electronics

Article

Application of Enterprise Architecture and Artificial Neural Networks to Optimize the Production Process

Zbigniew Juzoń [1], Jarosław Wikarek [2] and Paweł Sitek [2,*]

[1] Doctoral School, Kielce University of Technology, 25-314 Kielce, Poland; zjuzon@tu.kielce.pl
[2] Department of Applied Computer Science, Kielce University of Technology, 25-314 Kielce, Poland; j.wikarek@tu.kielce.pl
* Correspondence: sitek@tu.kielce.pl

Abstract: Production optimization is a complex process because it must take into account various resources of the company and its environment. In this process, it is necessary to consider the enterprise as a whole, taking into account the interaction between its key elements, both in the technological and business layer. For this reason, the article proposes the use of enterprise architecture, which facilitates the interaction of these layers in the production optimization process. As a result, a proprietary meta-model of enterprise architecture was presented, which, based on good practices and the assumptions of enterprise architecture, facilitates the construction of detailed optimization models in the area of planning, scheduling, resource allocation, and routing. The production optimization model formulated as a mathematical programming problem is also presented. The model was built taking into account the meta-model. Due to the computational complexity of the optimization model, a method using an artificial neural network (ANN) was proposed to estimate the potential result based on the structure of the model and a given data instance before the start of optimization. The practical application of the presented approach has been shown based on the example of optimization of the production of an exemplary production cell where the cost of storage and the number of unfulfilled orders and maintenance are optimized.

Keywords: enterprise architecture; production optimization; meta-model; mathematical programming; ANN

check for updates

Citation: Juzoń, Z.; Wikarek, J.; Sitek, P. Application of Enterprise Architecture and Artificial Neural Networks to Optimize the Production Process. *Electronics* **2023**, *12*, 2015. https://doi.org/10.3390/electronics12092015

Academic Editor: Maciej Ławryńczuk

Received: 27 January 2023
Revised: 27 March 2023
Accepted: 24 April 2023
Published: 26 April 2023

1. Introduction

In Industry 4.0, a very important role is played not only by automation and robotization, but also by full computerization and the use of various types of methods and IT techniques such as mathematical and constraint programming, data mining, and artificial intelligence which make effective industry management and optimization of production processes possible [1]. It is necessary to take into account various uncertain business and market factors, such as fluctuations in the value of global economic indicators, changes in the market environment, fluctuations in the value of revenues and costs, etc. Taking these factors into account necessitates the development of new methods and systems to support decision-making. The enterprise must be considered as a whole, taking into account the relationship between its technological and business layers. The change resulting from the assumptions of Industry 4.0 allows modern production systems to identify and group various production resources, define types of individual relationships between products and resources, and manage a set of these relationships. However, one specific challenge is the sharing of architecture knowledge between different organizations, and even within a single enterprise (between its different layers and component units). Architecture models are prepared in different languages and environments such as ArchiMate, UML, and BPMN [2–4], and with different levels of detail. As a result, there are difficulties with consistency when building decision-making and optimization models that take into account enterprise architecture [5] (the larger and more complex the organization, the greater the challenges in this area).

Our motivation for undertaking the research was the observation that the current approaches to optimizing production systems [6–8] usually boil down to building a mathematical model based on data extracted directly from the technological layer without referring to the business layer. These models do not allow feedback from the technology layer to the business layer, e.g., through Service Level Agreement (SLA). What is more, in the absence of a solution to the models described in [6–8] due to, for example, contradictory constraints, the decision maker does not receive detailed information about the reasons for the lack of a solution. What caused this contradiction of constraints and how can it be remedied? In the proposed model, we not only take into account the business layer, but also choose the structure of the model in such a way, e.g., by appropriate selection of decision variables, that it always has a solution.

Detailed models of production optimization are characterized by a large number of decision variables, which are at least partially discrete in nature, and numerous constraints [9,10]. If the business layer is additionally included in the modelling, this number will further increase. All this results in high computational complexity of the modelled problems. Nevertheless, the inclusion of the business layer in the modelling process is an opportunity. It enables an in-depth analysis of the company's functioning and the identification of the elements that have a real impact on the functioning of the production system in order to propose an appropriate detailed model for optimizing the production system. Therefore, it was decided to leverage this opportunity by including the enterprise's corporate architecture in the modelling process. The use of the assumptions of enterprise architecture is proposed to develop an architecture meta-model in order to select key data and information that should be included in the modelling process and to build a detailed model of optimization of the production system. The meta-model itself is defined as an explicit model for constructing domain models [11]. An additional contribution was to propose a method that uses an artificial neural network (ANN) [12,13] to estimate the potential outcome based on the structure of the model and given data instance before starting the optimization. The proposed method helps to answer the key questions that arise for anyone dealing with discrete optimization of complex problems with constraints. Is there an optimal solution for a given model and given data instance? In other words, is it worth starting an expensive optimization process for your data? These questions are not unfounded for problems classified as discrete Constrained Optimization Problems (COPs) [14]. It often happens that, after lengthy calculations, a No Solution Found (NSF) situation arises. This situation may occur repeatedly for subsequent data instances. Therefore, for some data instances of the modelled problem, it is not worth investing resources and time. The rest of the article was organized as follows. The second section presents the assumptions of enterprise architecture and the original meta-model of such architecture. The third section is an illustrative example that has been presented for an easier understanding of the presented approach. The fourth section formulates a research problem and analyses an illustrative example. The next section contains a formalization of a detailed mathematical model of production optimization for the illustrative example. The remaining sections present computational experiments and conclusions.

2. Meta-Model of Enterprise Architecture

The Open Group Architecture Framework (TOGAF) officially introduced the architecture content meta-model [15]. In earlier versions of TOGAF, it was usually prepared as part of the customization of the architecture framework. It took place in the introductory phase of TOGAF cycle and it was the so-called consultancy work, not a formal approach, approved by The Open Group. If one wanted to define the purpose of creating a content meta-model, it could be synthesized as a precise way of defining key concepts that appear in architecture models and the relationships between them. The meta-model covers all four architecture domains (business, data, application, and engineering) [16]. Therefore, it can be precisely indicated that the enterprise will have a described business architecture along with an indication of information that must be included in the models describing this field of architecture and what the relationship between the application architecture and the technical architecture, etc. Contrary to appearances, despite the daunting name (with the so-called "academic" quality), the content meta-model brings the greatest benefit to the client of the enterprise architecture (if they entrust it to a third company). The architecture content meta-model can be considered as a formal specification of requirements for architecture products—on its basis, it will be possible to easily verify the completeness and consistency of the models provided (completeness in the context of the meta-model, not the problem description). The content meta-model takes the form of a class diagram drawn up in UML notation—individual classes denote entities appearing in architecture models, and the associations between them reflect the relationships between these entities. These relationships occur both between entities from the same architecture domain (e.g., within the data architecture: data entity and logical data component) and between different architecture domains (e.g., a business service from the domain of business architecture with an information system service from the domain of architecture application). Additionally, each of the meta-model classes is marked with an appropriate colour, which indicates whether the entities corresponding to a given class must be identified when creating architecture models (then this class is marked in white on the meta-model), or whether they can be identified (then the class is marked in a colour other than white). It is a practical implementation of the so-called extension mechanism introduced in TOGAF. It specifies a certain minimum set of information that must be collected each time to describe an enterprise architecture. Information beyond this set is collected as needed (e.g., information about the location of the organization's headquarters is optional and is only collected when the enterprise architecture is used during IT consolidation). There are several levels of detail in TOGAF specification to describe the architecture content meta-model. Each of them has a different degree of complexity and a different purpose. The first one includes the classes themselves (without relationships) assigned to individual architecture domains. It is characterized by a low level of complexity and is easy to explain to recipients who have no experience with enterprise architecture. The second level of detail is classes with relationships (relationships are named, but their number is not specified). Here, it is necessary to take the time to discover all the nuances and understand the consequences of the occurrence of certain relationships (especially the work they generate when creating and updating models). The third level of detail, which is not explicit, includes classes with relationships and additionally with reference attributes. Such a diagram is not explicitly included in the TOGAF. Based on TOGAF and the approach presented in [11], a proprietary meta-model of enterprise architecture has been proposed (1).

$$M = (R_s,\ P,\ R) \tag{1}$$

where:

- R_s—resources (all material and non-material elements of the production process that are necessary to produce products, e.g., machines, raw materials, employees, tools, etc.);
- P—processes (all phenomena and deliberately undertaken actions which result in the gradual occurrence of the desired changes in the subject of work subject to their influence);
- R—relationships (all connections and interdependent that affect the manufacture or maintenance of products or services).

By using the proposed meta-model of enterprise architecture, it is possible to ensure a common understanding of the concepts by the stakeholders involved in the process of building an enterprise architecture. This meta-model of enterprise architecture defines the semantics and use of concepts that appear in the planning, scheduling, routing, resource allocation, and similar models. In practice, it also facilitates the construction of detailed models of the aforementioned processes. The application of the proposed meta-model for building a detailed model of optimizing a simple production process will be presented in an illustrative example (Section 3).

3. Illustrative Example of Production Process

To show the implementation possibilities of the proposed meta-model (Chapter 2) and the method of evaluating potential solutions using ANN, a simple illustrative example was presented. It covers a part of the production process for an example production cell (Figure 1).

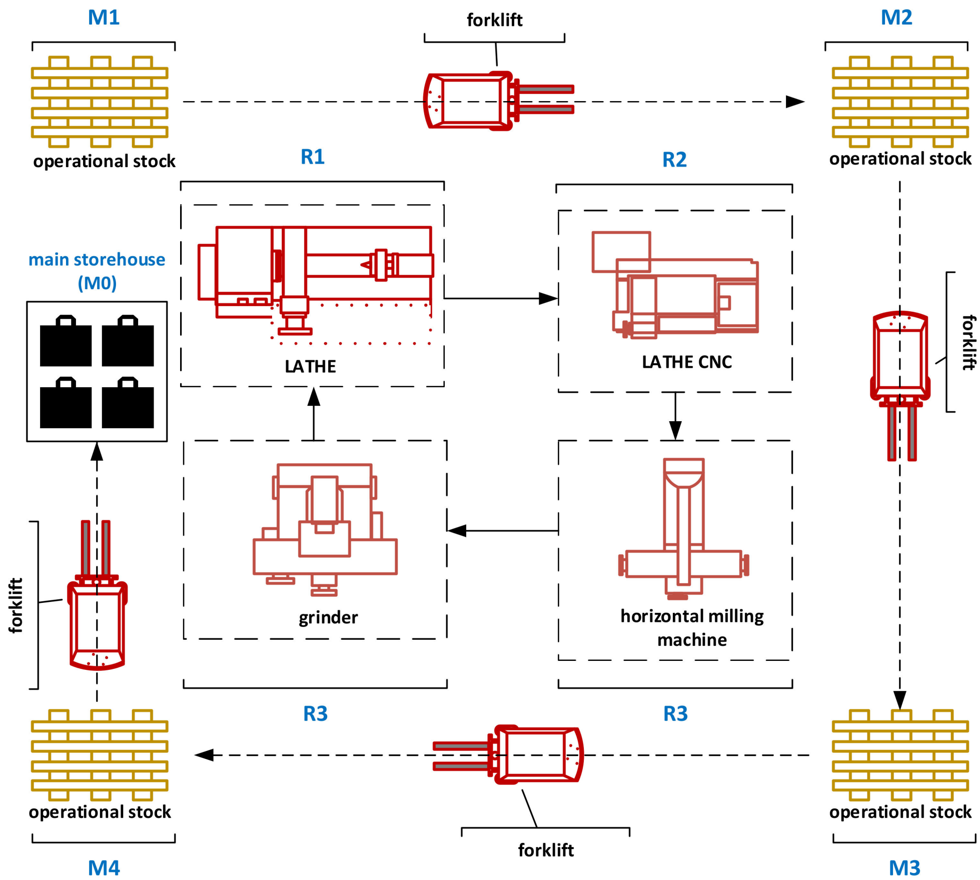

Figure 1. Diagram of the production cell.

The production cell produces eight products (***P*1**..***P*8**) using four types of machines (***R*1**..***R*4**). During the production process, products (***P*1**..***P*8**) are stored in production warehouses marked with symbols (***M*1**..***M*4**) indicating the places where products are stored during the production process. At the level of the production system, the following problems are identified: excessively long order fulfilment time, uneven load of production cells, lack of a maintenance plan, and excessively large inter-operational inventory.

The production volume of each manufactured product was determined (Table 1 and Figure 2). In addition, the production of each product requires a certain amount of time on each machine. The production time is shown in Table 2. A dash means that the production process of the product does not require a machine. The current level of ***R*1**..***R*4** machines utilisation is presented in Table 3. Table 4 presents the availability of warehouses, while the size of ***P*1**..***P*8** products is presented in Table 5.

Table 1. Sales plan (the quantity of the product is given in pieces)—first record from Table A1.

Month	Products							
	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**
Initial stock	1	0	1	1	1	1	0	0
January	45	-	30	-	40	30	20	20
February	-	-	10	-	-	30	10	10
March	-	25	-	-	20	30	20	-
April	-	10	30	20	400	-	20	-
May	20	20	22	-	-	10	10	-
June	10	-	40	-	10	20	-	-

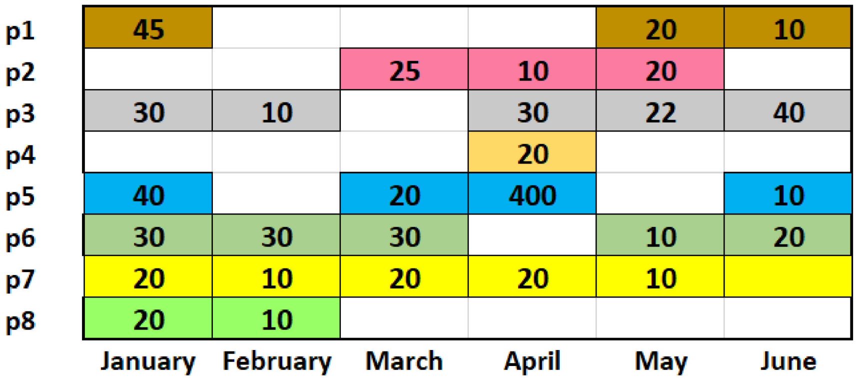

Figure 2. Sales plan (the quantity of the product is given in pieces).

Table 2. Production time (data in the table are given in minutes).

Resources	Products							
	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**
***R*1**	4	5	-	-	-	-	-	5
***R*2**	-	3	2	2	5	-	-	-
***R*3**	-	2	-	-	7	7	-	-
***R*4**	-	-	-	-	-	4	1	5

Table 3. Level of utilisation of R1..R4 per month (historical data.

Production Resource	The Current Level of Use of R1..R4
***R*1**	30.67%
***R*2**	75.57%
***R*3**	8.25%
***R*4**	45.78%

Table 4. Warehouse availability.

Type of Warehouse	Maximum Capacity
*M*0	1000 square metres/height 5 m
*M*1..*M*4	100 square metres

Table 5. Product size.

Products	Volume
*P*1..*P*8	0.05 cubic meter

The illustrative example shows a common production planning and resource allocation problem that is typically modelled and solved using a classical approach such as mathematical programming, dynamic programming, etc. [17] From the business perspective, a key general scientific question can be formulated for the above illustrative example: is an increase in productivity possible (Q1)? Usually, additional specific questions are required to be formulated and answered for question Q1. In order to identify all research questions for the illustrative example, the assumptions of enterprise architecture resulting from TOGAF standard and the proposed meta-model (1) were followed. A review of the business layer was carried out to define business requirements. The latter, combined with information from the production system, allows for determining, among other things, the profitability of production. The algorithm for collecting information as a result of the review of the business layer is illustrated in the form of a diagram (Figure 3).

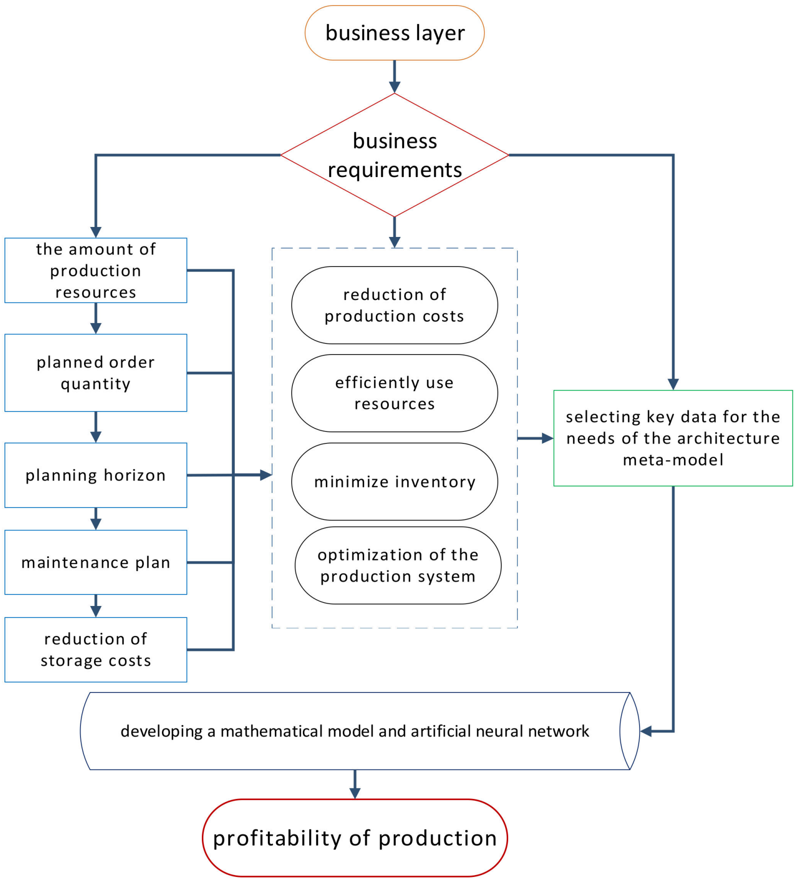

Figure 3. Algorithm for determining additional research questions.

The use of the above algorithm enabled the formulation of detailed research questions Q2..Q4 for the illustrative example.

Q2: How to efficiently use the $\boldsymbol{R_1}..\boldsymbol{R_4}$ resources?
Q3: What is the maintenance plan for $\boldsymbol{R_1}..\boldsymbol{R_4}$?
Q4: How to minimize inventory?

To answer questions Q1..Q4, it is necessary to take into account a key piece of information derived from the meta-model of the enterprise architecture; namely, what relationships exist between the machine and the product. In the illustrative example under consideration, the relationships between the product and the machine are presented in Figure 4. The analysis of the diagram showing the flow of products at the level of the production system provides the basis for considering the possibility of introducing elements of automation in the transport of $\boldsymbol{P1}..\boldsymbol{P8}$ products between $\boldsymbol{R_1}..\boldsymbol{R_4}$ production outlets. By merely introducing a modification in this area, we can have a real impact on reducing the operation time of the production system. An example of a possible modification of the current production system could be as that shown in Figure 5.

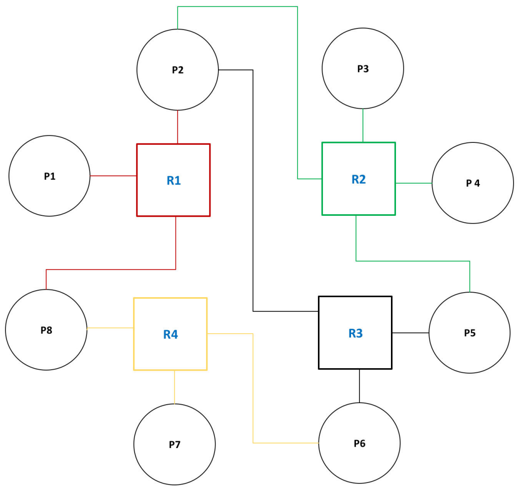

Figure 4. Set of relationships (machine—product).

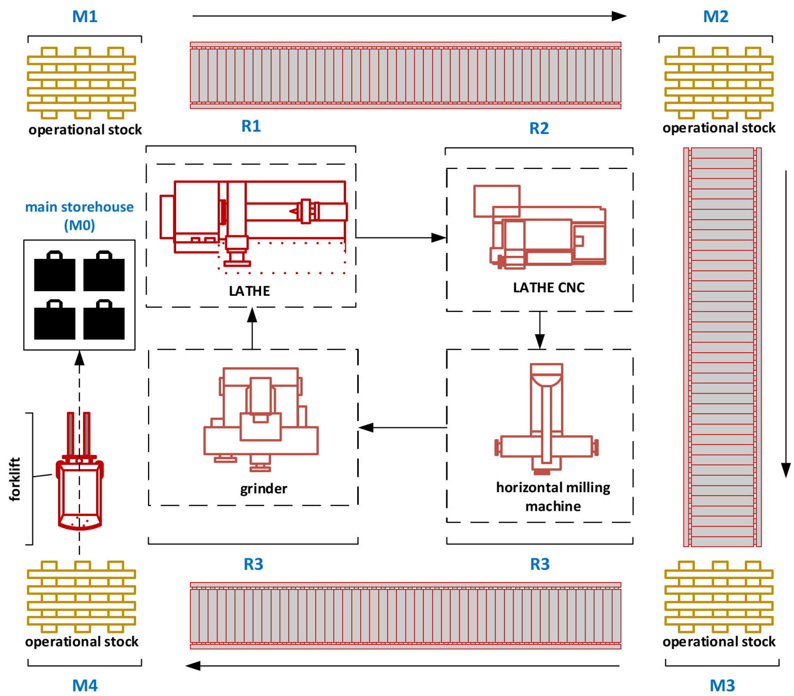

Figure 5. Diagram: possible modification of the flow aimed at reducing the time of moving products ***P*1..*P*8** between machines ***R*1..*R*4**.

Analysing the diagram of the relationship (product–machine) presented in Figure 4, it can be concluded that it will also be necessary to change the routes of movement for products ***P*2, *P*5, *P*6** and ***P*8** between warehouses ***M*0..*M*4** (Figure 5). The modification of product movement routes ***P*2, *P*5, *P*6** and ***P*8** allows us to propose a reduction in the distance that products need to travel between machines ***R*1..*R*4** (Figure 6).

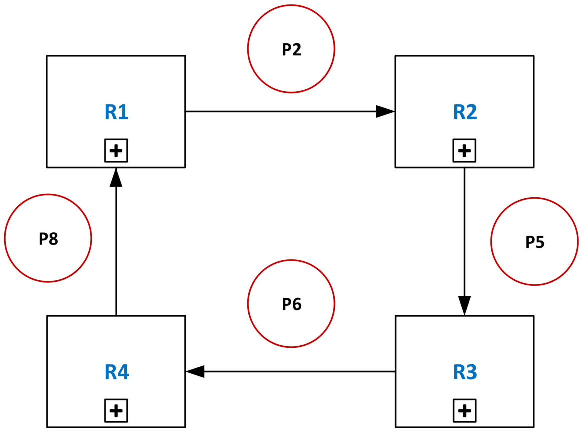

Figure 6. Diagram: possible modification of the flow of ***P*2**, ***P*5**, ***P*6** and ***P*8** products between warehouses ***M*0**..***M*4**.

Based on the analysis conducted, it can be concluded that it is possible to reduce working time by introducing—in the production system—modifications to the flows at the route level within the manufacturing process (Figure 7).

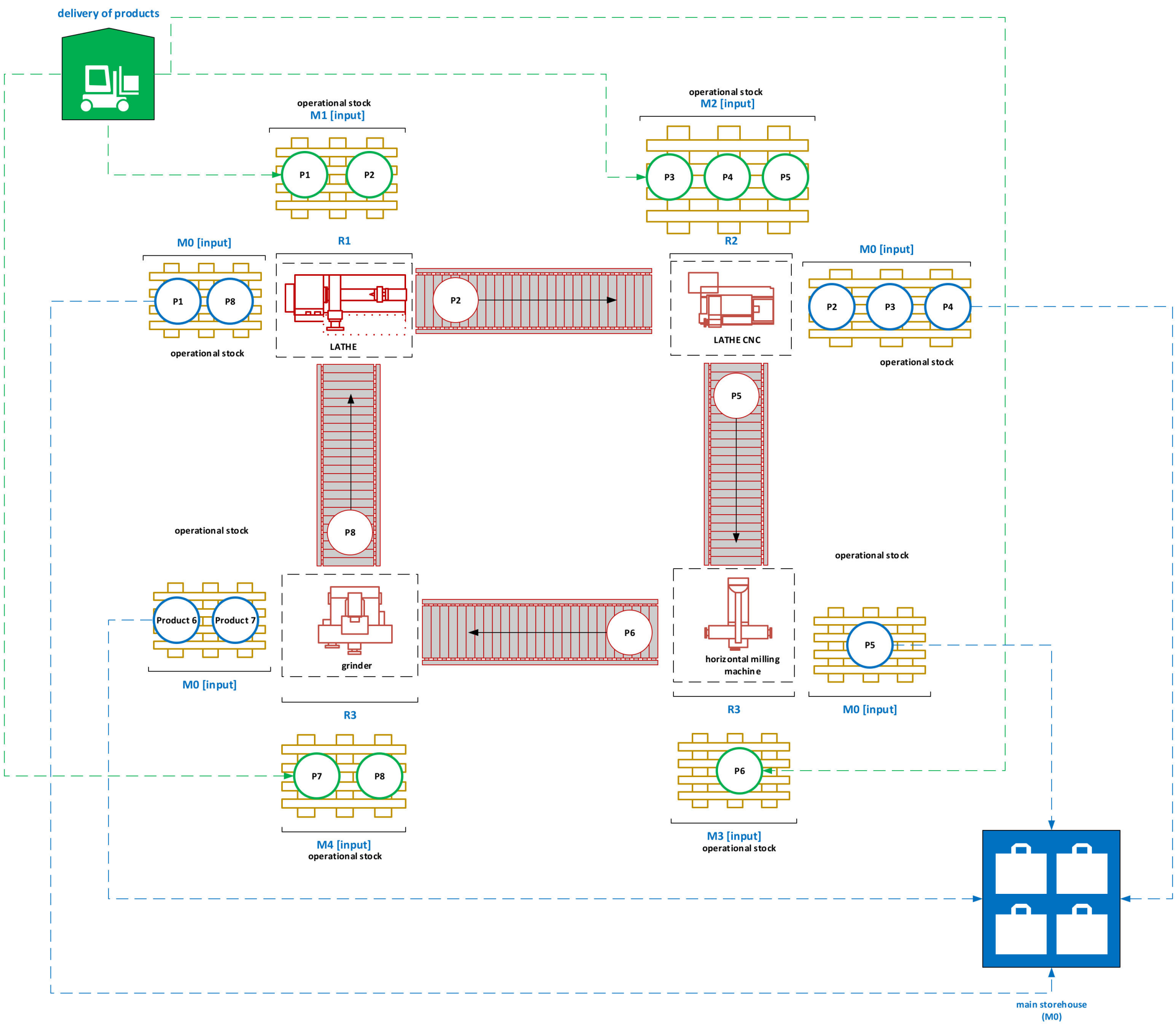

Figure 7. Diagram: recommended modification at route level for production flows.

4. Implementation Framework Using Enterprise Architecture

In order to find answers to the formulated research questions Q1..Q4, an implementation framework was proposed. In the proposed solution, the process is as follows Step 1: all the above-mentioned components of the architecture meta-model (Figure 8), identification of production resource types, identification of relationship types, and identification of types of production processes for the needs can then be considered as an argument for investment programming. Step 2: the process of building an architecture model taking into account the approach, the service, and the layer is not an automatic process and during its construction; the first thing to do is to build a set of good practices for modelling enterprise architecture that has proven successful in various projects. The process starts with creating an architecture and building a model of the motivational layer that leads to the construction of enterprise services (Figure 8) in order to get the answer to questions Q2 and Q4. Step 3: to answer questions Q1, Q2 and Q4, a detailed planning and resource allocation model was formulated for an exemplary production process. LINGO mathematical programming environment was used to solve this problem. However, ANN was used to evaluate potential solutions obtained using the proposed model for specific data instances. Step 4: it boils down to building an enterprise data bus. The enterprise data bus (Figure 8) means a data repository, a place where data obtained using architecture and results obtained by solving decision and optimization problems are stored. Information gathered at the enterprise data bus level can be reused at the organizational level, at the design stage of a future SLA understood as an agreement to maintain and systematically improve the agreed service level between the service provider and the recipient of the service quality level through a constant production cycle. SLAs are created to document the obligations towards customers that need to be fulfilled.

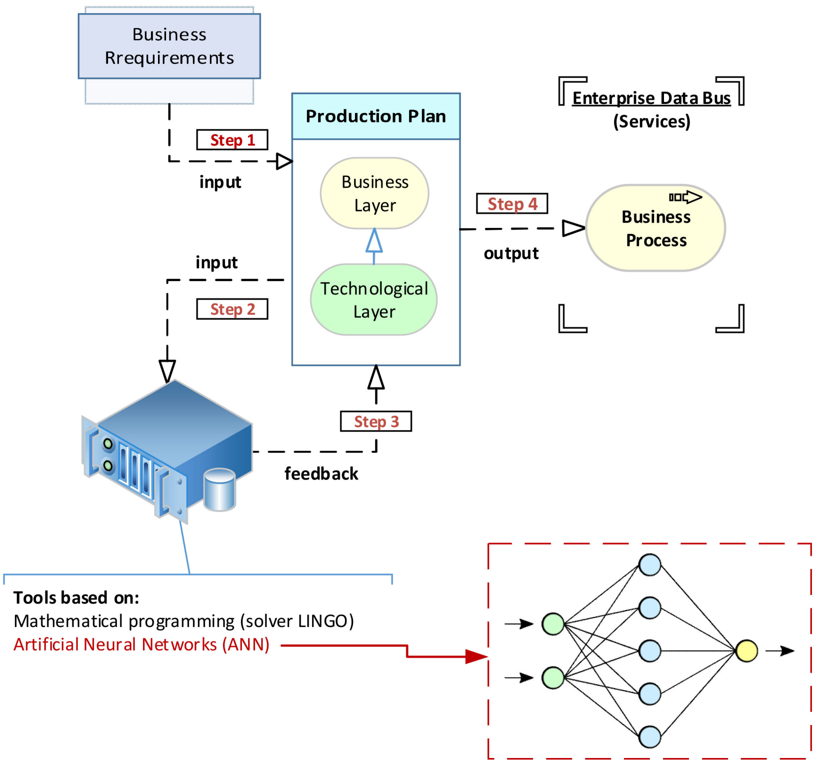

Figure 8. The elements of implementation framework using enterprise architecture.

5. Formalizing a Mathematical Model of Production Planning, Maintenance and Resource Allocation

The mathematical model of production planning, maintenance and resource allocation was formulated as a mixed-integer linear programming (MILP) problem. The meta-model

(1) was used in the process of building the mathematical model (2)..(12). Table 6 presents the elements of the mathematical model (parameters, constraints, decision variables) and the mapping of these elements to the components of the meta-model (Rs, P, R). In the proposed model, we optimize the cost of storage and try to perform all orders and planned maintenances. Therefore, the objective function was formulated as the sum of storage costs and penalties for non-execution of orders and failure to fulfil the orders and to provide maintenances. This function is minimized (11). The model has many constraints related to order fulfilment, production capacity, warehouse capacity, etc.

Table 6. Defining elements of the mathematical model.

Meta-Model Elements	Symbol	Model Elements	Description
R_s	P	Sets & indexes	product $p \in P$
	R		machine $r \in R$
	T		time period $t \in T$, t_o—initial period, t_k—end period
P	$X_{p,t}$	Decision variables	production volume of the product p in the period t.
	$Z_{p,t}$		what order quantity for the product p in period t we are not able to fulfill
	$W_{r,t}$		if the machine is not to be maintenance in period t $W_{r,t} = 1$ else $W_{r,t} = 0$.
	B_r		if the overhaul of the machine r cannot be performed, then $B_r = 1$ else $B_r = 0$
R	$A_{p,t}$	Determined values	stock of product p at the end of period t
	Cost		product storage cost
	K_{r1} K_{r2}		penalties for non-fulfillment of orders and maintenances
	C_1		how many scheduled maintenances have not been carried out
	C_2		how many products in total have not been comp. for all orders
	$g_{p,t}$	Parameters	fulfilment of customer orders
	ep		initial stock of the product p
	np		product storage volume
	hp,r		factor $h_{p,r}$ determines how much time the product p must be processed on the machine r. If hp,r1 $\neq 0$ i hp,r2 $\neq 0$ means that the product is made p must be processed on a machine r1 i r2. Value hp,r = 0 means product p does not need to be processed on a machine r.
	$o_{r,t}$		Any machine r in the period t has a specific production capacity (parameter value $o_{r,t}$)
	m_r		Factor m_r = 1 means that the maintenance of machines should be planned r where $m_r = 0$ means that such maintenance is not planned.
	m_r		planned machine maintenance in the planning period
	kp		kp—the maximum number of product p that may be in stock
	v_m		The total volume of the warehouse v_m
	fp		product unit p in the warehouse for a time unit is associated with the incurring cost

An important feature of the proposed model is the fact that it will always have a solution. That is, there will never be an NSF. This is due to the introduction of an additional decision variable [Z_(p,t)] and constraint (9).

Constraints

Fulfillment of customer orders

$$\begin{gathered} \mathrm{A}_{\mathrm{p},t-1} + \mathrm{X}_{\mathrm{p,t}} + \mathrm{Z}_{\mathrm{p,t}} = \mathrm{g}_{\mathrm{p,t}} + \mathrm{A}_{\mathrm{p,t}}\ \forall \mathrm{p} \in \mathrm{P}, \mathrm{t} \in \mathrm{T} - \{\mathrm{t_o}\} \\ e_p + X_{p,t} + Z_{p,t} = g_{p,t} + A_{p,t}\ \forall p \in P, t = t_o \end{gathered} \tag{2}$$

The load of the machines only within the permitted limits

$$\sum_{\mathrm{p} \in \mathrm{P}} \left(\mathrm{h_{p,r}} \cdot \mathrm{X_{p,t}}\right) = \mathrm{o_{r,t}} \cdot \mathrm{W_{r,t}}\ \forall \mathrm{r} \in \mathrm{R}, \mathrm{t} \in \mathrm{T} \tag{3}$$

Execution of scheduled maintenance

$$\sum_{t \epsilon T} (1 - \mathrm{W}_{\mathrm{r,t}}) = \mathrm{m_r} - \mathrm{B_r}\ \forall \mathrm{r} \in \mathrm{R} \tag{4}$$

Only the permitted number of maintenance during the period

$$\sum_{\mathrm{r} \in \mathrm{R}} (1 - \mathrm{W}_{\mathrm{r,t}}) = \mathrm{m\_r}\ \forall \mathrm{t} \in \mathrm{T} \tag{5}$$

The quantity of the product in the warehouse does not exceed the allowed value

$$\mathrm{A_{p,t}} \leq \mathrm{k_p}\ \forall \mathrm{p} \in \mathrm{P}, \mathrm{t} \in \mathrm{T} \tag{6}$$

Total warehouse storage capacity not exceeded

$$\sum_{\mathrm{p} \in \mathrm{P}} \left(\mathrm{A_{p,t}} \cdot \mathrm{n_p}\right) \leq \mathrm{v\_m}\ \forall \mathrm{t} \in \mathrm{T} \tag{7}$$

Calculation of warehouse costs

$$\mathrm{Cost} = \sum_{\mathrm{p} \in \mathrm{P}} \sum_{\mathrm{t} \in \mathrm{T}} \left(\mathrm{A_{p,t}} \cdot \mathrm{f_p}\right) \tag{8}$$

Value determination C_1

$$\mathrm{C_1} = \sum_{\mathrm{r} \in \mathrm{R}} \mathrm{B_r} \tag{9}$$

How many total products were not made for all orders

$$\mathrm{C_2} = \sum_{\mathrm{p} \in \mathrm{P}} \sum_{\mathrm{t} \in \mathrm{T}} \mathrm{Z_{p,t}} \tag{10}$$

Objective Function

$$Min(Cost + \mathrm{K}_{r1} * \mathrm{C_1} + \mathrm{K}_{r2} * \mathrm{C_2}) \tag{11}$$

Binary and integerity

$$\begin{gathered} \mathrm{X_{p,t}} \in \mathrm{N}\ \forall \mathrm{p} \in \mathrm{P}, \mathrm{t} \in \mathrm{T} \\ \mathrm{Z_{p,t}} \in \mathrm{N}\ \forall \mathrm{p} \in \mathrm{P}, \mathrm{t} \in \mathrm{T} \\ \mathrm{A_{p,t}} \in \mathrm{N}\ \forall \mathrm{p} \in \mathrm{P}, \mathrm{t} \in \mathrm{T} \\ \mathrm{W_{r,t}} \in \{0,1\}\ \forall \mathrm{r} \in \mathrm{R}, \mathrm{t} \in \mathrm{T} \\ \mathrm{B_r} \in \{0,1\}\ \forall \mathrm{r} \in \mathrm{R} \end{gathered} \tag{12}$$

6. Computational Experiments

In order to select the optimal production system for the illustrative example, a numerical (manual) experiment was carried out to determine, using the mathematical model, the values of the production plan $[X_{p,t}]$, what can not be produced from the sales plan $[Z_{p,t}]$, machine maintenance plan $[W_{r,t}]$, whether machine maintenance is possible $[B_r]$, Storage Costs, and Production Costs (Table 6). Table A1 presents a set of data that was used at the stage of this experiment (record no. 1). This experiment took about 30 min because solving this problem manually, even for small data instances, is time-consuming and error-prone. The obtained solution is presented in Table 7. Then, computational experiments were carried out using the proposed framework. Numerous computational experiments were carried out to test the proposed framework and model. The plan of computational experiments is presented in Figure 9.

Table 7. Results of the manual experiment.

$X_{p,t}$							$Z_{p,t}$							$W_{r,t}$							B_r	Costs Storage
t	1	2	3	4	5	6	t	1	2	3	4	5	6									
p1	0	0	0	0	0	3	p1	40	4	0	0	20	7									
p2	5	0	0	14	7	0	p2	0	0	20	0	9	0	t	1	2	3	4	5	6		
p3	5	11	9	6	0	0	p3	24	0	3	17	22	40	r1	0	1	1	1	1	1	r1 0	2000
p4	0	5	6	13	7	0	p4	0	0	0	0	0	0	r2	1	1	1	1	0	1	r2 0	
p5	39	0	20	42	0	10	p5	0	0	0	358	0	0	r3	1	1	1	1	1	1	r3 1	
p6	0	6	4	0	0	0	p6	29	24	26	0	10	20	r4	1	1	1	1	1	0	r4 0	
p7	20	13	17	20	11	0	p7	0	0	0	0	9	0									
p8	0	0	0	0	0	0	p8	22	8	0	0	0	0									

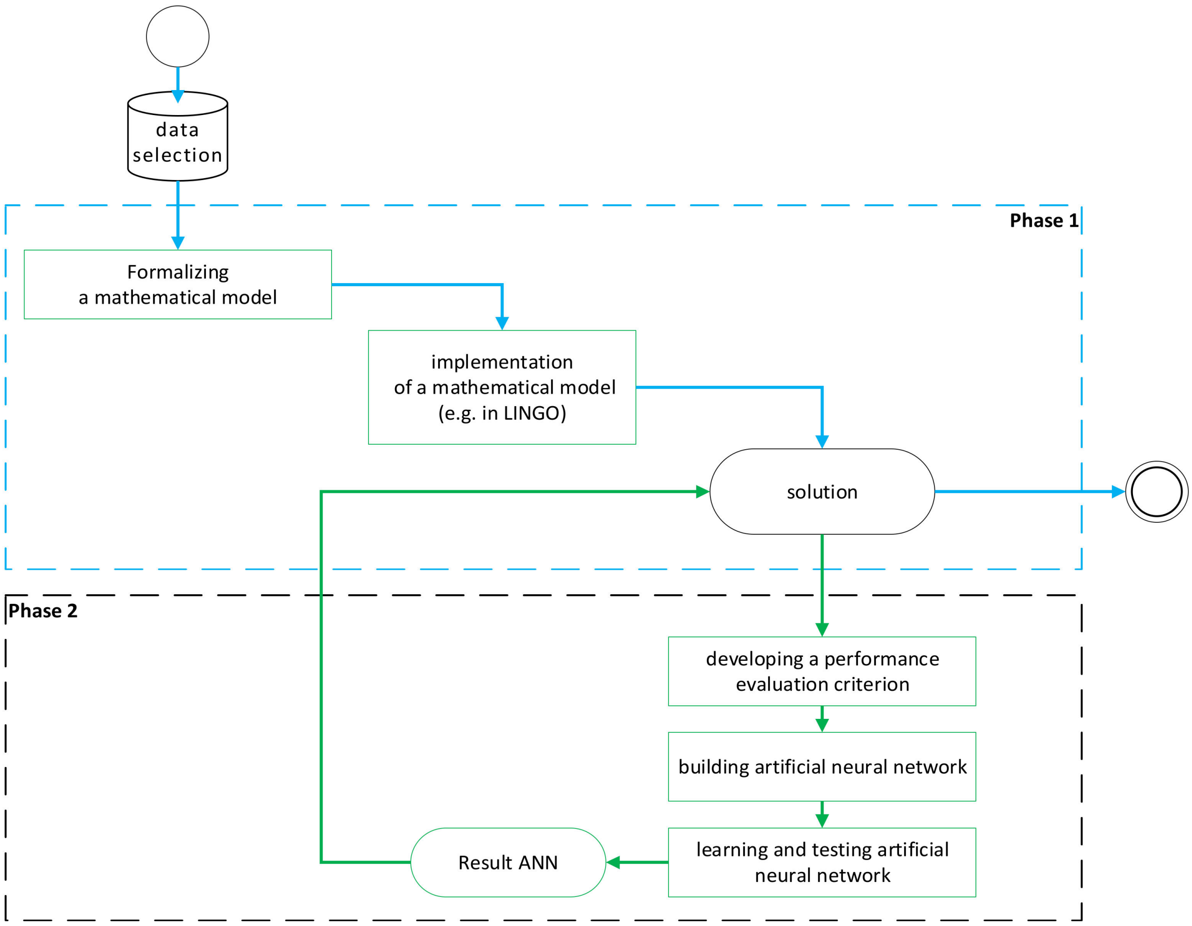

Figure 9. Plan of computational experiments.

The algorithm (Figure 9) shows the structure of experiments, which has been divided into two phases (phase 1 contains the sequence of activities that should usually be performed in the process of optimizing the production system, while the activities included in phase 2 take into account a new approach to data analysis and acquisition using the meta-model of architecture and the use of ANN to estimate input data for which it is worth making the effort to carry out complex and costly calculations).

As part of "Phase 1", Experiment_1 was conducted for 50 data instances (Table A1) using the framework presented in Chapter 5. LINGO ver.12.0 software was used as a solver. A computer with the following parameters was used for the experiments: Processor: Intel(R) Core (TM) i7 1-700K CPU @ 3.80 GHz 3.79 GHz; RAM: 16 GB; Windows 11; 64 processor. The obtained results are presented in Table A3. A detailed comparative analysis was carried out for one instance of input data (Table 1, which is equal to 1 record in Table A1) and the results obtained (Table 7 and 1 record in Table A3). In the first case, manual calculations were made; in the second, optimization using LINGO was used. Firstly, the solution time was reduced from 30 min (manual solution) to 2 s (LINGO). Secondly, the cost of storage decreased from 2000 (manual solution) to 431 (LINGO). The obtained sales/production plans and unfulfilled orders were also presented in the form of Gantt diagrams in Figure 10, Figure 11 (manual solution), Figure 12, and Figure 13 (LINGO), respectively. All the obtained results indicate that the use of the proposed model and the LINGO solver, compared to the manual one, gives better results in every aspect. For several Experiment_1 data instances, the size of the data instances was increased, i.e., from (T = 6, P = 8, R = 4) to (T = 6, P = 12, R = 6) and (T = 6, P = 20, R = 8) which resulted in the extension of the calculation time to over 600 s in the first case. In the second case, it took as long as 3600 s to obtain a feasible solution. This is due to the nature of the problems modelled as MILP.

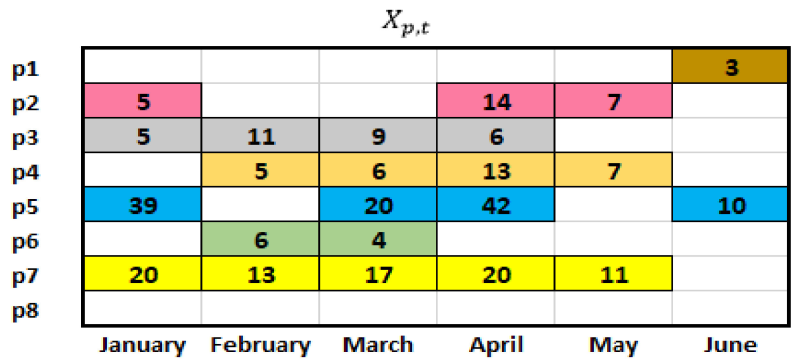

Figure 10. Recommended production plan for products p1..p8 in particular periods t (values of *Xp,t* variables). The experiment was done manually.

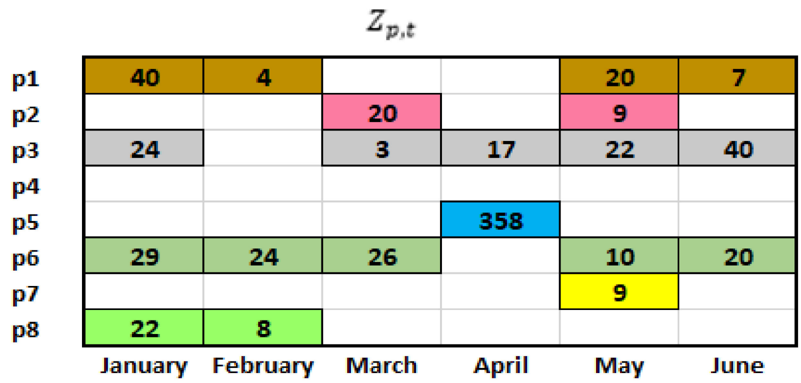

Figure 11. List of unfulfilled orders for products p1..p8 in individual periods t (values of *Zp,t* variables). The experiment was done manually.

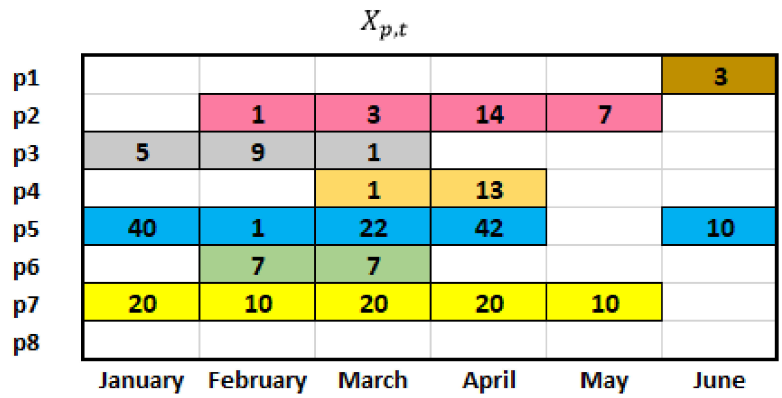

Figure 12. List of products p1..p8 in individual periods t (values of variables *Xp,t*). The experiment was performed using the LINGO solver.

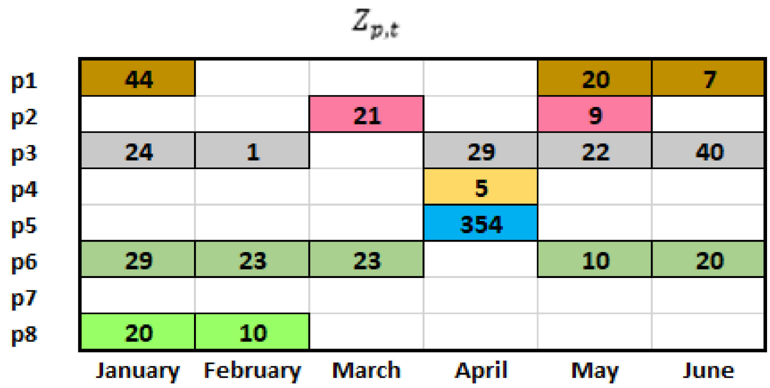

Figure 13. List of unfulfilled orders for products p1..p8 in individual periods t (values of *Zp,t* variables). The experiment was performed using the LINGO solver.

A detailed analysis of the optimization process for individual data instances (Table A1) and the results obtained (Table A3) for Experiment_1 showed that, depending on the data instance (values of individual parameters, sizes, etc.), it was not always possible to obtain a satisfactory result within an acceptable timeframe. This resulted in the implementation of "Phase 2" of conducted experiments (Figure 9). The main idea of the proposed approach was to use ANN to evaluate a given data instance in the context of obtaining a satisfactory solution for it using the proposed model and MP solver. More precisely, it was about obtaining, before starting the optimization, often very time-consuming and cost-intensive, an answer as to whether it is advisable to start optimization for this data instance. According to the "Phase 2" algorithm, an evaluation criterion and a training pattern for ANN were developed, and then ANN was built. All these steps are described in the following sections.

6.1. Evaluation Criterion

At the stage of reviewing the business layer, the "Evaluation Criteria" should be defined. In the illustrative example considered, in order to determine the profitability of production costs, it is necessary to determine the share of storage costs in the entire production process. Next, weights for the selected evaluation criterion are set (Table 8), followed by an evaluation of the results obtained from solving the mathematical model.

Table 8. Evaluation criterion.

Share of Storage Costs in the Production Process	Classification Label
storage costs <50%	1
storage costs >50% and storage costs = 0	0

The next step will be to strengthen the decision-making process by using artificial intelligence and building ANN to assess for which data sets the mathematical model presented (Table 6) should be solved.

6.2. Developing a Training Pattern for ANN

Before the process of building ANN, the structure of the teaching pattern for the neural network should be built. When learning with a teacher, the network gives examples of correct operations that should be followed in its current activity (during the exam). The example should be understood as meaning that the teacher gives certain inputs and outputs, showing what the required network response is for a specific input configuration. We are dealing with a pair of values—an exemplary input signal and the desired (expected) output, i.e., the required network response to this input signal. As an illustrative example (Figure 14), the scheme of the learning pattern along with the presentation of sample data is presented. The collection of examples collected for use in network learning is usually called the training string. Thus, in a typical learning process, the network receives the learning sequence from the teacher and learns from it the correct operation using one of the many learning strategies known today. The schema of the learning pattern with sample data for the considered example is presented in Figure 14.

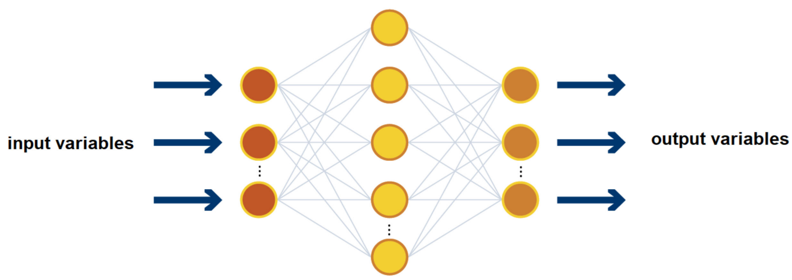

Figure 14. ANN model built in Python system using the KERAS library.

To build a training pattern for ANN, it is necessary to classify the training data set (Table A5) according to the evaluation criterion (Table 8). After completing the classification, the target training set (Table A6) is built for ANN, which will be used during (Experiment_2). Thus, in a typical learning process, the network receives the learning sequence from the teacher and learns correctly from it using one of the many learning strategies known today. The training pattern (Table 9) for ANN that will be used for the illustrative example consists of two elements: input variables and output variables.

Table 9. Training pattern structure (input variables).

Input Variables/Output Variables
T—number of products
P—planning horizon
R—number of machines
k_{r_1}—penalty for failure to perform maintenances
k_{r_2}—penalty for failure to fulfil orders
v_m—total storage capacity
n_p—product storage volume
e_p—initial stock of the product
f_p—product storage cost p per time unit
g_p—**product storage cost p per time unit**
m_r—is there any planned overhaul of the machine?
$g_{p,t}$—sales of P product in the period T
$o_{r,t}$—machine production capacity R in the period T
$h_{p,r}$—how much time does it take to process the product P on the machine R?
Rating Costs Storage—assessment of storage costs according to the criterion

6.3. Building an ANN

In order to strengthen the decision-making process and reduce its time, the process of building an ANN together with learning and testing the ANN will be carried out for the illustrative example. The implementation was made in the Python system using the KERAS library. Keras is a powerful and easy-to-use, free open-source Python library for developing and evaluating deep learning models. Using the Keras library in Python, a neural network model is defined by trial and error (Figure 14). The selection of ANN weights in the process of teaching ANN is carried out during the training of ANN.

We teach the purpose of the neural network model until we obtain output results closest to those expected as the response from the neural network. In the initial period of network training, it must be always taken into account that the response from the neural network will always be burdened with error, and therefore it is necessary to modify the network model configuration by changing the weights. The neural network in Python is tested using the so-called learning pattern (Table A6) supervised teaching. The testing process can be assessed on the training data set (Table A6) by checking whether the learned neural network model at the output responds with the data expected from the network response resulting from the training data sets (Table A7). After the learning process and testing of ANN, the data set should be assessed (Table A2). The learned ANN evaluates the data (Table A2) that has not been used in the learning and testing process. The result of the evaluation is shown in Table A8. During the evaluation of new data (Table A2), the learned ANN from 10 records subjected to the evaluation procedure selected eight records by assigning them to the "1" classification and adding the value "1 (expected 1)" in the "Classification label" column, for which it is reasonable to carry out complex calculations in LINGO. On the other hand, for the two records contained in Table 2, the evaluation using the learned ANN meant assigning them to the "0" classification and adding the value "0 (expected 0)" in the "Classification label" column. Assigning the value "0" means that performing complex calculations in LINGO will not be possible. Therefore, after the process of evaluating the data contained in Table A2, another experiment was carried out in LINGO, as a result of which a solution was obtained for eight records (evaluated positively by ANN), while for two records that were assigned to the "0" LINGO category during the evaluation, LINGO presented the result "Infeasible" in each case. The above experiment showed the usefulness of the proposed approach, which indicates for which data instances (our method labels them 1) it is advisable to perform optimization using the MP solver. The values of all data instances used during both experiments, as well as the training data, and the results obtained, i.e., full contents of Tables A1–A8, are provided in [18].

7. Conclusions

The paper proposes an implementation framework for modelling and solving decision-making and optimization problems in the area of production. On the one hand, the framework uses an enterprise architecture (compliant with TOGAF assumptions) and on the other, mathematical programming methods supported by ANN. An important element of the proposed approach is the proposed meta-model of enterprise architecture (1), the use of which facilitates the construction of detailed decision-making and optimization models. The production and maintenance optimization model (Section 4) was built on the basis of the enterprise architecture meta-model. The proposed optimization model (Chapter 5) enables the optimization of storage costs while maximizing completed orders and performing scheduled maintenances. It also ensures better machine utilization. This model is only one of the few that can be built based on the proposed meta-model. These can be models related to distribution, routing, scheduling and distribution of loads, etc. The modelled problem, like most problems of this type known in the literature, is classified as a discrete-constrained optimization problem [9–11]. Such problems are characterized by high computational complexity, which results in the involvement of considerable resources and time to solve them. An interesting idea in this context is the idea of evaluating, whether for a given instance of data and for a given model, there is a chance to find a satisfactory solution before starting the calculations. For this purpose, the authors proposed the use of ANN (Section 6) for the classification of input data, bearing in mind that the accuracy of classification in ANN depends to a large extent on its architecture [18,19]. In the proposed framework, ANN is responsible for obtaining an evaluation of a potential solution to the problem before running the solver [20]. This is a great way to save time and resources necessary in the calculation process because calculations are run only on selected data instances. In future works, it is planned to use not only mathematical programming methods to solve the problem, but also methods of constraint programming, hybrid methods [21,22] and selected heuristics such as genetic algorithms [22], ant algorithms, etc. It is also planned to use the proposed framework to model and solve problems in the area of logistics, UAV deliveries [23,24] supply chains, and multimodal processes [25,26]. Maintaining the quality of a firm's products at the highest level is very important for keeping an edge over the competition [27,28]. The conclusion is as follows: before making a business decision, it is worth carrying out in advance a simulation of whether it will be possible to perform the order with technological constraints, minimizing the risk of contractual penalties as a result of non-performance of the order.

Author Contributions: Conceptualization, Z.J.; methodology, Z.J. and J.W.; software, J.W.; validation, P.S. and J.W.; formal analysis, J.W.; investigation, Z.J. and J.W.; resources, Z.J.; data curation, J.W.; writing—original draft preparation, P.S.; writing—review and editing, Z.J. and J.W.; visualization Z.J.; supervision P.S.; project administration, P.S.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Appendix A Data for Experiment_1 and Experiment_2

Table A1. Data for *Experiment_1*.

Record Number	T	P	R	k_{r_1}	k_{r_2}	v_m	m_r	k_p	f_p	e_p	n_p	m_r	g_{pt}	$o_{r,t}$	$h_{p,r}$
1	*6*	*8*	*4*	*466,426*	*50*	*45*	*1*	*22 5 55 95 11 72 60 52*	*6 6 0 6 8 9 2 2*	*1 0 1 1 1 1 0 0*	*13 39 31 39 7 24 35 9*	*1 1 0 1*	*45 0 0 0 20 10 0 0 25 10 20 0 30 10 0 30 22 40 0 0 0 20 0 0 40 0 20 400 0 10 30 30 30 0 10 20 20 10 20 20 10 0 20 10 0 0 0 0*	*15 80 59 67 38 14 95 78 96 85 1 91 18 45 76 61 81 0 91 23 18 70 36 10*	*4 5 0 0 0 0 0 5 0 3 2 2 5 0 0 0 0 2 0 0 7 7 0 0 0 0 0 0 0 4 1 5*
...															

Table A1. *Cont.*

Record Number	T	P	R	k_{r_1}	k_{r_2}	v_m	m_r	k_p	f_p	e_p	n_p	m_r	g_{pt}	$o_{r,t}$	$h_{p,r}$
50	6	8	4	724,775	45	552	1	62 61 85 64 40 12 33 83	4 7 4 4 1 8 2 8	1 0 1 0 0 1 1 0	46 0 41 33 39 21 49 28	1 0 0 1	1 3 9 9 8 4 5 5 2 5 8 9 5 1 3 3 2 3 0 7 3 0 3 4 5 1 9 4 5 9 2 5 4 9 3 7 2 4 4 3 8 8 3 1 1 0 6 5	86 62 72 85 15 60 12 26 9 19 95 94 30 77 53 59 99 23 24 78 81 38 65 58	7 6 0 5 6 2 0 3 1 5 9 9 7 9 7 3 5 8 8 0 5 3 2 7 8 2 0 5 5 0 5 9

Table A2. Data for *Experiment_2*.

Record Number	T	P	R	k_{r_1}	k_{r_2}	v_m	m_r	k_p	f_p	e_p	n_p	m_r	g_{pt}	$o_{r,t}$	$h_{p,r}$
1	6	8	4	368,888,888,000	1,800,000,000	2227	2	46 46 61 74 8 73 43 42	70 30 40 60 10 90 10 80	0 1 0 0 1 1 1 0	4 17 33 39 33 4 3 45	1 1 1 1	9 0 5 0 3 1 3 6 0 4 7 3 1 4 8 5 7 3 7 1 0 5 8 5 2 1 7 6 8 9 9 7 3 4 4 5 3 7 4 7 3 1 4 8 7 4 4 8	93 85 40 28 4 18 73 79 14 10 58 59 59 18 9 23 84 36 39 25 17 10 83 46	6 2 1 7 4 8 3 5 0 3 0 2 3 0 2 8 3 2 3 0 2 7 8 6 9 1 9 3 0 7 8 7
…															

Table A2. *Cont.*

Record Number	T	P	R	k_{r_1}	k_{r_2}	v_m	m_r	k_p	f_p	e_p	n_p	m_r	g_{pt}	$o_{r,t}$	$h_{p,r}$
10	*6*	*8*	*4*	*466,426*	*1,800,000,000*	*2227*	*2*	*46 46 6 74 80 730 400 420*	*70 300 400 600 10 900 100 800*	*0 1 0 0 1 1 1 0*	*4 17 33 39 33 4 3 45*	*1 1 1 1*	*9 0 5 0 3 1 3 6 0 4 7 3 1 4 8 5 7 3 7 1 0 5 8 5 2 1 7 6 8 9 9 7 3 4 4 5 3 7 4 7 3 1 4 8 7 4 4 8*	*93 85 40 28 4 18 73 79 14 10 58 59 59 18 9 23 84 36 39 25 17 10 83 46*	*6 2 1 7 4 8 3 5 0 3 0 2 3 0 2 8 3 2 3 0 2 7 8 6 9 1 9 3 0 7 8 7*

Table A3. Recommended production plan (results of the Experiment_1).

Record Number	$X_{p,t}$	$Z_{p,t}$	$W_{r,t}$	B_r	Costs Storage
1	*t = 1 2 3 4 5 6 p1 0 0 0 0 0 3 p2 0 1 3 14 7 0 p3 5 9 1 0 0 0 p4 0 0 1 13 0 0 p5 40 1 22 42 0 10 p6 0 7 7 0 0 0 p7 20 10 20 20 10 0 p8 0 0 0 0 0 0*	*t = 1 2 3 4 5 6 p1 44 0 0 0 20 7 p2 0 0 21 0 9 0 p3 24 1 0 29 22 40 p4 0 0 0 5 0 0 p5 0 0 0 354 0 0 p6 29 23 23 0 10 20 p7 0 0 0 0 0 0 p8 20 10 0 0 0 0*	*t = 1 2 3 4 5 6 r1 0 1 1 1 1 1 r2 1 1 1 1 0 1 r3 1 1 1 1 1 1 r4 1 1 1 1 1 0*	*r1 0 r2 0 r3 0 r4 0*	*431*
		…			
50	*t = 1 2 3 4 5 6 p1 2 0 2 0 0 0 p2 2 3 0 0 0 0 p3 1 4 1 0 7 3 p4 0 0 0 1 0 3 p5 1 1 2 5 0 3 p6 0 0 0 0 1 0 p7 4 0 0 0 0 0 p8 0 0 0 0 4 2*	*t = 1 2 3 4 5 6 p1 7 0 3 0 3 1 p2 0 3 0 4 7 3 p3 0 0 7 5 0 0 p4 7 1 0 4 8 2 p5 0 0 5 1 8 6 p6 8 7 3 4 3 5 p7 0 5 4 7 3 1 p8 4 8 7 4 0 6*	*t = 1 2 3 4 5 6 r1 1 1 1 1 1 1 r2 1 1 1 1 1 1 r3 1 1 1 1 1 1 r4 1 1 1 1 1 1*	*r1 0 r2 0 r3 0 r4 0*	*6*

Table A4. Recommended production plan (results of the Experiment_2).

Record Number	$X_{p,t}$	$Z_{p,t}$	$W_{r,t}$	B_r	*Costs Storage*
1	*t* = *1 2 3 4 5 6* *p1 0 0 0 0 0 0* *p2 4 0 0 0 0 0* *p3 6 12 0 0 7 3* *p4 0 0 2 0 1 0* *p5 9 0 0 5 0 6* *p6 0 0 0 0 0 0* *p7 2 0 0 0 0 0* *p8 0 0 0 0 5 2*	*t* = *1 2 3 4 5 6* *p1 9 0 5 0 3 1* *p2 0 4 0 4 7 3* *p3 0 0 0 0 0 0* *p4 7 1 0 3 7 5* *p5 0 0 0 1 8 3* *p6 8 7 3 4 4 5* *p7 0 7 4 7 3 1* *p8 4 8 7 4 0 5*	*t* = *1 2 3 4 5 6* *r1 1 0 1 1 1 1* *r2 1 1 0 1 1 1* *r3 1 0 1 1 1 1* *r4 1 1 1 0 1 1*	*r1* *1* *r2* *1* *r3* *1* *r4* *1*	*1411*
		...			
8	*t* = *1 2 3 4 5 6* *p1 5 0 0 0 0 0* *p2 0 1 0 0 0 0* *p3 1 9 4 0 10 0* *p4 0 0 0 0 1 5* *p5 18 5 0 5 0 0* *p6 0 0 0 0 0 0* *p7 0 0 0 0 0 0* *p8 0 0 0 0 4 0*	*t* = *1 2 3 4 5 6* *p1 4 0 5 0 3 1* *p2 2 5 0 4 7 3* *p3 0 0 0 4 0 0* *p4 7 1 0 5 7 0* *p5 0 0 0 0 0 4* *p6 8 7 3 4 4 5* *p7 2 7 4 7 3 1* *p8 4 8 7 4 0 8*	*t* = *1 2 3 4 5 6* *r1 1 1 0 1 1 1* *r2 1 1 1 1 1 0* *r3 1 1 0 1 1 1* *r4 1 1 1 0 1 1*	*r1* *1* *r2* *1* *r3* *1* *r4* *1*	*2607*

Table A5. Sample data to build a learning pattern for ANN *(Experiment_2)*.

Record Number	T	P	R	k_{r_1}	k_{r_2}	*v_m*	*m_r*	k_p	f_p	e_p	g_p	m_r	g_{pt}	$o_{r,t}$	h_{pr}
1	*6*	*8*	*4*	*466,426*	*50*	*45*	*1*	*22 5* *55 95* *11 72* *60 52*	*6 6 0* *6 8 9* *2 2*	*1 0 1* *1 1 1* *0 0*	*13 39* *31 39* *7 24* *35 9*	*1 1 0* *1*	*0 1 0* *1 5 7* *4 9 6* *9 9 3* *7 9 0* *8 8 1* *0 9 4* *3 5 0* *1 6 3* *2 9 9* *2 3 2* *8 6 6* *7 5 7* *4 8 1* *2 7 0* *5 5 0*	*15 80* *59 67* *38 14* *95 78* *96 85* *1 91* *18 45* *76 61* *81 0* *91 23* *18 70* *36 10*	*1 2 3* *1 1 2* *9 5 6* *0 6 8* *8 6 0* *0 4 5* *9 2 8* *4 4 8* *2 0 5* *7 4 0* *4 1*
							...								
50	*6*	*8*	*4*	*724,775*	*45*	*552*	*1*	*62 61* *85 64* *40 12* *33 83*	*4 7 4* *4 1 8* *2 8*	*1 0 1* *0 0 1* *1 0*	*46 0* *41 33* *39 21* *49 28*	*1 0 0* *1*	*1 3 9* *9 8 4* *5 5 2* *5 8 9* *5 1 3* *3 2 3* *0 7 3* *0 3 4* *5 1 9* *4 5 9* *2 5 4* *9 3 7* *2 4 4* *3 8 8* *3 1 1* *0 6 5*	*86 62* *72 85* *15 60* *12 26* *9 19* *95 94* *30 77* *53 59* *99 23* *24 78* *81 38* *65 58*	*7 6 0* *5 6 2* *0 3 1* *5 9 9* *7 9 7* *3 5 8* *8 0 5* *3 2 7* *8 2 0* *5 5 0* *5 9*

Table A6. Classification of data according to the evaluation criterion (Experiment_2).

Record Number	***T***	***P***	***R***	k_{r_1}	k_{r_2}	*v_m*	*m_r*	k_p	f_p	e_p	g_p	m_r	g_{pt}	$o_{r,t}$	h_{pr}	***Classification Label***
1	*6*	*8*	*4*	*466,426*	*50*	*45*	*1*	*22 5* *55* *95* *11* *72* *60* *52*	*6 6 0* *6 8 9* *2 2*	*1 0 1* *1 1 1* *0 0*	*13* *39* *31* *39 7* *24* *35 9*	*1 1 0* *1*	*0 1 0* *1 5 7* *4 9 6* *9 9 3* *7 9 0* *8 8 1* *0 9 4* *3 5 0* *1 6 3* *2 9 9* *2 3 2* *8 6 6* *7 5 7* *4 8 1* *2 7 0* *5 5 0*	*15* *80* *59* *67* *38* *14* *95* *78* *96* *85 1* *91* *18* *45* *76* *61* *81 0* *91* *23* *18* *70* *36* *10*	*1 2 3* *1 1 2* *9 5 6* *0 6 8* *8 6 0* *0 4 5* *9 2 8* *4 4 8* *2 0 5* *7 4 0* *4 1*	***1***
									…							
50	*6*	*8*	*4*	*724,775*	*45*	*552*	*1*	*62* *61* *85* *64* *40* *12* *33* *83*	*4 7 4* *4 1 8* *2 8*	*1 0 1* *0 0 1* *1 0*	*46 0* *41* *33* *39* *21* *49* *28*	*1 0 0* *1*	*1 3 9* *9 8 4* *5 5 2* *5 8 9* *5 1 3* *3 2 3* *0 7 3* *0 3 4* *5 1 9* *4 5 9* *2 5 4* *9 3 7* *2 4 4* *3 8 8* *3 1 1* *0 6 5*	*86* *62* *72* *85* *15* *60* *12* *26 9* *19* *95* *94* *30* *77* *53* *59* *99* *23* *24* *78* *81* *38* *65* *58*	*7 6 0* *5 6 2* *0 3 1* *5 9 9* *7 9 7* *3 5 8* *8 0 5* *3 2 7* *8 2 0* *5 5 0* *5 9*	***1***

Table A7. Verification ANN—(Experiment_2).

Record Number	T	P	R	k_{r_1}	k_{r_2}	v_m	m_r	k_p	f_p	e_p	g_p	m_r	g_{pt}	$o_{r,t}$	h_{pr}	*Classification Label*
1	*6*	*8*	*4*	*466,426*	*50*	*45*	*1*	*22 5* *55* *95* *11* *72* *60* *52*	*6 6 0* *6 8 9* *2 2*	*1 0 1* *1 1 1* *0 0*	*13* *39* *31* *39 7* *24* *35 9*	*1 1 0* *1*	*0 1 0* *1 5 7* *4 9 6* *9 9 3* *7 9 0* *8 8 1* *0 9 4* *3 5 0* *1 6 3* *2 9 9* *2 3 2* *8 6 6* *7 5 7* *4 8 1* *2 7 0* *5 5 0*	*15* *80* *59* *67* *38* *14* *95* *78* *96* *85 1* *91* *18* *45* *76* *61* *81 0* *91* *23* *18* *70* *36* *10*	*1 2 3* *1 1 2* *9 5 6* *0 6 8* *8 6 0* *0 4 5* *9 2 8* *4 4 8* *2 0 5* *7 4 0* *4 1*	***1*** ***(expected 1)***
									…							
50	*6*	*8*	*4*	*724,775*	*45*	*552*	*1*	*62* *61* *85* *64* *40* *12* *33* *83*	*4 7 4* *4 1 8* *2 8*	*1 0 1* *0 0 1* *1 0*	*46 0* *41* *33* *39* *21* *49* *28*	*1 0 0* *1*	*1 3 9* *9 8 4* *5 5 2* *5 8 9* *5 1 3* *3 2 3* *0 7 3* *0 3 4* *5 1 9* *4 5 9* *2 5 4* *9 3 7* *2 4 4* *3 8 8* *3 1 1* *0 6 5*	*86* *62* *72* *85* *15* *60* *12* *26 9* *19* *95* *94* *30* *77* *53* *59* *99* *23* *24* *78* *81* *38* *65* *58*	*7 6 0* *5 6 2* *0 3 1* *5 9 9* *7 9 7* *3 5 8* *8 0 5* *3 2 7* *8 2 0* *5 5 0* *5 9*	***1*** ***(expected 1)***

Table A8. Result ANN (rating input variables)—(Experiment_2).

Record Number	T	P	R	k_{r_1}	k_{r_2}	v_m	m_r	k_p	f_p	e_p	g_p	m_r	g_{pt}	$o_{r,t}$	h_{pr}	Classification Label
1	*6*	*8*	*4*	*368,888,888,000*	*1,800,000,000*	*2227*	*2*	*46* *46* *61* *74* *8* *73* *43* *42*	*70* *30* *40* *60* *10* *90* *10* *80*	*0* *1* *0* *0* *1* *1* *1* *0*	*4* *17* *33* *39* *33* *4* *3* *45*	*1* *1* *1* *1*	*9 0* *5 0* *3 1* *3 6* *0 4* *7 3* *1 4* *8 5* *7 3* *7 1* *0 5* *8 5* *2 1* *7 6* *8 9* *9 7* *3 4* *4 5* *3 7* *4 7* *3 1* *4 8* *7 4* *4 8*	*93* *85* *40* *28* *4* *18* *73* *79* *14* *10* *58* *59* *59* *18* *9* *23* *84* *36* *39* *25* *17* *10* *83* *46*	*6 2* *1 7* *4 8* *3 5* *0 3* *0 2* *3 0* *2 8* *3 2* *3 0* *2 7* *8 6* *9 1* *9 3* *0 7* *8 7*	***1*** ***(expected 1)***
						…										
10	*6*	*8*	*4*	*466,426*	*1,800,000,000*	*2227*	*2*	*46* *46* *6* *74* *80* *730* *400* *420*	*70* *300* *400* *600* *10* *900* *100* *800*	*0* *1* *0* *0* *1* *1* *1* *0*	*4* *17* *33* *39* *33* *4* *3* *45*	*1* *1* *1* *1*	*9 0* *5 0* *3 1* *3 6* *0 4* *7 3* *1 4* *8 5* *7 3* *7 1* *0 5* *8 5* *2 1* *7 6* *8 9* *9 7* *3 4* *4 5* *3 7* *4 7* *3 1* *4 8* *7 4* *4 8*	*93* *85* *40* *28* *4* *18* *73* *79* *14* *10* *58* *59* *59* *18* *9* *23* *84* *36* *39* *25* *17* *10* *83* *46*	*6 2* *1 7* *4 8* *3 5* *0 3* *0 2* *3 0* *2 8* *3 2* *3 0* *2 7* *8 6* *9 1* *9 3* *0 7* *8 7*	***1*** ***(expected 1)***

References

1. Scrimieri, D.; Afazov, S.M.; Ratchev, S.M. Design of a self-learning multi-agent framework for the adaptation of modular production systems. *Int. J. Adv. Manuf. Technol.* **2021**, *115*, 1745–1761. [CrossRef]
2. Stiehl, V. *Process-Driven Applications with BPMN*; Springer: Berlin/Heidelberg, Germany, 2014; ISBN 978-3-319-07218-0. [CrossRef]
3. Opekunova, L.A.; Opekunov, A.N.; Kamardin, I.N. Modeling Enterprise Architecture Using Language ArchiMate. In Proceedings of the International Scientific Conference "Digital Transformation of the Economy: Challenges, Trends, New Opportunities", Samara, Russia, 26–27 April 2019; ISBN 978-3-030-27014-8. [CrossRef]
4. Rumpe, B. *Modeling with UML*; Springer: Berlin/Heidelberg, Germany, 2016; ISBN 978-3-319-81635-7. [CrossRef]
5. Maher, G.M.; Abdolrasol, S.M.; Hussain, S.; Sarker, M.R.; Hannan, M.A.; Mohamed, R.; Ali, J.A.; Milad, A. Artificial Neural Networks Based Optimization Techniques: A Review. *Electronics* **2021**, *10*, 2689. [CrossRef]
6. Laisupannawong, T.; Jeenanunta, C. Improved Mixed-Integer Linear Programming Model for Short-Term Scheduling of the Pressing Process in Multi-Layer Printed Circuit Board Manufacturing. *Mathematics* **2021**, *9*, 2653. [CrossRef]
7. Cedillo-Robles, J.A.; Smith, N.R.; González-Ramirez, R.G.; Alonso-Stocker, J.; Alonso-Stocker, J. *A Production Planning MILP Optimization Model for a Manufacturing Company, Communications in Computer and Information Science*; Springer: Berlin/Heidelberg, Germany, 2021; ISBN 978-3-030-76307-7. [CrossRef]
8. Belil, S.; Kemmoé-Tchomté, S.; Tchernev, N. MILP-based approach to mid-term production planning of batch manufacturing environment producing bulk products. *IFAC-PapersOnLine* **2018**, *51*, 1689–1694. [CrossRef]
9. Vahidreza, G.; Saidi-Mehrabad, M.; Makui, A.; Sadjadi, S.J. Optimization and Mathematical Programming to Design and Planning Issues in Cellular Manufacturing Systems under Uncertain Situations. 2021. Available online: https://www.igi-global.com/chapter/optimization-mathematical-programming-design-planning/69302 (accessed on 20 March 2023).
10. Kallrath, J. *Business Optimization Using Mathematical Programming*; Springer: Berlin/Heidelberg, Germany, 2021; ISBN 978-3-030-73237-0. [CrossRef]
11. Sobczak, A. Enterprise architecture. In *Theoretical Aspects and Selected Practical Applications, Center for the Study of the Digital State*; Ośrodek Studiów nad Cyfrowym Państwem: Lodz, Poland, 2013; ISBN 978-83-936383-0-7.
12. Da Silva, I.N.; Spatti, D.H.; Flauzino, R.A.; Liboni, L.H.B.; Alves, S.F.D.R. *Artificial Neural Networks A Practical Course*; Springer: Cham, Switzerland, 2018; ISBN 978-3-319-82751-3. [CrossRef]
13. Nagy, B.; Galata, D.L.; Farkas, A.; Nagy, Z. Application of Artificial Neural Networks in the Process Analytical Technology of Pharmaceutical Manufacturing—A Review. *AAPS J.* **2022**, *24*, 74. [CrossRef] [PubMed]
14. Kumar, K.; Paulo Davim, J. *Optimization for Engineering Problems*; Wiley: Hoboken, HJ, USA, 2019; ISBN 9781786304742.
15. The Open Group Architecture Framework (TOGAF) Standard. 2018. Available online: https://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html (accessed on 20 March 2023).
16. Greefhorst, D.; Proper, E. *Architecture Principles—The Cornerstones of Enterprise Architecture*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1867–8920. [CrossRef]
17. Geurtsen, M.; Didden, J.B.; Adan, J.; Atan, Z.; Adan, I. Production, maintenance and resource scheduling: A review. *Eur. J. Oper. Res.* **2023**, *305*, 501–529. [CrossRef]
18. Available online: https://drive.google.com/drive/folders/1sehC4LHEdt1F4dnFzffQGuCfrugmtBsB?usp=share_link (accessed on 20 March 2023).
19. Ünal, H.T.; Başçiftçi, F. Evolutionary design of neural network architectures: A review of three decades of research. *Artif. Intell. Rev.* **2022**, *55*, 1723–1802. [CrossRef]
20. *Sudharsan Ravichandiran Hands-On Deep Learning Algorithms with Python: Master Deep Learning Algorithms with Extensive Math by Implementing Them Using TensorFlow*; Packt Publishing: Birmingham, UK, 2019; ISBN 978-1789344158.
21. Nazari-Heris, M.; Asadi, S.; Jebelli, H.; Sadat-Mohammadi, M.; Mohammadi-Ivatloo, B.; Abdar, M. *Application of Machine Learning and Deep Learning Methods to Power System Problems*; Springer: Berlin/Heidelberg, Germany, 2021; ISBN 978-3-030-77696-1.
22. Sitek, P.; Wikarek, J. A multi-level approach to ubiquitous modeling and solving constraints in combinatorial optimization problems in production and distribution. *Appl. Intell.* **2018**, *48*, 1344–1367. [CrossRef]
23. Sitek, P.; Wikarek, J. A Hybrid Programming Framework for Modeling and Solving Constraint Satisfaction and Optimization Problems. *Sci. Program.* **2016**, *2016*, 5102616. [CrossRef]
24. Aurambout, J.P.; Gkoumas, K.; Ciuffo, B. Last mile delivery by drones: An estimation of viable market potential and access to citizens across European cities. *Eur. Transp. Res. Rev.* **2019**, *11*, 30. [CrossRef]
25. Thibbotuwawa, A.; Nielsen, P.; Zbigniew, B.; Bocewicz, G. Factors Affecting Energy Consumption of Unmanned Aerial Vehicles: An Analysis of How Energy Consumption Changes in Relation to UAV Routing. In *Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology—ISAT 2018*; Świątek, J., Borzemski, L., Wilimowska, Z., Eds.; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2018; Volume 853. [CrossRef]
26. Świć, A.; Wołos, D.; Gola, A.; Kłosowski, G. The Use of Neural Networks and Genetic Algorithms to Control Low Rigidity Shafts Machining. *Sensors* **2020**, *20*, 4683. [CrossRef] [PubMed]

27. Bocewicz, G.; Nielsen, P.; Banaszak, Z.A.; Dang, V.Q. Cyclic Steady State Refinement: Multimodal Processes Perspective. In *Advances in Production Management Systems. Value Networks: Innovation, Technologies, and Management: APMS 2011*; Frick, J., Laugen, B.T., Eds.; IFIP Advances in Information and Communication Technology; Springer: Berlin/Heidelberg, Germany, 2012; Volume 384. [CrossRef]
28. Md, A.Q.; Jha, K.; Haneef, S.; Sivaraman, A.K.; Tee, K.F. Overview of data-driven quality prediction in the manufacturing process using machine learning for Industry 4.0. *Processes* **2022**, *10*, 1966. [CrossRef]