ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20220818161430
Original scientific paper

Development of a Neural Network Algorithm for Estimating the Makespan in Jobshop Production Scheduling

İncilay YILDIZ, Abdülvahap SAYGIN, Selçuk ÇOLAK, Fatih ABUT*

Abstract: Since production scheduling is considered a short-term plan for future production planning, the advantages of effective scheduling and control and their contribution to the production process are numerous. Efficient use of resources improves productivity and ensures that customer orders are met on time. Even the simplest scheduling system has a complex solution structure. Long lead times also make it difficult to estimate the demand accurately. Therefore, it is important to solve scheduling problems effectively for such difficult-to-manage production processes. Job shop scheduling (JSS) problems are among the combinatorial problems in the NP-hard problems class. As constraints increase in such problems, the solution space starts to go to infinity, making it increasingly difficult to find the exact optimum solution. For this reason, metaheuristic algorithms have been used to solve such problems in recent years. This study aims to develop an artificial neural network (ANN)-based application to produce an optimal or near-optimal solution for JSS. Using the job shop type production data of Taillard comparison problems, the total processing time (i.e., makespan) has been calculated with the proposed ANN application. The results have been compared with the results of related studies in the literature, and the algorithm's efficiency has been evaluated in detail.

Keywords: artificial neural networks; makespan; optimization; scheduling problems

1 INTRODUCTION

Production systems were first proposed in 1943 as a general computational mechanism. The system has shown great improvement in the literature and has been applied to diverse problems [1]. Production scheduling is determining which resources (machine, labor, material, etc.) should do the jobs to meet the production plan or the delivery dates of the orders. Production scheduling problems are among the combinatorial optimization problems in which limited resources are assigned to various tasks at certain times. Therefore, it is difficult to find optimal solutions for scheduling problems. Three elements are planned in production scheduling problems: delivery date, completion time, and machine usage order. Since the delivery date in production scheduling is important in terms of not delaying customer orders, it is aimed to minimize the production completion time by using the appropriate machinery, manpower, and equipment in the most effective way [2]. Optimization techniques are used to solve such problems. Thanks to these optimization techniques used in scheduling, companies can increase customer satisfaction and the efficiency of their resources, as well as create plans that minimize costs.

Processing a job on a resource is called an operation. The processing of job i on resource j is called O_{ij} operation. Scheduling can be defined as the task of discovering an entry and end time for each O_{ij}. Production Scheduling, on the other hand, is about finding the combination that makes a certain objective function optimal in relation to the extracted boundaries of the problem. The purpose of scheduling production systems is to minimize the waiting time in the production process or the total completion time of the existing works (i.e., makespan) by considering the constraints (maximum capacity of the facility, number of available machines, etc.) [3].

Optimization in production scheduling plays an important role in design, scheduling, and costing activities in businesses. It ensures efficient use of resources, saving time and increasing product quality. Especially in gigantic systems, optimization of dynamics with human effort is difficult, so such operations are performed with the help of computer programs by creating algorithms. In this study, an ANN algorithm is proposed to estimate the optimum total processing time of the job shop scheduling (JSS) problem.

The data set examined in this study includes the data of the JSS problem. Job shop type production systems consist of machines placed according to the product or job. Systems of this form are often used to produce small quantities of a wide variety of products. The job data entered into the system on order is transformed in appropriate machines in the appropriate part of the system according to the production purpose. Every product produced and every job to be done has a unique machine route. This route may vary depending on which machine is empty and which is full at the time determined in the system. In jobshop type production systems: (a) there is a finite set of n jobs, (b) there is a finite set of m machines, (c) each job consists of a chain of operations, (d) each machine can do at most one operation at a time; (e) each operation must be processed on a given machine for a certain length of uninterrupted time, (f) each job has a machine queue and a particular route, (g) the purpose is a schedule where the jobs processed on the machines have a minimum total time, taking into account the time intervals of the operations to create.

If the problem P (algorithm running in polynomial time) is an NP-class problem, then the P problem is called an NP-hard problem. Most combinatorial optimization problems fall under the class of NP-hard, polynomial time-boundless problems [4]. Regardless of the alternative ways to be decided in combinatorial optimization problems or the problem that is defined and sought to be solved, the general goal is to optimize the result. Therefore, the process of obtaining the optimum result in such problems is called optimization [5]. Since JSS problems are an NP-hard problem type, it is necessary to minimize the overall total time by providing as many restrictions as possible in order to create an objective program.

JSS problem is very important for both the industrial sectors. Companies that make good scheduling can save both money and time in terms of production. JSS is the problem of planning the operational process for a certain number of jobs on a specific number of machines, with each job having a specific route. In order to schedule such a system, it is necessary to create a model first. Once a machine, operator, or workstation completes an operation, a decision must be made about what to do next. This decision is made according to a scheduling rule. In this study, five scheduling methods were used: giving priority to the shortest processing time, giving priority to the longest processing time, first come first served, last come first served, and random assignment.

Upon examining the literature, it has been observed that heuristic methods are commonly employed to tackle the Taillard JSS problem. These methods include Particle Swarm Optimization (PSO), Taboo Search (TS), Annealing Simulation (SA), Path Relinking (PR), Genetic Algorithms (GA), Viable Neighbor Solution (VNS), and Artificial Immune Systems (AIS). PSO, which was introduced by Kennedy and Eberhart in 1995 [6], is inspired by the movement of bird and fish flocks in two-dimensional space. TS, which was coined by Glover in 1986, employs adaptive memory and deterministic probabilistic exploration to solve problems [7, 8]. The AIS method, which aims to develop an algorithm based on the principles of the biological immune system, was first proposed by Robert Shaw and Julie Miller in 1987 [9]. The SA algorithm, first used by Kirkpatrick et al. in 1983 [10], is capable of escaping local minima by transitioning to sub-solutions during the search process. PR is typically used in hybrid algorithms, and was first proposed by Carlos Cotta in 1996 [11, 12]. VNS method was first introduced by Mladenovic and Hansen in 1997. The basic idea of VNS is to start with an initial solution and then search for better solutions by applying different types of local search techniques [13]. GA, inspired by biological evolution and inheritance mechanisms, were first introduced by John Holland in the early 1970s [14].

In the literature, heuristic methods are generally used to develop hybrid methods that combine multiple techniques to address the limitations of individual methods. However, this can add complexity. In this study, we utilized a single method with the proposed Artificial Neural Network (ANN) algorithm, which led to optimal results.

In this article, the total completion time of the job has been tried to be minimized by using the data on the number of machines, the number of jobs, the completion times of the jobs on the machine and the machine queues with the ANN method, which was coded considering the JSS problem constraints. It is aimed to plan the most effective scheduling in production with optimal completion time. In order to measure the effectiveness of the ANN method developed in the study, the method was applied to Taillard's jobshop type scheduling data. The results obtained were compared with the studies using these data in the literature, and the effectiveness of the method was evaluated in detail. Within the scope of the study, the ANN algorithm was coded with the Visual Studio 2019 C# program to find the shortest total completion time (makespan) in scheduling 15 jobs and 15 machines, 20 jobs and 20 machines, 30 jobs and 15 machines in Taillard's data set. The results obtained using the job shop type scheduling data shared in Taillard's study were compared with those of academic publications.

The rest of the paper is organized as follows. Section 2 summarizes some important related studies. In Section 3, the methodology and the proposed ANN-based algorithm are presented. Results and discussion are presented in Section 4. Finally, in Section 5 the paper is concluded with possible future research directions.

2 RELATED WORKS

D. Y. Sha and C. Y. Hsu [8] conducted a study in 2006 that suggested the PSO method for JSS and used the TS method to improve the quality of this method. It was emphasized that a hybrid algorithm including PSO and TS methods is better than other traditional metaheuristic methods. This hybrid method, applied to Taillard test problems, basically produces solutions by being influenced by the first solution. The main purpose of the process in the PSO method is to provide good and varied initial solutions to the TS method. The results of the study showed that this hybrid method outperformed both TS and PSO methods and the mean difference was 0.356% lower than PSO. It is stated that the maximum calculation time of the hybrid method is between 3 and 103 seconds and 99% of the calculation time is spent on the local search process of the hybrid method. Since the computation time of particles applying a local search procedure could not be reduced by reducing the number of iterations.

Chandrasekaran et al. [9] conducted a study in 2007 proposing the computational AIS algorithm to minimize the total transaction completion time for the JSS problem. The proposed AIS has been shown to be an effective algorithm that gives better results than the TS shift routing procedure with the best derailleur occlusion procedure of Balas and Vazacopoulos. The efficient AIS method, based on the immune system principles, was used with two principles: The clonal selection principle and the affinity maturation principle. In the clonal selection principle, each program has a production value that expresses the affinity value of this program. The affinity value of each program is calculated from the affinity function. The affinity function is defined as 1/makespan. Two methods were used in the clonal selection principle: test mutation and double change mutation. When the findings obtained with the AIS are compared with the studies in the literature testing the same problems, it has been reported that the proposed AIS is a more effective problem-solving technique.

In 2008, Zhang et al. [10] proposed the TS method, which has been used as an effective algorithmic approach for the JSS problem in recent years. The main principle of the proposed hybrid approach is the use of SA to find outstanding solutions within the solution space. This hybrid algorithm has been tested on Taillard datasets and compared with other approaches. The computational results show that the proposed algorithm can obtain high-quality solutions in reasonable computation times. A robust diversification strategy using the SA procedure to find good solutions is equipped with core TS in the proposed hybrid algorithm. Core TS directs the concentrated search to explore other regions of the solution space. More precisely, starting from a random initial solution, the core of the hybrid algorithm executes the TS procedure, and good enough solutions found by simulated annealing are followed in the search procedure. These good solutions found by the SA procedure are stored in the elite solution stack *L*. When discovered, each new good solution is added on top of the solution stack *L*. Such solutions can then be removed from the *L* stack as new built-in solutions where a concentrated search is performed over a predetermined number of iterations. Appropriate temperature *T* should be considered, and solutions in the elite solution stack should not run out. The algorithm ends when it reaches the total number of iterations and the solution obtained is considered optimal.

M. M. Nasiri and F. Kianfar [11] found optimal makespan values for the scheduling problem by using a complex distribution search algorithm with TS and PR in their study in 2011. Distribution research is an evolutionary method that produces new solutions by systematically combining existing ones. In the first step of this method, an initial population is generated. In the second step, the reference set was created, the solutions of the 1st set containing the old low makespan b_1 solutions and the 2nd set containing the high diversity solutions were called the reference solutions. Subsets, each containing two reference solutions, were created, and a new solution was generated by combining the two solutions of each subset and using the path reconnection algorithm. The TS method is applied to the solutions, and the resulting solution is added to the population. Optimal results were obtained with this study using the Taillard dataset.

In 2015, B. Peng et al. [12] proposed a hybrid algorithm using TS and PR methods for the JSS problem. This algorithm has a number of distinctive features, such as a special mechanism reference solution for determining solutions as a distance-based path solution construction procedure. Also, this method is run continuously between a path reconnection method used to generate optimal solutions on the trajectory established from an initial solution to a guide solution and a TS procedure that improves the optimal solution produced. In the proposed hybrid method, the initial population is created as follows: Starting from zero, a randomly viable solution is generated, and then the solution is optimized using the optimization method. If the optimal solution is found or the best objective value for a given TS iteration has not been improved, the TS is concluded. The proposed hybrid method has been applied to Taillard's JSS dataset, and the results confirmed the effectiveness of the method.

In 2017, Bürgy [13] suggested the Viable neighbor solution (VNS) method that can be applied to the JSS problem. VNS is produced by removing a job from a given solution and relocating it to a neighbor location. In a sense, this neighbor generation is included in the TS method. A tabu list *L* is used to store entries for its last iterations. Initially, the *L* list is empty. In one iteration, an *S* value is selected from neighboring solutions. Whether or not this selected *S* value will be added to *L* is evaluated by making formulaic choices. Finally, a neighbor with the lowest objective value is selected. The concept of critical springs is used to generate neighbors that potentially improve the existing solution. For any selection *S*, the longest path logic to a node representing a process, with the goal of makespan, which is the last completion time of a job's production process, is called the critical path. Usually, only a transaction with the most recent completion time contributes to objective value, but some (subsets) transactions have been noted to determine objective value for a particular selection. It has been stated that the results obtained with the Variable Neighborhood Search (VNS) and TS hybrid method applied to Taillard's data set are effective.

In 2022, Jin Xie et al. [14] presented a hybrid algorithm called HA that combines Genetic Algorithms (GA) and Taboo Search (TS) to solve the JSS problem with a makespan criterion. GA is known for its excellent global search capability, while TS has strong local search capabilities. By combining their advantages, HA achieves a well-balanced approach between intensification and diversification. In the GA section, the paper introduces a crossover operator that incorporates a path reconnection procedure and a mutation operator based on critical paths. In the TS section, the paper adopts a previously proposed neighborhood structure, which effectively expands the search space of neighborhood solutions. To evaluate the optimization effect of their method, the authors selected 135 benchmark samples. The results confirm that the method outperforms the compared algorithms in terms of both computational efficiency and solution quality.

When the relevant studies in the literature are examined, it is observed that for the job shop type production scheduling problems, an additional method is used besides the ANN method, and a solution is sought by combining more than one method. In this study, unlike the literature, a single method, the artificial neural network algorithm, was used. With ANN, it is aimed to obtain near-optimal solutions with a simple algorithm in a short period of time, such as an average of one hour, with the need for fewer iterations compared to other studies. With this artificial neural network developed in the C# program, fast and near-optimal solutions have been obtained for scheduling the job shop type production system, especially for problems with less work and few machines.

3 METHODOLOGY AND THE PROPOSED ANN-BASED ALGORITHM

Taillard [15] addressed various production scheduling problem types and created a database of problem sets to facilitate comparisons among researchers. This dataset is publicly available on the web [16]. In this study, we employed the Visual Studio 2019 C# language and an ANN algorithm to find the shortest total completion time (makespan) for scheduling 15 jobs and 15 machines, 20 jobs and 20 machines, and 30 jobs and 15 machines in Taillard's dataset. We compared our results with those from academic publications that utilized the same workshop-type scheduling data as Taillard's study.

The proposed ANN model in this study is comprised of three layers. The input layer includes four headings: number of jobs, number of machines, machine queues, and completion time of jobs on the machine. The output layer of the ANN model retrieves data on the total completion time of the jobs (i.e., makespan).

The middle layer of the ANN model defines a weighting factor for each operation, which is used in lieu of make-up times. Priority is given to operations with the shortest processing time, and the first weight value is assigned as 1 in the algorithm. The weight value is then randomly assigned and multiplied by the makespan, and the resulting values are sorted in ascending order. Next, the jobs are assigned to machines according to their predetermined routes. The first result is recorded as the best result after all jobs have been assigned to machines. These processes are repeated in subsequent iterations, with the weights changing each time. Each iteration result is compared to the best result obtained so far, and the best value is updated when the output data improves. Generally, the best results were obtained after 1000 iterations in this study.

The data in the layers of the ANN model used in this study is explained in Fig. 1.

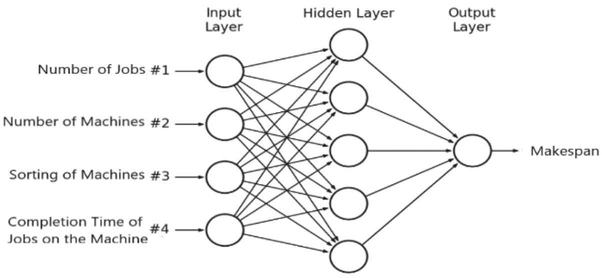

Figure 1 The proposed ANN model

Three processes take place in the hidden layer. First of all, the weight value is assigned, then the weighted processing times are calculated, and the learning coefficient is applied.

Application of learning coefficient: The learning coefficient plays a critical role in the weight-strengthening process. It controls the degree of weight change during each iteration, thereby allowing for the granularity of the search to be controlled. The value of the learning coefficient is randomly selected from a range of 0.01 to 0.99. In this study, a learning coefficient of 0.05 was found to be an optimal average value for all iterations. If there is no change in the results after 10 iterations, the learning coefficient is increased by 0.01. This approach helps to fine-tune the weight adjustments for more accurate and efficient results.

The program is run until it reaches 1000 iterations. When a better makespan value is obtained than the first result due to iterations, this result is accepted as the best and replaced with the first recorded result. The program is tested with a certain number of iterations. The best weights in the determined iteration are recorded and the best processing time is displayed. Parameters can be listed as follows:

- Jobset {1, 2, …, n): n.
- Set of m machines{1, 2, …, m}: m.
- Sorting of machines: Mor_{ij}.
- Processing time of operation: P_{ij}.
- Weight associated with operation process: W_{ij}.
- Weighted processing time: WP_{ij}.
- Number of Iterations: k.
- Learning Coefficient: a.
- Best Total Processing Time (Makespan): BMS.
- Best Weights: BW_{ij}.

Mathematical modelling of ANN method is shown in Eq. (1) through Eq. (3) [17].

$$n > 0, m > 0, i > 0, j > 0$$

$$P_{ij} > 0, Mor_{ij} > 0, 0 < w_{ij} \leq WP_{ij} > 0$$

$$\sum_{i=0,j=0}^{i=n,j=m} P_{ij} \tag{1}$$

$$\sum_{i=0,j=0}^{i=n,j=m} W_{ij} \cdot P_{ij} \tag{2}$$

$$W_{ij_{k+1}} = \begin{cases} RND > 0.05, W_{ij\,k} + RND \cdot a \cdot P_{ij} \\ RND < 0.05, W_{ij\,k} + RND \cdot a \cdot P_{ij} \end{cases} \tag{3}$$

This study employed prioritization techniques to assign jobs to machines. Specifically, five techniques were utilized for this purpose. The techniques can be listed as follows:

- In scheduling, which is done by giving priority to the one with the shortest processing time, the times are ordered from smallest to largest. The smaller one is given priority.
- In scheduling, which is done by giving priority to the one with the longest processing time, the times are ordered from largest to smallest. The big one is given priority.
- In first come first serve scheduling, the machine queue is first served first.
- In last-come, first-served scheduling, the machine queue is first served first.
- Jobs are randomly assigned to machines.

The operation of the ANN algorithm is explained with flowchart in the Fig. 2.

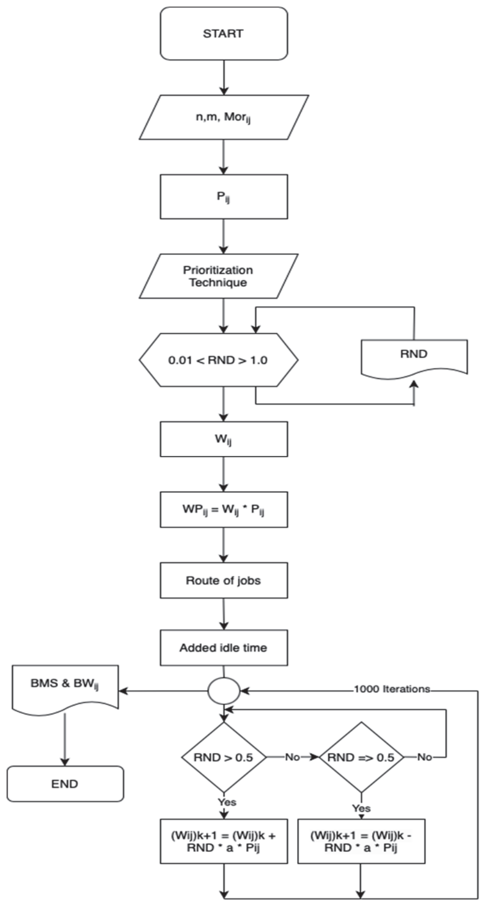

Figure 2 Flowchart of the ANN model

4 RESULTS AND DISCUSSION

Tab. 1 presents the results of Taillard's study on production scheduling problems according to the data of 15 jobs and 15 machines, 20 jobs and 20 machines, 30 jobs and 15 machines in jobshop type scheduling, and the results obtained in 1000 iterations with our proposed ANN method.

When calculating the improvement rates (*IR*) presented in the tables, the result obtained from the literature study was subtracted from the result obtained using the proposed ANN method. The difference was then divided by 100, and the percentage rate was computed. A positive improvement rate indicates the proximity of the best result, while a negative improvement rate indicates the percentage of better results achieved.

This study was able to generate solutions that closely approximated optimal results using a simpler algorithm with fewer iterations compared to the methods described in Tab. 2 of the literature. The ANN method employed in this study offers the advantage of being efficient in terms of scheduling, as it can produce optimal results quickly for problems with a low workload. For instance, 15 × 15 Job shop problems yielded better outcomes than 30 × 15 Job shop problems. For problems with a high number of machines and jobs, a hybrid approach and an increased number of iterations can be used to enhance the network's learning process, leading to improved results. Moreover, the proposed ANN method can be applied to other scheduling problems by adjusting the algorithm's constraints with ease.

When Tab. 1 through 4 are examined, the findings can be outlined as follows:

- Upon examination of Tab. 1, a comparison between the optimal results of the Taillard dataset and the results obtained in this study reveals that, for the first 20 models, the second model completed the work in the same time as the optimum results. However, for eight models, the work was completed in a shorter time. In the remaining 11 models, operations were scheduled with an average time difference of 0.63. Notably, the ANN method was successful in updating eight new bounds for the first 20 problems of the Taillard dataset.
- By using the proposed ANN algorithm in this study, near-optimal results were obtained in a short time of 1 hour, 4 minutes and 28 seconds on average and with 1000 iterations. Unlike the studies in the literature, a new ANN method has been proposed. While coding the ANN algorithm, C# language was used and a 6-step algorithm that can be easily integrated into other scheduling problems was used.
- When Tab. 2 is examined, it is observed that other studies that used the same data set in recent years worked with at least two methods, except for the 2007 study, that we had equal iterations with the 2006 study, and that the number of iterations was higher in all other studies. On the other hand, the algorithm proposed in this study achieved successful results by using only a single method and fewer iterations than other studies.

Table 1 Comparison of Results Obtained by ANN Method with Taillard's Results

No	ANN	Time	Taillard [15]	*IR* / %
01 15 × 15	1235	01:01:06	1231	0.04
02 15 × 15	1244	01:01:01	1244	0
03 15 × 15	1218	01:04:24	1222	−0.04
04 15 × 15	1217	01:00:55	1181	0.36
05 15 × 15	1224	01:00:57	1233	−0.09
06 15 × 15	1238	01:01:07	1243	−0.05
07 15 × 15	1261	01:01:02	1228	0.33
08 15 × 15	1432	00:58:06	1220	2.12
09 15 × 15	1398	00:57:08	1282	1.16
10 15 × 15	1398	00:56:40	1259	1.39
11 20 × 20	1658	01:02:47	1663	−0.05
12 20 × 20	1636	01:03:07	1626	0.10
13 20 × 20	1559	01:02:24	1574	−0.15
14 20 × 20	1654	01:02:54	1660	−0.06
15 20 × 20	1691	01:03:16	1598	0.93
16 20 × 20	1662	01:03:09	1657	0.05
17 20 × 20	1705	01:02:29	1704	0.01
18 20 × 20	1620	01:22:20	1626	−0.06
19 20 × 20	1682	01:22:26	1629	0.53
20 20 × 20	1599	01:21:56	1614	−0.15
21 20 × 20	2597	01:21:36	1770	8.27
22 30 × 15	2529	01:21:42	1841	6.88
23 30 × 15	2579	01:21:55	1832	7.47
24 30 × 15	2693	01:22:01	1851	8.42
25 30 × 15	2375	01:22:44	2007	3.68
26 30 × 15	2460	01:21:43	1844	6.16
27 30 × 15	2672	01:21:27	1815	8.57
28 30 × 15	2499	01:21:59	1700	7.99
29 30 × 15	2341	01:21:52	1811	5.30
30 30 × 15	2656	01:21:41	1720	9.36

Table 2 Comparison of this Study with the Studies Conducted in Recent Years

Study	Method	Number of iterations	Year
Sha and Hsu [8]	PSO ve TS	None	2006
Chandrasekaran et al. 2007 [9]	AIS	1500	2007
Zhang et al. [10]	SA and TS	10.000	2008
Nasiri and F.Kianfar [11]	TS and PR	5.000.000 - 10.000.000	2011
Peng et al. [12]	TS and PR	500 - 12.500	2015
Bürgy [13]	Neighborh oodandTS	If there is no improvement, the iteration is terminated.	2017
Xie et al. [14]	GA and TS	800 - 12000	2022
This study	ANN	1000	2023

- When Tab. 3 is examined, it has been observed that close to optimal results were obtained in 30 models with an average error rate of 3.67 according to the 2006 study, 2.67 according to the 2007 study, and with an average of 3.20 error rate according to the 2008 study. When Tab. 4 is examined, it is observed that close to optimal results are obtained in 30 models with an average time difference of 3.76 according to the 2010 study, 3.20 according to the 2015 study, and 2.50 according to the 2017 study.
- When Tab. 3 and Tab. 4 are examined, optimal total processing times with less than 4% time differences were obtained using a single method with a simpler algorithm and using relatively less number of iterations compared to the studies in the literature.

When Tab. 3 and Tab. 4 are examined, it is observed that the proposed method alone gives close to optimal results in jobshop type scheduling, when the results of the first 20 models applied to ANN are examined.

It is observed that the total processing time increases when 30 jobs and 15 machines are used. Therefore, it can be said that the proposed ANN method is more suitable for JSSP with less work and less machinery.

Table 3 Comparison of the Results Obtained through the ANN Method with those of Other Studies in the Literature (Part 1)

No	Proposed ANN	2006 Sha & Hsu [6]	*IR* / %	2007 Chandrasekaran et al. [7]	*IR* / %	2008 Zhang et al. [8]	*IR* / %
01 15 × 15	1235	1231	0.04	1231	0.04	1231	0.04
02 15 × 15	1244	1244	0	1244	0	1244	0
03 15 × 15	1218	1218	0	1206	0.12	1218	0
04 15 × 15	1217	1175	0.42	1170	0.47	1175	0.42
05 15 × 15	1224	1224	0	1215	0.09	1224	0
06 15 × 15	1238	1238	0	1210	0.28	1238	0
07 15 × 15	1261	1228	0.33	1223	0.38	1228	0.33
08 15 × 15	1432	1217	2.15	1187	2.45	1217	2.15
09 15 × 15	1398	1274	1.24	1297	1.01	1274	1.24
10 15 × 15	1398	1249	1.49	1241	1.57	1241	1.57
11 20 × 20	1658	1658	0	1649	0.09	1644	0.14
12 20 × 20	1636	1614	0.22	1627	0.09	1600	0.36
13 20 × 20	1559	1559	0	1556	0.03	1560	−0.01
14 20 × 20	1654	1654	0	1624	0.3	1646	0.08
15 20 × 20	1691	1616	0.75	1580	1.11	1597	0.94
16 20 × 20	1662	1662	0	1672	−0.1	1647	0.15
17 20 × 20	1705	1690	0.15	1688	0.17	1680	0.25
18 20 × 20	1620	1617	0.03	1602	0.18	1603	0.17
19 20 × 20	1682	1634	0.48	1583	0.99	1627	0.55
20 20 × 20	1599	1589	0.10	1573	0.26	1584	0.15
21 20 × 20	2597	1766	8.31	1764	8.33	1764	8.33
22 30 × 15	2529	1823	7.06	1824	7.05	1795	7.34
23 30 × 15	2579	1818	7.61	1829	7.5	1796	7.83
24 30 × 15	2693	1844	8.49	1841	8.52	1831	8.62
25 30 × 15	2375	2007	3.68	2009	3.66	2007	3.68
26 30 × 15	2460	1825	6.35	1825	6.35	1819	6.41
27 30 × 15	2672	1795	8.77	1796	8.76	1778	8.94
28 30 × 15	2499	1681	8.18	1699	8	1673	8.26
29 30 × 15	2341	1796	5.45	1803	5.38	1795	5.46
30 30 × 15	2656	1698	9.58	1684	9.72	1676	9.80

Table 4 Comparison of the Results Obtained through the ANN Method with those of Other Studies in the Literature (Part 2)

No	Proposed ANN	2011 Nasiri &Kianfar[9]	*IR* / %	2015 Penget al. [10]	*IR* / %	2017 Bürgy [11]	*IR* / %	2022 Xie et al. [15]	*IR* / %
01	1235	1249	−0.14	1231	0.04	1232	0.03	1231	0.04
02	1244	1157	0.87	1244	0	1245	−0.01	1244	0
03	1218	1087	1.31	1218	0	1223	−0.05	1218	0
04	1217	1131	0.86	1175	0.42	1177	0.4	1175	0.42
05	1224	1161	0.63	1224	0	1232	−0.08	1224	0
06	1238	1162	0.76	1238	0	1242	−0.04	1238	0
07	1261	1100	1.61	1228	0.33	1228	0.33	1228	0,33
08	1432	1086	3.46	1217	2.15	1218	2.14	1217	2.15
09	1398	1139	2.59	1274	1.24	1283	1.15	1274	1.24
10	1398	1076	3.22	1241	1.57	1245	1.53	1241	1.57
11	1658	1452	2.06	1644	0.14	1660	−0.02	1642	0.16
12	1636	1365	2.71	1600	0.36	1619	0.17	1600	0.36
13	1559	1332	2.27	1557	0.02	1566	−0.07	1557	0.02
14	1654	1550	1.04	1645	0.09	1661	−0.07	1644	0.10
15	1691	1519	1.72	1595	0.96	1604	0.87	1595	0.96
16	1662	1403	2.59	1647	0.15	1659	0.03	1645	0.17
17	1705	1583	1.22	1680	0.25	1694	0.11	1680	0.25
18	1620	1485	1.35	1613	0.07	1624	−0.04	1603	0.17
19	1682	1517	1.65	1625	0.57	1633	0.49	1625	0.57
20	1599	1398	2.01	1584	0.15	1603	−0.04	1584	0.15
21	2597	1780	8.17	1764	8.33	1766	8.31	1764	8.33
22	2529	1829	7.00	1787	7.42	1829	7	1785	7.44
23	2579	1733	8.46	1791	7.88	1818	7.61	1791	7.88
24	2693	1849	8.44	1829	8.64	1843	8.5	1829	8.64
25	2375	1745	6.30	2007	3.68	2007	3.68	2007	3.68
26	2460	1780	6.80	1819	6.41	1828	6.32	1819	6.41
27	2672	1774	8.98	1771	9.01	1800	8.72	1771	9.01
28	2499	1674	8.25	1673	8.26	1688	8.11	1673	8.26
29	2341	1686	6.55	1795	5.46	1805	5.36	1795	5.46
30	2656	1642	10.14	1671	9.85	1706	9.50	1640	10.15

When compared with other studies using the same data set based on the first 20 models, the findings are as follows:

- The study by Sha & Hsu in 2006 used a hybrid method consisting of two methods and the same number of iterations as this study. According to this study, summarized in Tab. 3, when the first 20 models were examined, it was observed that the same results were obtained in 8 models. The operations were scheduled in the remaining 12 models with an average time difference of 0.48.
- In the study conducted by Chandrasekaran et al. in 2007, a single method, the MIS algorithm, was proposed, but 1500 iterations were used. According to this study, summarized in Tab. 3, when the first 20 models are examined, the works were completed in a shorter time in the 16th model, while the works were completed in the same time in the 2nd model. In the remaining 18 models, operations were scheduled with an average time difference of 0.53.
- In the study by Zhang et al. 2008, a hybrid method consisting of two and 10,000 iterations was used. According to this study, which is summarized in Tab. 3, when the first 20 models are examined, the works were completed in a shorter time in the 13th model, while the works were completed in the same time in the 4 models. In the remaining 15 models, operations are scheduled with a time difference of 0.56.
- In the 2010 study by Nasiri & Kianfar, 5-10 million iterations were used using three hybrid methods. According to this study, summarized in Tab. 4, when the first 20 models are examined, the works were completed in a shorter time in the 1st model, while the operations in the remaining 19 models were scheduled with a time difference of 1.79.
- According to the study by Peng et al. 2015, between 500 and 12500 iterations were used while using a hybrid method consisting of two methods. According to this study, which is summarized in Tab. 4, when the first 20 models are examined, the works were completed in the

same time in 4 models, while the operations were scheduled with an average time difference of 0.53 in the remaining 16 models.

- In the study conducted by Bürgy in 2017, when using a hybrid method consisting of two methods, the number of iterations was stopped when there was no improvement in the results. According to this study, which is summarized in Tab. 4, when the first 20 models are examined, the works are completed in a shorter time in 9 models, while the operations in the remaining 11 models are scheduled with an average time difference of 0.65.
- In the 2022 study conducted by Xie et al., a hybrid method comprising two methods was employed, with the number of iterations ranging from 800 to 12000. Tab. 4 summarizes the study, revealing that four of the first 20 models yielded the same results, with an average time difference of 0.43.

5 CONCLUSION AND FUTURE WORKS

In JSS, the wide variety of products and the small amount of incoming orders require high planning and concentration. Efficient use of resources is important for time and financial savings. The effectiveness of scheduling is of great importance for increasing product quality and efficient resource use. As the number of jobs and machines increases, it becomes more difficult to solve JSSP and mathematical methods are insufficient. For this reason, researchers use heuristic methods in JSSP solution. To evaluate the performance of algorithms, a famous benchmark has been presented from 1993 Taillard.

In this study, the ANN algorithm was developed as a solution approach for data set Taillard. When the best results were obtained in Taillard's 15 job and 15 machine, 20 job and 20 machine problems are compared with the results obtained with the ANN method we developed in this study, it has been observed that our method gives better results. Results close to optimal results were obtained in 30 job and 15 machine problems.

The proposed ANN method in this study successfully updated eight new bounds to Taillard's first 20 problems. In the past 30 years, studies cited in the literature have also provided new limits. As such, in Section 4, we compare the results of the ANN method to those of these studies. Compared to hybrid methods, the ANN method is much less complex and easier to implement. It achieved superior performance in Taillard problems, which are among the most challenging problems, with just 1000 iterations. The ANN method, which we propose in this study, can be preferred compared to the methods used in the literature, since it is an effective and practical method that can be easily adapted, produces optimal or near-optimal fast solutions in a short time.

The proposed ANN algorithm offers several advantages, such as the ability to calculate makespan close to optimal within an average of one hour, as we recommend. Moreover, the ANN algorithm is easy to understand, effortless to use, and can be easily adapted without requiring large-scale calculations, lengthy coding, or complex simulations. The ANN method is particularly well-suited for small-scale JSSP problems, characterized by a low number of jobs and machines. In such cases, the ANN method yields fast and optimal results. As a result, the study's conclusion supports the aim of the study, as close to optimal results were obtained using a single method for production scheduling with fewer jobs and machines, within a shorter period of time.

According to the results obtained in this study, ANN is a method that gives effective results in scheduling problems. In future studies, the ANN method can also be applied to different scheduling problems to further generalize its effectiveness.

6 REFERENCES

[1] Gever, B., Güneş, B. N., Gülel, D., Ataseven, E. B., Saatcıoğlu, Ö., & Bebek, Z. S. (2015). Bir Üretim İşletmesinin Verileri Kullanilarak Çarpimsal Deney Tasarimi İle Taguchi Yaklaşiminin Karşilaştirilmasi. *Endüstri Mühendisliği*, *26*(1), 2-26.

[2] Küçükkoç, İ. (2018). http://ikucukkoc.baun.edu.tr/lectures/-EMM4129/EMM4129-S1.pdf%20. Avaliable 30.09.2018.

[3] Akkaya, G. & Gökçen, T. (2006). Job-Shop Scheduling Design with Artificial Neural Networks. *Journal of Engineering And Natural Sciences*, *4*, 121-130

[4] Reeves, C. R. (1993). *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc.

[5] Gass, S. I. (2000). Making decisions with precision. *Business Week October*, *30*, 2000.

[6] Eberhart, R. & Kennedy, J. (1995). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43. IEEE.

[7] Glover, F. & Laguna, M. (1997). Tabu Search Kluwer Academic Publishers. *Boston, MA*. https://doi.org/10.1007/978-1-4615-6089-0

[8] Sha, D. Y. & Hsu, C. Y. (2006). A hybrid particle swarm optimization for job shop scheduling problem. *Computers & Industrial Engineering*, *51*(4), 791-808. https://doi.org/10.1016/j.cie.2006.09.002

[9] Chandrasekaran, M., Asokan, P., Kumanan, S., Balamurugan, T., & Nickolas, S. (2006). Solving job shop scheduling problems using artificial immune system. *The International Journal of Advanced Manufacturing Technology*, *31*(5-6), 580-593. https://doi.org/10.1007/s00170-005-0226-3

[10] Zhang, C. Y., Li, P., Rao, Y., & Guan, Z. (2008). A very fast TS/SA algorithm for the job shop scheduling problem. *Computers & Operations Research*, *35*(1), 282-294. https://doi.org/10.1016/j.cor.2006.02.024

[11] Nasiri, M. M. & Kianfar, F. (2011). A hybrid scatter search for the partial job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, *52*, 1031-1038. https://doi.org/10.1007/s00170-010-2792-2

[12] Peng, B., Lü, Z., & Cheng, T. C. E. (2015). A tabu search/path relinking algorithm to solve the job shop scheduling problem. *Computers & Operations Research*, *53*, 154-164. https://doi.org/10.1016/j.cor.2014.08.006

[13] Bürgy, R. (2017). A neighborhood for complex job shop scheduling problems with regular objectives. *Journal of Scheduling*, *20*(4), 391-422. https://doi.org/10.1007/s10951-017-0532-2

[14] Xie, J., Li, X., Gao, L., & Gui, L. (2022). A hybrid algorithm with a new neighborhood structure for job shop scheduling problems. *Computers & Industrial Engineering*, *169*, 108205. https://doi.org/10.1016/j.cie.2022.108205

[15] Taillard, E. (1993). Benchmarks for basic scheduling problems. *European journal of operational research*, *64*(2), 278-285. https://doi.org/10.1016/0377-2217(93)90182-M

[16] Dataset: http://mistic.heigvd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html. Available 10.10.22

[17] Agarwal, A., Colak, S., & Eryarsoy, E. (2006). Improvement heuristic for the flow-shop scheduling problem: An adaptive-learning approach. *European Journal of Operational Research*, *169*(3), 801-815. https://doi.org/10.1016/j.ejor.2004.06.039

Contact information:

İncilay Yildiz, Res. Asst.
Department of Management Information Systems,
Faculty of Applied Sciences,
Altınbaş University,
34381 Şişli/Istanbul, Turkey
E-mail: incilay.yildiz@altinbas.edu.tr

Abdülvahap Saygin, Lecturer
Department of Computer Technology,
Adana Vocational School,
Çukurova University,
01160 Çukurova/Adana, Turkey
E-mail: asaygin@cu.edu.tr

Selçuk Çolak, Prof. Dr.
Department of Business Administration,
Faculty of Economics and Administrative Sciences,
Çukurova University,
01330 Sarıçam/Adana, Turkey
E-mail: scolak@cu.edu.tr

Fatih ABUT, Assoc. Prof. Dr.
(Corresponding author)
Department of Computer Engineering,
Faculty of Engineering,
Çukurova University,
01330 Sarıçam/Adana, Turkey
E-mail: fabut@cu.edu.tr