materials along the line, by transferring the materials through the flow line using labor or equipment and combining the operations on the part, considering the constraints such as priority relations and cycle time between them. Workers on the line workstations go through one or more of the work items related with them as the semi-finished product to be processed into products pass in front of them. As a result of this process, the incoming parts and semi-finished products become outputs as products after all necessary works are implemented.

When assembly line production is designed for one or more products, the problem of balancing processing times for production line workstations will appear. The aim to solve this problem is to distribute the processes to the stations so that the installed assembly line can be operated efficiently, and each assembly is left with little idle time or no idle time. In other words, the goal is to minimize the total processing time differences among workstations, as there are many operations and a high production speed under existing constraints. On the other hand, the problem of line balancing arises in the planning of the layout of continuous production systems. The case where the work to be implemented during the product formation is assigned to the assembly workstations in such a way that the loss periods are reduced to the lowest is called Assembly Line Balancing or Line Balancing to distribute the work items to the work stations.

The Probabilistic Line Balancing method is used in the problems developed by El-Sayed & Boucher (1985), where the processing times are stochastic (variable). This method is a line balancing method that allows reliable assignment of work items to workstations and provides reliable results since normalized distribution receivers with μ average and σ standard deviation of work item durations are accepted (Baskak, Kalaoglu, & Eryuruk, 2011).

Artificial Neural Networks (ANN) is a logical programming technique developed by imitating the working mechanism of the human brain that aims to realize the biological processes of the human brain with a specific software. It can be defined as an algorithm that can perform the operations that brain does; e.g. make decisions, draw conclusions, arrive at the result of inadequate data in case of insufficient data, accept the continuous data entry, learn and remember in the computer environment (Ozdemir, 2013).Studies on ANN mainly allow the understanding and mathematical modeling of biological neural systems and require the understanding of the physiological structures of biological neural networks (Terence, 1999). The human neural system is a very complex network. Brain is the central element of this system and it is foreseen that there are about $10^{1º}$ neurons (nerve cells) connected to each other by subnets and they have more than $6x10^{13}$ connections. Neural cells are specialized cells for carrying information by an electrochemical process and are generally composed of four different regions: dendrite, soma, axon and synapse (Anderson & McNeill, 1992). The transmission between neurons can be explained briefly as follows. Dendrite receives input signals from other neurons. These signals are electrical responses that are transmitted through the synaptic spaces between dendrites. The axon, which is long and unique, conveys the output signals to other neurons. The combination of axon and dendrite is called synapse. They evaluate the signals received from the neurons and transmit those to the next cell if there is an input above the threshold value (Arıkan Kargı, 2015). Scientists have developed ANN by using the structure and properties of the neural networks formed by biological cells, and there are some structural similarities among them. Such similarities are given at Table 1. Source: (Sagıroglu, Besdok, & Erler, 2003).

Biological Neural System	Artificial Neural Networks
Neuron	Processor unit (Input)
Dendrite	Aggregation function
Cell body	Transfer function
Axons	Artificial neuron output
Synapses	Weights

Table 1. Similarities Between Neural System and Artificial Neural Networks

An artificial neural cell (neuron) is the smallest information processing unit that forms the basis of ANN's work (Arıkan Kargı, 2015). The basic structure of an artificial neuron is indicated on Figure 1 in its general status.

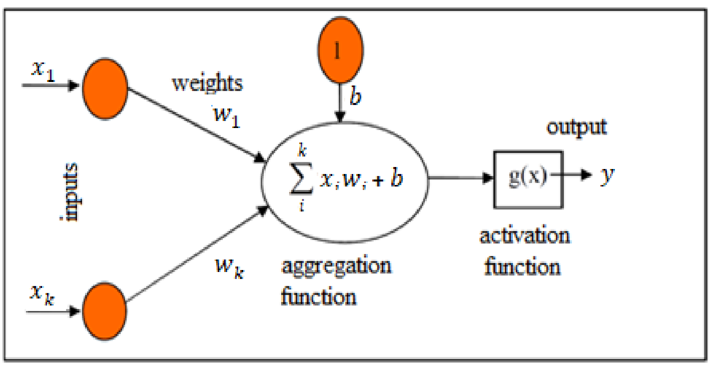


Figure 1. Structure of an Artificial Neural Basic Cell

An artificial nerve cell consists basically of five parts: inputs, weights, aggregation function, activation function and output. Inputs are the date taken inside the artificial neural cell from outside. Weights are the values that indicate the importance of information that comes to an artificial cell and their impact on the cell. The aggregation function is a function that calculates the net input to a cell. Activation (transfer) functions are also called learning curves. Activation functions allow ANN to limit the output amplitude of the neuron to the desired values. The output are the values obtained from the transfer function (Arıkan Kargı, 2015), (Oztemel, 2015). Artificial nerve cells come together to form ANN. Generally, cells form a network by combining three layers (input layer, intermediate / hidden layers, output layer) and in parallel within each layer (Oztemel, 2015). An **input layer** is a unit that receives inputs from the outside and transmits them to the next layer without performing any operation. **Intermediate layers** are layers that allow the data coming from the input layer to be processed and transferred to the output layer. The **output layer** is the layer at the edge of the network. It is the layer that contains the neurons that output from the hidden layer to the outside or to another network by functioning with the function that the network uses (Arıkan Kargı, 2015). The **multi-layer network model** is a typical network that consists of several layers of interconnected neurons. It is the most well-known artificial neural network model which finds wide application areas due to the solutions of nonlinear problems they produce. Multilayer networks are successfully applied to problems that require classification, prediction, recognition and generalization using the backpropagation algorithm. According to the learning algorithm used, the error between the net output and the desired output is propagated backwards again, changing the net weights until the error reaches a minimum threshold. In this way, it is ensured that the input data defined in the system reaches to the most appropriate solution and given out as output from the system. The multi-layer artificial neural network model has various learning algorithms. The algorithms chosen for network design are entirely at the discretion of the user. However, the Levenberg-Marquardt algorithm, which has a low convergence rate and a high risk of captured by local minimum, is known as the most used and best-performing algorithm. The Levenberg-Marquardt algorithm is a combination of the best features of Gauss-Newton and step-reduction algorithms and a highly sensitive technique based on the first-order derivative (Hessian) approach. This learning algorithm has the feature of rapid learning and good convergence (Zhang, Patuwo, & Michael, 1998).

2. Identification of The Problem and The Method Applied

2.1. Identification of the Problem and Purpose

The Company where the application is made is one of Turkey's leading companies in the global production of jeans. The company makes production in 30,000 m^2 closed area, with a capacity of 4.3 million pieces of jeans per year. From 4.22 million products produced between 2015-2016, 865.000 have become second quality and / or useless. 739,000 of these poor-quality products occurred at final transactions assembly line. When final transactions assembly line is considered, the most mistakes were made by robot automats in the spraying and sanding process of 518,000 pieces. This error, which occurs in robot automats, causes yarn or fabric bursts in the crotch, side and plier’s areas due to the inflated model which can be inflated with compressed air. These explosions can be sent back to the sewing line for repair work, and some of the products are assessed as second quality.

In this study, error / fault prediction was made through ANN in order to prevent explosion in the crotch, side and plier’s areas which were formed during the production of this jeans model, considering the production of standard 5 pocket models produced in a jeans (denim) factory. After the prediction process, it is necessary to prevent the mistakes with high error rate, sewing line with high quality in the sewing line, high resistance of the press and high strength, and sewing only where the explosions will occur. This disrupts the current sewing line balance due to new processes. According to the new situation, there is a problem of balancing the line.

In the study, it is aimed to estimate the crotch, side and dart seam areas error rates which are higher than 5% in jeans production and to reduce the existing errors to 2.00% by adding sewing process added to the assembly line after estimation.

3. Method

Levenberg-Marquardt Learning Algorithm which is a Multi-Layer Network model and Probabilistic Line Balancing Method in sewing line balancing are used in predicting error rate.

3.1. Levenberg - Marquardt Learning Algorithm

The Levenberg-Marquardt learning algorithm consists of two phases that calculate the output of the network and calculate "forward" and "backward" by changing the weights. The Levenberg-Marquardt (LM) application process for a single hidden layer and forward feed multi-layer network is described below.

Step 1: Samples are collected, initial values of weights are assigned randomly after the determination of topological structure of the network and learning parameters.

Step 2: Repeat steps 3 to 9 until the necessary condition is met to finish the training.

Step 3: In order to calculate the net weights for each training cluster data, transactions between steps 4 to 8 are applied.

Forward Calculation:

Step 4: It begins by showing the inputs to the input layer of network (G1, G2, ...) from a sample selected from the training set. Inputs are forwarded to the intermediate layer without any processing. The output of the k. neuron at input layer is in the form $Ç_k^I = G_k$.

Step 5: Each input to the neurons in the hidden layer is multiplied by weights $\{w_1, w_2, \ldots w_n\}$ and the net input is calculated as:

$$\mathbf{NET}^a_j = \sum_{k=1}^{n} \mathbf{w_{kj}} Ç^I_k \qquad (1)$$

w_{kj}: it indicates the weight value that connects the k. input layer unit to j. hidden layer unit.

When the sigmoid function (a derivable function) is used as the activation function, the output is as given below. β_j in this equation is the weight of threshold value for j. unit at hidden layer.

$$Ç^a_j = \mathbf{1} \Big/ \mathbf{1} + e^{-\left(NET^a_j + \beta^a_j\right)} \qquad (2)$$

These calculations are made in all neurons, and finally the output values for the output layer are found and the forward calculation phase ends (Karacameydan, 2009).

Backward Calculation:

The first step to renew the weights is to obtain the Hessian matrix. The Hessian matrix is formed by taking second grades derivatives according to performance weights.

$$\boldsymbol{h} = \begin{bmatrix} \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_1^2} & \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_1 \partial \mathrm{w}_2} \cdots & \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_1 \partial \mathrm{w}_n} \\ \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_2 \partial \mathrm{w}_1} & \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_2^2} \cdots & \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_2 \partial \mathrm{w}_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_n \partial \mathrm{w}_1} & \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_n \partial \mathrm{w}_2} \cdots & \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}_n^2} \end{bmatrix} \quad (3)$$

h = Hessian matrix
p = Performance function
w = Synaptic weight of network
m = Number of steps
and the Hessian function is as follows.

$$\boldsymbol{h}(\boldsymbol{m}) = \frac{\partial^2 \mathrm{P(m)}}{\partial \mathrm{w}^2 (\mathrm{m}-1)} \quad (4)$$

Computation of the Hessian matrix is difficult for artificial neural networks. For this reason, the Levenberg-Marquardt algorithm uses the approximate value of the Gauss-Newton matrix and gives the approximate value of the Hessian matrix.

$$\mathbf{h(m)} \approx \mathbf{J^T(m)\, J(m)} + \boldsymbol{\mu}\, \mathbf{I} \quad (5)$$

J(m) = Jacobian matrix
$\mathbf{J^T(m)}$ = Inverse jacobian matrix
I = Unit matrix
m = Number of steps
μ = Marquardt parameter.

The Jacobian matrix is preferred because it is easier to calculate than the Hessian matrix. It consists of first derivatives according to weights of network faults. The Jacobian matrix is defined as follows.

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial \mathrm{w}_1} & \frac{\partial e_1}{\partial \mathrm{w}_2} \cdots & \frac{\partial e_1}{\partial \mathrm{w}_n} \\ \frac{\partial e_2}{\partial \mathrm{w}_1} & \frac{\partial e_2}{\partial \mathrm{w}_2} \cdots & \frac{\partial e_2}{\partial \mathrm{w}_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial e_i}{\partial \mathrm{w}_1} & \frac{\partial e_i}{\partial \mathrm{w}_2} \cdots & \frac{\partial e_i}{\partial \mathrm{w}_n} \end{bmatrix} \quad (6)$$

e = the gradient of the network with error value is obtained by the following equation;

$$\mathbf{G(m) = J^T(m)\;\; E(m)} \quad (7)$$

The weight change correlation for the **LM algorithm** is:

$$\mathbf{a\,(m+1) = a(m)\text{-}\;[J^T(m)\;\; J(m) + \;\mu\, I]^{-1} J^T\,(m)\; E(m)} \quad (8)$$

In this method;

First Rule: If μ is zero, the Newton algorithm based on the Hessian matrix is used.

Second Rule: If μ is a big number, a small step weight gradient reduction algorithm is used (Deveci, 2012).

These operations are repeated until the optimum weights are reached (until the stop criterion is reached).

The steps of recalculating the LM algorithm can be briefly summarized as follows:
Step 6: Calculate the Jacobian matrix and the gradient of the network,
Step 7: Find the Hessian matrix and calculate the μ value using Hessian. If μ = 0, use the newton algorithm, if μ> 0, use the network gradient,
Step 8: Change the weights,
Step 9: Continue with the repeating transaction until the stop criterion is obtained.

3.2. Probabilistic Line Balancing Method

The statistical distribution of task periods in the Probabilistic Line Balancing method is unknown, only μ average and σ standard deviation values are known. This is referred to as Stochastic situation. The compliance of duration of the task (t_i); to a normal distribution with μ average and σ standard deviation also provides the Probabilistic status. In this case, the P (preliminary items) and F (a priori items) matrices of the processes involved in an assembly line production process are generated by using the work flow priority diagrams related to the process. P Matrix contains the transactions performed before the transaction / task, and F Matrix contains the transactions to be performed after the task. After the matrices are generated in this way, the transactions will begin, and the following steps will apply:
Step 1: The rows of the matrix P that contain only zeros will be taken. If this is the case for more than one lines, the task element with the longest duration is selected (each row corresponds to one task element / transaction). This task element is assigned to the work station if the time is appropriate.
Step 2: If the selected task element is assigned, then one will go to F matrix line that has the same line number with this line and the numbers in this line will be obtained; then there will be a return to P matrix, and 0 (zero) values will be written instead of the latest assigned task element among the primary elements in P matrix and the transaction in step 1 will be repeated for the new situation. If not assigned, return to step 1 to either open a new station or select a new task element.
Step 3: By adhering to (E_{max} $t_i \leq T \leq C$) limitation, steps 1 and 2 will continue until all lines in P matrix are used (T: Work Station Duration, C: Cycle Duration).
By adhering to this transaction alignment, transactions begin to be assigned to the stations.
Step 4: By using the "If two independent random variables comply with the normal distribution with μ_1 and μ_2 average, σ_1^2 and σ_2^2 variance respectively, then their aggregate will comply with (μ_1+μ_2) average and (σ_1^2+ σ_2^2) variance normal distribution" theorem; if there are two work elements to assign at one station, when the first work element is assigned, the probability of station duration not exceeding the cycle duration will be calculated and this will be compared with the probability of the station duration not exceeding the cycle duration when the second work element is assigned to the station.
Step 5: Using the z value calculated through the following equation, the value of P (T ≤ C) is read from the normal distribution table. If this probability is smaller than the predetermined probability, this operation is assigned to the station and a value P(T≤C) is calculated for a second operation. This probability continues to be assigned to the station until it exceeds the 0.80 Reliability Grade value. (Here, high-priority transactions are given priority as much as possible). These calculations continue until all transactions are

assigned to stations (Baskak, Kalaoglu, & Eryuruk, 2011), (Eryuruk, Baskak, & Kalaoglu, 2008), (Eryuruk, 2005).

$$\mathbf{z} = \frac{\mathbf{T} - \mathbf{C}}{\boldsymbol{\sigma}_{\mathbf{station}}} \qquad (9)$$

4. Research and Findings

In this chapter, in order to determine whether the sewing of additional operations (crotch, side and dart seam) should be done by using the error ratio estimation process and prediction results with ANN by using Levenberg-Marquardt Algorithm in the production of standard 5 pocket models in denim pants factory and the results of performing the probabilistic line balancing are provided in the table below.

4.1. Research and Findings on Error Rate Estimation

Data realized during the production of the business subject to the application study between January 2016 and December 2016 are used in order to predict the crotch, side and dart seam bursts that occur in robot automats. MATLAB R2015b program is used for error rate estimation.

Variables causing fault; the inputs are indicated by X and the unwashed thickness of the fabric in oz / yd^2 is indicated by X_1 (the weight in ounces of the fabric in 1 yard^2 area), washed thickness of the fabric in oz/yd^2 is indicated by X_2, warp pull percentage is indicated by X_3, muffler pull percentage is indicated by X_4, warp tear endurance in grf unit is indicated by X_5, muffler tear endurance in grf unit (gram force) is indicated by X_6, warp tear endurance in kgf (kilogram force) unit is indicated by X_7, muffler pull endurance in kgf unit is indicated by X_8, elasticity percentage is indicated by X_9, maximum growth percentage is indicated by X_{10}, warp elongation percentage is indicated by X_{11}, cotton percentage in cloth is indicated by X_{12}, woven knitting type is indicated by X_{13} and yarn denier number is indicated by X_{14}. Y (output) values indicating error percentages are also trained in the network as output values. 308 pieces of network data were transferred from the Excel program to the MATLAB program. Then, 70 % of the data were randomly divided into training data and 30 % as test data and the input and output variables were normalized between -1 and +1 using a linear transformation to the range [a, b] for normalization before data sets were given to the network. The fact that the normalized data is between -1 and +1 affects the transfer function used between layers in models. For this reason, it is preferred to use the hyperbolic **tangent sigmoid function** (tansig) as the transfer function in the generated model. The maximum **number of iterations** (epochs) is taken as 1000 while training is implemented on the program, and parameters are used to stop the iteration when the network learns. This is set to prevent the program to overrun and to reach the conclusion in a short period of time. The performance criterion is the average of error squares, and the objective is to have this value closest to zero or zero. For the learning coefficient, a value between 0 and 1 introduced to the network is the coefficient for accelerating or decreasing the learning process. If this number is close to 1, then the network might learn too much and memorize the structure. Therefore, in the classic nnTool program, the initial value of 0.001 was used. Training transaction is implemented after the model was built.

Immediately after the training, the model was tested to determine the most appropriate model. R, R^2 and MAPE criterion were used as error performance in determining the best model. After the model was established, Multilayer Artificial Neural Network model was established for the training of the data. In the input layer of the model, 14 variables are described, and the output error rate is introduced to the network. However, the hidden layer of the network and the number of neurons in this layer are not definite. Timothy Masters (1993) has revealed that having more than one hidden layer does not help in practice and slows learning. For this reason, the number of hidden layers is taken as 1. To determine how many hidden neurons are in the hidden layer, this layer was given 1 to 20 neurons and each model was trained 20 times to try to determine the best model. The largest R value is 0.9451 and the R^2 value is 0.8933. Smallest MAPE value is 0.0367. As line 6 has these values, the result was reached as that neuron number to be 6. Therefore, the most suitable model is selected as the 14-6-1 network structure model and the network training process is completed. The 30 % test data that had previously been separated and never defined to the network was used to measure whether the network learned as a result of the network training completion. In this model for error rate estimation, R = 0.9651 in the training data and R = 0.9451 in the test data. These results indicate that the estimate of 96.55 % for training data and 94.51 % for testing is correct. At the same time, Table 2 shows the actual values of the test data, which are set to estimate the error rate, and the estimated values in the network after the machine learning has taken place. When the table is examined, it is seen that the estimated values are clearly like the actual values. In this case it also shows that the network established is successful and that it can also succeed in a real estimation process.

ER	**ER**	**ER**	**ER**	**ER**	**ER**	**ER**	**ER**	**ER**	**ER**	**ER**	**ER**
,522	,7	,002	,8	,206	,4	,180	,55	,128	,1	,949	,45
,978		,993	,95	,496	,75	,442	,95	,174	,1	,98	,9
,925	,25	,563	,35	,831	,5	,106	,2	,220		,612	,55
,707	,9	,012	,4	,242	,8	,209	,5	,257		,832	,6
,946		,406	,1	,190	,4	,01	,5	,377	,65	,461	,45
,008	,5	,542	,45	,56	,7	,358	,7	,581	,55	,404	,6
,651	,6	,367	,25	,034	,2	,142	,7	,991	,9	,764	
,371	,35	,492	,05	,011	,8	,912	,6	,040	,05	,572	,6
,371	,35	,717	,8	,726	,7	,275	,5	,963	,9	,415	,3
,415	,5	,724	,8	,713	,7	,591	,6	,802	,7	,222	,95
,740	,7	,862	,95	,501	,85	,938		,740	,5	,017	
,989	,2	,194		,737	,7	,191	,15	,099		,218	,8
,992	,05	,391	,45	,677	,7	,718	,8	,213	,05	,242	,15
,534	,5	,612	,65	,011	,95	,107	,1	,01	,95	,659	,25

Table 2 Actual Error Rates (AER) Used as Test Data and Estimated Error Rates (EER) Produced

In adapting the forecasting process, error rate estimation for 2 production orders arrived in January 2017, followed by additional processing of denim models with error rates greater than 5 %. Line balancing was done in order to re-balance the new line formed in the model with additional operation. The 2 incoming orders are standard 5-pocket jeans and the number of units to be produced is 10,000. The input data to be presented to the prediction file is as indicated in Table 3.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}
1st jeans model	8,52	9,86	-3,6	-15	3000	2000	55	25	60	5	0	89	0	120
2nd jeans model	9,44	9,76	-2,3	-4,7	6250	5400	75	65	30	30	30	75,5	1	50

Table 3. Data of Models for Which Prediction Transaction will be Implemented

The estimated error value of the first jeans model was found to be 5.4923 % when the predicted data were run on the previously designed network. The estimated error estimate for the second jeans model was found to be 2,5010 %. In this case, it was decided to apply the crotch, side and dart seam additional sewing and line stabilization work in the first jeans model that is outside the 95 % confidence limits (Tahsin, 2017).

4.2. Sewing Line Balancing with Probabilistic Line Balancing Method

The technological priority diagram for the production process of the first jeans model is given in Fig.2. As indicated in the figure, the production process in the jeans production line consists of 57 processes and workstations including front preparation, front group, rear preparation, rear group, assembly input and assembly output operation groups before the improvement works. P and F matrix is generated by using Figure 2. These matrices created and their task numbers, operations performed, standard task durations, standard deviation values are given in Table 3. On the other hand, the crotch, side and dart seam strengthening, which is the improvement transaction to eliminate the problem of faulty production, is placed between the transactions 54 and 55 in the line "Additional Operation" in Table 4. The aim of this placement is to ensure that the reinforcement is done after the assembly of the denim parts with the machines and that the newly added transaction is prior to the manual operation such as quality control, cleaning, so that this placement is more beneficial for balancing the line. After the P and F matrices are arranged according to the order of precedence, it is necessary to find the cycle duration and the minimum number of stations.

C = T (total time worked in one day) / US (total number to be produced in one day)

It is planned to sew 10.000 products of jeans with 5 pockets in 9,5 days. In this case, the amount to be produced per day is 1053. The total working hours in a day is 9 hours, i.e. 540 minutes.

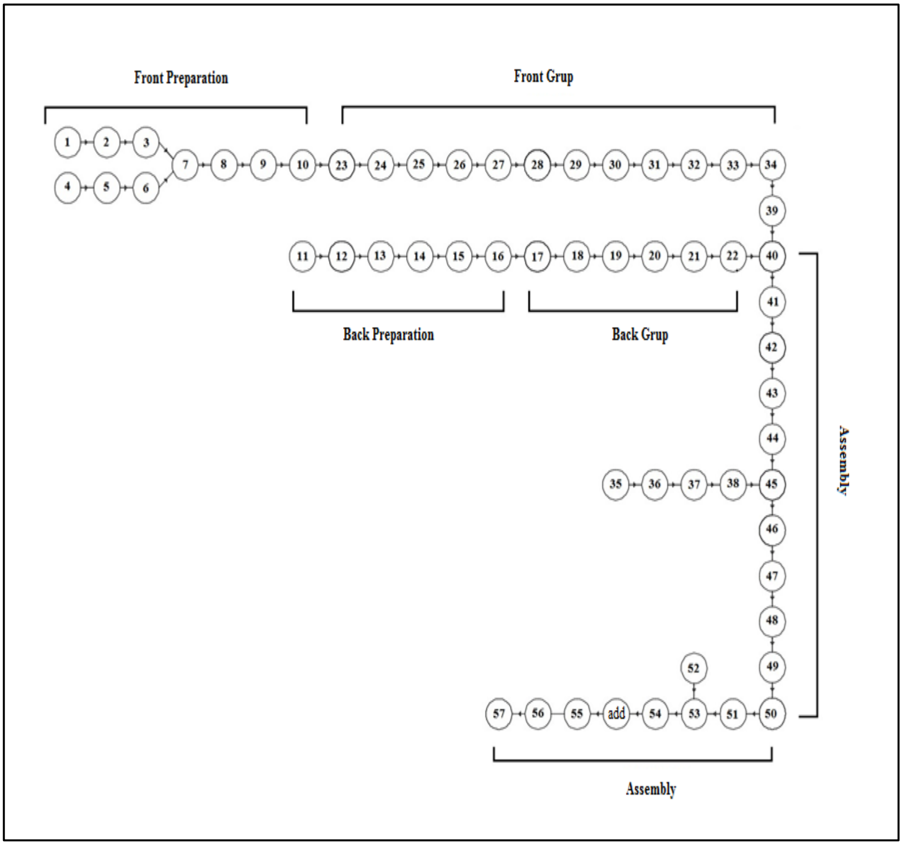
Cycle duration is (C) = 540/1053 = 0.5128 ≈ 0.52 min.

The minimum number of stations (n_{mim}) and total time of work ($\sum t_i$);

$$N_{mim} = [\sum t_i / C] = 12,74 / 0,52 = 24,5 \approx 25. \quad (10)$$

Processes were assigned to the work stations by continuing with the Probabilistic Line Balancing Method given above and the assignment results are given in Table 5.

Figure 2. Priority Diagram for First Jeans Model



Op. No	Operations	Op. Times (min.)	Standard Deviation	P Matrix	F Matrix
1	Fly overlock (Left)	0,06	0,0085	0 0 0	2 0 0
2	Fly overlock (Right +reversing)	0,11	0,0156	1 0 0	3 0 0
3	Front center overlocks (x1)	0,09	0,013	2 0 0	7 0 0
4	Coin pocket bending (right)	0,09	0,013	0 0 0	5 0 0
5	Pressing coin pocket (right)	0,26	0,0371	4 0 0	6 0 0
6	Coin pocket bartacking	0,12	0,0171	5 0 0	7 0 0
7	Stitching coin pocket to front pocket facing	0,06	0,0085	3 6 0	8 0 0
8	Stitching front pocket to front side (x2)	0,17	0,0241	7 0 0	9 0 0
9	Front pocket bag bagging + topstitch (x2)	0,23	0,0126	8 0 0	10 0 0
10	Front pocket bag reinforcement + edge St. x2	0,27	0,0145	9 0 0	23 0 0
11	Back pocket location mark (x2)	0,15	0,0016	0 0 0	12 0 0
12	Back pocket hem bending (x2)	0,06	0,0007	11 0 0	13 0 0

13	Back pocket fancy seam	0,10	0,0012	12 0 0	14 0 0
14	Back pocket stictching (x2)	0,45	0,0051	13 0 0	15 0 0
15	Back pocket press	0,40	0,0047	14 0 0	16 0 0
16	Back pocket asembling (x2)	0,49	0,0054	15 0 0	17 0 0
17	Yoke assembling (x2)	0,27	0,0031	16 0 0	18 0 0
18	Back panel stitching (x2)	0,36	0,0042	17 0 0	19 0 0
19	Back panel overlock (x2)	0,27	0,0031	18 0 0	20 0 0
20	Yoke top stitch (x2)	0,11	0,0012	19 0 0	21 0 0
21	Back center stitching	0,20	0,0023	20 0 0	22 0 0
22	Back top stitch	0,15	0,0016	21 0 0	40 0 0
23	Fly overlock +asembling and lock stitch	0,25	0,0028	10 0 0	24 0 0
24	Fly zipper signing	0,18	0,002	23 0 0	25 0 0
25	Fly top stitch	0,20	0,0023	24 0 0	26 0 0
26	Front pocket binding seam (x2)	0,30	0,0035	25 0 0	27 0 0
27	Front pocket safe stictching (x2)	0,32	0,0038	26 0 0	28 0 0
28	Assembleing the wash label	0,19	0,0019	27 0 0	29 0 0
29	Fly's zipper safe stitch	0,20	0,0023	28 0 0	30 0 0
30	Front asembling	0,27	0,0031	29 0 0	31 0 0
31	Flont center stitching	0,25	0,0028	30 0 0	32 0 0
32	Front bartacking (x2)	0,16	0,0019	31 0 0	33 0 0
33	Front panel stitching (x2)	0,29	0,0031	32 0 0	34 0 0
34	Front panel overlok (x2)	0,22	0,0027	33 0 0	39 0 0
35	Waistband preparing	0,15	0,0016	0 0 0	36 0 0
36	Waistband's fiiling matarials cuts(x2)	0,12	0,0015	35 0 0	37 0 0
37	Matarials stitching	0,07	0,0008	36 0 0	38 0 0
38	Matarials assembling	0,16	0,0019	37 0 0	45 0 0
39	Front fly topstitches	0,11	0,0012	34 0 0	40 0 0
40	Front and back panel asembling	0,04	0,0005	22 39 0	41 0 0
41	Inside leg center stitch	0,33	0,0035	40 0 0	42 0 0
42	Leg center top stitch	0,18	0,002	41 0 0	43 0 0
43	Side overlock	0,45	0,0051	42 0 0	44 0 0
44	Side edge stitch	0,40	0,0047	43 0 0	45 0 0
45	Waistband sign	0,18	0,002	44 38 0	46 0 0
46	Waistband assembling	0,32	0,0038	45 0 0	47 0 0
47	Waistband edge and inside cleaning	0,34	0,0039	46 0 0	48 0 0
48	Waistband edge unseaming	0,35	0,0039	47 0 0	49 0 0
49	Waistband edge stitching	0,40	0,0047	48 0 0	50 0 0
50	Turning inside out of trousers	0,30	0,0035	49 0 0	51 0 0
51	Leg bending	0,37	0,0043	50 0 0	53 0 0
52	Waistband loop preparing	0,07	0,0008	0 0 0	53 0 0
53	Waistband loop assembling	0,24	0,0028	52 0 0	54 0 0
54	Waistband loop side bending (x2)	0,20	0,0023	53 0 0	Add 0 0
Add. Op.	Crotch, side and dart seam areas strengthening	0,15	0,0016	54 0 0	55 0 0
55	Quality control	0,26	0,0031	Add 0 0	56 0 0
56	Fabric and yarn cleaning	0,16	0,0019	55 0 0	57 0 0
57	Waistband buttonhole	0,09	0,0008	56 0 0	0 0 0
	Total	12,74			

Table 4. Probabilistic Line Balancing Data for First Jeans Model

Stations No (k)	Op. No (i)	t_i	T_k	Idle Time
1	1	0,06	0,49	0,03
	2	0,11		
	3	0,09		
	7	0,06		
	8	0,17		
2	4	0,09	0,47	0,05
	5	0,26		
	6	0,12		
3	9	0,23	0,5	0,02
	10	0,27		
4	23	0,25	0,43	0,09
	24	0,18		
5	25	0,2	0,5	0,02
	26	0,3		
6	27	0,32	0,51	0,01
	28	0,19		
7	29	0,2	0,47	0,05
	30	0,27		
8	31	0,25	0,41	0,11
	32	0,16		
9	33	0,29	0,51	0,01
	34	0,22		
10	39	0,11	0,5	0,02
	40	0,04		
	21	0,2		
	22	0,15		
11	11	0,15	0,31	0,21
	12	0,06		
	13	0,1		
12	14	0,45	0,45	0,07
13	15	0,4	0,4	0,12
14	16	0,49	0,49	0,03
15_16	17	0,27	1,01	0,03
	18	0,36		
	19	0,27		
	20	0,11		
17	41	0,33	0,51	0,01
	42	0,18		
18	43	0,45	0,45	0,07
19	44	0,4	0,4	0,12
20	45	0,18	0,5	0,02
	46	0,32		
21	35	0,15	0,5	0,02
	36	0,12		
	37	0,07		
	38	0,16		
22	47	0,34	0,34	0,18

23	**48**	**0,35**	**0,35**	**0,17**
24	**49**	**0,4**	**0,4**	**0,12**
25	**50**	**0,3**	**0,3**	**0,22**
26	**51**	**0,37**	**0,37**	**0,15**
27	**52**	**0,07**	**0,51**	**0,01**
	53	**0,24**		
	54	**0,2**		
28	**Add**	**0,15**	**0,41**	**0,11**
	55	**0,26**		
29	**56**	**0,16**	**0,25**	**0,27**
	57	**0,09**		

Table 5. Probabilistic Line Balancing Results for First Jeans Model

In an Assembly Line problem solution, it is expected to have minimum loss of Balance and high rates of line efficiency. With the establishment of 29 stations as a result of Probabilistic Line Balancing, the line is balanced and the total idle time at stations is 2.34 min.

$$\textbf{LB}\ (\%) = \frac{[(\boldsymbol{n} \times \boldsymbol{C}) - \sum \boldsymbol{ES_i})]}{\boldsymbol{n} \times \boldsymbol{C}} \boldsymbol{x100} \tag{11}$$

$$\textbf{LB}\ (\%) = \frac{[(\mathbf{29} \times \mathbf{0,52}) - \mathbf{12,74}]}{\mathbf{29} \times \mathbf{0,52}} \boldsymbol{x100} = \%\ \mathbf{15,5} \tag{12}$$

$$\boldsymbol{LE}(\%) = \frac{\sum \boldsymbol{ES_i}}{\boldsymbol{n} \times \boldsymbol{C}} \boldsymbol{x100} = \frac{\mathbf{12,74}}{\mathbf{29 * 0,52}} \boldsymbol{x100} = \%\ \mathbf{84,5} \tag{13}$$

5. Conclusion and Evaluation

Prediction is a process of estimating what will happen in the future, and those processes are not final. However, all administrative and production decisions and plans are based on forward-looking predictions. This aspect allows for the estimation of future uncertainties, especially for manufacturing companies, to produce better quality, more functional and more profiting products. Prediction of errors of newly produced products with the help of data of products produced in the past makes production easier to intervene beforehand. Improving the processes by focusing on the product with the higher error rate will produce more products, lower the cost, increase the productivity and profitability. The more balanced and efficient the line at the end of the Probabilistic Line Balancing allows for more products to be manufactured in a shorter time and thus the cost of production can be reduced. The fact that idle periods are reduced to a minimum and distributed uniformly ensures equal distribution of workers workloads. This means that production is continuous, and the production period is kept stable and balanced.

In the study, it is aimed to estimate the crotch, side and dart seam areas error rates which are higher than 5% in jeans production and to reduce the existing errors to 2.00% by adding sewing process added to the assembly line after estimation. When the artificial neural network designed for forecasting is run, the estimated error value of line balancing for the jeans model is found to be 5.4923. It is known that after the production of the first denim model in the current production line with the current production process the crotch, side and dart seam areas error rate is 5.70%. In this case, when the test data taken for the performance criterion and post-production error rates are examined, it is seen that the artificial neural network works well, and its performance is high. Due to the addition of new sewing operations, the existing line will be out of balance and the production line will have to been re-balanced. According to the literature, in assembly line balancing exercises where priority relations such as sewing lines are present and manual or machine operations are very, the probabilistic line balancing method is slightly less efficient than other methods used for solving such problems, but it allows work items to be precisely assigned to workstations (Eryuruk, Baskak, & Kalaoglu, 2008), (Baskak, Kalaoglu, & Eryuruk, 2011).The line balancing used previously for the 5 pocket models of the company is again made with the probabilistic method and the balance loss is 13.6% and the line efficiency is 86.4%. In the new production line balanced by the probabilistic line balancing method, the equilibrium loss is 15.5% and the line efficiency is 84.5%. According to the previous production line, the new line efficiency appears to be about 1.9% lower. This shows that while the line balancing process does not re-balance the line, the additional operations distort the line balance very little. However, this addition, which will provide better quality jeans production and prevent customer complaints and losses, will not be a significant cost loss and it is obvious that this loss will be very little when compared with the alternative costs that will arise from possible customer losses. As a matter of fact, the next 1000 pieces of the first denim model made with the corrected production process by applying the sewing operation to the crotch, side and dart seam areas were tested in order production process. And it has been determined that there are 18 cases of explosion. Accordingly, an error rate of 1.8% achieved less than the target value 2%. However, the error rate in the previous production process is 5.7%, which means that the number of defective products in 1000 production is 57. Monthly production is about 10,000 pieces.

	Current Production Process	**Improved Production Process**
The actual error rate (%)	5,7	1,8
Estimated error rate with ANN (%)	5,4923	-
Loss of balance in production line (%)	13,6	15,5
Line Efficiency in production line (%)	86,4	84,5
Total time for one product (min.)	12,59	12,74
Number of operations	57	58
Number of stations	28	29
Idle time for one product (min).	1,97	2,34
Monthly number of defective products	570	180
Monthly loss (Unit cost price 12.5 Euro)	7.125 Euro	2.250 Euro

Table 6. Current and Improved Production Process Results Comparison Table.

As a result, this study, which is based on error estimation with artificial neural networks, results in a decrease in error rate and costs, resulting in an increase in production efficiency. As a matter of fact, as summarized in Table 6, the total process time, the number of processes, the number of stations, the idle time and the balance loss for a product in the improved production process increase at a small rate, but the number of defective products and the number of defective products decrease. Using the improved production cycle of the first denim model, a reduction of 390 units, which will be achieved in about a month, will provide a cost reduction of € 4.875. Although the additional processing time leads to the production of about 198 fewer products per month, the increase in quality provided by production will lead to higher quality image and increase in potential sales. On the other hand, if the work is applied to other products of the company, the cost reductions to be provided will be much higher.

REFERENCES

ANDERSON, D., MCNEILL, G. (1992). Artificial Neural Networks Technology, New York: Kaman Science Corporation.

ARIKAN KARGI, V. S. (2015). Yapay Sinir Ağ Modelleri ve Bir Tekstil Firmasında Uygulama. Bursa: Ekin Yayınevi.

BASKAK, M., KALAOGLU, F. ve ERYÜRÜK, S. H. (2011). Konfeksiyon Üretiminde İstatistiksel Yöntemle Montaj Hattı Dengeleme. Tekstil ve Konfeksiyon Dergisi. 21.(1), 65-71.

DEVECI, M. (2012). Yapay Sinir Ağları ve Bekleme Süresinin Tahmininde Kullanılması. (MSc Thesis). Ankara: Gazi University.

EL-SAYED, E.A. and BOUCHER, T.O. (1985). Analysis and Control of Production Systems, Prentice Hall Inc., New Jersey.

ERYURUK, S. H. (2005). Bir Konfeksiyon İşletmesinde Montaj Hattı Dengeleme. (MSc Thesis) İstanbul Teknik University.

ERYURUK, S. H. BASKAK, M. ve KALAOGLU, F. (2008). Assembly Line Balancing in a Clothıng Company. FIBRES and TEXTILES in Eastern Europe. 16(1), 66.

KARACAMEYDAN, F. (2009). Yatırım Fonları Net Varlık Değerlerinin Yapay Sinir Ağları Yöntemiyle Tahmin Edilmesi (MSc Thesis). Bozok University.

MASTERS, T. (1993). Practical Neural Network Recipes in C++, 1th ed., Toronto: Academic Press.

OZDEMIR H. (2013). Yapay Sinir Ağları ve Dokuma Teknolojisinde Kullanımı, Tekstil Teknolojileri Elektronik Dergisi. 7.(1), 51-68

OZTEMEL, E. (2015). Yapay Sinir Ağları, 3. Baskı, İstanbul: Papatya Yayıncılık.

SAGIROGLU, S., BESDOK, E. ve ERLER, M. (2003). Mühendislikte Yapay Zekâ Uygulamaları-1, Kayseri: Ufuk Yayıncılık.

TAHSIN, A. (2017). Montaj Hattı Sistemlerinin Yapay Sinir Ağları ile Çözümlenerek Hat Dengelenmesi ve Uygulama. (MSc Thesis). İstanbul Arel University.

TERENCE, F. L. (1999). Feedforward Neural Network Methodology. New York: Springer-Verlag Inc.

ZHANG, G.; PATUWO, B. E. ve MICHAEL Y. H. (1998). Forecasting with Artificial Neural Networks, The State of the Art. İnternational Journal of Forecasting. 14, 35-62.

Özet

Tahmin, gelecekte ne olacağını tahmin etme sürecidir ve bu işlemler nihai değildir. Ancak, tüm idari ve üretim kararları ve planları ileriye dönük tahminlere dayanmaktadır. Bu özellikle imalat şirketleri için daha iyi kalitede, daha işlevsel ve daha kârlı ürünler üretmek için gelecekteki belirsizliklerin tahminine olanak sağlamaktadır. Geçmişte üretilen

ürünlerin verileriyle yeni üretilen ürünlerin hatalarının önceden tahmin edilmesi, üretime önceden müdahale edilmesini kolaylaştırır. Prosesleri daha yüksek hata oranına sahip ürüne odaklanarak geliştirmek daha fazla ürün üretecek, maliyeti düşürecek, verimliliği ve karlılığı artıracaktır. Çalışmada hata oranı tahmini, mevcut hatta hataları yerinde yok etmek amaçlı yapılan iyileştirme çalışmasının, hangi kot (denim) modellerinde uygulanması gerektiği ile ilgili bilgiler vermektedir. Montaj hattı dengelemede ise tahmin işlemi sonrası iyileştirme çalışmasının etkisiyle, dengesi bozulan hattın yeniden dengelenmesi hedeflenmiştir. Yapılan uygulama çalışması sonucunda, seçilen beş cepli değişik iki kot modeli için Yapay sinir ağları ile hata tahmin işlemi gerçekleştirilmiş ve bu tahminlere dayalı olarak Probabilistik Hat Dengeleme yöntemiyle iş öğelerinin, iş istasyonlarına hassas ve güvenilir bir biçimde atanması sağlanmıştır. Aynı zamanda Probabilistik Hat Dengeleme yönteminin kullanılmasıyla hattın daha dengeli ve etkin olması, daha kısa sürede daha çok ürün ortaya çıkarılabilmesine ve buna bağlı olarak üretim maliyetinin düşmesine olanak sağlamaktadır. Boş sürelerin, en aza indirilip düzgün bir şekilde dağılması sayesinde de işçilerin çalışma yüklerinin eşit dağılması sağlanmaktadır. Bu da üretimin sürekli ve üretim süresinin kararlı ve dengeli tutulması anlamına gelmektedir.

Uygulamada, kot pantolon üretiminde % 5'ten daha yüksek olan Ağ, yan ve pens dikiş alanları hata oranlarının tahmin edilmesi ve tahmin sonrası montaj hattına eklenen dikiş işleminin eklenmesiyle, mevcut hataların % 2,00'e düşürülmesi amaçlanmıştır. Tahmin için tasarlanan yapay sinir ağı çalıştırıldığında, kot modeli için hat dengelemesinin tahmini hata değeri % 5.4923 olarak bulunmuştur. Mevcut üretim hattında ilk denim modelinin mevcut üretim prosesiyle üretilmesinden sonra Ağ, yan ve pens dikiş alanları hata oranının % 5,70 olduğu bilinmektedir. Bu durumda performans kriteri için alınan test verileri ve üretim sonrası hata oranları incelendiğinde, yapay sinir ağının iyi çalıştığı ve performansının yüksek olduğu görülmektedir. Yeni dikiş işlemlerinin eklenmesiyle mevcut hattın dengesi bozulmuş ve üretim hattı yeniden dengelenmek zorunda kalmıştır. Literatüre göre, dikiş hatları gibi öncelikli ilişkilerin mevcut olduğu ve manuel veya makine işlemlerinin çok olduğu montaj hattı dengeleme alıştırmalarında, Probabilistik hat dengeleme yönteminin bu tür problemleri çözmek için kullanılan diğer yöntemlerden biraz daha az etkili, ancak iş istasyonlarına iş öğelerinin düzgün olarak atanmasına izin veren bir yöntem oluğu ve güvenilir sonuçlar ve elde edildiğini ispatlamışlardır. Uygulama yapılan şirkette mevcut 5 cepli modeller için kullanılan hatta denge kaybı 13,6 % ve hat etkinliği 86,4 %'tür. Probabilistik hat dengeleme sonuçları kıyaslandığında, hat etkinliğinin yaklaşık 1,9 % oranda güncel hat sıralamasına göre daha düşük olduğunu göstermektedir. Bu da hat dengeleme işleminde hattın yeniden dengelemesine karşın ek işlemlerin hattın dengesini çok küçük oranda bozduğunu göstermektedir. Ancak daha kaliteli kot üretimi sağlayacak, müşteri şikayetleri ve kayıplarını önleyecek şekilde yapılan bu eklemeyle önemli bir maliyet kaybı yaşanmayacağı gibi bu kaybın olası müşteri kayıplarından kaynaklanacak alternatif maliyetlerinin yanında çok küçük kalacağı açıktır. Nitekim ağ, yan ve pens bölgelerine dikim işlemi uygulanılarak düzeltilmiş üretim süreci ile yapılan daha sonraki 1000 adetlik birinci kot modeli sipariş üretiminde süreç test edilmiş ve 18 adette patlama olduğu tespit edilmiştir. Kısacası, % 1,8'lik bir hata oranı elde edilerek % 2'den daha düşük bir hedef değere ulaşmıştır. Bununla birlikte, önceki üretim sürecinde gerçekleşen hata oranı% 5.7'dir, bu da 1000 üretimdeki hatalı ürün sayısının 57 olduğu anlamına gelir. Aylık üretim yaklaşık 10.000 adettir. Sonuç olarak, yapay sinir ağları ile hata tahminine dayanan bu çalışma maliyetlerde düşüşe neden olarak üretim verimliliğinde artışa neden olmaktadır. İlk denim modeli için geliştirilen üretim döngüsünü kullanarak, bir ay içinde elde edilecek 390 birimlik bir azalma, € 4.875 değerinde kazanç sağlar. Her ne kadar ek işlem süresi ayda yaklaşık 198 ürünün daha az üretimine yol açsa da, üretimin kalitenin artması firmanın imajının ve potansiyel satışlarda artışa yol açacaktır. Öte yandan, uygulama şirketin tüm diğer ürünlerine uygulanırsa, genel olarak kazanç çok daha yüksek olacaktır.