ISSN 1611-1699 / eISSN 2029-4433

VILNIUS TECH
Vilnius Gediminas Technical University

JOURNAL of BUSINESS ECONOMICS & MANAGEMENT

2024
Volume 25
Issue 5
Pages 892–920
https://doi.org/10.3846/jbem.2024.22242

IMPROVING PREDICTION ACCURACY OF OPEN SHOP SCHEDULING PROBLEMS USING HYBRID ARTIFICIAL NEURAL NETWORK AND GENETIC ALGORITHM

Mohammad Reza KOMARI ALAEI [1], Reza ROSTAMZADEH [2✉], Kadir ALBAYRAK [3], Zenonas TURSKIS [4], Jonas ŠAPARAUSKAS [5]

[1]*Department of Industrial Engineering, Alghadir Institute of Higher Education, Tabriz, Iran*
[2]*Artificial Intelligence, Automation, Big Data Research Center, Urmia Branch, Islamic Azad University, Urmia, Iran*
[3]*Department of Business Administration, Gazi University, Ankara, Turkey*
[4]*Institute of Sustainable Construction, Faculty of Civil Engineering, Vilnius Gediminas Technical University, Vilnius, Lithuania*
[5]*Department of Construction Management and Real Estate, Faculty of Civil Engineering, Vilnius Gediminas Technical University, Vilnius, Lithuania*

Article History:
- received 9 September 2022
- accepted 9 September 2024

Abstract. Scheduling issues are typically classified as constrained optimization problems that examine the allocation of machines and the sequence in which tasks are processed. Regarding the existence of one machine, identification of works processing sequence forms a complete time schedule. Therefore, following a review of previous works, the goal of the present study is designing a mathematical model for open shop scheduling (OSS) problems using different machines aiming at minimizing the maximum time required to complete the works using an artificial neural network (ANN) and genetic algorithm (GA). The research data were driven from a Shoe company carried out between the years 2019 and 2020. The GA and ANN methodologies were employed to analyze and forecast the scheduling of activities within the shoe manufacturing sector. The findings indicated that the probability associated with the third population of the GA was 0.15. Furthermore, an examination of the average values of standard error revealed that the neural network model outperformed in terms of predictive accuracy. The estimated minimum time necessary for task completion, as determined by the neural network, was calculated to be 0.96699, facilitating an optimal condition for meeting the established objectives.

Keywords: open shop scheduling (OSS), different work stations, single machine problems, resource assignment, efficient production, artificial neural network (ANN), genetic algorithm (GA).

JEL Classification: C6, C61, C63, M11.

✉Corresponding author. E-mail: *Reza.Rostamzadeh@iau.ac.ir*

1. Introduction

Mathematical models are the basis for scheduling theory that explain scheduling processes. A theoretical framework facilitates a quantitative approach aimed at establishing a structure for addressing issues within the context of mathematical models. This process involves elucidating resources and activities while transforming decision-making objectives into a target function. Consequently, resources, activities, and goal functions are identified as fundamental

Copyright © 2024 The Author(s). Published by Vilnius Gediminas Technical University

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

components in mathematical scheduling models. Qualitative and quantitative capabilities determine the characteristics of resources. Each model represents the type and number of resources utilized. On the other hand, activities are described based on data such as resources required, the time to do the activities, starting and ending time of activities. Also, the target functions entail system costs to administer decisions related to assignment of resources to the activities. The key considerations in the scheduling process encompass the effective allocation of resources, prompt reactions to demand fluctuations, and the meticulous alignment of actual delivery times with established delivery schedules. Furthermore, the principal decisions within the scheduling framework involve the optimal utilization of resources, swift responses to market demands, and the precise adherence to delivery timelines as dictated by the planned delivery dates (Arroyo et al., 2011).

During some recent years, the topical issue of scheduling has been intensively investigated regarding some sub-branches of operational research such as investigations in management field, different categories of management like production and operation, theoretical computer science studies and so on. Thus, it could be stated that there are plenty of research literature in the field of scheduling (Coelho & Vanhoucke, 2018; Daniels et al., 2004). On the other hand, many different results were gained (Coelho & Vanhoucke, 2018; Ma et al., 2021). Usually, it can be isolated in two parts: static and dynamic (Ouelhadj & Petrovic, 2009; Baykasoğlu & Ozsoydan, 2018; Baykasoğlu et al., 2020). The challenges associated with production scheduling have been extensively examined as optimization issues, given their diverse applications across various industries. The OSS problem involves the allocation of a collection of jobs to multiple machines, with each task linked to a specific processing duration and without any predetermined order for the execution of operations (Anand & Panneerselvam, 2015).

The OSS refers to a scenario where a collection of tasks is executed across various machines, with the aim of achieving an optimized performance standard. The OSS has wide applications in various sectors. Examples of problems in the healthcare industry include: patient scheduling for coronary heart disease diagnosis, shoe factories, car factories, etc. In conclusion, the open shop problem involves the allocation of a series of jobs, each comprising multiple operations, across a variety of machines, with the stipulation that each machine can handle only one operation at any given time. Notably, the sequence in which jobs are processed on the machines is not predetermined (Ahmadian et al., 2021). The OSS framework is applicable to numerous industrial sectors, including plastic injection molding, chemical processing, the oil industry, food manufacturing, and pharmaceutical production. Additionally, in the service industry, this problem can be effectively applied to the scheduling of medical services, automotive maintenance, museum visits, and telecommunications (Lin et al., 2008; Naderi et al., 2010; Abreu et al., 2020). The OSS provides complete freedom for scheduling without any particular order. The objective of this task is to reduce the execution time of the OSS problem. There are two inherent constraints associated with OSS: a single machine can handle only one job at any given moment, and each job can be assigned to only one machine at a time. For the sake of simplicity, pre-emption is not permitted, ensuring that all tasks are executed without any interruptions. It has been established that OSS is NP-hard when the number of machines exceeds two. Furthermore, the OSS presents a broader solution space compared to other scheduling issues on the shop floor, as it allows for greater scheduling flexibility and the Makespan of optimal scheduling solutions tends to be more closely aligned. This is particularly evident at the lower bound, which is defined by the maximum total processing time for each task (Lee & Loong, 2019).

The open-shop scheduling problem (OSSP) represents a significant NP-hard problem class. Unlike the job-shop scheduling problem (JSP), which mandates a predetermined order of job operations, the OSSP utilizes a flexible sequence-processing approach. This flexibility in sequencing is designed to alleviate the difficulties posed by rigid operational sequences during job execution. As a result, the OSSP enhances the adaptability of manufacturing processes within organizations, leading to improvements in machine utilization and overall production efficiency. It is important to highlight that the OSSP offers a broader solution space than the conventional JSP, thus providing greater opportunities for optimization (Wang et al., 2024).

The most notable contributions concerning the OSSP were documented by Gonzalez and Sahni (1976); simulated annealing (SA) (Ahmadian et al., 2021); minimize the makespan (Liaw, 1999); a tabu search (TS) algorithm (Gueret & Prins, 1998); branch and bound (Harmanani & Ghosn, 2016); Naderi et al., (2010). Many meta-heuristics models are utilized to solve big problems but they lack the required precision. Thus, in the present research and to enhance prediction precision a GA and neural networks' algorithm were applied. Meta-heuristic algorithms are used in this work. Meta-heuristic algorithms represent a category of approximate optimization techniques characterized by their capacity to escape local optima, making them applicable to a diverse array of problems. However, these algorithms lack the precision necessary for accurate predictions. To enhance predictive accuracy and reduce production time in the shoe manufacturing sector, neural networks have been employed. Notably, there exists a gap in the existing literature on OSS, as prior research has largely overlooked the simultaneous consideration of human errors, preventive maintenance, and production scheduling. Therefore, the authors identified fourteen factors to investigate and evaluate the existing gap in order to determine the cause of the gap in maintenance and other discussions using the mentioned factors. This research work is comprised of the following sections: a literature review presented in second section. Next, the methodology has been introduced in section three, more specifically concentrating on GA and ANN. At the end, the data analysis and conclusions are prepared in sections four and five respectively.

2. Review of the related literature

The term scheduling means a decision-making process for the appropriation of the resources by achieving one or more goals in a given time frame (Pinedo, 2022; Lee & Loong, 2019). The scheduling system is designed to coordinate all operations across relevant machines in order to meet the constraints of process plans and enhance the established objectives. Scheduling serves as the connection between two phases of production: the preparation of processes and their subsequent implementation. Scheduling is the process of allocating resources to activities taking into account their respective time periods in order to optimize one or more objective functions. This process is considered to be the basis of many manufacturing and service industries as a decision-making process. Scheduling problems and their categorization have been defined through many different patterns. Numerous patterns are available for managing each operation related to scheduling issues, which arise from product manufacturing processes. These patterns are influenced by the quantity of operations required to complete a task and the number of machines involved in each process. In scheduling problems, the goal of finding the sequence of activities may be different. Some of the common objectives are minimizing the completion time of all tasks or minimizing the number of tasks that are late. One of the most important goals in many manufacturing industries is the timely delivery of goods and services, which plays a role in obtaining profit or lower costs. Scheduling problems are defined by considering resource allocation as job processing times that are controlled by changing the resource allocation to jobs. Typically, scheduling issues are formulated as constrained optimization problems, where the focus is on the decisions regarding machine allocation and the order in which tasks are processed (Li et al., 2018).

Our understanding of scheduling problems is related to the calculation of the total flow time, the number of tasks that are delayed and the total delay. The measure of total delays, in particular, has become a standard method for measuring compliance with the delivery date, however, the consequences of completing the work before the delivery date have been ignored and only those late works have been penalized. But this criterion began to change as the share of Just-In-Time (JIT) production grew, emphasizing that early delivery should be considered as inappropriate as late delivery. In a just-in-time schedule environment, work that is completed early by its due date must be held in inventory, while a job that is completed after its due date may interfere with meeting customer needs. Therefore, an ideal schedule is a schedule in which all tasks are completed exactly on the dates assigned to them (Behnamian et al., 2010). In workshop settings, scheduling challenges frequently arise when the time required for machine setup is overlooked or treated as part of the overall job processing duration. Such environments are typically modeled under the premise that setup times are negligible in comparison to processing times. Consequently, these times may be disregarded, or they may be considered independent of the sequence in which jobs are processed on the machines, allowing them to be incorporated into the processing durations. Nevertheless, in numerous industrial contexts, a sequence-dependent setup time is evident when transitioning between jobs on the same machines (Anders et al., 2005).

The challenges associated with scheduling in the context of resource allocation are characterized by the management of job processing durations through the modification of resource distribution to various tasks. Numerous researchers have contributed to this dynamic domain of scheduling studies, including investigations into JIT systems (Wang & Alidaee, 2019) and maintenance activities (Ji et al., 2015). A comprehensive discussion on scheduling challenges involving resource allocation, particularly with respect to controllable processing times, is presented by Shioura et al. (2018). Additionally, Liu and Xiong (2021) explored issues related to resource allocation scheduling on a single machine, focusing on deterioration and general positional effects.

- *Flow shop*

In the context of product or process flow, production systems are classified into two main types: continuous and intermittent. Intermittent systems include flow shop systems, also referred to as online production, and job shop systems, commonly known as task shops (Osorio Gómez et al., 2008). A flow shop is characterized by the presence of *n* jobs awaiting processing across *m* machines, with all jobs following the same route or sequence of operations (Caicedo-Rolon & Parra Llanos, 2021). Some scholars in the field have worked on this vivid area of flow shop (Wang et al., 2024; Xue et al., 2024; Mousighichi & Avci, 2024).

- *Open shop system*

The OSS represents a significant optimization challenge within the fields of computer science and operations research. It serves as a specific instance of optimal job scheduling. This scheduling problem is foundational, involving the allocation of a collection of jobs to a diverse array of resources, with the objective of optimizing performance outcomes. The OSSP has many functions in varied sections (Ahmadian et al., 2021). In fact, regarding job shop, job operations should be processed using a fixed (technological) order, generally different for every work. On the contrary, regarding open shop problems, activities of a single job can be processed using any desired order. A significant body of research has emerged in the domain of open shop scheduling since the seminal work of Gonzalez and Sahni (1976), who proposed that a minimum completion time for multi-processor OSS can be achieved within a multinomial timeframe. An important review of deterministic machine scheduling issues, which encompasses the open shop problem, was conducted by Chen and Strusevich (1993). This review addresses complexity results and presents both exact and heuristic algorithms tailored to various performance metrics, including maximum weighted earliness and tardiness, total weighted completion times, and the weighted count of late jobs. In their study, Noori-Darvish et al. (2012) focused on minimizing total weighted tardiness and completion time. Additionally, Tavakkoli-Moghaddam and Seraj (2009) examined the minimization of mean tardiness and completion time in open shop settings. The investigation of minimizing total completion time, earliness, or tardiness has also been explored by Doulabi (2010) and Anderson and Potts (2002). Sheykhalishahi et al., (2019) investigated multi-objective OSS by considering human error and preventive maintenance. Thus, the job-shop solution is suggested as a single processing range of operations on machines, while the open-shop solution is shown by two operations' processing orders within works and operations done on machines.

The OSS includes the machine (m) to do the work (m_1, m_2, m_3, ..., m_n) where each job is included in the activity. J work activity is indicated by ji with Q_{ij} that must be $pi_i \geq 0$ processed on the machine M_j for a unit of time. Therefore, there will be a total number of $n \times m$ (Q_{11}, Q_{12}, Q_{mn}) activities. The total processing time ji of the job is equal to $p_{ij} = \sum_j pij$ and the total time required for the machine M_j is equal to $M_j = \sum_j pij$. Also, a new parameter $H = \max(pij, M_i)$ is defined in such a way that obviously the end time of each scheduling program in the open shop will be at least as long as H.

The limitations of the open workshop problem are as follows:

- Each machine processes at most one activity at the same time,
- Each work is processed by at most one machine,
- Each activity Qi_j must be processed on the machine M_j for a pij unit of time (Blazewicz et al., 2007).

Ahmadian et al., (2021) reviewed many studies related to the OSS to minimize the duration. In a typical JSP, there are n jobs denoted as j_1, j_2, ..., j_n, each characterized by distinct processing times. These jobs must be allocated to m machines that possess varying processing capabilities, with the objective of minimizing the makespan, defined as the total duration required to complete all jobs. A specific variant of this problem, referred to as the OSS, involves each job comprising a series of operations, O_1, O_2, etc., which can be executed in any arbitrary order. The OSS is also known as the moderated job shop scheduling problem (Panneerselvam, 1999) and represents a subset of both the flow shop scheduling problem (FSSP) and the job shop scheduling problem. In the OSS framework, there are n jobs, each containing a maximum of m operations, and the sequence in which the operations of each job are processed is flexible. For instance, if a job includes three operations labelled 1, 2, and 3, it can be executed in any of the six possible sequences.

Sequence 1: 1 – 2 – 3;
Sequence 2: 1 – 3 – 2;
Sequence 3: 2 – 1 – 3;
Sequence 4: 2 – 3 – 1;
Sequence 5: 3 – 1 – 2;
Sequence 6: 3 – 2 – 1.

In the case where a problem is composed of n jobs, each associated with a maximum of m operations of the OSS, a generalized data format for the processing times is presented in Tables 1 and 2 (Anand and Panneerselvam, 2015). Specifically, in Table 2, a positive value tij indicates that job i necessitates t_{ij} units of processing time for operation j, whereas a value of zero signifies that job i does not require operation j.

Table 1. Operations of each job in any sequence

Sequence 1	1	2	3
Sequence 2	1	3	2
Sequence 3	2	1	3
Sequence 4	2	3	1
Sequence 5	3	1	2
Sequence 6	3	2	1

Table 2. An open shop problem's generalized data format regarding the process time

	Opeation j									
		1	2	3	.	.	j	.	.	M
Job i	1	t_{11}	t_{12}	t_{13}	.	.	t_{1j}	.	.	t_{1m}
	2	t_{21}	t_{22}	t_{23}	.	.	t_{2j}	.	.	t_{2m}
	3	t_{31}	t_{32}	t_{33}	.	.	t_{3j}	.	.	t_{3m}
	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.
	i	t_{i1}	t_{i2}	t_{i3}	.	.	t_{ij}	.	.	T_{im}
	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.
	n	T_{n1}	T_{n2}	T_{n3}	.	.	t_{nj}	.	.	t_{nm}

- *Flow Shop problem*

In the context of a permutation flow shop configuration, machines are arranged in a sequential manner, ensuring that jobs are processed in a consistent order without skipping any machines. Flow shop scheduling is a prevalent approach utilized across numerous industries, characterized by an organization of processes or shops that adheres to a specific flow, with all jobs following an identical processing sequence (Fekri et al., 2024). The FSSP is comprised of locating a processing order for n jobs on m sequential machines in a way that specific criterion is optimized. If all the machines are limited to the same processing order of jobs, the issue of permutation flow shop scheduling problem (PFSSP) occurs. The PFSSP using the makespan minimization criterion is NP-Complete and a commonly investigated problem considering Operations (Fernandez-Viagas et al., 2017). A new tiebreaker in the NEH (Nawaz, Enscore, and Ham) heuristic for the permutation flow shop scheduling issue was proposed by Benavides (2018). They suggested a new tiebreaker according to the reckoning of the variation of idle times created with the insertion of the new job, and containing reversibility property of the PFSSP. Findings show that this tiebreaker outperforms the currently common heuristics.

- *Definition of prediction*

Prediction is a key element for managerial decisions. In a decision, a sequence of the effective effects of this decision and the developments that may occur after the decision is taken into account. The ability to estimate these uncontrollable effects will improve selection and decision making. For this reason, management systems need foresight for designing and controlling the operators of their formations.

- *Prediction methods*

Forecasting methods are typically categorized into two main types: qualitative models and quantitative models. Qualitative methods include Delphi method, brainstorming method, nominal group method, etc. In addition to the above four methods in qualitative forecasting, there are other methods such as subjective estimation methods, comparative methods, appropriate tree, and morphological research, which are subjective forecasting methods. Quantitative methods, including general univariate or multivariate models, which in univariate models, such as simple smooth methods, Box-Jenkins, autoregressive integrated moving average (ARIMA), trend analysis, etc., in multivariate models, methods such as regression and econometric analysis are of interest, but in new methods, neural networks have been considered (Azer & Rajabzadeh, 2012).

- *ANNs method*

Nowadays, ANNs are considered as one of the new methods in prediction. The use of ANNs or in general "neural networks" was started by McCulloch and Pitts (1943). Since the purpose of artificial intelligence is the development of paradigms or algorithms used by humans to be used in machines, ANNs, as one of the most important methods of artificial intelligence, seek to imitate the functioning of the human brain. ANNs are new computing systems and methods that are used for machine learning, displaying knowledge, and finally applying knowledge to predict the output responses from the complex system. Neural networks are composed of a number of neurons or cells that have the ability to learn, which relate the input set to the output (Minhaj, 2017). The ANN model employs the feed-forward backpropagation algorithm, a widely utilized method in machine learning for the training of artificial neural networks. Optimization of the model was achieved through careful selection of the number of hidden layers, the number of neurons within each layer, and the choice of activation functions (Gautam et al., 2023). ANNs facilitate a non-linear mapping of relationships between inputs and outputs, frequently serving as black-box models for systems characterized by unknown or intricate underlying dynamics. Learning structure in ANNs using optimization. ANNs have many names such as: communication, parallel processing and distribution, neural computing, natural intelligence systems, machine learning algorithm and ANNs. The basic operation of an artificial nerve consists of collecting weighted input signals and using an activity function to produce an output. The artificial nerve, which is the main unit in the artificial nerve networks, is modeled after the four special functions of the biological nerve, but the artificial nerves are much simpler than the biological nerves. The artificial nerve has a fixed number of inputs (n) and each input is connected to another nerve by a weighted connection (wi). The inputs (xn) entered into the network are multiplied by the weight (wn) of the connection, and in the simple case, these products are added and transformed into an output through the transfer function to produce the result (Alwani & Hosseinpour, 2016).

An ANN has several inputs (x_i) and one output (y_i). Each input has its own weight that controls the intensity of the connection (W_{ij}) and is usually a real number. Weights can be inhibiting or encouraging (that is, have positive or negative values). The input of the network is calculated by the sum of the input values multiplied by the corresponding weights.

$$net_t = \sum_j x_j W_{ij}; \tag{1}$$

$$y_i = f_i(net_t). \tag{2}$$

The output value calculates the application of the activity function that is used in the network.

- *Simple perceptron*

The most influential network in neural networks was invented by Frank Rosenblatt in the 1960s under the name of perceptron. Minsky and Papert (1969) described the limitations of single-layer perceptron. The result showed that some neural network researchers lost their interest. This book showed that single-layer perceptron could not perform basic pattern recognition operations, until in the 80s, with proper training, it was determined that perceptron multi-level systems can perform this operation.

- *Feed forward Multilayer Neural Network (MLP)*

In feedforward neural networks, nodes are organized into sequential layers with unidirectional communication. When an input pattern is introduced to the network, the initial layer computes its output values and relays them to the subsequent layer, with each node transmitting a signal to the nodes in the next layer. Multi-layer perceptron networks exemplify this category of networks. Typically, feedforward neural networks exhibit several defining features: they consist of three types of layers; input layers, hidden layers, and output layers, without a restriction on the number of hidden layers. In these networks, neurons in each layer communicate with those in the following layers. The nodes are interconnected through links, each possessing adjustable weights. The arrangement of nodes occurs in parallel layers, with connections limited to adjacent layers. Each neuron functions akin to a processor, receiving input from the preceding layer via connections, executing processing tasks, and transmitting the output to the subsequent layer through output connections. The simultaneous operation of these nodes facilitates a parallel processing system. Upon receiving an input vector, the nodes in the input layer accept it and forward it to the first hidden layer without any processing. The hidden layer nodes then process the incoming information and pass the results to the next layer until the output layer nodes generate a final output vector. This characteristic of information flow is what designates the network as a feedforward network. Although the processing within the nodes can be intricate, it fundamentally arises from straightforward series and parallel operations across the various layers of the network. Generally, a neuron is designed to accommodate multiple inputs (Minhaj, 2017).

- *Applications of neural networks in management*

ANNs are considered as one of the branches of artificial intelligence that provide us with a lot of knowledge to improve management decisions. A fundamental responsibility of management is the process of decision-making, wherein the critical component is the acquisition of accurate information. This information serves to illuminate potential future scenarios, thereby facilitating improved decision-making outcomes. In addition, these artificial intelligence techniques AI and AAN ANNs provide as sources of knowledge for organizations that have guiding and managerial responsibility (Iqbali & Moghadspour, 2017). ANNs have the potential to greatly enhance supply chain management (SCM) by optimizing several critical components, including demand forecasting, inventory control, logistics efficiency, and risk assessment. These computational models, which draw inspiration from the neural networks found in the human brain, possess the ability to learn from data, enabling them to identify patterns and generate predictions (Zhu & Liu, 2023).

- *GA*

GA represent a methodological approach for addressing various problem-solving scenarios. This approach is applicable to a diverse array of economic challenges, including but not limited to the optimization of stock portfolios, forecasting bankruptcy, and predicting stock market trends. Allen and Karjalainen (1999) predicted the efficiency of different combinations of trading techniques. They used GA on S&P500 data between 1928 and 1995 to reveal trading methods. But this method did not have a good yield compared to the old methods. Although the simulated research was done to improve the previous research. Iba and Sasaki (1999) implemented a GA program to predict the value of stocks in the Japanese stock market. They tried to predict the best investment situation when they intend to buy or sell stocks, they implemented this GA on the neural network designed, so that they can make the most profit in take over the market (Defersha & Chen, 2010). In similar research, Rimcharoen et al., (2005) presented the comparative evaluation strategy and the GA in order to be able to evaluate the movement trend of 5 economic indicators (MLR, gold price, Hang Seng, Nikkei, Dow Jones). In the end, they considered the method used by Hang Seng to predict MLR as the best prediction method.

- *Neural network combined with GA*

The GA can be combined with the neural network for three different purposes. To determine the amount of input variables that are needed, to find the optimal neural network topology, training in neural network. Also, the GA can remove the irrelevant variables of the input. In the field of prediction by neural networks, Yao and Tan (2001) presented a 7-step guide, which are named in order:

I. Initial data processing, II. Selection of input and output, III. Sensitivity analysis in order to find the most sensitive index in the output, IV. Data organization, V. Building the model, VI. Analysis after doing the work VII. Model recommendations. In the above guide, nothing is mentioned about the design of the neural network training method. Therefore, the design of the neural network training method should also be considered as a main step in neural network design.

It is believed that optimizing OSS is known as a prerequisite issue for organizational productivity and lots of researchers have been busy with probes on it. For the purpose of gaining a typically efficient calendar to extract components/products, a bunch of algorithms have been suggested, including exact algorithms, heuristics, GAs and neural network. Regarding heuristics, many scholarly papers have been presented in different conferences and research journals during some recent years. Chen and Strusevich (1993) introduced a linear time heuristic to investigate about OSS. he findings indicate that the worst-case performance ratio of the heuristic, which stands at 3/2, is significantly more favorable than that of the greedy algorithm, recorded at 5/3. In a further effort to address this challenge, Colak and Agarwal (2005) employed a non-greedy heuristic alongside Augmented Neural Networks (AugNN) to enhance makespan optimization. Naderi et al., (2010) introduced four insertion and reinsertion heuristics aimed at mitigating redundancy within the permutation list. In the context of GA, it is posited that they can serve as a potent approach when integrated with other algorithms to tackle OSS. Additionally, Bello et al. (2016) proposed an innovative model that merges neural networks with reinforcement learning, utilizing negative tour length as a reward and achieving model optimization through the AC algorithm. Concurrently, Zang et al. (2019) suggested the implementation of a hybrid deep neural network scheduler (HDNNS) to facilitate the convolution operations in deep learning, specifically addressing the JSP.

Based on what was pointed above, it can be observed that researchers have utilized these methods for optimization, finding suitable performance in tasks, assigning *n* machines to *m* tasks to the open shop scheduling problems in neural networks, combined models of genetic neural networks and reinforcement learning framework of critical actors in order to alleviate open shop scheduling problems. On the other hand, the authors aimed to minimize the maximum time to complete work in shoe production lines. To achieve such a goal, the GA and neural networks have compared to recognize which one has a more predictive accuracy. The utilization of our proposed method could fill the gap observed in the previous research projects. When we put an end to the experiment and analysis stages in the present study, it was observed that the neural networks have a high predictive accuracy in comparison with the GA. Furthermore, to observe this possibility occurrence, it is referred to the research gap in the introduction section to stress the fact that our goal in this project is to minimize the maximum time to complete the work in shoe production lines. Additionally, through the use of the integrated model of GA and neural networks, we could be able to compare a more accurate method throughout the two options mentioned after the analyses being carried out.

3. Research method

In addressing the research problem of optimizing OSS resolution across various machines with the objective of minimizing the maximum completion time of tasks, the researcher initially explored the factors influencing scheduling and the sequencing of operations, particularly within the context of neural networks and GA in the discovery studies section and relevant literature. Then, they were analyzed using neural network and GA patterns. As a consequence of such research projects, it could be gained a theoretical framework through which the OSS would be resolved. Also, to resolve this technique MATLAB software utilized. The Figure 1 represents the different stages proposed in the current research.

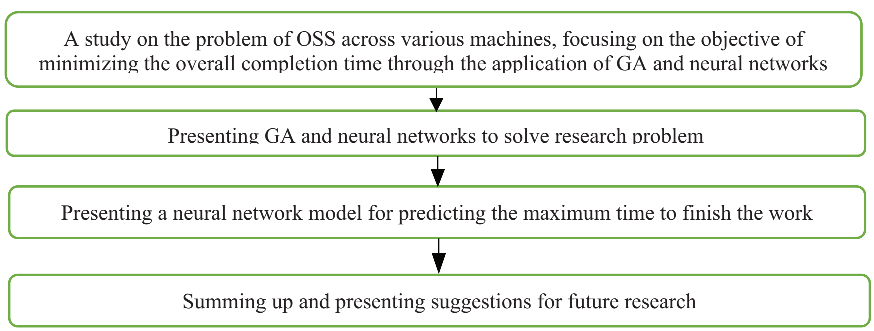

Figure 1. Model for implementation research (source: authors, 2024)

In addition, to evaluate the various criteria for scheduling activities and their role in forecasting the completion times of these activities, all relevant data and information concerning the scheduling of manufacturing line activities within the shoe industry, specifically pertaining to the Sanat-Gostar-e-Hamgam shoe manufacturing company, have been collected. This study has utilized current market data from Iran covering the period from 2019 to 2020. The order of using the data is that the first 10 activities were used for training and 2 activities for research modeling and the next 2 activities were used for testing and prediction.

4. The conceptual model validation

In order to establish the relationship between the variables in the study, the following tasks were executed in a sequential manner: Utilizing the variance analysis method and employing SPSS, the potential existence of a relationship among the variables and the impact of these variables were assessed (see Table 11).

4.1. GA

GAs is mainly driven from Darwin's theory. Regarding Darwin's theory, generations with superior traits and characteristics achieve higher survival and reproduction, and their superior traits and characteristics will be delivered to future generations. The response to a problem solved using GAs is permanently getting better. The GA commences with a series of responses that are proposed by chromosomes. Using such an algorithm, the answers gained from one population are used to produce the next population, along with a combination or mutation. During such a process, it is believed that the new population will be better than the preceding one. The choice of some chromosome responses from the total answers of the parents aiming at creating new responses for the children stems from their desirability, which is carried out using the fitting function. Naturally more proper responses have a better chance of reproducing. It lasts until a presupposed condition is observed (like the number of populations or the rate of improvement of the response) (Nemati et al., 2016). Totally, GAs is comprised of the following constituents. Many of inventions by human beings are inspired by the nature. ANNs are among most prominent examples in such inventions. Another invention is the development of the idea referred to GA. GAs simulate the completion process in nature aiming at finding the best possible resolution for a problem probing through Candidate Solution Space. Within the process of the search for an optimal solution, first a set or a population of the primitive resolutions are produced. Then, through a consecutive set of generations a set of changed responses are produced (in each of GAs, certain changes are made on genes of the chromosomes of the population). The primitive responses usually change in a way that in every generation, the population of resolutions converge into an optimal response. This branch of Artificial Intelligence is based on the mechanism of living beings' evolution and the production of more successful and fit species inspired by nature. In other words, the major idea for GAs refers to Survival of the Fittest. A chromosome refers to a long and complicated string of Deoxyribonucleic Acid or DNA. The heredity factors identifying the personal characteristics or features of an individual form these chromosomes. Each of the characteristics of individuals is coded based on a combination of DNA in human genes. There are four bases in living beings' body to produce chromosomes using DNA as follows:

- Base A or Adenine;
- Base B or Cytosine;
- Base G or Guanine;
- Base T or Thymine.

As alphabet comprise the structure of a language, a meaningful combination of chromosomes (and its bases) produces specific manuals for cells. The changes in chromosomes occur during reproduction process. Parents' chromosomes are exchanged randomly through a specific process called integration or Crossover. Therefore, the offspring inherit some of the characteristics or traits of father and some of the characteristics of mother and represent them.

4.2. Designing ANN

Considering rate of products shoe and local consumption amount, the changes in product were calculated. A neural network with two input layer neuron (demand rate and imports and the amount of production and exports) and an output layer neuron (maximum time required to complete works) was designed.

MLP network has been used for prediction and also different threshold functions were assessed to find the optimal states which are as follows:

$$Y_i = \frac{1}{1 + \exp\left(X_j\right)}. \tag{3}$$

Logarithmic Sigmoid function

- *Sigmoid function*

This nonlinear activation function changes its input into an amount within the range of 0 and 1. The greater amount of the input leads the output of the function to be closer to 1. Meanwhile, when the amount of this function is very small (negative number), the output amount of sigmoid function becomes closer to 0.

Y_i – neuron output, X_i – neuron input, exp – negation expressive.

The range from 0 to 1 lead to optimal output or binary amounts or it is between 0 and 1.

$$Y_i = \frac{2}{\left(1 + \exp\left(-2X,\right)\right) - 1}. \tag{4}$$

The function above is a bipolar sigmoid function similar to tangent hyperbolic function: Y_i – neuron output, X_i – neuron input, *exp* – negation expressive. Range between –1 and 1.

Sigmoid and Gaussian functions are used widely in neural networks. Sigmoid is defined as follows:

$$f(N_{eti}) = 11 + e - N_{eti}Tf(N_{eti}) = 11 + e - N_{eti}T. \tag{5}$$

For example, by selection of T = –1.0 and T = 1.0, the figure of this function will be as follows (see Figure 2):

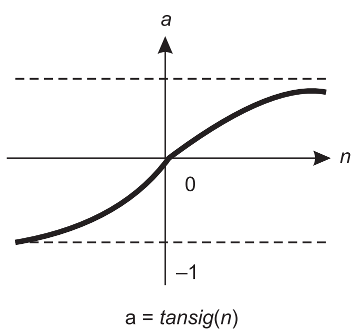

Figure 2. Hyperbolic tangent sigmoid function

One of the problems occurring while neural network is being taught is over-teaching the network. In this case, while network is being taught, error reaches acceptable rate, but when assessment is done, network error becomes more than training data errors. There are two solutions to avoid over teaching: *a* – rapid stop of teaching, *b* – the selection of the least neurons in hidden layer. In this research the second method is used. To teach the network, first the data were divided into three parts in a way that 70 percent was used for teaching, 15 percent of the data was used for assessment, and 15 percent of the data was used to test the network. While teaching the network, when the training inter data error and assessment is increasing, the teaching process is stopped. To find neural network or appropriate topology with the help of teaching algorithm, mean squares error (MSE) used through which the goal is to minimize the error mentioned that could be defined as follows:

$$MSE = \sum_{P}^{M}\sum_{P}^{M}(S_{ip}(S_{ip} - T_{ip})2, \tag{6}$$

where, MSE is mean squares error in testing stage, S_{ip} represents network output in ith neuron and pth pattern, T_{ip} determines the target output in ith neuron and pth pattern, N shows the number of output layer neurons, and M refers to the number of test patterns. In order to measure the network chosen in the previous stage identification criterion used (R^2) for the data predicted while teaching the network.

To increase the precision and speed of the ANN, input and output data could be calculated using the following equations in normalized format in the range of [0, 1]:

$$T_n = \frac{T_i - T_{\min}}{T_{\max} \quad T_{\min}}; \tag{7}$$

$$X_n = \frac{X_i}{X_t}; \tag{8}$$

$$t_n = \frac{t_i}{50}, \tag{9}$$

where, T_n, X_n, t_n are the normalized amounts, X_i is the real completion work time, X_n is the primary completion work time and $T_{\max}$ refers to maximum completion work time, $T_{\min}$ is the minimum completion work time, and T_i is the real work time. T_i is the real time span. After network teaching, the network with the least test error would be chosen as the best network.

5. Data analysis

5.1. Analysis of data obtained using GA

A comprehensive representation of the neural network shown in MATLAB software. Also, the distinct stages of a neural network are represented in Figure 3. The transmission of data from the initial layer to the intermediate layers occurs through a sequence of mathematical computations, culminating in the observable output at the final layer. In this research, there are 14 input layers, 8 hidden layers and one output layer.

The Figure 4 shows the first and second input layers of neural networks and GA, which include gates Forget gate, the input gate, cell state and output gate.

The Figure 5 shows the gates in the structure of the neural network, including Forget Gate, Input Gate, Cell State, and Output Gate, which are combined with GA.

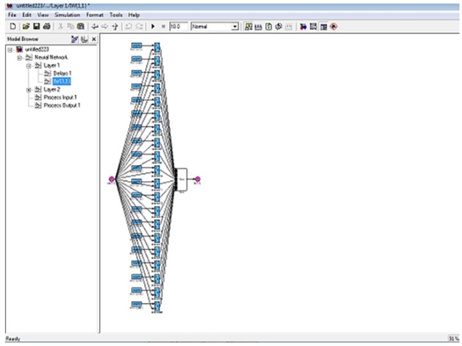

Figure 3. Structure of recurrent ANN and GA combination (source: authors, 2024)

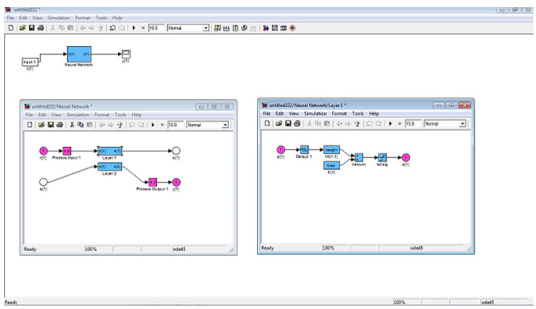

Figure 4. First and second input layers of neural networks and GA (source: authors, 2024)

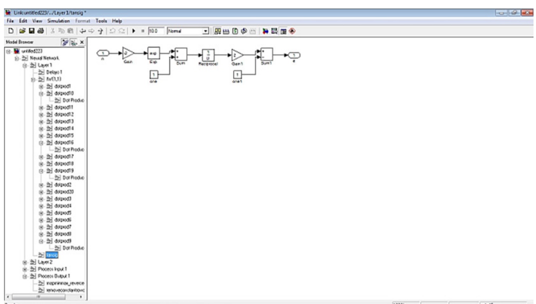

Figure 5. Related to the processing layer of neural networks and combined GA (source: authors, 2024)

5.1.1. Proposed model

In this section, the proposed mathematical model of heuristic and meta-heuristic algorithm with integer programming approach for the investigated problem is presented. The description of the indices and input parameters, decision variables, limitations and its objective function are discussed. In the first stage, the workshop flow shown with S_1 and the second stage with S_2. It also assumed that there are m_1 parallel machines in the first stage, which denoted as $[M_{11}, M_{12}, \ldots, M_{1,} m_1]$. There are m_2 parallel machines in the first stage, which denoted as $[M_{21}, M_{22}, \ldots, M_{2,} m_2]$. Also, $P_{11}, P_{12}, \ldots P_{1}, n_1$ represents the processing times of the first stage and $P_{21}, P_{22}, \ldots P_{2,} n_2$ represents the processing times of the second stage. The due dates of the works displayed with $d_{11}, d_{12} ..., d_m$. Finally, $r_1, r_2 ..., r_n$ defined as the time of availability of work, or in other words, the time of entering the work into the workshop. L represents the set of scheduled tasks and k is the set of remaining tasks for scheduling.

5.1.2. Roulette wheel

Since the boundaries of Roulette-wheel are marked based on fitness of each of the strings or chromosomes, it is expected that $\sqrt{FF1/F}\quad F_i$ copy of ii^{th} string is produced in mating pool by this operator. The average fitness of chromosomes' population will be calculated as follows.

$$F = n\sum_0 i = \sqrt{FiF} = \sum_0 i = 1nF_i. \tag{10}$$

Roulette-Wheel is marked by the chromosomes present in current population and based on their fitness values. In other words, the selection probability of each of the chromosomes of the current generation population would be equal to:

$$p_i = F_i \sum ni = 1F_i p_i = F_i \sum i = 1nF_i. \tag{11}$$

There have been 14 variables proposed through the GA in Table 3. They are supposed as chromosomes in the GA. Furthermore, variables are written in parentheses for each chromosome and represented in Table 3.

In this problem, each chromosome has 14 genes which are in fact the values for ITS, RT, CLT, DST, MT, CT, LAC, AT, MO, TCS, TCM, STL, TLF and PA represented in Table 3. Every chromosome gene selected has ITS (Installation time depends on the sequence and machine), RT (Rest time), CLT (Cleaning time), DST (Device setting time), MT (Moving time), CT (Control time), LAC (Limited access to the car), AT (Access time), MO (Molding), TCS (Time to combine the color of shoes), TCM (Time to cut leather for making posts), STL (Sewing time on leather), TLF (Leather loading time for final shaping of shoes) and PA (Packaging).

Table 3. Scheduling process chromosomes

ITS	RT	CLT	DST	MT	CT	LAC	AT	MO	TCS	TCM	STL	TLF	PA

The Table 4 shows Roulette-Wheel selection operator for each of the living beings or chromosomes present in the population. As it can be observed, the chromosomes' fitness is different for each one. Since the third chromosome has a higher fitness than other chromosomes in the population, it can be expected that selection operator (reproduction) based on roulette-wheel can choose these chromosomes more than other chromosomes and put them into the (mating pool) tank.

Table 4. Average fitness for chromosomes' population

Population	Fitness
ITS	5
TR	8
CLT	15
DST	9

End of Table 4

Population	Fitness
MT	5
CT	7
LAC	8
AT	6
MO	10
TCS	6
TCM	3
STL	6
TLF	7
PA	3

Considering the population in Table 5 select nn living beings or chromosomes from among the current generation population, a number of nn random numbers between zero and one will be produced. Then, for each of the random values produced, the following conditions will be checked:

If the random value produced is less than the integrative probability of the n^{th} chromosome, chromosome n will be selected. For example, if the random value produced is less than the integrative probability of the first chromosome (0.05), chromosome 1 will be selected. If not, the next condition will be checked and shown in Table 5.

Table 5. Average fitness for chromo somes' population

Population	Fitness	Probability Choice	Probability Cumulative
1	5	0.05	0.05
2	8	0.08	0.13
3	15	0.15	0.28
4	9	0.09	0.37
5	5	0.05	0.42
6	7	0.07	0.49
7	8	0.08	0.57
8	6	0.06	0.65
9	10	0.10	0.75
10	6	0.06	0.81
11	3	0.03	0.84
12	6	0.06	0.90
13	7	0.07	0.97
14	3	0.03	1.00

Through investigating the GA method, the choice of different initial random populations and influencing the phenomenon (which are referred to as training data) for the purpose of learning the nature of the governing mechanism of the phenomenon, it was found that it is a complex model and this can increase the involved memory. To identify input variables, the GA is utilized. Based on Table 6, the MSE error value for learning, testing and validation data is equal to 0.084, 0.082 and 0.0031, respectively, measured through the square root of these numbers and divided by 2 (because the normalization in the interval of 2 units, from –1 to 1 was done) and multiplied by the calculated error percentage and the resulting percentages are equal to 14.491, 14.317 and 19.487, respectively.

Table 6. Data related to genetic and thier predictove accuracy

	MSE	RMSE	NMSE	R2
Learning data	0.084	0.041	0.487	0.91
Test data	0.082	0. 338	0.0418	0.952
Validation data	0.0031	0.0859	0.146	0.879

5.2. Neural network

In this section, investigating the statistical characteristics of research variables and diagnosis tests of these models conducted. Both estimation and neural networks calculated.

5.2.1. Teaching algorithms

Levenburg-Marquat teaching algorithm is one of the teaching algorithms to make weights in ANNs in time considered as one of the most commonly utilized algorithms since they teach network rapidly and minimize current level of error. Of course, the mentioned algorithm can be used to enhance learning pace of the network designed according to the matrix of Hason.

Considering the data in Table 7, from 14 input data, 10 equals to 83.3% were trained using the software and 2 of them equal to 16.7% was utilized to test the prediction.

Table 7. Data taught

Percentage	Number			
83.3%	10		Teaching	Sample
16.7%	2		Test	
100%	12		Valid	
	2		Deleted	
	14		Total Amount	
	14		Total Amount	

Table 8. Neural network data

Raw production times	1	Factors	Input layer
14	Number of units		
8	Number of units		Hidden layers
Maximum	Activation function		
Total production times	1	Dependent variable	Output layer
1	Number of units		
Normal	Criterion dependent method		
Recognition	Activation function		
Sum of squares	Error function		

Based on the data in Table 8, 14 layers were considered as the input layers (nodes) and 8 layers were calculated as hidden layers with maximum activation function along with sum of error square's function and as shown in Table 9, the sum of error squares in teaching stage was equal to 2.198 and the calculated relative error in teaching stage was equal to 0. 84.

Table 9. Results obtained in the training stage

Training	Sum of square error	2.198
Test	Relative error	0.469
	Training duration	671.10:00:00
	Mean squared error	0.582
	Relative error	0.840

5.2.2. Neural networks and network teaching analysis stages

5.2.2.1. Simple ANN

The method mentioned above was used to forecast the next day of the work completion in shoe manufacturing company. To predict the work completion time of the next day the data of *t*, *t*–1, *t*–1, *t*–1, *t*–1, *t*–1, *t*–1 for each variable was utilized. T is the day the prediction is being carried out. The following graphs represent the outputs of simple neural networks (see Figure 8). It should be noted that the blue lines show the prediction and the red lines represent the real trend. The red circles represent the real data amounts and the blue squares show the predicted amounts.

5.2.2.2. Studying the generalizability of neural networks

In order to assess the generalizability of the network, the network results against data not experienced were utilized. The tests done in this stage were investigated to predict 30 forthcoming days. The work's finish time in the firm was identified using 24 timing data of these variables through which an ANN was designed equal to 6×24 or 144 inputs. Also, there were 10 nodes in hidden layer and the exit layer utilized for this purpose. Considering the accessibility of 10 tests, the results of simulating the best neural network model were compared with experimental results. Based on the previous part, MLP1 network model (with one input layer, one hidden layer including 8 nodes, and an exit layer) and the structure in input layer entailing data *t* to *t*–7, *t*–15, *t*–30, *t*–60, and *t*–150 were selected as the most appropriate model for the variables mentioned above. The graph resulted from such a comparison is represented in Figure 6. As it can be observed, the network has had an acceptable performance in simulations unlike the data not experienced. The 30 days for work finish prediction has been considered as the most important output of the test in this neural network in the presence of an acceptable error level. Therefore, the neural network model has had an acceptable success in interpretation of the test of work finish in the firm.

As it can be observed, all circles representing the data have been lied on the blue line and this shows that there exist low errors and there is a good learning. This figure shows how MSE becomes as repetition in each of the set's increases. This means that as the number of opaque increases, the error decreases and over fit state occurs. Therefore, the network becomes fit on learning data. In the set for validation, first the error becomes more and then it decreases and this occurs in the same form within the test set too. The amount of MSE for learning data, test data and validation data were equal to 1.8241 *e*–007, 0.0195, and 0.0075, respectively and shown in Table 10.

Table 10. Prediction precision data using a simple ANN in 3 stages

	MSE	RMSE	NMSE	R2
Learning data	1.8241e-007	0.00043	5.3808e-007	1
Test data	0.0195	0.13964	0.04	0.96
Validation data	0.0075	0.08660	0.02154	0.97846

5.2.2.3. Reflexive neural network

In this method 3 data clusters were tested to predict 30 forthcoming days of work completion time in company which were teaching data, test data, and validation data of a day ending in 30th of October 2020 and the learning data comprised 70 percent while the rest of 30 percent remaining data were test data. To predict using this method different number of layers along with the variety of the node number were tested and the best structure was found to be a 3 layers' structure (a hidden layer, one input layer, and a single output layer). In such a method, the data were normalized within a range of –1 to 1 and a tansig function was utilized and presented in Figure 6.

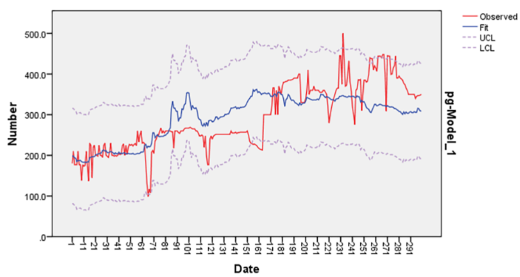

Figure 6. Prediction graph for validation data (source: authors, 2024)

The researcher tested several repetitions for this trend and the best observation result for one repetition of this trend was recognized. After learning stage ended, the researcher substituted the mean squares of errors of standard error in learning stage for all test and validation data as input data. Results of several tests on nodes' number in each hidden layer and hidden layers' number showed that the best result was gained by a three layer structure (an input layer, a hidden layer including 8 nodes, an output layer) and this structure in the input layer entailed data for t to t–7, t–15, t–30, t–60, t–150 for the variables mentioned above (in this research 2 variables were investigated as the variables affecting neural network and to predict the next 30 days, the work completion time in company utilized 24 time data of these variables through which the designed ANN was equal to 6×24, that is 144 inputs). Also, there were 10 nodes in the hidden layer and in the output layer there was an output which represented the prediction for 30 next days to complete the work in company. In this method, the identification coefficient obtained from the validation stage was equal to 0.96699 and presented in Table 11.

Table 11. Standard error mean squares of the error in the model

Model name	Square mean of standard error
The designed neural network	0.96699

Regarding the amounts of standard error mean square amounts in Table 11, it could be stated that the designed model has been approved through the model proposed, there has been a better prediction and the maximum time required to complete the work using neural network was estimated to be equal to 0.96699 and it has had an optimal estimation in achieving the predetermined goals in Table 11. Li et al. (2020) investigated about how to solve OSS using graph attention neural network. They proposed that OSS minimizing makespan has been commonly utilized a lot. Complex limitations and large solution space result in great difficulty for the achievement of optimal solutions.

5.3. Neural network learning with GA

In this method three sets of tests are used to predict the next day when the work is finished in the company as follows: learning data, test data, validation data. The validation data are gained from the day ending on 30th October 2020 and entail 70% test data and 30% the rest of data (an input layer, a hidden layer, and an output layer). In this method, they were normalized in the range –1 to 1 and the tansig activation function was used. In this method, the number of generations is 4 and the population is 10 and in order to be in the local minimum error, random generation is used along with the natural method. The Figure 7 shows the abnormal outputs with the help of the algorithm network. It should be noted that the blue line represents the forecast and the red lines show the actual trend.

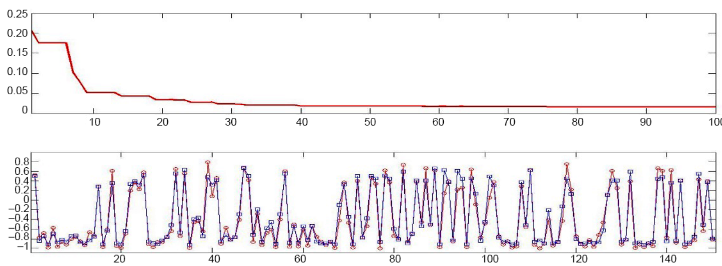

Figure 7. Phase output with-genetic neural network (source: authors, 2024)

As it can be seen in the Figure 8, there is some error in the forecast. To better understand this error, the following figure shows the errors in the event.

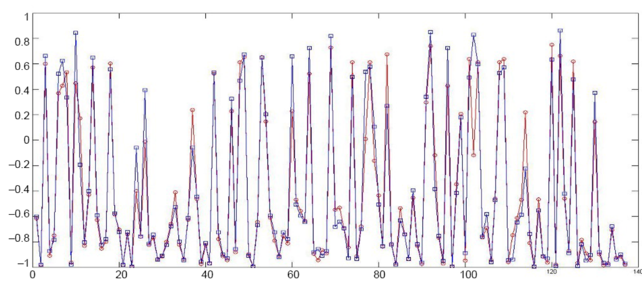

Figure 8. Test stage output with simple genetic neural network (source: authors, 2024)

According to the Figure 9, it can be seen that some data are around the blue line, which the greater the distance, the more error indicates. Regarding the test data, the forecast chart is as follows. As can be seen in the figure above, there is some error in the forecast. To better understand this error, the following figure shows the errors of the event.

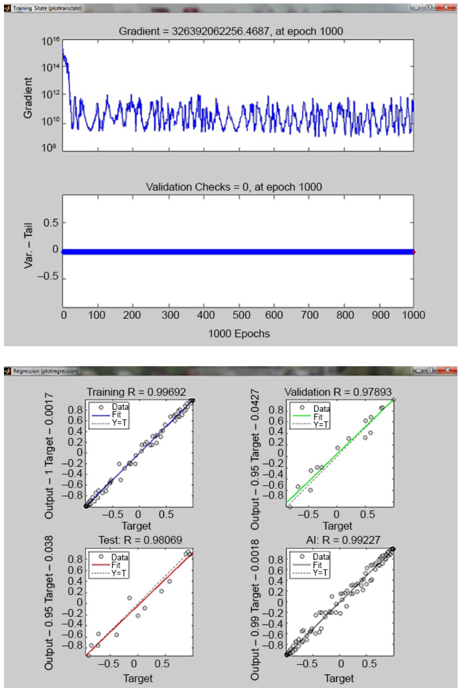

Figure 9. Learning stage output with genetic neural network displays predictive differences with reality (source: authors, 2024)

In Figure 10, it could be obviously seen that the more congruence on blue line means more congruence and regularity of the data. The farther distance represents more scatter and increase of errors and when the congruence is greater, the optimal solution will be approached. According to the Figure10, it can be seen that some data know a large distance from the blue line, which the greater the distance, the more error indicates. For validation data, the forecast chart is shown in Figure 11. Also, the Figure 12 shows the errors in predicting validation data

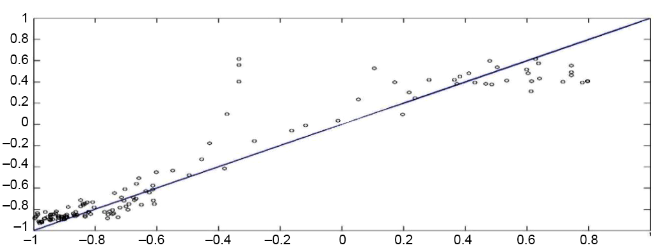

Figure 10. The output of the test phase with the artificial-genetic neural network shows the predicted difference with reality (source: authors, 2024)

There have been 14 activities utilized based on the structure of neural networks for training and testing the variables from among 20–80 formats proposed which were mentioned in the present study, and they consisted of 10 activities for training, 2 activities for modeling

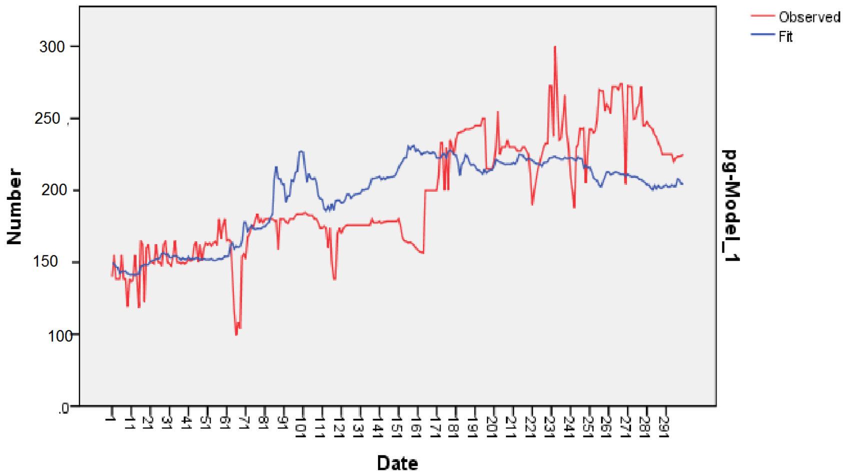

Figure 11. Validation stage output with genetic neural network (source: authors, 2024)

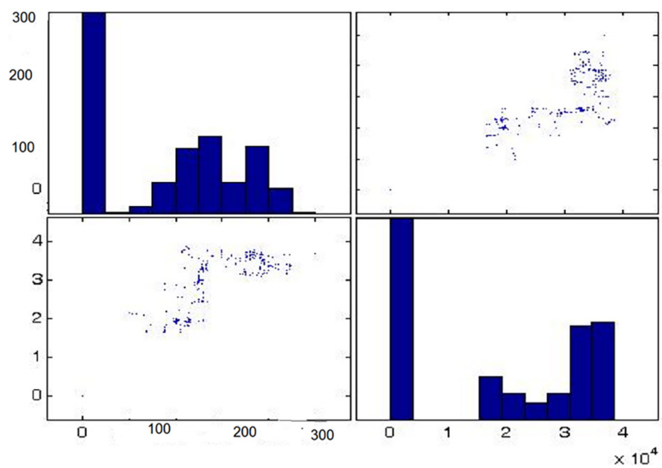

Figure 12. The output of the validation stage with genetic neural network displays the predicted difference with reality (source: authors, 2024)

research and 2 activities for testing, which are equal to the variables proposed in Table 3. Furthermore, in neural networks for data validation, SPSS software is needed to calculate MSE error rate in relation with learning data, test data and validation data. The outcomes of the comparison of GA and neural network are represented in Table 12. The results showed that the amount of MSE error for learning data, test data, and validation data are equal to 1.8241e-007, 0.0195 and 0.0031, respectively.

These measures were achieved using the second root of the related numbers divided by 2 (since normalization was done within the range of 2 units, from –1 to 1) and then they were multiplied by the percentage of error and these percentages were equal to 67.529, 14.317 and 2.783, respectively. It could be concluded that the amounts gained were good error amounts and it can be easily generalizing the neural network outcomes.

Table 12. A comparison of GA and neural network Methods (source: authors, 2024)

		R^2	NMSE	RMSE	MSE
Learning data	GA	0.91	0.487	0.041	0.084
	ANN	1	5.3808e-007	0.00043	1.8241e-007
Test data	GA	0.952	0.0418	0.338	0.082
	ANN	0.96	0.04	0.13964	0.0195
Validation data	GA	0.879	0.146	0.0859	0.0031
	ANN	0.97846	0.02154	0.08660	0.0075

5.4. Parameter for algorithm consistency

Previous discussions have indicated that a higher degree of similarity in results from consecutive algorithm implementations contributes to greater consistency. To evaluate this criterion, the algorithm underwent ten separate implementations. The variance in the final solutions produced by the algorithm was assessed across these ten different execution times. For enhanced clarity and comparability, the analysis utilized normalized data variance. This normalization ensures that the variance remains within a range of 0 to 1, with values closer to 0 indicating a more consistent algorithm. The calculated variance of the normalized solutions across the ten neural network implementations was found to be 0.96699, suggesting a nearly satisfactory level of algorithmic consistency. Regarding what was gained through the data in Table 11, the MSE values of the neural network are good error values in comparison with the GA. Furthermore, the results gained can be generalized to the neural network and it could also be claimed that these values are better than the values of the GA. Additionally, the goal of the present study is gaining a comparison of the GA and the neural network considering the minimization and the maximum work completion time within the shoe production lines, and thus it can conclude that the neural network has more accurate predictions than the GA, and the proof for such a belief has been provided through different sections of the current study.

5.5. Managerial Implication

According to the results obtained from the fourteen activities selected in this work, management applications for the managers of the shoe manufacturing company were given as follows:

In order to minimize the time to complete the work, the operators should be given new practical training for the time of installation and arrangement of materials and parts to the machines in the production lines. This practice was recommended for the company by making the operators agile, making the operators flexible to minimize the loading of materials on the production machines, observing the man-machine ergonomics to match the operators with the devices in the production lines. In order to rest the production equipment, the loading and unloading time of the production equipment must be in accordance with the exact schedule provided by the production planning unit and industrial engineers, according to the detailed information about the working time of the production equipment and the capabilities of the operators in the production lines. Also, for the accurate repair and maintenance of the production equipment, the maintenance and repair unit should give a detailed schedule for the accurate repair and restoration of the production equipment so that the downtime of the equipment is not too long. The cleaning time of the devices in the production lines should be done according to a fixed and flexible program in the production lines. Because this action increases the lifespan of the production equipment. Also, during cleaning, the repair or replacement of parts in case of failure is seen better, and the replacement of defective parts is done in the shortest possible time, and the increase in waste produced by the equipment is prevented. The setting time of production devices in production lines, firstly, the time of loading and unloading of parts should be done in accordance with the relevant catalog from the manufacturer of the devices, secondly, in case of case modifications for the repair and maintenance of the devices to avoid the loss of stoppage of the production lines, the time of setting the devices is one of the influential factors for the company. The movement of machines between the production lines should be done without any interruption and obstacles. To do this, it is necessary to observe the systematic layout planning of the arrangement of devices in production lines. In addition, the use of conveyor belts for better movement and transfer of materials between production lines is effective in reducing waste time for materials and semi-finished parts to reach the machines. Access time refers to the timely production of shoes according to the codified program by the production planning unit. Because in this program, the production method of the entire shoe production process is described separately by production units. Also, in order to reach the access time for shoe production in the production lines, the quality control must be controlled by operators in order to prevent more production waste. In addition, randomly and without the knowledge of operators, the employees of the quality control department to control the quality of shoes production in every device to control the production of shoes at every stage. Molding is one of the most important factors for the production of any product in all factories. Because the precise modeling of the work process is designed and built by mechanical engineers in each production unit. The cost of each mold is much higher and is considered as one of the strategic processes of the manufacturing company. Therefore, the company should be more sensitive in the molding process. The color mixing time of shoes is considered one of the most basic tasks in shoe manufacturing companies. If the color mixing time is not done correctly when dyeing the shoes produced by the dyeing machine, the desired color will not be sprayed correctly on the shoes and this This will cause the dyeing machines to shut down for corrective measures and to find the cause of the correct dyeing. This will also increase the time to complete the dyeing process on the manufactured shoes, and on the other hand, it will increase the waiting time for the dye on the manufactured shoes. Therefore, it is necessary to use experts for the correct combination of colors on shoes. The leather cutting time for making the posts should be accurately given by the shoe production planning unit to the leather cutting department for making the posts. For this, it should be cut according to the designed pattern. This step is one of the basic steps in cutting on embroidered posts. Because a clean and neat cut will increase the quality of the desired posts. It is better to spend the appropriate time for cutting with great care. Because after preparing the desired product, the cutting quality can be measured from its sides. The sewing time on leather should also be taken into account by the shoe production planning unit. Since the accuracy of the person in arranging the desired pattern on the cutable leather is limited to the natural capabilities of the human senses, usually some of the leather is wasted during cutting, which is undesirable for the manufacturers due to the expensiveness of the processed leather. Another result of using the traditional cutting system is low cutting speed. Since cut leather is the input of other activity units in the factory, the cutting stage is considered a bottleneck in the factory production process. Making a fully automatic leather cutting machine is a technological solution for the above problems. Also, with the use of these devices, the loading time of the leather for the final shaping of the shoe is reduced, and it is possible to better produce it with better quality. Finally, the packaging is considered as the most important feature of the product. The transformation of market dynamics on a global scale has compelled manufacturers and packaging designers to concentrate on specific consumer packaging sectors, including shoe packaging. To enhance both the efficacy and appeal of their products, manufacturers of shoe packaging are increasingly attentive to the preferences and requirements of consumers. By purchasing shoe wraps, shoes are protected from moisture, dust, damage, etc. The ability to print on this carton establishes an effective connection between the consumer and the business brand to increase the overall value of the customer. Corrugated cartons have long served as the primary packaging solution for footwear. Nevertheless, shoe manufacturers often incorporate various layers of primary packaging, including plastic covers, lightweight protective wraps, and bubble packs, to enhance the durability and protective qualities of shoe packaging. Therefore, the company was advised to use plastic packaging to prevent humidity.

6. Conclusions

The shop flow scheduling problem involves determining the order of operations within a manufacturing system. Given that effective scheduling can significantly reduce a company's costs, the model presented in this study is an integer programming framework designed to address OSS across multiple machines, with the objective of minimizing the maximum time required to complete tasks. This is achieved through the application of ANN and GA. The constraints considered include various operational factors such as installation time, which varies based on the sequence and machine used, as well as rest periods, cleaning durations, device setup times, transportation times, control intervals, limited vehicle access, access times, molding processes, color combination times for shoes, leather cutting durations for post production, sewing times for leather, leather loading times for final shoe shaping, and packaging times.

The findings of this research indicate that the variance of the normalized solutions across ten implementations of the neural networks was calculated to be 0.96699, suggesting a high level of consistency within the algorithm. Additionally, the fitness probability value for the third population in the GA was determined to be 0.15. Notably, the neural network demonstrated superior performance compared to the GA, as evidenced by its value of 0.96699. Consequently, the neural network has been employed to compare and predict the minimum time necessary for task completion within the company.

Every research is formed in a space of limitations, which inevitably affects a part of the authors' ability, and this will show itself over time. In this work, despite spending time on the part of the authors, the problems caused by this limitation, there have been limitations as follows. The most important limitation is related to the subject matter of the article. Solving the OSS using the combined method of GA and neural networks in the company under study is considered. This has caused the authors to face a challenge due to the traditional administration in different departments of the company under study. The second and perhaps one of the most important limitations of the authors was caused by the data collection tools. Due to the fact that in the current research, the authors have used the questionnaire tool and it has its own problems and inadequacies, the authors faced limitations in collecting and using the results in the company under study. Considering that the studied community included: industrial engineers, production planning managers, production planning department and quality control managers in shoe production and considering the prevailing atmosphere in the authors' company in collecting the data, especially from experts It was faced with many limitations and actually wasted a large part of the writers' times. Future works can investigate the resolution for OSS using different machines using meta-heuristic algorithms such as SA and TS.

Disclosure statement

The authors did not report any potential conflict of interest.

References

Abreu, L. R., Cunha, J. O., Prata, B. A., & Framinan, J. M. (2020). A genetic algorithm for scheduling open shops with sequence-dependent setup times. *Computers & Operations Research*, *113*, Article 104793. https://doi.org/10.1016/j.cor.2019.104793

Ahmadian, M. M., Khatami, M., Salehipour, A., & Cheng, T. C. E. (2021). Four decades of research on the open shop scheduling problem to minimize makespan. *European Journal of Operational Research*, *295*(2), 399–426. https://doi.org/10.1016/j.ejor.2021.03.026

Allen, F., & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Economics*, *51*(2), 245–271. https://doi.org/10.1016/S0304-405X(98)00052-X

Alwani, S. M., & Hosseinpour, D. (2016). Application of artificial neural networks in strategic decision-making. *Management Studies Quarterly* (Improvement and Transformation), *18*(4), 1–28.

Anand, E., & Panneerselvam, R. (2015). Literature review of open shop scheduling problems. Intelligent Information Management, *7*, 33–52. https://doi.org/10.4236/iim.2015.71004

Anderson, E. J., & Potts, C. N. (2002). On-line scheduling of a single machine to minimize total weighted completion time. *Mathematics of Operations Research*, *29*(3), 548–557. https://doi.org/10.1287/moor.1040.0092

Anders, C. R., Albarracín, J. M., & Tormo, G. (2005). Group technology in a hybrid flow shop environment: A case study. *European Journal of Operational Research*, *167*, 272–281. https://doi.org/10.1016/j.ejor.2004.03.026

Arroyo, J. E. C., dos Santos Ottoni, R., & de Paiva Oliveira, A. (2011). Multi-objective variable neighborhood search algorithms for a single machine scheduling problem with distinct due windows. *Electronic Notes in Theoretical Computer Science*, *281*, 5–19. https://doi.org/10.1016/j.entcs.2011.11.022

Azer, A., & Rajabzadeh, A. (2012). Evaluation of hybrid forecasting methods: With classical neural network approaches in the field of economics. *Journal of Economic Research*, *63*, 87–14.

Baykasoğlu, A., Madenoglu, F. S., & Hamzadayi, A. (2020). Greedy randomized adaptive search for dynamic flexible job-shop scheduling. *Journal of Manufacturing Systems*, *56*, 425–451. https://doi.org/10.1016/j.jmsy.2020.06.005

Baykasoğlu, A., & Ozsoydan, F. B. (2018). Dynamic scheduling of parallel heat treatment furnaces: A case study at a manufacturing system. *Journal of Manufacturing Systems*, *46*, 152–162. https://doi.org/10.1016/j.jmsy.2017.12.005

Behnamian, J., Fatemi Ghomi, S. M. T., & Zandieh, M. (2010). Development of a hybrid metaheuristic to minimise earliness and tardiness in a hybrid flow shop with sequence dependent setup times. *International Journal of Production Research*, *48*(5), 1415–1438. https://doi.org/10.1080/00207540802556817

Bello, I., Pham, H., Le, Q. V., Norouzi, M. & Bengio, S. (2016). *Neural combinatorial optimization with reinforcement learning*. arXiv. https://doi.org/10.48550/arXiv.1611.09940

Benavides, A. J. (2018). *A new tiebreaker in the NEH heuristic for the permutation flow shop scheduling problem*. (No 440). EasyChair preprint. https://doi.org/10.29007/ch1l

Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. (2007). *Handbook on scheduling: From theory to applications*. Springer.

Caicedo-Rolón, A. J., & Parra Llanos, J. W. (2021). Production sequencing in a flow shop system using optimization and heuristic algorithms. *Gestão & Produção*, *28*(1), Article e3886. https://doi.org/10.1590/1806-9649.2020v28e3886

Chen, B., & Strusevich, V. A. (1993). Approximation algorithms for three machine open shop scheduling. *Informs Journal on Computing*, *5*(3), 321–326. https://doi.org/10.1287/ijoc.5.3.321

Coelho, J., & Vanhoucke, M. (2018). An exact composite lower bound strategy for the resource-constrained project scheduling problem. *Computers & Operations Research*, *93*, 135–150. https://doi.org/10.1016/j.cor.2018.01.017

Colak, S., & Agarwal, A. (2005). Non-greedy heuristics and augmented neural networks for the open-shop scheduling problem. *Naval Research Logistics*, *52*(7), 631–644. https://doi.org/10.1002/nav.20102

Daniels, R. L., Mazzola, J. B. & Shi, D. (2004). Flow shop scheduling with partial resource flexibility. *Management Science*, *50*(5), 658–669. https://doi.org/10.1287/mnsc.1040.0209

Defersha, F. M., & Chen, M. (2010). A parallel genetic algorithm for a keller problem with sequence dependent setups. *International Journal of Advanced Manufacturing Technology*, *49*, 263–279. https://doi.org/10.1016/j.ejor.2011.01.011

Doulabi, S. H. H. (2010). A mixed integer linear formulation for the open shop earliness-tardiness scheduling problem. *Applied Mathematical Sciences*, *4*, 1703–1710.

Fekri, M., Heydari, M., & Mahdavi, M. (2024). Bi-objective optimization of flexible flow shop scheduling problem with multi-skilled human resources. *Engineering Applications of Artificial Intelligence*, 133(Part C), Article 108094. https://doi.org/10.1016/j.engappai.2024.108094

Fernandez-Viagas, V., Ruiz, R., & Framinan, J. M. (2017). A new vision of approximate methods for the permutation flow shop to minimize makespan: state-of-the-art and computational evaluation. *European Journal of Operational Research*, *257*(3), 707–721. https://doi.org/10.1016/j.ejor.2016.09.055

Gautam, V. K., Pande, C. B., Moharir, K. N., Varade, A. M., Rane, N. L., Egbueri, J. C., & Alshehri, F. (2023). Prediction of sodium hazard of irrigation purpose using artificial neural network modelling. *Sustainability*, *15*, Article 7593. https://doi.org/10.3390/su15097593

Gonzalez, T., & Sahni, S. (1976). Open shop scheduling to minimize finish time. *Journal of the ACM*, *23*(4), 665–679. https://doi.org/10.1145/321978.321985

Gueret, C., & Prins, C. (1998). Classical and new heuristics for the open-shop problem: A computational evaluation. *European Journal of Operational Research*, *107*(2), 306–314. https://doi.org/10.1016/S0377-2217(97)00332-9

Harmanani, H. M., & Ghosn, S. B. (2016). An efficient method for the open-shop scheduling problem using simulated annealing. In S. Latifi (Eds.), *Advances in intelligent systems and computing: Vol. 448. Information technology: New generations.* Springer, Cham. https://doi.org/10.1007/978-3-319-32467-8_102

Iba, H., & Sasaki, T. (1999). Using genetic programming to predict financial data. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)* (Vol. 1, pp. 244–251). IEEE. https://doi.org/10.1109/CEC.1999.781932

Iqbali, H., & Moghadspour, A. (2017). *The use of artificial neural networks in business management decision-making* [Conference presentation]. The second international conference on management and fuzzy systems.

Ji, M., Yao, D. L., Yang, Q. Y., & Cheng, T. C. E. (2015). Single-machine common flow allowance scheduling with aging effect, resource allocation, and a rate-modifying activity. *International Transactions in Operational Research*, *22*(6), 997–1015. https://doi.org/10.1111/itor.12121

Lee, T. S., & Loong, Y. T. (2019). A review of scheduling problem and resolution methods in flexible flow shop. *International Journal of Industrial Engineering Computations*, *10*(1), 67–88. https://doi.org/10.5267/j.ijiec.2018.4.001

Liaw, C. F. (1999). A tabu search algorithm for the open shop scheduling problem. *Computers & Operations Research*, *26*(2), 109–126. https://doi.org/10.1016/S0305-0548(98)00056-2

Li, J., Dong, X. Y., Zhang, K., & Han, S. (2020). Solving open shop scheduling problem via graph attention neural network. In *IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 277–284). IEEE. https://doi.org/10.1109/ICTAI50040.2020.00052

Li, X. Y., Gao, L., Pan, Q. K., Wan, L., & Chao, K. M. (2018). An effective hybrid genetic algorithm and variable neighborhood search for integrated process planning and scheduling in a packaging machine workshop. *IEEE Transactions on Systems Man Cybernetics Systems*, *49*(10), 1933–1945. https://doi.org/10.1109/TSMC.2018.2881686

Lin, H. T., Lee, H. T., & Pan W. J. (2008). Heuristics for scheduling in a no-wait open shop with movable dedicated machines. *International Journal of Production Economics*, *111*(2), 368–377. https://doi.org/10.1016/j.ijpe.2007.01.005

Liu, C. L., & Xiong, C. H. (2021). Single machine resource allocation scheduling problems with deterioration effect and general positional effect. *Mathematical Biosciences and Engineering*, *18*(3), 2562–2578. https://doi.org/10.3934/mbe.2021130

McCulloch, W. W., & Pitts, W. (1943). A logical calculus of ideas imminent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*, 115–133. https://doi.org/10.1007/BF02478259

Ma, R., Guo, S. N., & Miao, C. X. (2021). A semi-online algorithm and its competitive analysis for parallel-machine scheduling problem with rejection. *Applied Mathematics and Computation*, *392*, Article 125670. https://doi.org/10.1016/j.amc.2020.125670

Minhaj, M. B. (2017). *Neural networks*. Islamic Azad University Publications.

Minsky, M., & Papert, S. (1969). Review of "Perceptrons: An Introduction to Computational Geometry" (Minsky, M., and Papert, S.; 1969). *IEEE Transactions on Information Theory*, *15*(6), 738–739. https://doi.org/10.1109/TIT.1969.1054388

Mousighichi, K., & Avci, M. G. (2024). The distributed no-idle permutation flow shop scheduling problem with due windows. *Computational & Applied Mathematics*, *43*, Article 179. https://doi.org/10.1007/s40314-024-02702-w

Naderi, B., Ghomi, S. M. T. F., Aminnayeri, M., & Zandieh, M. (2010). A contribution and new heuristics for open shop scheduling. *Computers & Operations Research*, *37*(1), 213–221. https://doi.org/10.1016/j.cor.2009.04.010

Noori-Darvish, S., Mahdavi, I., & Mahdavi-Amiri, N. (2012). A bi-objective possibilistic programming model for open shop scheduling problems with sequence-dependent setup times, fuzzy processing times, and fuzzy due dates. *Applied Sof Computing*, *12*, 1399–1416. https://doi.org/10.1016/j.asoc.2011.11.019

Nemati, K., Refahi, S., Kord Roostemi, S., & Amir, H. (2016). Optimization of parallel algorithms scheduling using genetic algorithms. *Journal of Operational Research and its Applications*, *13*(2), 35–52.

Osorio Gómez, J. C., Castrillón Montenegro, O. E., Toro Cardona, J. A., & Orejuela Cabrera, J. P. (2008). Modelo de programación jerárquica de la producción en un Job shop flexible con interrupciones y tiempos de alistamiento dependientes de la secuencia. *Revista Ingeniería e Investigación*, *28*(2), 72–79. https://doi.org/10.15446/ing.investig.v28n2.14896

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, *12*(4), 417–431. https://doi.org/10.1007/s10951-008-0090-8

Panneerselvam, R. (1999). Heuristic for moderated job shop scheduling problem to minimize makespan. *Industrial Engineering Journal*, *28*, 26–29.

Pinedo, M. L. (2022). *Scheduling theory, algorithms, and systems*. Springer. https://doi.org/10.1007/978-3-031-05921-6

Rimcharoen. S., Sutivong, D., & Chongstitvatana, P. (2005). Soft computing in the forecasting of the stock exchange of Thailand. In *Proceedings of the fourth IEE international conference on management of innovation and technology*. IEEE. https://doi.org/10.1109/ICMIT.2008.4654554

Sheykhalishahi, M., Eskandari, N., Mashayekhi, A., & Azadeh, A. (2019). Multi-objective open shop scheduling by considering human error and preventive maintenance. *Applied Mathematical Modelling*, *67*, 573–587. https://doi.org/10.1016/j.apm.2018.11.015

Shioura, A., Shakhlevich, N. V., & Strusevich, V. A. (2018). Preemptive models of scheduling with controllable processing times and of scheduling with imprecise computation: A review of solution approaches. *European Journal of Operational Research*, *266*(3), 795–818. https://doi.org/10.1016/j.ejor.2017.08.034

Tavakkoli-Moghaddam, R., & Seraj, O. (2009). A tabu search method for a new bi-objective open shop scheduling problem by a fuzzy multi-objective decision making approach (research note). *International Journal of Engineering, Transactions B: Applications*, *22*(3), 269–282.

Xue, H., Meng, L., Duan, P., Zhang, B., Zou,W., & Sang, H. (2024). Modeling and optimization of the hybrid flow shop scheduling problem with sequence-dependent setup times. *International Journal of Industrial Engineering Computations*, *15*(2), 473–490. https://doi.org/10.5267/j.ijiec.2024.1.001

Wang, H. B., & Alidaee, B. (2019). Effective heuristic for large-scale unrelated parallel machines scheduling problems. *Omega*, *83*, 261–274. https://doi.org/10.1016/j.omega.2018.07.005

Wang, X., Ren, T., Wang, X., Bai, D., Chu, F., Lu, X., Weng, Z., Li, J., & Li, J. (2024). Hybrid flow shop scheduling with learning effects and release dates to minimize the makespan. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *54*(1), 365–378. https://doi.org/10.1109/TSMC.2023.3305089

Yao, J. T., & Tan, C. L. (2001, November 14–18). Guidelines for financial forecasting with neural networks. In *Proceedings of International Conference on Neural Information Processing* (pp. 757–761), Shanghai, China.

Zang, Z., Wang, W., Song, Y., Lu, L., Li, W., Wang, Y., & Zhao, Y. (2019). Hybrid deep neural network scheduler for job-shop problem based on convolution two-dimensional transformation. *Computational Intelligence and Neuroscience*, *2019*, Article 172842. https://doi.org/10.1155/2019/7172842

Zhu, T., & Liu, G. (2023). A. Novel Hybrid methodology to study the risk management of prefabricated building supply chains: an outlook for sustainability. *Sustainability*, *15*, Article 361. https://doi.org/10.3390/su15010361