Solving Flexible Job Shop Scheduling Problem with Transportation Time Based on Neuro - Fuzzy Suggested Metaheuristics

Aleksandar Stanković, Goran Petrović, Danijel Marković, Žarko Ćojbašić

Faculty of Mechanical Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, e-mail: aleksandar.stankovic@masfak.ni.ac.rs, goran.petrovic@masfak.ni.ac.rs, danijel.markovic@masfak.ni.ac.rs, zcojba@ni.ac.rs

Abstract: The flexible job shop problem (FJSP) represents an extension of the classical job shop problem (JSP). The paper deals with a FJSP in an available set of machines with additional transportation time between machines. This type of problem belongs to the group of NP-hard problems. To solve the FJSP, artificial intelligence was used by applying three improved metaheuristic algorithms: Particle Swarm Optimization (PSO) algorithm, Artificial Bee Colonies (ABC) algorithm and Genetic Algorithm (GA). The new approach in solving planning and scheduling problems with additional transportation time in combination with artificial intelligence and the developed neuro-fuzzy system represents the main research subject in the paper. The aim of the paper is to reduce the objective function with regard to time and increase productivity. Based on the case study optimization, experimental results show that the proposed mathematical model and the metaheuristic algorithms lead to an efficient outcome.

Keywords: flexible job shop problem; transportation time; metaheuristic algorithms; anfis; machine learning

1 Introduction

The main motive and aim of production in today's world of resource planning and scheduling is to achieve the most efficient results possible in the sense of increasing productivity. Modern economy has witnessed a rise in the importance of solving resource planning and scheduling problems (*PPSR*). Resource scheduling represents one of the key factors in a production system. Appropriate production resource scheduling encompasses correct planning of jobs in production and adequate scheduling of all jobs at the right time so that the production process itself becomes as efficient as possible. Scheduling problems in production comprise several phases and represent an appropriate scheduling of several operations within multiple jobs on machines, all with the aim of minimizing the criterion function and increasing productivity. Production planning is one of the initial phases of production system management, where initial planning objectives are set, a management strategy of the entire company is outlined and work methodology implemented so as to make the system efficient and meet the given criteria of company goals [1]. Resource planning and scheduling within a company is a very complicated and long-term process and it belongs to the group of complex processes. The planning and scheduling process consists of several phases, as follows: long-term planning, medium-term planning and short-term planning. All phases of the planning and scheduling process depend on initial conditions and objectives during the optimization process and the application of work methodology in the planning process. Furthermore, the planning process also depends on the mission, vision and strategy of a company [2], [3].

The paper aims to apply metaheuristic algorithms in the example of job shop scheduling with additional transportation time between machines (*FJSPT*). The main purpose of the paper is to show the optimality and behavior of algorithms for different scheduling problems with regard to the scope of the problem based on the *ANFIS* system. Scientific literature contains numerous examples of solving *FJSP* by applying one metaheuristic algorithm. This is where the motivation lies for applying multiple metaheuristic algorithms to *FJSPT* for different optimization problems starting from the speed of the convergence solution through a series of iterations to the testing of small dimension problems and robust *FJSPT* with a large dataset. Furthermore, it should be noted that *FJSP* represents one of the most difficult *NP*-hard problems in combinatorial optimization. It is particularly important to emphasize the part of transportation time that is taken into consideration as one of the constraints, which further complicates the model and brings it closer to a real situation in practice. The proposed modification of the algorithms and the addition of transportation time between machines in combination with artificial intelligence and the *ANFIS* system represents the main innovation in the paper. The new approach to solving robust models with a large set of data can greatly contribute to saving time and increasing productivity in the future. The novelty of the approach also lies in the development of the neuro-fuzzy system whose main purpose is to recommend the optimal method for solving *PPSR* problems on the basis of input parameters.

The paper comprises several parts and is presented in six sections. In the introductory part in Section 1 the vision of the paper is represented through the *PPSR* process and the main motivation behind applying multiple metaheuristic methods in combination with the *ANFIS* system is given. Section 2 presents a detailed literature review through the classification of methods used in the area of *PPSR*. Section 3 provides a detailed description of the mathematical model, as well as an example of the application of the described mathematical model with input data in Tables 1 and 2 and output results in the form of a Gantt chart. Section 4 presents the applied work methodology and the implementation of well-known metaheuristic algorithms for solving *PPSR* problems with additional transportation time. Section 5 proposes an adaptive neuro-fuzzy system capable of recommending the optimal metaheuristic method for solving *PPSR* with additional transportation time on the basis of the input data of the considered problem. Section 6 uses the developed *ANFIS* system, and based on the developed method, the *GA* method was used in the example of *PPSR* operations in the production environment with real data in a footwear company. In the conclusion section the experimental results show that the proposed mathematical model and the metaheuristic algorithms lead to optimal results.

2 Background

Based on a detailed analysis and literature review in the field of resource planning and scheduling in a manufacturing environment, it can be concluded that different methods are used to solve *FJSP*. The most well-known methods used today to solve *PPSR* are: exact methods, heuristic methods, metaheuristic methods and simulation methods. Within said methods, various software packages are employed to present the mathematical model that can be used to solve a problem on a concrete example with real data. The classification of methods employed in solving *PPSR* is shown in Figure 1 [2].

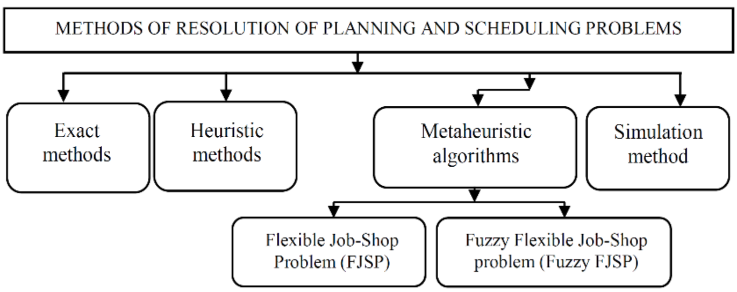

Figure 1
PPSR resolution methods

2.1 FJSP and Fuzzy FJSP

In what follows, a review of literature on *FJSP* and *Fuzzy FJSP* (*FFJSP*) in combination with the *ANFIS* system will be presented. Artificial intelligence and fuzzy sets have a wide application in resolving these *PPSR* problems. The division of *FFJSP* depends on the objective function of the presented model, type of problem being solved and given constraints for the defined problem. *FJSP* can be divided into problems with partial flexibility and problems with total flexibility, depending on the problem being solved and the objective function [4], [5]. The classification of *FFJSP* is given in detail in Figure 2 [6].

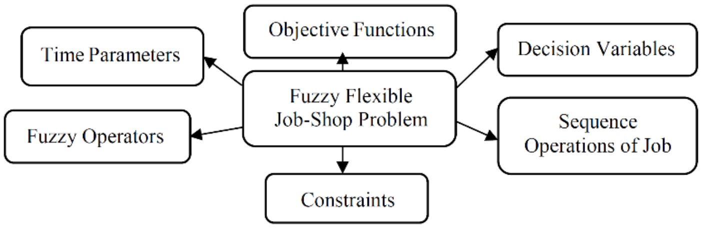

Figure 2
FFJSP classification and parameters

Haojie Ding and Xingsheng Gu [7] present the application of an improved particle swarm optimization algorithm for solving flexible job shop problems. The improvement of this algorithm is seen in better communication between particles during the search of the local space of possible solutions. Alejandro Vital-Sotoa et al. [8] describe a flexible job shop problem in detail and present the applied mathematical model. The paper employs a hybrid algorithm in combination with simulated annealing. The mentioned algorithm is improved by the authors using a local search method that is based on manipulation with critical operations. K. Dehghan-Sanej et al. [9] consider a job shop planning and scheduling problem with reverse flows and unstable time of operation processing on a set of machines. The authors recommend a model for solving such problems, where the main parameter during scheduling optimization is processing time. To solve such a problem the authors propose a simulated annealing algorithm. O. Bilkay et al. [10] present the application of flexible job shop scheduling in a dynamic environment with the optimization time for operation processing based on fuzzy sets. In the case of any type of failure, the proposed algorithm can change the sequence of operations and continue working, which is proof of the dynamic environment. Sezgin Kılıç et al. [11] shows the application of fuzzy sets in the example of the scheduling process optimization. Input operation times are not fixed. The authors propose an ant colony optimization algorithm for problem resolution. Qun Niu et al. [12] present a combination of two metaheuristic methods and processing time optimization based on fuzzy sets. The application of fuzzy sets relates to the selection of optimal operation processing time to make the optimization process as real as possible. By combining two methods, particle swarm optimization and genetic algorithm, in the crossover and mutation phases, the authors claim that the used methodology is successful in comparison with other metaheuristic algorithms. Zhongshi Shao et al. [13] observe the problem as a flow of material without mutual constraints in terms of machines, with the entire system observed as a single whole. Due to the very configuration of the production system, processing time is uncertain and represented on the basis of a fuzzy logic system. This way the fuzzy system chooses the appropriate processing time. The main aim of the paper is to minimize the total product manufacturing time between the sequence of production machinery based on the fuzzy system. To reach the optimal solution and avoid a drop in the local optimum the authors used the methodology of the improved acceptance criteria and fuzzy system. Tibor Dulai et al. [14] present a genetic algorithm for solving resource planning and scheduling problems in a manufacturing environment. The goal of the proposed algorithms is to find the optimal sequence and minimize the total manufacturing time with maximum productivity. Guohui Zhang et al. [15] present an improved genetic algorithm for solving flexible job shop problems with multiple time constraints. The authors mention a very important segment within time constraints during scheduling that points to time constraints during the transport to another machine, as well as the preparation of a machine for a different operation. The first part of the paper presents the problem, while the second part shows the proposed algorithm and optimization objective in the sense of minimizing manufacturing time, as well as minimizing the total time of transporting operations between machines. The improved algorithm is implemented through several phases in the form of initial population, adopted good solutions with artifical intelligence and a mechanism with a dynamic change in the mutation and the scope of search for possible solutions.

The major innovation of this paper is reflected in the modification of the proposed algorithms with additional transportation time and part of simulation that needs to be applied. Also, the combination with artificial intelligence and the *ANFIS* system further complicates the model, providing more concrete results and suggestions for the next phase during the optimization process. The novelty of the approach in this paper also includes the developed neuro-fuzzy system that is trained to suggest an optimal metaheuristic method for *FJSP* solving, on the basis of the input parameters of the considered problem.

3 Problem Description and Mathematical Modeling

This part presents the *FJSPT* between machines. The essence of flexible scheduling can be described as n jobs that need to be performed on a set M of machines. Each job can have a different number of operations as well as processing time for each operation. Also, each operation can be performed on any machine if that machine is available at that moment and if there is a possibility for that machine to perform the given operation. When an operation is finished on a selected machine, the job moves to another machine where the next operation of that job is scheduled. The job is completed when the set of operations of that job is performed on a machine. The optimization goal is to obtain the minimal time of the last operation of all jobs C_{max} on a set of machines M, as well as the minimal operation transportation time from one machine to another [15]. Basic constraints that have to be addressed when scheduling operations on a set of machines with additional transportation time between machines are presented in the following manner:

- the same job can be processed by only one machine at a time,
- a job cannot be stopped once the processing begins,
- one machine can process multiple jobs,
- each machine can be used with the beginning in 0,
- all jobs can be performed in the starting moment in 0,
- the sequence of operations of a job is predefined, i.e. an operation will be sent to the next machine for processing as soon as processing finishes,
- the time of each operation is different due to the difference in selected machines for performing the given operation,
- the total processing time of every operation on a machine is known,
- the distance between two different machines during the transportation of one operation to another machine is called transportation time.

The set of all jobs $J = \{J_1, J_2, ... J_p, ... J_n\}$ is a set where each job J_p is predefined by the sequence of operations. The set of all machines is $M = \{M_1, M_2, ... M_{ai}, ... M_m\}$, with M_i representing a symbol where *i* is a machine. O_{jhs} marks and represents *hs* operation of job *j* and defines it using $O_{j(hs-1)}$ as the previous operation O_{jhs}, while $O_{j'hs'}$ represents the previous operation of the machine on which O_{jhs} is processed. F_{jhs} represents the total end time of job *j*. T_{ijhs} defines the time needed for the *hs* operation of job *j* on machine *i*. S_{ijhs} is the time needed to initiate operation *hs* of job *j* on machine *i*. C_{ijhs} is the time needed to process operation *hs* of job *j* on machine *i*. $TransTimes_{ie}$ represents the total transportation time from one machine M_i to another machine M_e. C_j represents the time needed to complete job *j*. C_{max} is the objective function in this case and it represents the total time after optimization and scheduling of all jobs on a set of machines. Taking into account the minimization of the criterion function, the constraints are as follows (presented in detail in [15]):

$$C_{max} = min(max_{1\le j\le n}(C_j)) \tag{1}$$

$$C_{ijhs} = S_{ijhs} + T_{ijhs} \tag{2}$$

$$C_{ijhs} - C_{ij'hs'} \ge T_{ijhs} \tag{3}$$

$$\begin{cases} C_{ej(hs-1)} + TransTime_{ie} & C_{ij'hs'} < C_{ej(hs-1)} + TransTime_{ie} \\ C_{ij'hs'} & C_{ij'hs'} > C_{ej(hs-1)} + TransTime_{ie} \end{cases} \tag{4}$$

Equation (1) shows the objective function of the defined problem. The next equation (2) shows the time needed to perform an operation and it is equal to the total time at the very beginning of the operation and the total time needed to perform a single operation. On the basis of equation (3) resource constraints during the *PPSR* process can be determined. Equation (4) shows that the total processing time of a single machine is shorter than the duration of the entire process. An example of such a model can be seen in [15], on the basis of which the *FJSPT* between machine M_i and machine M_e is studied. The detailed analysis of [15] and testing of advanced *ABC, PSO* and *GA* algorithms on the same example with the same input data leads to the conclusion that the algorithms proposed in the paper provide sufficiently good results. Drawing on such a course of research, the idea is to apply three metaheuristic algorithms in a small manufacturing environment.

The used algorithms and the part of the *ANFIS* system are presented in detail in the next section with all of the employed parameters. Also, what follows is an example of the functioning of the *FJSPT* system between machines. Table 1 shows the input parameters in the form of the operation processing time on each machine separately, while Table 2 shows the total time needed after an operation ends on one machine to start the next operation on another machine.

Table 1
Input parameters of the FJSPT problem

Jobs	Operations	Processing time					
		M_1	M_2	M_3	M_4	M_5	M_6
J_1	O_{11}	5	2	-	3	5	8
	O_{12}	4	2	1	7	-	-
J_2	O_{21}	2	7	5	4	9	-
	O_{22}	-	-	4	7	3	5
J_3	O_{31}	4	2	-	-	-	6
	O_{32}	6	7	8	2	-	3
J_4	O_{41}	2	3	8	2	4	1
	O_{42}	4	8	9	3	4	-

Table 2
Total transport time between sets of different machines

Machines	Transportation time					
	M_1	M_2	M_3	M_4	M_5	M_6
M_1	0	4	2	4	2	3
M_2	4	0	2	1	3	5
M_3	2	2	0	1	2	4
M_4	4	1	1	0	4	3
M_5	2	3	2	4	0	4
M_6	3	5	4	3	4	0

The obtained results are a consequence of the optimization of input parameters on the basis of the applied metaheuristic algorithms. Given that the input dataset is small, which can be seen from Table 1 and Table 2, it should be noted that all three algorithms provide optimal scheduling. The next section contains the presentation of the working methodology based on metaheuristics and the *ANFIS* system. Figure 3 shows a Gantt chart of the detailed scheduling of operations on a set of machines.

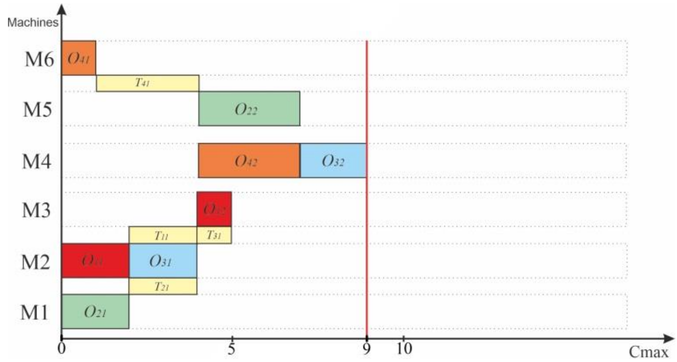

Figure 3
The Gantt chart with transportation time

4 Algorithms for Solving the FJSP with Transportation Time

Metaheuristics and part of simulation within the *ANFIS* system are used to solve *FJSPT* in this paper. Metaheuristics represent a higher level of heuristics and are used to obtain the local minimum depending on the objective function. Metaheuristics are used within metaheuristic algorithms, and the following algorithms are used in this paper: Genetic Algorithm, Particle Swarm Optimization Algorithm and Artificial Bee Colonies Algorithm. Metaheuristics were first applied in 1986 by Fred Glover [2], [16]. Metaheuristics are used to solve complex problems during optimization. Algorithms used today within metaheuristics represent one of the most important approaches to solving practical and complex problems in combinatorial process optimization. The next section presents the three metaheuristic algorithms used in solving *FJSPT*.

4.1 Particle Swarm Optimization

The basic principle and manner of functioning of the Particle Swarm Optimization (*PSO*) algorithm is founded on the swarm of particles while solution convergence is based on particle orientation. The *PSO* algorithm was first proposed by Kennedy and Eberhart in 1995 [2]. When searching for the best optimal solution, *PSO* is characterized by one phase during the search for the optimal minimum through a sequence of iterations. The main factors that are essential to the way a *PSO* algorithm functions are: the speed of particle movement and particle location. The algorithm forms optimal solutions through a sequence of iterations during the search based on particle location and input parameters.

During the solution finding process and the movement of the particle swarm through a sequence of iterations, the algorithm collects information on the basis of particle movement as follows: it is necessary to record the best optimal value achieved by each particle at a specific moment and the best value of the objective function through a sequence of iterations.

The position of particles at any given time is presented on the basis of variables $X_i = (x_{i1},x_{i2},\dots,x_{iD})$, while the speed of particles is presented using expression $V_i=(v_{i1},v_{i2},\dots,v_{iD})$. The position of all particles that move during a search for the best solution X^i_{k+1} I, $k+1$ is calculated on the basis of the following formula [2]:

$$X^i_{k+1}=xik+v^i_{k+1}\Delta t, \tag{5}$$

where the speed of the i-th particle in a sequence of iterations is V^i_{k+1}, $k+1$, while Δt represents the time interval during the solution search. The equation of the speed of particle movement is given by the following expression [2]:

$$V^i_{k+1}=\omega V^i_k+c_1r_1*(P_{Bi}-X_{ik})/\Delta t + c_1r_1*(P_g-X_{ik})/\Delta t \tag{6}$$

Based on the particle movement and the speed at which the particle swarm moves at the best local optimum, the algorithm finds the optimal solution through a sequence of iterations. Furthermore, parameters c_1 and c_2 should be mentioned since they represent the parameters for data learning during particle movement. Initial parameters of a *PSO* algorithm can be changed before the optimization process and the search for the optimal solution. The pseudocode of the *PSO* algorithm is presented in Table 3 [17].

Table 3

Pseudocode of Particle Swarm Optimization

Pseudo code of PSO
Objective function f(x), $x = (x_1,\dots,x_p)^T$
Initialize locations x_i and velocity v_i of n particles.
Find g^* from min{$f(x_1),\dots,f(x_n)$} (at t = 0)
While (criterion)
t = t+1 (pseudo time or iteration counter)
for loop over all n particles and all p dimensions
Generate new velocity v^{t+1}_i using equation
Calculate new locations $x^{t+1}_i=x^t_i+v^{t+1}_i$
Evaluate objective functions at new locations x^{t+1}_i
Find the current best for each particle x^*_i
end for
Fin the current global best g^*
end while
Output the final results x^*_i and g^*

4.2 Artificial Bee Colony Optimization

The Artificial Bee Colony Optimization (*ABC*) algorithm mimics the natural process by which a colony of bees searches for food and finds the best path during that search. During a search, the colony of bees is divided into three groups as follows: working bees exclusively in charge of finding food, observer bees with the role of monitoring other bees and scout bees that scan the entire field and look for new places with food sources [18]. The *ABC* algorithm was first proposed and applied by Karaboga and Basturk in 2005. The basis of objective function can be presented using the following equation [2]:

$$P_i = \frac{F(x_i)}{\sum_{j=1}^{S} F(x_j)} \tag{7}$$

where $F_{(x)}$ represents the total amount of collected nectar during the bee search and the objective function during the process search and optimiation, while P_i represents the probability of the best food source at location x_i. S represents the number of locations and food sources. The pseudocode of the *ABC* algorithm is presented in Table 4 [17].

Table 4

Pseudocode of Artificial Bee Colony Optimization

Pseudo code of ABC
Objective function f(x), x = (x1,...., x_n)T constraints
Encode f(x) into virtual nectar levels
Define dance routine (strength, direction) or protocol
While (criterion)
for loop over all n dimensions
(or nodes for routing and scheduling problems)
Generate new solutions
end for
Communicate and update the optimal solution set
end while
Decode and output the best results

4.3 Genetic Algorithm

Genetic algortihms were first developed by John Holland and his associates in the 1960s and 1970s. When solving a *FJSP* using *GA*, it is necessary to adapt the algorithm itself during problem solving, from input parameters to the basic objective function, which is the foundation for solving the defined problem. In setting the planning and scheduling problem *GA* includes several phases: problem setting, individual selection, crossover, mutation and assessment [2]. Problem setting consists of defining initial parameters and determining the initial population. The selection of individuals from the size of population is done according to the objective function, which represents the impetus for an individual from the population. The *GA* pseudocode in shown in Table 5 [17].

Table 5

Pseudocode of Genetic Algorithm

Pseudo code of GA
Objective function $f(_x)$, $x = (x_1, \dots, x_n)^T$
Encode the solution into chromosomes (binary strings)
Define fitness F (e.g., $F \infty f(x)$ for maximization)
Generate the initial population
Initial probabilities of crossover (p_c) and mutation (p_m)
While (t < Max number of generations)
Generate new solution by crossover and mutation
If p_c > rand, Crossover; **end if**
If p_m > rand, Mutate; **end if**
Accept the new solutions if their fitness increase
Select the current best for new generation
end while
Decode the results and visualization

GA crossover represents a process of combination of existing individuals so as to obtain completely new individuals, such as parents and their offspring [18], [19]. Mutation includes a random alteration in the genes of individuals in the population. Mutations are generally applied at the level of the genes of individuals in the population. This practically stands for the order of random sequence changes. The main aim is to get an individual that cannot be obtained in other search phases. Fitness, selection, crossover and mutation processes are repeated until a predefined number of iterations is reached.

5 Neuro-Fuzzy Selection of the Suitable Metaheuristic Algorithm for the Considered Problem

In this section, an adaptive neuro fuzzy system is proposed for suggesting which of the considered metaheuristic methods is preferred for the case at hand. Training of the neuro fuzzy *ANFIS* system is based on the extended data from Table 6. The complete dataset comprises 109 different cases distinguished by a different number of jobs/operations/machines, while all cases were solved with the *ABC, PSO* and *GA* algorithms. A combination of problem parameters and best results (preferred optimization method) was used to train the adaptive neuro fuzzy system, which could then be used further to suggest a suitable metaheuristic algorithm among the ones considered for the future observed problem. System inputs had two job characteristics, i.e. the number of operations and the number of machines, while a single output was the suggested optimization algorithm that is one of the three classes (*ABC, PSO, GA*). For each combination a single optimization method was marked as preferred on the basis of the obtained results (*CPU time* and C_{max}, Table 6).

Table 6
The computational results using the ABC, PSO and GA algorithm

No.	Size (j · m)	ABC		PSO		GA	
		CPU time	C_{max}	CPU time	C_{max}	CPU time	C_{max}
$FJSP_{T1}$	06 · 06	3	55	3	55	1	55
$FJSP_{T2}$	10 · 05	31	665	30	666	10	666
$FJSP_{T3}$	10 · 10	290	905	280	893	282	890
$FJSP_{T4}$	15 · 10	625	1153	627	1147	670	1147
$FJSP_{T5}$	20 · 10	725	1362	664	1362	937	1359
$FJSP_{T6}$	10 · 10	120	916	108	912	98	913
$FJSP_{T7}$	10 · 10	125	891	122	875	125	876
$FJSP_{T8}$	15 · 05	392	937	370	926	385	926
$FJSP_{T9}$	15 · 15	906	1592	910	1566	900	1554
$FJSP_{T10}$	20 · 05	537	1242	600	1238	588	1228
$FJSP_{T11}$	20 · 10	1218	1352	1200	1307	1259	1308
$FJSP_{T12}$	20 · 15	1350	756	1336	750	1361	750
$FJSP_{T13}$	20 · 20	2189	1075	2179	1165	2168	1072
$FJSP_{T14}$	30 · 10	1788	1927	1685	1922	1784	1918
$FJSP_{T15}$	50 · 10	2868	3693	2746	3659	2747	3658

Adaptive neuro-fuzzy systems represent a specific combination of artificial neural networks and fuzzy logic, thus combining the learning ability of artificial neural networks with the knowledge representation capability of fuzzy logic systems. *ANFIS* (Adaptive Neuro Fuzzy Inference System) as proposed by Jang [20], [21], consists of many layers of nodes (neurons), each of which performs a particular function (node function) on incoming signals as well as a set of parameters pertaining to this node.

ANFIS can be seen as a structure equivalent to a Radial Basis Function (*RBF*) neural network. However, constructed to make use of some organizational principles resembling those of the human brain it is a hybrid structure of both fuzzy system and artificial neural network. An *ANFIS* network has all the advantages of these systems and, besides, its hybrid learning algorithm offers superior training results in comparison to other methods. The basic *ANFIS* architecture with two inputs, two outputs, two rules and five layers is presented in Figure 4.

ANFIS structure – Each node in layer 1 generates membership grades of a linguistic label. Each node in layer 2 calculates the firing strength of each rule via multiplication. Nodes of layer 3 calculate the ratio of the rule's firing strength to the sum of all rules' firing strengths. Parameters of layer 4 are referred to as the consequent parameters. Finally, the single node layer 5 computes the overall output as the summation of all incoming signals, producing the classification result.

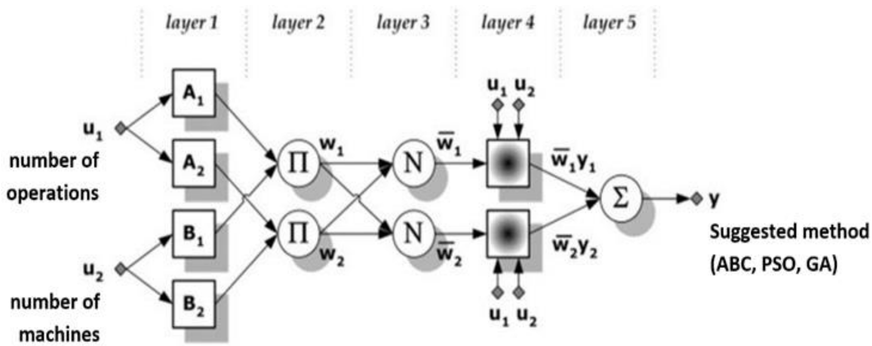

Figure 4
Basic two input – five layer - single output ANFIS architecture used for suggesting the optimal optimization procedure among the ones considered

The hybrid learning algorithm – The hybrid learning algorithm of *ANFIS* consists of two alternating parts:

- Back propagation/gradient descent (*BP/GD*) which calculates error signals (defined as the derivative of the squared error with respect to each node output) recursively from the output layer backward to the input nodes, and
- The recursive least squares estimation (*RLSE*) method, which finds a feasible set of consequent parameters.

MMC clustering – The purpose of clustering is to distill natural groupings of data from a large dataset, producing a concise representation of a system's behavior. The clustering of input/output data produces a set of cluster centers, and each cluster center acts as a prototypical data point that describes a characteristic mode of the system, and can be considered the nucleus of a fuzzy if-then rule. In that way partitioning of the inputs and determination of the initial minimal rule base for *ANFIS* can be performed.

The *MMC* clustering technique was used for determining the initial *ANFIS* classifier structure, prior to training. Training was performed through the above described *ANFIS* hybrid *BP/RLSE* learning algorithm. The described *ANFIS* system was successfully trained and used to suggest the most suitable of the three considered optimization algorithms, but this methodology can be effectively extended to include a broader selection of optimization methods for solving considered job shop scheduling problems with transportation time.

6 Case Study and Computational Experiments

The *ANFIS* system developed in the previous section was used to come up with a suggestion about which of the considered methods is suitable for solving the actual examined *FJSPT*, and based on that suggestion the *GA* method was applied to solve the observed problem. The case study represents an example of planning and scheduling the sequence of operations in a manufacturing environment of a footwear company. It should be noted that the study case is based on real data. The company examined in this paper manufactures various types of footwear depending on the needs and objectives.

The motivation behind this type of research in this manufacturing environment lies in the increase in the footwear manufacturing productivity. The manufacturing environment is observed as a *FJSPT* between machines. The footwear manufacturing process includes several machines and a series of operations of one job, and the process itself is completed when the final operation within one job is finished. The manufacturing process of a single final product comprises several operations, as follows: footwear design, die manufacturing, cutting, stitching, glueing and assembling the final product. After designing the footwear, the first phase of the manufacturing process consists of machine cutting of all outer and inner parts for each product separately. This type of operation in this phase is called cutting.

The cutting procedure is performed on special presses using manufactured knives and taking care of material stretching and quality in certain parts of footwear. It should be noted that the highest quality parts of the material are used in making the front of the footwear withthe least material waste as possible. After the initial cutting phase, the next phase is called stitching. The stitching operation consists of preparing the material for stitching and the main stitching operation, where parts are placed on a conveyor belt and the material is prepared at the same time. This phase contains the largest number of operations. It should be noted that the stitching procedure in this case study is observed as a single operation with a set of suboperations, and as such it is called stitching with cumulative time within a single stitching operation.

The next operation in the footwear manufacturing process is the product assembly operation and preparation for the final product. The last operation is the final product control and packaging operation. The conducted analysis of the operation sequence in the footwear manufacturing company shows that the operation sequence is not clearly defined. The application of artificial intelligence in the footwear manufacturing process provides great advantages in the sense of increasing manufacturing productivity with the minimal criterion function C_{max}, which represents the goal of this paper.

All input parameters in this manufacturing environment are shown in Tables 7 and 8. Input parameters, in the form of processing time on machines and transportation time between machines, are obtained on the basis of specific measurements of each operation for each job and machine.

Table 7
Input parameters of the problem

Jobs	Operations	Processing time					
		M_1	M_2	M_3	M_4	M_5	M_6
J_1	O_{11}	5	8	9	6	6	7
	O_{12}	5	6	6	-	8	9
	O_{13}	2	3	2	4	5	3
	O_{14}	62	63	-	70	58	67
	O_{15}	14	16	17	15	-	12
	O_{16}	10	11	13	12	13	15
J_2	O_{21}	6	7	5	4	3	7
	O_{22}	2	-	4	6	3	5
	O_{23}	2	5	3	4	7	9
	O_{24}	26	24	20	28	30	-
	O_{25}	14	15	16	18	17	13
	O_{26}	9	7	8	10	10	8
J_3	O_{31}	7	8	6	9	10	11
	O_{32}	6	7	-	2	8	3
	O_{33}	5	7	6	9	4	5
	O_{34}	20	-	23	27	24	-
	O_{35}	13	15	10	11	13	16
	O_{36}	7	8	12	6	9	10
J_4	O_{41}	9	8	8	6	4	7
	O_{42}	7	-	5	-	6	-
	O_{43}	4	3	5	2	4	6
	O_{44}	40	42	-	35	37	41
	O_{45}	15	10	12	11	16	14
	O_{46}	10	9	7	12	14	11

The mark „-“ in Table 7 signifies that the operation cannot be processed on that machine, e.g. job J_1 with operation O_{12} cannot be performed on machine M_4. As it can be seen and concluded on the basis of the case study and input data in Tables 7 and 8, it is the case of the manufacturing environment of a footwear manufacturing company.

Table 8 shows different times between machines that represent the total transportation time between different machines. As a result in Figure 5 the sequence of operations is obtained with the minimal duration of the manufacturing cycle in the form of the criterion function C_{max} and the maximal manufacturing productivity, e.g. job J_1 with operation O_{11} on machine M_1 needs to be transported to machine M_2 to perform the next operation O_{12}, which includes transportation time T_{11}.

Table 8
Transportation time between different machines

Machines	Transportation time					
	M_1	M_2	M_3	M_4	M_5	M_6
M_1	0	1	2	3	2	1
M_2	1	0	4	2	3	1
M_3	2	4	0	1	2	2
M_4	3	2	1	0	1	2
M_5	2	3	2	1	0	2
M_6	1	1	2	2	2	0

The results are presented graphically in Figure 5.

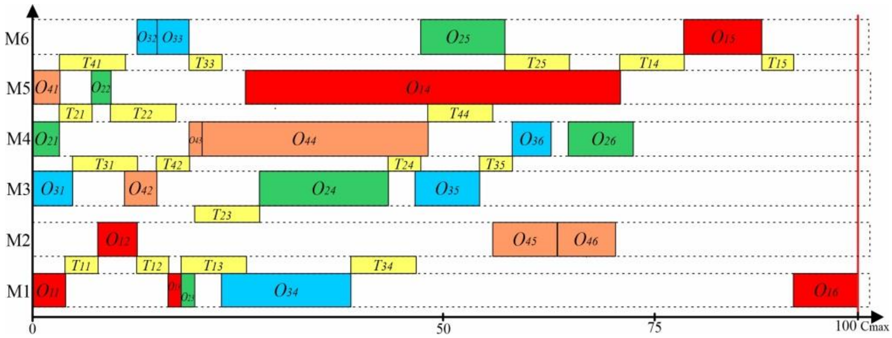

Figure 5
Graphic representation of the results of scheduling operations with transportation time

The average time of completion of the last operation within one job also represents the total time needed to finish all products. Based on the previous results of the company and the time of completion of the final job, the criterion function C_{max} was 117 min, while the total time after optimization was 100 min, which can clearly be seen graphically in Figure 5 in the form of a Gantt chart. On the basis of the results and the scheduling process optimization as an output within these results, manufacturing productivity was increased by 14.5 %, which justifies the application of the *ANFIS* system and the selection of *GA* for further optimization.

Conclusions

The application of artificial intelligence combines metaheuristics with the *ANFIS* system for *PPSR* process optimization and represents a new approach in solving this type of problem. At the beginning of the paper the methodology based on three metaheuristic algorithms was presented in detail. The improvements introduced within the algorithms are clearly defined and shown in a model with an additional matrix that describes the transportation time from one machine to another. To select the most appropriate algorithm, the *ANFIS* system was used with a complete dataset comprising 109 different cases distinguished by a different number of jobs/operations/machines, while all cases were solved with the *ABC*, *PSO* and *GA* algorithms and presented in detail in Section 5. Based on the *ANFIS* system, *GA* was used for further optimization in the example of a *FJSPT* in a footwear manufacturing company. Figure 6 shows the results of the previous state in the company and the results obtained after the application of artificial intelligence.

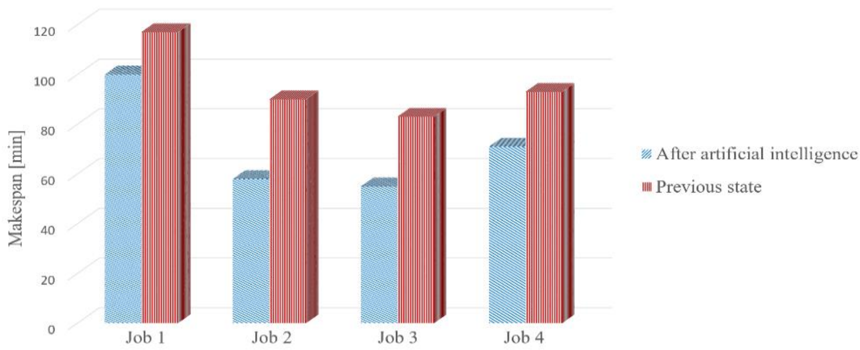

Figure 6

Graphical representation of the previous state of manufacturing and the state after optimization based on artificial intelligence

What can be observed from the graphical results in Figure 6 is the improvement in the working process and the increase in manufacturing productivity. The total time needed to perform every job J_1, J_2, J_3, J_4 is minimized, which was the goal of this paper, and the consequences of such results are great savings in time, higher productivity and profit. The importance of the application of artificial intelligence should also be mentioned here. Furthermore, such a planning and scheduling system can be applied in different spheres of resource scheduling, which emphasizes the flexibility of the presented model. As future research directions within *FJSPT*, new constraints may be considered in the form of the machine unavailability period and a further expansion of the *ANFIS* system so that it includes a wider selection of optimization methods for future resource planning and scheduling problems with additional constraints.

Acknowledgement

This research was financially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

References

[1] Marković, D., Petrović, G., Ćojbašić, Ž., Marinković, D. (2019) Metaheuristic Approach to the Waste Collection Vehicle Routing Problem with Stochastic Demands and Travel Time, Acta Polytechnica Hungarica 16 (7), pp. 45-60

[2] Stanković, A., Marković, D., Petrović, G., Ćojbašić, Ž. (2020) Metaheuristics for the waste collection vehicle routing problem in urban areas, Working and Living Environmental Protection 17(1), pp. 1-16

[3] Marković, D., Petrovć, G., Ćojbašić, Ž., Stanković, A. (2020) The vehicle routing problem with stochastic demands in an urban area – a case study, Facta Universitatis, Series: Mechanical Engineering 18 (1), pp. 107-120

[4] Chaudhry, I. A., Khan, A. A. (2016) A research survey: review of flexible job shop scheduling techniques, International Transactions in Operational Research, 23(3), pp. 551-59

[5] Kacem, I. S., Hammadi., Borne, P, (2002) Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32(1), pp. 1-13

[6] Abdullah, S., Abdolrazzagh-N., M. (2014) Fuzzy job-shop scheduling problems: A review, Information Sciences 278(10), pp. 380-407

[7] Ding, H., Gu, X. (2020) Hybrid of human learning optimization algorithm and particle swarm optimization algorithm with scheduling strategies for the flexible job-shop scheduling problem, Neurocomputing, 414(3), pp. 313-332

[8] Vital-Soto, A., Azab, A., Baki, M. F. (2020) Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible jobshop scheduling problem with sequencing flexibility, Journal of Manufacturing Systems, 54, pp. 74-93

[9] Dehghan-Sanej, K., Eghbali-Zarch, M., Tavakkoli-Moghaddam, R., Sajadi, S. M., Sadjadi, S. J. (2021) Solving a new robust reverse job shop scheduling problem by meta-heuristic algorithms. Engineering Applications of Artificial Intelligence, 101, 104207

[10] Bilkay, O., Anlagan, O., Kilic, S. E. (2004) Job shop scheduling using fuzzy logic, The International Journal of Advanced Manufacturing Technology 23, pp. 606-619

[11] Kılıç S., Kahraman C. (2006) Metaheuristic Techniques for Job Shop Scheduling Problem and a Fuzzy Ant Colony Optimization Algorithm, In: Kahraman C. (eds) Fuzzy Applications in Industrial Engineering. Studies in Fuzziness and Soft Computing, 201. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-33517-X_17

[12] Qun N., Bin J., Xingsheng G. (2008) Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time, Applied Mathematics and Computation 205(1), pp. 148-158

[13] Zhongshi S., Weishi S., Dechang P. (2020) Effective heuristics and metaheuristics for the distributed fuzzy blocking flow-shop scheduling problem, Swarm and Evolutionary Computation, 59, 100747

[14] Tibor D., György D., Ágnes W.,-S. (2018) Multi-Project Optimization with MultiFunctional Resources by a Genetic Scheduling Algorithm, Acta Polytechnica Hungarica 5(4), pp. 101-119

[15] Guohui Z., Jinghe S., Xing L., Guodong W., Yangyang Y. (2019) Solving flexible job shop scheduling problems with transportation time based on improved genetic algorithm. Mathematical Biosciences and Engineering 16(3), pp. 1334-1347, doi: 10.3934/mbe.2019065

[16] Glover, F. W., Kochenberger, G. A. (2006) Handbook of metaheuristics. Springer Science & Business Media, 57

[17] Yang, X., S. (2010) Engineering Optimization: An introduction with metaheuristic applications, University of Cambridge, United Kingdom

[18] Leonid V. L., Bolodurina, I., P. (2020) An Effective Scheduling Method in the Cloud System of Collective Access, for Virtual Working Environments, Acta Polytechnica Hungarica 17(8), pp. 180-195

[19] Balázs M., Soma M., S., Ágota D.,-K. (2021) Application of a Genetic Algorithm for Minimum Zone Method of Flatness, Acta Polytechnica Hungarica 18(3), pp. 107-126

[20] Jang, J., Sun, C., Mizutani, E. (1997) Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, New Jersey, U.S.A.

[21] Marjan S., Vesna O., L. (2020) Feature Space Reduction, using PCA in the Algorithm for Epilepsy Detection, using an Adaptive Neuro-Fuzzy Inference System and Comparative Analysis, Acta Polytechnica Hungarica 17(10), pp 90-107