applied sciences

MDPI

Article

Assembly Sequence Planning Using Artificial Neural Networks for Mechanical Parts Based on Selected Criteria

Marcin Suszyński * and Katarzyna Peta

Institute of Mechanical Technology, Poznan University of Technology, 60-965 Poznan, Poland; katarzyna.peta@put.poznan.pl
* Correspondence: marcin.suszynski@put.poznan.pl

Abstract: The proposed model of the neural network describes the task of planning the assembly sequence on the basis of predicting the optimal assembly time of mechanical parts. In the proposed neural approach, the k-means clustering algorithm is used. In order to find the most effective network, 10,000 network models were made using various training methods, including the steepest descent method, the conjugate gradients method, and Broyden–Fletcher–Goldfarb–Shanno algorithm. Changes to network parameters also included the following activation functions: linear, logistic, tanh, exponential, and sine. The simulation results suggest that the neural predictor would be used as a predictor for the assembly sequence planning system. This paper discusses a new modeling scheme known as artificial neural networks, taking into account selected criteria for the evaluation of assembly sequences based on data that can be automatically downloaded from CAx systems.

Keywords: assembly sequence planning (ASP); modelling; artificial neural networks

check for updates

Citation: Suszyński, M.; Peta, K. Assembly Sequence Planning Using Artificial Neural Networks for Mechanical Parts Based on Selected Criteria. *Appl. Sci.* **2021**, *11*, 10414. https://doi.org/10.3390/app112110414

Academic Editor: Christian W. Dawson

Received: 15 September 2021
Accepted: 3 November 2021
Published: 5 November 2021

1. Introduction

The technological assembly process is the final and the most important stage of the production process, which determines its labor consumption and the final production costs. For this reason, the development of the most favorable technology to join parts with the given conditions is a difficult task with multi-criteria but is extremely important. Optimization or improvement of assembly at the production planning stage concerns the determination of components having a direct impact on this process.

One of the most important problems at this level is the determination of the most advantageous sequence [1–5] of the assembly and components of the production cycle but also the problem of assembly line balancing (ALB) in linear systems, which in principle are also part of activities occurring at the production process stage. These issues are fundamentally related to the degree of process automation but also to the production conditions in a given enterprise. It should be emphasized that in recent times the issues of determining the assembly sequence based on artificial intelligence methods were not very frequent, despite the rapid development of this field of knowledge and the significance of the problem [2,6,7].

Planning the assembly sequence is crucial because it relates to many of its aspects, including the number of necessary tool changes, the number of assembly directions, or even the design of mounting brackets and other instrumentation, for the analyzed assembly sequence. It also has a major impact on the overall efficiency of the process. These features of the assembly process, along with many others, have a decisive impact on the efficiency of its course, but some of them may also be criteria for assessing assembly sequences for its improvement or optimization. Assembly sequence planning (ASP) consists of determining the feasibility and at the same time, finds the most advantageous, under certain criteria: order of combining assembly units, parts, and assemblies into more complex units, which leads to obtaining a final product or a product that meets all design and functional assumptions. Due to the high complexity of the issue of choosing the appropriate assembly sequence from among all acceptable choices and at the same time remaining feasible is a difficult and complex task. This is due to a large number of possible combinations of the assembly order, as the theoretical number of variants increases exponentially with the number of parts joined. In many industrial cases, when planning the assembly process, no analysis of the sequence or selection of assembly sequences is performed, and this choice is often based only on the engineering knowledge of people directly involved in planning the assembly process, although this area often contains large reserves allowing for improvement and optimizations. This state of affairs results mainly from the difficulty of evaluating even the already generated ones, due to the constraints of the constructional nature of assembly sequences.

In the literature on the subject, the assessment and selection of the most favorable sequence are made according to various criteria, depending on the specificity of plants, availability of devices, etc. Such criteria may be: assembly time, number of changes in assembly direction, number of tool changes, degree of difficulty in reaching the next process state, degree of complexity of assembly unit movements, degree of difficulty in reaching the next process state, the necessary number of reorientations of the base unit during assembly, stability of assembly units, correctness of the assembly course itself, technological production capacity, and economy of the process. Sequence evaluation criteria may also include aspects of safety, reliability, weight, operating economy, technology, ergonomics, aesthetics, or ecology. Importantly, selected data regarding the criteria for evaluating assembly sequences can be obtained automatically from CAD assembly models, for example, the direction of joining parts obtained in this way is related to the number of changes in assembly direction for a specific sequence. Very important for this process are assembly features, which also have a direct impact on the assembly order of parts. Figure 1 presents a summary of the most commonly used criteria for optimizing the assembly process in the selection of assembly sequences in the published and analyzed scientific studies.

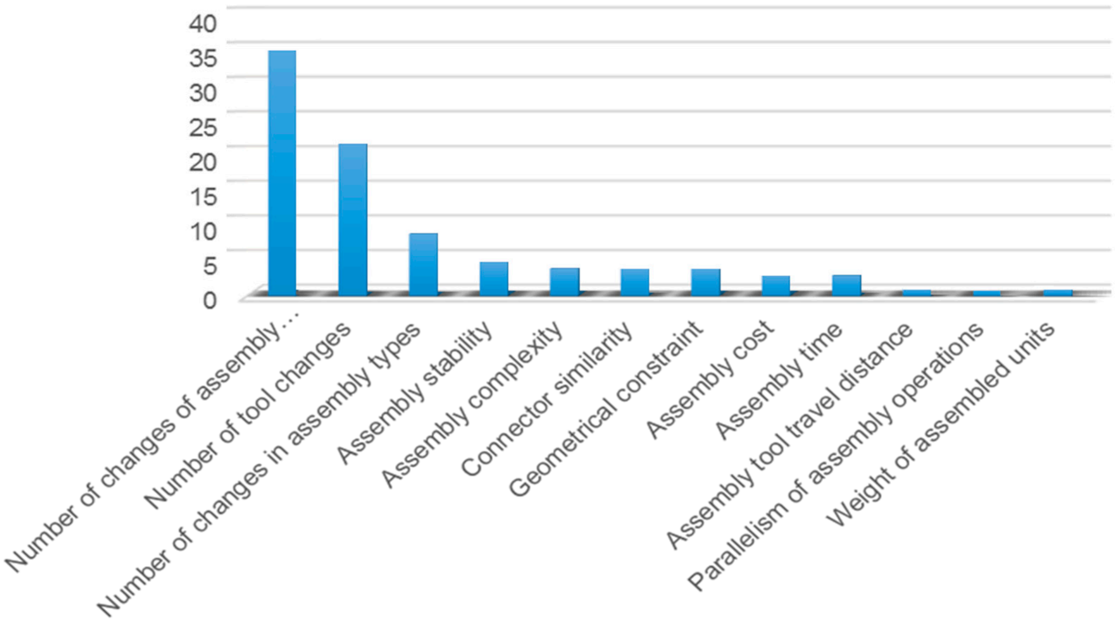


Figure 1. Criteria for the evaluation of assembly sequences in the analyzed scientific publications concerning ASP issues.

The assembly sequence planning problem belongs to a general class of optimization problems known as NP-complete. For this kind of problem, it is necessary to query the whole set of permissible solutions to ensure that the optimal assembly sequence is found. Nevertheless, because this search strategy is very time consuming and impractical in many industrial applications which are complex, have multiple criteria, and often contain issues that prove difficult to optimize, other heuristic techniques are often applied to find a solution close to the optimal one. One solution is also artificial neural networks, which is an information processing paradigm inspired by the natural biological nervous system. The very topic of assembly sequence planning using neural networks was covered in recent years by only a limited number of publications [8,9].

The input vector in neural network is multiplied by the synaptic weights, which are the weight vector. This activity is related to the implementation of the function of the postsynaptic potential and the determination of the value of the y signal calculated based on the sum of input signals multiplied by synaptic weights. The models of artificial neurons can be perceived as mathematical models. What we consider to be the first model of a neural network is the neuron model proposed by W. McCulloch and W. Pitts in 1943 and inspired by the biological model, following the pattern [9]:

$$\text{Transfer function} \;\Rightarrow\; y = f\left(\sum_{i=1}^{k} x_i w_i + w_0\right) \tag{1}$$

Usually, the signal path between neurons (processing units) is as shown in Figure 2, where x_n are the neuron input signals (or the external system input data), w_n are the weights of the edge-connections (synapses), w_o is the neuron's sensitivity threshold (i.e., bias), and $f(\cdot)$ is a simple non-linear function, e.g., a sigmoid or logistic one. Activation (transfer) (AF) functions are possible for each of the hidden and output layers [3].

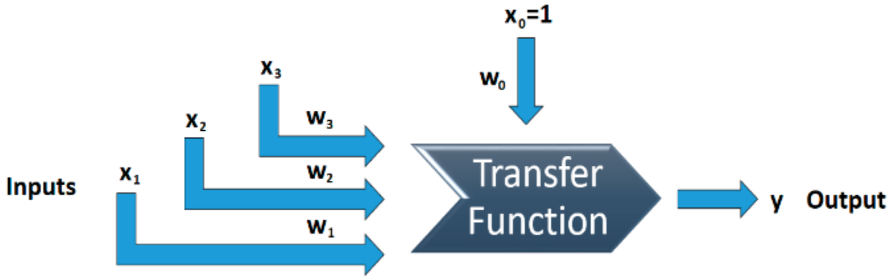


Figure 2. Schematic representation of the artificial neural network.

These studies are aimed at showing the possibility of predicting the assembly time of mechanical products based on variable factors influencing this parameter. The advantage of using artificial neural networks over other optimization algorithms is the ability to predict the assembly time without knowing the mathematical model that describes this phenomenon. This allows to obtain adequate results, also in the conditions of having incomplete production data. This procedure is indirectly aimed at indicating the assembly sequence by selecting the least time-absorbing solution. The article focuses on the application of artificial neural networks as a universal tool of artificial intelligence to support predictive tasks in the area of assembly of machine and device parts. The authors did not find any articles in which the issue of minimizing assembly time, which is important from the point of view of production efficiency, is solved with the use of art neural networks or other methods corresponding to the current trends in the use of artificial intelligence methods. Difficulties in developing an assembly time prediction procedure are mainly focused on providing an appropriate number of examples teaching the neural network. This was achieved by experimentally testing the operation of the network after each set of 100 examples was prepared. The criterion for accepting the network model for further analysis was the achievement of network efficiency during verification at a level greater than 90%. This publication should contribute to a better explanation of the relationship between the determinants of the technological process and its time consumption.

This paper discusses a modelling scheme known as artificial neural networks. The neural network approach has been used for analyzing all feasible assembly sequences. This network structure is suitable for this kind of problem. Proposed assembly planning system is a graph-based approach in the representation of product.

2. Related Works

One of the most important issues in determining the assembly sequence is the appropriate data structure, which means graph representations, mainly directed graphs or hypergraphs. This kind of structure can be considered as formalisms to encode the feasible assembly sequences. To determine all feasible sequences an appropriate graph search algorithm is necessary. The commonly used algorithm for directed graphs or hypergraphs is a heuristically guided search algorithm A*. Although exhaustive search is the simplest and most popular strategy ensuring the complete of the task, it is quite often impractical. This approach is usually used in cases where the number of parts is small (simple assembly objects). In the case when the number of parts increases, these strategies may have limitations due to the problem of combinatorial explosion.

Studies on ASP have implemented different heuristics optimization algorithms such as genetic algorithm, simulated annealing, evolutionary algorithm, ant colony optimization algorithm, and immune and other heuristic methods [10–17].

In paper [10] to solve the assembly sequence planning of a certain type of product, first of all, the rule of nomenclature is designed. Secondly, geometric feasibility and coherence are designed as constraint conditions and these two are combined with each other as the objective function. Finally, authors proposed a novel method under the name of immune particle swarm optimization algorithm. The results show that the immune particle swarm algorithm can be effective and useful in solving the problem of planning the assembly sequence.

Authors of [12] address assembly sequence planning problem and propose an improved cat swarm optimization (CSO) algorithm and redefine some basic CSO concepts and operations according to ASP characteristics. The feasibility and the stability of this improved CSO are verified through an assembly experiment and compared with particle swarm optimization.

Paper [13] proposes an ASP algorithm based on the harmony search (HS), which has an outstanding global search ability to obtain the global optimum. To solve the sequence planning problem, an improved harmony search algorithm is proposed in four aspects: (1) an encoding of harmony is designed based on ASP problems; (2) an initial harmony memory (HM) is established using the opposition-based learning (OBL) strategy; (3) a particular way to improvise a new harmony is developed; and (4) a local search strategy is introduced to accelerate the convergence speed. The proposed ASP algorithm is verified by two experiments.

In paper [17], an attempt is made to generate optimal feasible assembly sequences using design for assembly concept by considering all the assembly sequence testing criteria from obtained feasible assembly sequences. To generate all sets of assembly sequences a simulated annealing technique is used. Sequences consist of $n - 1$ levels during assembly, which are reduced by the DFA concept. DFA uses functionality of the assembled parts, material of the assembled parts, and liaison data to reduce the number of levels of the assembly by considering the directional changes as the objective function.

In this article, an assembly sequence planning system is proposed. The neutral network structure is suitable for this kind of problem. The network is capable of predicting the assembly time, which allows one to choose the best assembly sequence from all the feasible sequences.

3. Methodology

3.1. The Scope of Research Studies

The following research tasks were performed:

- Indication of determinants affecting the assembly time (number of tool changes, the number of changes in assembly direction and its stability);
- Measurement of assembly time on an example mechanical part;
- A set of prepared input and output data has been implemented in the neural network;
- Determination of constant parameters of the neural network model:

1. 3 input neurons (number of tool changes, number of changes in assembly directions, and stability of the assembly unit) and 1 output neuron (assembly time);
2. Percentage of teaching (80%), testing (10%), and verification (10%) examples;
3. Regression model (determination of the quantitative and floating-point numerical values).

- Development of the most effective model of neural network:
    1. Changing network learning algorithms (steepest gradient, scaled conjugate gradient, Broyden–Fletcher–Goldfarb–Shanno, and RBFT radial basis function teaching);
    2. Network topography (multilayer perceptron and network with radial basis functions);
    3. Activation functions (linear, sigmoidal, exponential, hyperbolic, and sine);
    4. Number of hidden neurons (1–12).
- Selection of the most effective network model, taking into account the error of the sum of squared differences generated by the network;
- Introduction of previously untested data to the network, allowing verification of the effectiveness of prediction of assembly time.

This methodology first defined general research tasks and then performed network testing based on a specific example of a mechanical part. The graphic concept of the proposed method is presented in Figure 3.

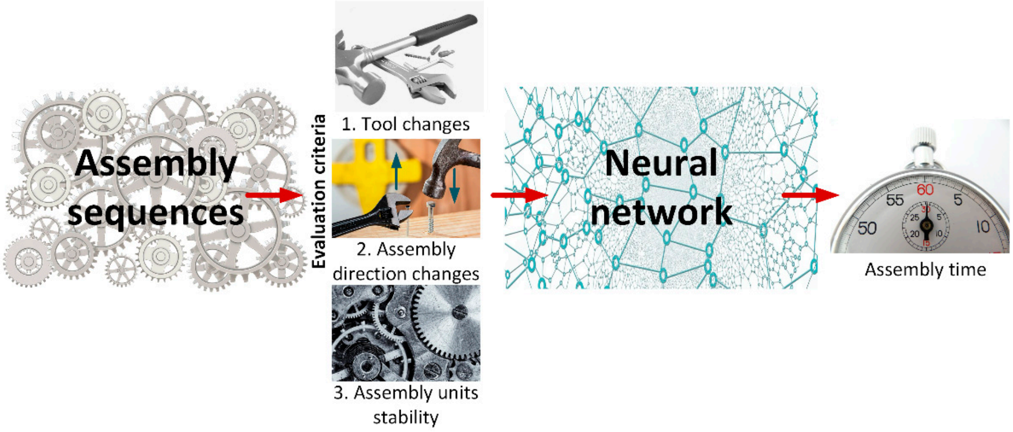


Figure 3. The graphic concept of the proposed method of prediction of assembly time.

In the future, this research may be extended to the verification of other parts of machines and devices based on the developed model of the neural network. The selected criteria determining the assembly time are universal and it is assumed that they are also adequate to other solutions.

3.2. *Assessment Criteria for the Assembly Sequence*

The proposed tool, based on artificial neural networks, has the objective to support the determined sequence for manual assembly (although it is also possible to apply it, albeit after modifications, to an automated process). It was assumed that at the current stage of research it is used in a specific mechanical production company, where the conditions of the assembly process for newly introduced products are subject to ASP analysis, and the processes implemented were used to teach the network. This applies to issues related to, for example, the available machine park, production organization, process control and supervision, or the level of training of employees, especially in the aspect of manual assembly.

The following assembly sequence evaluation criteria were used as input to the process:

- Number of tool changes for the respective assembly sequence.
  This criterion indicates the number of tool changes during assembly operations. Operation constitutes the main structural element of a technological assembly process. In

this work, operations should be understood as, for example, activities such as riveting, drilling, fitting, and screwing, which are related to changing tools. Depending on the type of parts to be installed, the required tools can be assigned to them in a simple manner, from the set of tools utilized in the considered assembly process.

- The number of changes in assembly direction for the respective assembly sequence. It is the most frequent optimization criterion in ASP. This criterion is connected with the direction in which the parts are attached during their assembly. There are 6 main assembly directions, along the 3 main axes: ± X, ± Y, and ± Z.
- Number of stable and unstable units for the specific assembly unit. Stability criterion determines the number of stable and unstable units for a particular assembly sequence. We assume that a stable unit is such a unit that remains in an assembled state, regardless of the force applied to it. The applied forces may be the force of gravity or the forces associated with the movement of parts or an assembly unit.

We justify the adoption of these criteria for the evaluation, among others, with the fact that, as one of the few, they can be automatically obtained from the CAD assembly model, although it is also assumed that the data can be completed manually.

The purpose of the system is to assist in the estimation of time for all acceptable sequences under constructional constraints (i.e., feasible ones) and thus enable the selection of the most favorable one under existing manufacturing conditions. Under these evaluation criteria it is the sequence with the lowest number of tool changes, the smallest number of changes in assembly directions, and the smallest possible number of unstable states that will likely be indicated as the most favorable one; however, it is practically impossible to obtain such values with these criteria for a single sequence. This is related, for example, to the weights of individual criteria in relation to the specific assembly process.

3.3. Neural Network Assumptions

Artificial neural networks were used to evaluate the sequence of combining assembly units. For this purpose, the input and output features of the network were selected and a set of teaching examples was prepared. The input data were the number of tool changes, the number of changes in the assembly direction, and assembly stability, while the assembly time was classified as the group of output data. An important task is to provide an appropriate number of training samples and identify connections between data, which when combined allow for obtaining sufficient results and network efficiency [18]. In order to prepare the training dataset, the numerical values of individual features were normalized, allowing one to obtain independence between all analyzed data and to ensure equivalence. The numerical values of the features initially appearing in different ranges were scaled to values in the range <0.1> using a linear transformation. The task of data normalization was performed by the min-max function, calculating the difference between the scaled value and minimum value and scaling it by the range of numerical data according to the formula:

$$X^* = \frac{X - min(X)}{\max(x) - min(x)} \tag{2}$$

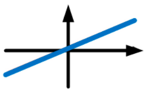
To obtain adequate efficiency, neural network training is performed, consisting of minimizing the prediction error function determined by the sum of squares (*SOS*) as defined by the formula:

$$SOS = \sum_{i=1}^{n} (y_i - y_i^*)^2 \tag{3}$$

where: n is the number of training examples, y_i is an expected network output value, and y_i^* is an actual network output value.

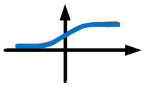
The error surface is paraboloid-shaped with one distinct minimum, it is associated with the neurons belonging to the output layer, and it is calculated after each epoch—repeating the training algorithm. The error is related to a discrepancy between the values obtained at the network output and the reference values included in the training dataset. Errors are also determined for neurons in hidden layers by backpropagation, which consists of adjusting the weight values depending on the assessment of the neuron error in a multilayer network, using gradient optimization methods. The error backpropagation algorithm is implemented in the direction from the output layer to the input layer, which is the opposite direction to the information flow. The effectiveness of a neural network is directly related to the error function and is calculated as the ratio of correctly classified or approximated cases to all cases included in the dataset. In order to obtain the highest efficiency of prediction, the parameters describing the neural network model were changed and empirically selected: the number of layers (input, output, and hidden) and the included neurons, the presence of an additional neuron—bias and network learning rules, including the learning algorithm and activation function. The input layer consists of neurons to which the input signals are sent to the first hidden layer. The set of input data is divided into three groups: (1) training data string that allows reflection on prediction tasks, (2) test data, which check the operation of the network, (3) verification data, which evaluate the network performance based on new, previously unused set of numerical data. The number of neurons and hidden layers is selected empirically, enabling a compromise between its extensive structure and the correct generalization of the processed data. The output layer of the network is a collection of neurons representing the output signals. The number of neurons in the output layer is identical to the number of output data points constituting the result of the network. In addition, in the model of the neural network there may be an additional neuron bias, called the artificial signal generator, constituting an additional input for the neuron with a value of +1 and improving the stability of the network during the training process. The effectiveness of the network is determined by the activation function of hidden and output neurons, which take the following form: linear (directly transmitting the excitation value of the neuron to the output), logistic (sigmoidal curve with values greater than 0 and less than 1), exponential (with a negative exponent), and hyperbolic (hyperbolic tangent curve with values greater than −1 and less than 1). To verify the threshold value of the input signal needed to activate the neuron, the activation functions *f(x)* are used:

- Linear with output values in the range from $-\infty$ to ∞:



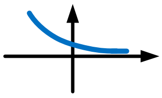
$$f(x) = x \tag{4}$$

- Logistic (sigmoidal) with output values in the range from 0 to 1:



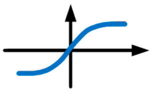
$$f(x) = \frac{1}{1-e^{-1}} \tag{5}$$

- Exponential with output values in the range from 0 to ∞:



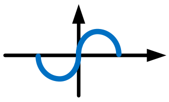
$$f(x1) = e^{-x} \tag{6}$$

- Hyperbolic (hyperbolic tangent) with output values in the range from -1 to 1:



$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{7}$$

- Sine with output values from the range from −1 to 1:



$$f(x) = \sin(x) \tag{8}$$

The selection of the neural network learning algorithm affects its effectiveness. The general principle of the learning algorithms is to minimize the error function by iteratively modifying the weights assigned to neurons. The learning process involves entering successive learning cases containing information and correct network responses to a set of input values. The iterative algorithm is stopped when the ability to generalize the learning results deteriorates. There are many neural network learning algorithms. In this study, the methods of steepest descent, gradient scaling, and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm were used. In the steepest descent method, after specifying the search direction, the minimum value of the function in this direction is determined, as opposed to the simple gradient method, which uses a shift with a constant step. An important feature of the steepest descent method is that each new direction towards the function optimum is orthogonal to the previous one. Movement in one direction continues until this direction turns out to be tangent to a certain line of constant value of the objective function. The principle of the steepest slope, when designating subsequent search directions, requires carrying out a large number of searches along the successively proposed straight lines. In this situation, a neural network teaching method based on conjugate directions is a better solution. The algorithm determines the appropriate direction of movement along the multidimensional error surface. Then a straight line is drawn over the error surface in this direction and the minimum value of the error function is determined for all points along the straight line. After finding the minimum value along the initially given direction, a new search direction is established from this minimum and the whole process is repeated. Accordingly, there is a constant shift towards decreasing values of the error function until a point is found which corresponds to the function minimum. The second derivative determined in this direction is set to zero during the next learning steps. To maintain the second derivative value of zero, the direction's conjugate to the previously chosen direction is determined. Moving in the conjugate direction does not change the fixed (zero) value of the second derivative computed along the previously selected direction. Determining the conjugate direction is associated with the assumption that the error surface has a parabolic shape. The Broyden–Fletcher–Goldfarb–Shanno algorithm refers to a quasi-Newton algorithm that modifies the weights of the interneural connections after each epoch based on the mean error gradient. The principle of operation is based on the search for the minimum squared error function with the use of a Hessian matrix (a matrix of partial derivatives of the second-order), the inverse of which is generated by an algorithm that initially uses the steepest descent method, and in the next step it refers to the estimated Hessian. For radial networks, standard learning procedures are used, including k-means center determination, k-neighbor deviation, and then output layer optimization. The k-means method is a method that consists of finding and extracting groups of similar objects (clusters). Thus, k different clusters are created; the algorithm allows one to move objects from one cluster to another until the variations within and between clusters are optimized. The similarity of data in a cluster is supposed to be as large as possible and separate clusters should differ as much as possible from each other. In the k-neighbor method, each dataset is assigned a set of n values that characterize it and then placed in an n-dimensional space. Assigning data to an existing group consists of finding the k-nearest objects in n-dimensional space and then selecting the most numerous group.

The different types of neural network topologies differ in structure and operating principles, the basis of which are the multilayer perceptron (MLP) and the network with radial basis functions (RBF). The multilayer perceptron consists of many neurons arranged in layers that calculate the sum of the inputs, and the determined excitation level is an argument of the activation function and then the calculated network output value. All neurons are arranged in a unidirectional structure in which the transmission of signals takes place in a strictly defined direction—from input to output. A key task in MLP network design is to determine the appropriate number of layers and neurons, usually performed empirically. A network with radial base functions often has only one hidden layer, containing radial neurons having a Gaussian character. On the other hand, a simple linear transformation is usually applied to the output layer. The task of radial neurons is to recognize the repetitive and characteristic features of input data groups.

In order to elaborate on the best model of the network, a number of constant and variable parameters were determined, tested by the multiple random sampling method, resulting in 10.000 network variants. The error of the sum of squared differences generated for each set of test parameters was established as the criterion for network effectiveness. The constant parameters of the artificial neural network are:

1. 3 input neurons (number of tool changes, number of changes in assembly directions, and stability of the assembly unit) and 1 output neuron (assembly time);
2. Percentage of teaching (80%), testing (10%), and verification (10%) examples;
3. Regression model (determination of the quantitative and floating-point numerical values).

Variable network parameters that were altered randomly during the generation of network models were:

1. Network learning algorithms (steepest gradient, scaled conjugate gradient, Broyden–Fletcher–Goldfarb–Shanno, and RBFT radial basis function teaching);
2. Network topography (multilayer perceptron and network with radial basis functions);
3. Activation functions (linear, sigmoidal, exponential, hyperbolic, and sine);
4. Number of hidden neurons (1–12).

4. Results and Discussion

4.1. Product Structure and Results

The structure of the product intended for assembly is presented in the form of a modified directed graph of assembly states. Moreover, we assumed that parts in the directed graph (digraph) (or the assembled units) are marked as vertices, while the directed edges demonstrate the possible sequences (paths) for assembling them. It is further assumed that the assembly of further elements takes place by adding a part or subassembly consisting of more parts (treated as a single assembled part) to the nth stage assembly. The directed edges connecting the vertices contain information about the stability of the newly formed assembly state, the direction of attachment of the parts, and the tool applied. The described digraph can be generated automatically, based on the CAD assembly drawing.

The basis for executing ASP according to the defined criteria for a specific assembly process is the determination of all assembly sequences that are feasible due to constraints of a constructional nature. The matrix record (e.g., in the form of an assembly states matrix or an assembly graph matrix) of assembly units enables us to determine all variants of assembly sequences using the appropriate algorithm (this procedure is not discussed here and it is reduced to finding all the paths in the digraph leading from the starting vertex x_s, constituting the base part, to the final vertex x_e, i.e., the last state of the assembled product—$x_s, \ldots, x_e$).

The task of determining the sequence of assembly using artificial neural networks was performed for a sample product—a forklift door, consisting of eight main assembly units:

- 1: door welded construction;
- 2: lock;
- 3: cassette lock;
- 4: door reinforcement bar with passenger's handle;
- 5: lock cover;
- 6: door seal;
- 7: lower glass;
- 8: upper glass.

In the first stage, using the construction documentation, the base part was determined in the form of assembly unit no. 1. Then, a digraph of the structural limitations of the assembly states was constructed, shown in Figure 4.

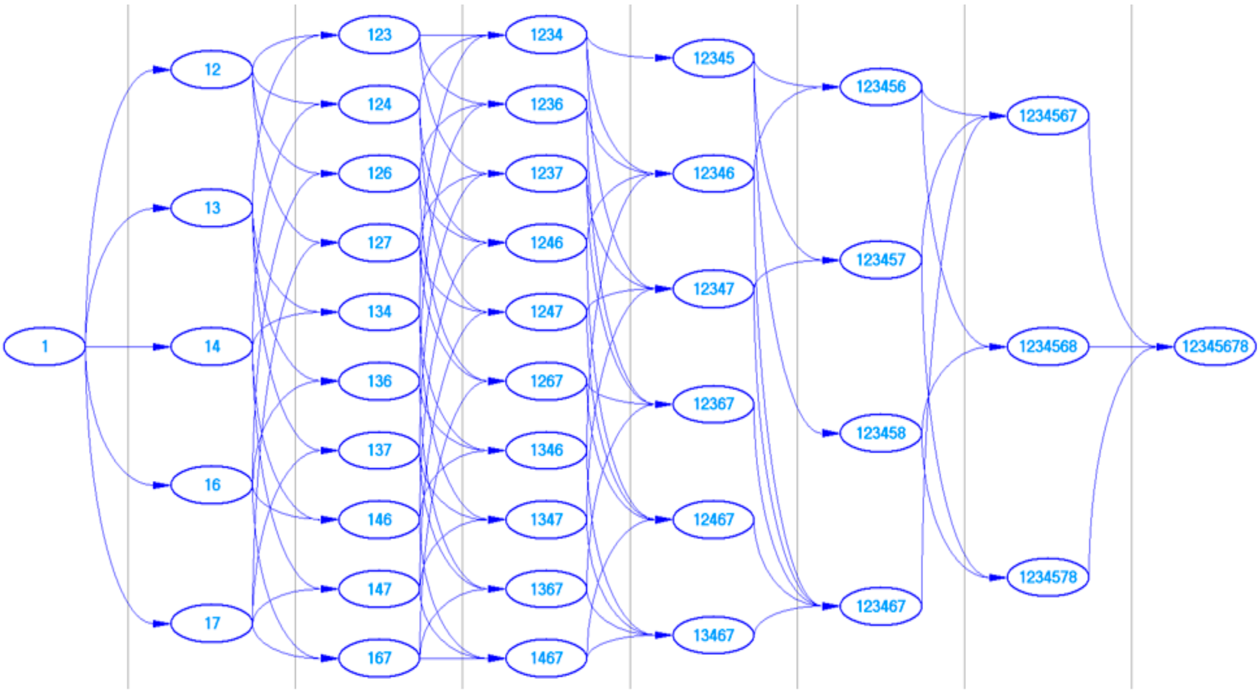


Figure 4. Digraph of the structural constraints of the forklift door assembly states.

It was assumed that the assembly of subsequent units takes place by adding another assembly unit to the assembly state of the nth stage. Based on the constructed digraph recorded in the form of the assembly state matrix, we determined, with the use of a selected graph search algorithm 252, those assembly sequences that were possible under the constraints of the structural nature (Table 1) [19,20], which constitute the basis for further analysis.

Table 1. Selected feasible assembly sequences generated due to design constraints.

No.	Start	1	2	3	4	5	6	STOP	RESULTING SEQUENCE
1	1	12	123	1234	12345	123456	1234567	12345678	12345678
2	1	12	123	1234	12345	123456	1234568	12345678	12345687
3	1	12	123	1234	12345	123457	1234567	12345678	12345768
4	1	12	123	1234	12345	123457	1234578	12345678	12345786
5	1	12	123	1234	12345	123458	1234568	12345678	12345867
6	1	12	123	1234	12345	123458	1234578	12345678	12345876
7	1	12	123	1234	12346	123456	1234567	12345678	12346578
8	1	12	123	1234	12346	123456	1234568	12345678	12346587
9	1	12	123	1234	12346	123467	1234567	12345678	12346758
10	1	12	123	1234	12347	123457	1234567	12345678	12347568
11	1	12	123	1234	12347	123457	1234578	12345678	12347586
12	1	12	123	1234	12347	123467	1234567	12345678	12347658
13	1	12	123	1236	12346	123456	1234567	12345678	12364578
14	1	12	123	1236	12346	123456	1234568	12345678	12364587
15	1	12	123	1236	12346	123467	1234567	12345678	12364758
…	…	…	…	…	…	…	…	…	…
252	1	17	167	1467	13467	123467	1234567	12345678	17643258

Table 2 presents the most effective neural networks for predicting the assembly time of the discussed product. By assessing the values of the sum of squared differences error and the effectiveness of the selected neural networks, it was found that the best results were obtained for network no. 9—the 3-8-1 RBF (Figure 5). We selected it for further analysis (a network with radial basis functions with three input, eight hidden, and one output neurons), in which hidden neurons were activated by a Gaussian function, and output neurons by a linear one, obtaining about 99% efficiency for the group of verification data.

Table 2. Values of neural network parameters that were found best for prediction of assembly time.

Network No.	**Network Name**	**Effectiveness (Learning)**	**Effectiveness (Testing)**	**Effectiveness (Verification)**	**SOS Error (Learning)**	**SOS Error (Testing)**	**SOS Error (Verification)**	**The Learning Algorithm**	**Activation (Hidden Neurons)**	**Activation (Output Neurons)**
1	RBF 3-7-1	0.4146	0.7848	0.9926	0.0229	0.0587	0.0050	RBFT	Gaussian	Linear
2	RBF 3-9-1	0.4381	0.7643	0.9958	0.0241	0.0698	0.0112	RBFT	Gaussian	Linear
3	RBF 3-8-1	0.4050	0.9764	0.9929	0.0231	0.0456	0.0033	RBFT	Gaussian	Linear
4	RBF 3-2-1	0.0794	0.9668	0.9913	0.0274	0.0636	0.0090	RBFT	Gaussian	Linear
5	RBF 3-7-1	0.4516	0.9759	0.9925	0.0220	0.0574	0.0042	RBFT	Gaussian	Linear
6	RBF 3-2-1	0.0794	0.9668	0.9913	0.0274	0.0636	0.0090	RBFT	Gaussian	Linear
7	RBF 3-2-1	0.0794	0.9668	0.9913	0.0274	0.0636	0.0090	RBFT	Gaussian	Linear
8	RBF 3-2-1	0.0794	0.9668	0.9913	0.0274	0.0636	0.0090	RBFT	Gaussian	Linear
9	RBF 3-8-1	0.4522	0.9778	0.9942	0.0220	0.0574	0.0042	RBFT	Gaussian	Linear
10	RBF 3-6-1	0.4207	0.8487	0.9981	0.0227	0.0567	0.0038	RBFT	Gaussian	Linear

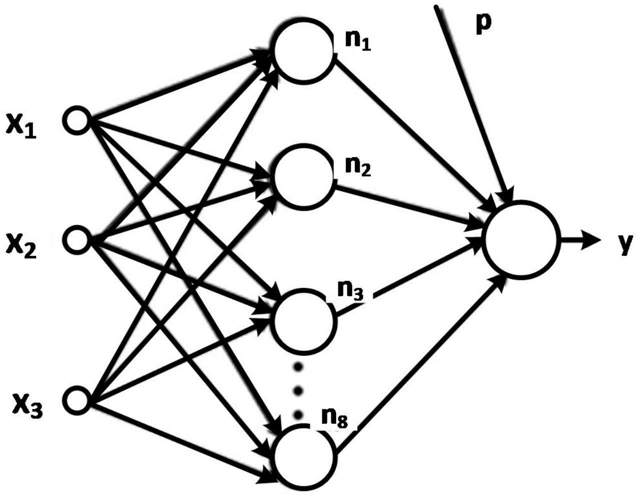


Figure 5. RBF network model (x_1 is the number of tool changes, x_2 is the number of changes in assembly directions, x_3 is a stability of the assembly unit, n1–n8 are hidden neurons, p is a bias, and y is an assembly time).

Figure 6 presents the changes in the value of the learning error of the selected RBF network depending on the number of learning cycles. The neural network was found in the first learning cycle—after the first iteration of the training algorithm. The stabilization of the error value occurred in the sixth learning cycle.

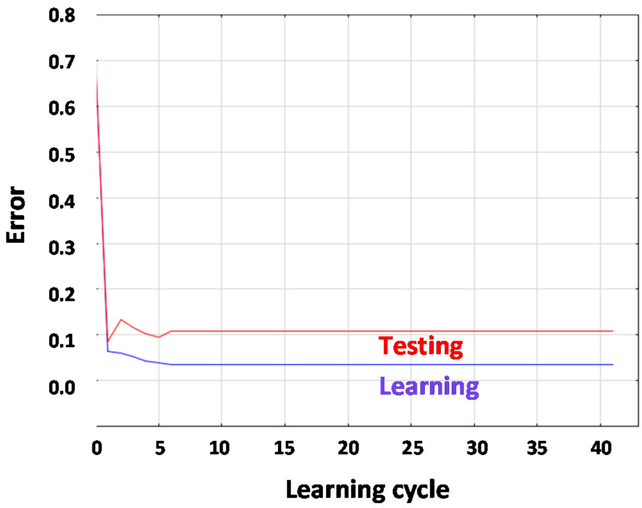


Figure 6. Changes in the value of network learning errors depending on the number of learning cycles.

In the learning process of the neural network, the weight values for all neurons are adjusted. This has an impact on the obtained results because the weights can weaken (negative values) or strengthen (positive values) the signals transferred by individual layers of the network. Table 3 presents the weight values generated for the analyzed RBF network.

Table 3. Neural network weights for prediction of assembly time and network parameters that were found best for prediction of assembly time.

Connections RBF 3-8-1	Weight Values	Connections RBF 3-8-1	Weight Values	Connections RBF 3-8-1	Weight Values
X_1—hidden neuron 1	0.400000	X_3—hidden neuron 5	1.000000	Radial range hidden neuron 5	0.640312
X_2—hidden neuron 1	0.500000	X_1—hidden neuron 6	0.600000	Radial range hidden neuron 6	0.200000
X_3—hidden neuron 1	1.000000	X_2—hidden neuron 6	0.500000	Radial range hidden neuron 7	0.200000
X_1—hidden neuron 2	0.00000	X_3—hidden neuron 6	1.000000	Radial range hidden neuron 8	0.200000
X_2—hidden neuron 2	0.00000	X_1—hidden neuron 7	0.400000	Hidden neuron 1—y	0.044928
X_3—hidden neuron 2	1.000000	X_2—hidden neuron 7	0.00000	Hidden neuron 2—y	−0.059589
X_1—hidden neuron 3	0.200000	X_3—hidden neuron 7	1.000000	Hidden neuron 3—y	−0.006650
X_2—hidden neuron 3	0.00000	X_1—hidden neuron 8	0.400000	Hidden neuron 4—y	0.074476
X_3—hidden neuron 3	1.000000	X_2—hidden neuron 8	0.500000	Hidden neuron 5—y	0.254939
X_1—hidden neuron 4	0.600000	X_3—hidden neuron 8	1.000000	Hidden neuron 6—y	−0.094136
X_2—hidden neuron 4	0.500000	Radial range hidden neuron 1	0.200000	Hidden neuron 7—y	0.006649
X_3—hidden neuron 4	1.000000	Radial range hidden neuron 2	0.200000	Hidden neuron 8—y	−0.045479
X_1—hidden neuron 5	1.000000	Radial range hidden neuron 3	0.200000	Data offset—y	0.541008
X_2—hidden neuron 5	1.000000	Radial range hidden neuron 4	0.200000		

Table 4 summarizes the actual and expected assembly time prediction values, whereas Figure 7 is a graphical interpretation of their dependencies. A set of verification data containing previously unused input and output data was selected for the analysis. The results of the analysis confirm the effectiveness of the prediction performed by the RBF 3-8-1 neural network. The obtained results, both expected and obtained at the network output, are comparable. The operation of the network was tested on the basis of 10 random assembly sequences and the result of assembly time was obtained for each of them. Based on the results presented in Table 4, it can be indicated which of the assembly sequences was characterized by the shortest assembly time, which is therefore the optimal solution.

Table 4. Assembly time values expected and obtained at the network output.

Case No.	Expected Network Value	Network Output Value
1	0.645	0.628
2	0.635	0.598
3	0.573	0.531
4	0.595	0.610
5	0.525	0.508
6	0.656	0.689
7	0.629	0.661
8	0.595	0.619
9	0.620	0.597
10	0.532	0.559

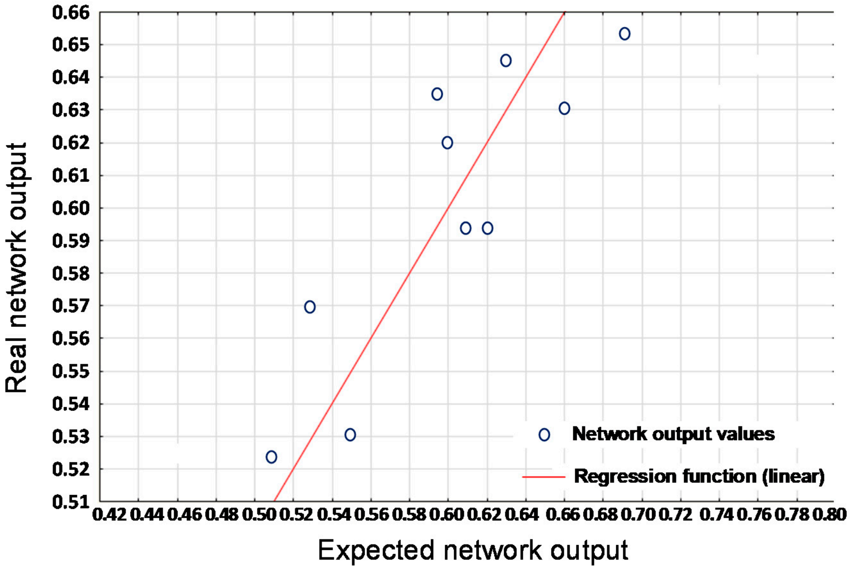


Figure 7. Comparison of the expected and obtained assembly time values at the network output.

4.2. Discussion

Presented method proposes the selection of the best assembly sequence based on the estimated assembly time for the selected product. It works on the basis of selected universal criteria for the evaluation of assembly sequences and their impact on the process time. In principle, its correct operation is based on constant production conditions, which is a prerequisite for its proper operation and correctness of the network learning process. Universal criteria for assessing the assembly sequence proposed in this paper can be effectively automatically retrieved from CAD documentation, although this is not the subject of the presented analysis. The obtained test results confirm that it is possible to develop procedures supporting the determination of the assembly sequence of mechanical products. The neural network model effectively predicts the time of the assembly process. Further research should focus on developing a more universal method and increasing the amount of data to enable network learning.

The effectiveness of the method depends mainly on the number of cases teaching the neural network that are able to generalize the knowledge and the neural network to different products to be assembled. At the moment, the effectiveness of the network in the data verification group is 99%. Entering new data into the network will improve the efficiency of the time sensitive tasks and universally the possibility of applying the procedure to new, not considered cases.

Thus, a network constraint may be a greater number of errors when predicting assembly time for other products. The aim of the authors is to develop the conducted research and verify the operation of the network on a wide range of products. A neural network model was developed to meet the requirements of all mechanical parts. The goal was to develop an overall model. Then, its effectiveness was verified on the basis of one selected product—the door of a forklift truck.

5. Conclusions

The article describes a mechanical assembly time prediction system operating in a neural network, determined by the criteria: the number of tool changes, the number of assembly direction changes, or the stability of the assembly units. The principle of operation and training of the network is its work in a specific mechanical production period; it allows one to determine the most advantageous workplace configuration, production organization, process control, or level of employee training. It is necessary for the best possible network search results.

The obtained results of the analyses confirmed the effectiveness of the previously developed model. The authors assumed that it would also be suitable for other mechanical products, and further studies will be carried out to prove these assumptions. The development of a universal model for selecting the least time-consuming assembly sequence will make it possible to improve many assembly processes. This is of particular importance for products consisting of many parts and in complex manufacturing processes.

The obtained test results confirm that it is possible to develop procedures supporting the determination of the assembly sequence of mechanical products. The model of the neural network, containing universal criteria determining the time of the assembly process, was verified on the example of the assembly of the door of a forklift truck, confirming its effectiveness. Further research should focus on checking the usefulness of the neural network also for other mechanical products. The effectiveness of the method depends mainly on the number of cases teaching the neural network that are able to generalize the knowledge and the neural network to different products to be assembled. Thus, a network constraint may be a greater number of errors when predicting assembly time for other products. The aim of the authors is to develop the conducted research and verify the operation of the network on a wide range of products.

Author Contributions: Conceptualization, M.S.; methodology, K.P. and M.S.; software, K.P. and M.S.; validation, M.S. and K.P.; formal analysis, M.S. and K.P.; investigation, M.S. and K.P.; data curation, M.S.; writing—original draft preparation, M.S. and K.P.; writing—review and editing, K.P. and M.S.; visualization, K.P. and M.S.; supervision, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Ministry of Science and Higher Education of Poland (No. 0614/SBAD/1529).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, G.A.; Bahubalendruni, M.R.; Prasad, V.V.; Ashok, D.; Sankaranarayanasamy, K. A novel Geometric feasibility method to perform assembly sequence planning through oblique orientations. *Eng. Sci. Technol. Int. J.* **2021**. [CrossRef]
2. Su, Y.; Mao, H.; Tang, X. Algorithms for solving assembly sequence planning problems. *Neural Comput. Appl.* **2021**, *33*, 525–534. [CrossRef]
3. Butlewski, M.; Suszyński, M.; Czernecka, W.; Pajzert, A.; Radziejewska, M. Ergonomic criteria in the optimization of as-sembly processes, Cristina Feniser. In *Proceedings of the 6th RMEE2018—Performance Management or Management Performance—Todesco*; Publishing House: Cluj-Napoca, Romania, 2018; pp. 424–431.
4. Sąsiadek, M. Planning and analysis of mechanical assembly sequences in design engineering—Part I: The Method. *Teh. Vjesn. Tech. Gaz.* **2015**, *22*, 337–342. [CrossRef]

5. Wu, W.; Huang, Z.; Zeng, J.; Fan, K. A decision-making method for assembly sequence planning with dynamic resources. *Int. J. Prod. Res.* **2021**, 1–20. [CrossRef]
6. Watanabe, K.; Inadaa, S. Search algorithm of the assembly sequence of products by using past learning results. *Int. J. Prod. Econ.* **2020**, *226*, 107615. [CrossRef]
7. Zhang, H.; Peng, Q.; Zhang, J.; Gu, P. Planning for automatic product assembly using reinforcement learning. *Comput. Ind.* **2021**, *130*, 103471. [CrossRef]
8. Chen, W.C.; Tai, P.H.; Deng, W.J.; Hsieh, L.F. A three-stage integrated approach for assembly sequence planning using neural networks. *Expert Syst. Appl.* **2008**, *34*, 1777–1786. [CrossRef]
9. Sinanoğlu, C.; Börklü, H.R. An assembly sequence-planning system for mechanical parts using neural network. *Assem. Autom.* **2005**, *25*, 38–52. [CrossRef]
10. Zhang, H.; Liu, H.; Li, L. Research on a kind of assembly sequence planning based on immune algorithm and particle swarm optimization algorithm. *Int. J. Adv. Manuf. Technol.* **2013**, *71*, 795–808. [CrossRef]
11. Biswal, B.B.; Pattanayak, S.K.; Mohapatra, R.N.; Parida, P.K.; Jha, P. Generation of optimized robotic assem-bly sequence using immune. In Proceedings of the ASME, International Mechanical Engineering Congress and Exposition, Houston, TX, USA, 9–15 November 2012; pp. 1–9.
12. Guo, J.; Sun, Z.; Tang, H.; Yin, L.; Zhang, Z. Improved Cat Swarm Optimization Algorithm for Assembly Sequence Planning. *Open Autom. Control. Syst. J.* **2015**, *7*, 792–799. [CrossRef]
13. Li, X.; Qin, K.; Zeng, B.; Gao, L.; Su, J. Assembly sequence planning based on an improved harmony search algorithm. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 2367–2380. [CrossRef]
14. Wang, D.; Shao, X.; Liu, S. Assembly sequence planning for reflector panels based on genetic algorithm and ant colony optimization. *Int. J. Adv. Manuf. Technol.* **2017**, *91*, 987–997. [CrossRef]
15. Xin, L.; Shang, J.; Cao, Y. An efficient method of automatic assembly sequence planningfor aerospace industry based on genetic algorithm. *Int. J. Adv. Manuf. Technol.* **2017**, *90*, 1307–1315. [CrossRef]
16. Zeng, C.; Gu, T.; Zhong, Y.; Cai, G. A Multi-Agent Evolutionary algorIthm for Connector-Based Assembly Sequence Planning. *Procedia Eng.* **2011**, *15*, 3689–3693. [CrossRef]
17. Murali, G.B.; Deepak, B.B.V.L.; Bahubalendruni, M.V.A.R.; Biswal, B.B. Optimal Assembly Sequence Planning To-wards Design for Assembly Using Simulated Annealing Technique. *Res. Des. Communities* **2017**, *1*, 397–407.
18. Tadeusiewicz, R.; Chaki, R.; Chaki, N. *Exploring Neural Networks with C#*; CRC Press: Boca Raton, FL, USA, 2015.
19. Suszyński, M.; Żurek, J. Computer aided assembly sequence generation. *Manag. Prod. Eng. Rev.* **2015**, *6*, 83–87. [CrossRef]
20. Suszyński, M.; Żurek, J.; Legutko, S. Modelling of assembly sequences using hypergraph and directed graph. *Teh. Vjesn. Tech. Gaz.* **2014**, *21*, 1229–1233.