FUZZY MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS FOR SOLVING MULTI-MODE AUTOMATED GUIDED VEHICLES BY CONSIDERING MACHINE BREAK TIME AND ARTIFICIAL NEURAL NETWORK

H. Nabovati[*], *H. Haleh*[†], *B. Vahdani*[*]

Abstract: In this paper, a novel model is presented for machines and automated guided vehicles' simultaneous scheduling, which addresses an extension of the blocking job shop scheduling problem. An artificial neural network approach is used to estimate machine's breakdown indexes. Since the model is strictly NP-hard and because objectives contradict each other, two developed meta-heuristic algorithms called "fuzzy multi-objective invasive weeds optimization algorithm" and "fuzzy multi-objective cuckoo search algorithm" with a new chromosome structure which guarantees the feasibility of solutions are developed to solve the proposed problem. Since there is no benchmark available on literature, three other metaheuristic algorithms are developed with a similar solution structure to validate performance of the proposed algorithms. Computational results showed that developed fuzzy multi-objective invasive weeds optimization algorithm had the best performance in terms of solving problems compared to four other algorithms.

Key words: *scheduling, AGV, MOIWO, MOCS, ANN*

Received: November 24, 2017 **DOI:** 10.14311/NNW.2018.28.016
Revised and accepted: June 8, 2018

1. Introduction

The novel machine scheduling problem presented in this paper is mainly designed to find an optimal schedule of the machines which work in the environment of a flexible manufacturing system. The main distinctive feature of the model presented in this paper is related to identifying machines and AGVs' simultaneous schedule in which machine breakdown is considered, and machines' operation can use a set of special tools; furthermore each job can be transported, using a set of different AGVs. In addition, the time of transportation between machines is not the same, and it depends on the type of the jobs; a soft due date interval time is considered

[*]Hojat Nabovati; Behnam Vahdani; Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

[†]Hassan Haleh – Corresponding author; Department of Industrial Engineering, Golpayegan University of Technology, Golpayegan, Iran, E-mail: hhaleh@qiau.ac.ir

in the proposed model. An artificial neural network algorithm is used in this paper to estimate the time spent between two machines' breakdowns as well as machines' repair times. The model presented in this paper is aimed to optimize the following objectives: The mathematic model is aimed to optimize the following objectives: minimizing total costs including processing, travelling, earliness and tardiness costs. It is also aimed to minimize total completion time. These objectives are conflicting each other, since decreasing total costs will lead to increasing completion time. In addition, in the next sections, it will be shown that the model presented in this paper is strictly NP-hard and that the exact algorithms are not capable of finding global optimum solution of the model's large-scale problems. So, metaheuristic algorithms will be developed in this paper to solve the model's different test problems in a reasonable computational time. The rest of this paper is organized as follows: Literature review of this paper is presented in the next section. Problem statement, assumptions and the proposed mixed integer non-linear model are included in Section 3. The proposed multi-objective metaheuristic algorithms are discussed in Section 4. Experimental results and the method of calibrating the model and algorithms' major parameters are presented in Section 5. Finally, the paper's conclusion is given in Section 7.

2. Literature review

Machines and AGVs' simultaneous scheduling problem in flexible manufacturing systems' environment has been studied by different researchers. [1] presented a novel mathematical formulation for modelling problem of concurrent scheduling machines and AGVs. The model aimed to identify the optimal schedule of machines and AGVs so that the total completion time of all jobs' is minimized. Since the model was strictly NP-complete, a genetic algorithm was proposed to find near optimal solutions of the model's various benchmark instances. [2] developed a multi-objective mathematical model for machines and AGVs scheduling problem in flexible manufacturing systems' environment. The model aimed to minimize three different objectives including makespan, mean flow time and mean tardiness. Since the model belonged to the class of NP-hard problems and because the model's objectives were in conflict, a novel multi-objective genetic-based algorithm was developed to find the model's feasible solutions. [3] presented a novel mathematical model for the problem of dispatching and routing AGV tools in a flexible manufacturing structure. They used benders decomposition algorithm to find optimal solutions for the model's various test problems. They respectively employed constrained programing and mixed integer programming formulations to decompose the original model into master and sub problems. This method was able to solve instances up to six AGV tools. [4] presented a novel mathematical formulation to model the problem of AGVs in a job shop environment. The main assumption of their model considered all the vehicle's bidirectional movement. In other words, they assumed that several vehicles are allowed to go in the same route. A dispatching algorithm was presented to accelerate vehicles movement and improve transportation efficiency. Finally, a simulation method was conducted to investigate the interactions of the model's major factors and their effect on predefined performance measures. [5] presented a novel mathematical model for mechanized container terminal scheduling problem. The model aimed to identify the optimal schedule of the terminals so that AGVs total travelling time is minimized. They presented a novel heuristic algorithm based on multi-layer genetic algorithm and maximum matching method to find optimal or near optimal solutions of the presented model. [6] developed a mathematical model for job shop scheduling problem with machine and AGVs' simultaneous schedule. The model aimed to identify the optimal schedule of the jobs processed on different machines so that total completion time is minimized. Finally, a neighborhood search mechanism was implemented in three different metaheuristic algorithms including simulated annealing, iterated local search and their hybridized solution procedure to solve 40 different benchmark problems. [7] presented a novel linear formulation for modelling a flexible manufacturing system with one vehicle. They imposed some constraints like buffers' limited capacity, upper bound of the jobs that can be transported at the same time and empty vehicle trips to make the model close to real world conditions. They presented several heuristic algorithms to solve the model's different instances. [8] presented a mathematical formulation for modelling the problem of scheduling machines and AGV tools in FMS systems. The model aimed to identify the optimal schedule of machines and identical AGVs so that total completion time of the jobs processed on different machines is minimized. They developed a differential evolutionary algorithm to find the model's near optimal solutions in a reasonable time. [9] presented a novel mathematical model for no wait multi-robot scheduling problem. They assumed that only two parts can be entered into a special cell at the same time. Additionally, they assumed that several robots can be employed on a single truck to move commodities among machines. They proposed an exact algorithm to identify optimal solutions of the model's various test problems in a reasonable computational time. [10] presented a novel mathematical model for simultaneous scheduling of machines and AGVs. The model aimed to minimize total makespan. They presented a genetic algorithm to find near optimal solution for the developed model. The proposed genetic algorithm was successfully employed for achieving suitable solutions for various 82 benchmark problems. [11] developed a mixed integer linear programming formulation for a cyclic job shop scheduling problem. An AGV tool was employed to change jobs' location and transfer them among various machines. The model is mainly designed to identify the optimal sequence of the jobs processed on different machines so that total cycle time is minimized. They used CPLEX optimization software to find optimal solutions of the benchmark test problems available in literature. [12] developed a mathematical model for a multi-product two-machine manufacturing system where AGVs are used for transporting various types of commodities between machines. The model was mainly designed to identify optimal sequence of AGVs and calculate the time that is required for processing various parts on each machine so that total cycle time is minimized. They proposed a novel two-stage heuristic algorithm to find near optimal solutions for the model's various test problems in a reasonable computational time. [13] proposed a disjunctive graph-based novel mathematical formulation for simultaneous scheduling of machines and AGVs. The novel formulation was presented in the context of jobs shop problem and AGVs were employed to transfer commodities among different machines. The model aimed to minimize total completion time. Finally, they proposed an improved memetic algorithm to solve the model's various test problems. [14] proposed a BJS–AGV problem which used blocking job shop problem by handling jobs between different machines, using a limited quantity of AGVs. The model aimed to minimize total completion time. Two-stage heuristic algorithm that incorporates an improving time tabling method and a local search is proposed. [15] developed a mathematical model for machines and AGV scheduling problem. The model aimed to minimize total tardiness cost. Since the model was strictly NP-hard, a binary particle swarm vehicle algorithm was presented to solve the model. [16] proposed a novel mathematical formulation to model the problem of concurrent scheduling machines and AGVs in FMS structure. The model aimed to minimize the completion time of all jobs. Since the model was strictly NP-hard and because the global optimal solution of the model's large-scale problems could not be obtained in a reasonable time, a Tabu-based heuristic algorithm was developed to solve the model. [17] presented a hybrid genetic algorithm to find near optimal solutions for hybrid scheduling, dispatching routing of the jobs in an FMS structure. The model aimed to optimize conflicting objectives of minimizing total completion time, AGVs' travelling time and tardiness costs. They employed an adaptive weight approach to identify objectives' optimal weight and compute fitness value on each generation. [18] developed a mathematical formulation for AGV-based job shop scheduling problem where AGV tools are mainly employed to transport a set of jobs between machines. They presented a novel genetic and Tabu-search-based metaheuristic algorithm to solve the model. [19] presented a novel mathematical formulation for modelling the problem of machines and mobile robots' concurrent scheduling in a modern manufacturing system. The model aimed to compute optimal schedule of machines and robots so that all the tasks' total completion time is minimized. They presented a genetic-based heuristic algorithm to solve the model's various test problems.

The main motivation of this paper, compared with related researches, is to present a novel mathematical formulation where the machines should be scheduled at the same time as a group of multi-task machines; and tools are required for completing predefined jobs. Also a group of multi-mode AGVs should be used for performing a special job. Furthermore, machines may be faced with failure; downtime and repair time are obtained using ANN, so it can be applied for modelling in real world problems. The summary of the aforementioned recent works is presented in Tab. I.

3. Problem statement

The job shop scheduling problem machine and AGVs, with no buffer constraints, can be described as follows: there is a set of m jobs ($i = 1, \dots, m$) with a set of n operations ($j = 1, 2, \dots, n$) needing to be processed on a set of K machines ($k = 1, 2, \dots, K$) and a set of L tools ($l = 1, 2, \dots, L$). The jobs are handled among different machines by a set of V AGVs ($v = 1, 2, \dots, V$). Manufacturing route of each job on various machines with at least one machine is known. Each AGV can only handle one job at the same time. The processing of a job on a machine is called as a machine operation. AGVs are employed in order to perform a transportation operation between machines M_k and $\mathrm{M}_{k'}$ when two consecutive operations are simultaneously performed on these machines. Since there is no buffer constraint

AUTERS	Simultaneous Scheduling	Routing	Multi-OF	Meta-Heuristic	Heuristic	FMS	Blocking Constraints	Tools Variety	AGV Variety	Machine down-time	ANN	Fuzzy	Solving Approach
Abdelmaguid et al. (2004)	✓	✓		✓	×	✓	×	×	×	×	×	×	GA
Reddy and Rao (2006)	✓	×	✓	✓	×	✓	×	×	×	×	×	×	Genetic Algorithm
Corréa et al. (2007)	✓	✓	×	×	×	✓	×	×	×	×	×	×	Hybrid, Constraint Programming
Kesen and Baykoç (2007)	✓	✓	×	×	×	✓	×	×	×	×	×	×	Simulation
Lau and Zhao (2008)	×	✓	×	×	✓	✓	×	×	×	×	×	×	Multi-layer Genetic Algorithm
Deroussi et al. (2008)	×	×	✓	✓	✓	✓	×	×	×	×	×	×	Heuristic based on SA
Brauner et al. (2009)	×	×		×	✓	✓	✓	×	×	×	×	×	local search
Gnanavel Babu et al. (2010)	✓	×	×	✓	×	✓	×	×	×	×	×	×	Evolutionary algorithm
Che et al. (2011)	×	×		×	×	✓	✓	×	×	×	×	×	GAMS
Chaudhry et al. (2011)	✓	×	×	✓	×	✓	×	×	×	×	×	×	Genetic Algorithm
Brucker et al. (2012)	✓	×	×	×	×	✓	×	×	×	×	×	×	CPLEX
Batur et al. (2012)	✓	×	×	×	✓	✓	×	×	×	×	×	×	Two Stage Heuristic
Lacomme et al. (2013)	✓	×	×	×	✓	✓	×	×	×	×	×	×	Memetic Algorithm
Zeng et al. (2014)	✓	×	×	×	✓	✓	✓	×	×	×	×	×	Two-stage Heuristic
Narendranath et al. (2014)	✓	×	✓	✓	×	✓	×	×	×	×	×	×	Particle Swarm Optimization
Nageswararao et al (2014)	✓	×	×	×	✓	✓	×	×	×	×	×	×	Binary Particle Swarm
Umar et al. (2015)	✓	×	✓	✓	×	✓	✓	×	×	×	×	×	Hybrid Genetic Algorithm
Mirabi (2015)	×	×	✓	✓	✓	✓	×	×	×	×	×	×	Hybrid Genetic Algorithm
Dang and Nguyen (2016)	✓	×	×	✓	×	×	×	×	×	×	×	×	Genetic Algorithm
Nouri et al. (2016)	✓	×	✓	✓	×	✓	×	×	×	×	×	×	Genetic Algorithm, Tabu Search
Salido et al. (2016)	×	×	✓	✓	×	✓	×	×	×	×	×	×	Heuristic
Al-Refaie, et al. (2017)	×	×	✓	✓	×	✓	×	×	×	×	×	✓	Heuristic
Rahman, et al. (2017)	×	×	×	✓	×	✓	×	×	×	×	×	×	Memetic Algorithm
The current work	✓	×	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	FMOIW, FMOCS, MOPSO, MOTLBO, NSGA-II

Tab. I *Summary of vehicles and machine's simultaneous scheduling problems.*

condition, the machine will be blocked until the product that is processed on the machine is released. A time interval should be imposed to the model to consider the waiting time of an AGV's arrival to transport the finished job. AGVs should wait until the next machine is released. The job will be unloaded on the machine when the next machine is released. An AGV should wait near the machine to transport a job when the machine is released. In addition, a job will be loaded on the next released machine. Various operations of each job are able to use a set of predefined tools. A special set of AGVs should be assigned for transporting each job and transportation time between the machines is independent from job types. The model includes soft interval due date. The mathematic model is aimed to minimize total costs including processing costs, travelling costs, earliness and tardiness costs, as well as minimizing total completion time. These objectives are in conflict since decreasing total costs will lead to increasing completion time. The main assumptions of the proposed mixed integer non-linear model is presented as follow.

An artificial neural network algorithm is used in this paper to estimate the machines' breaking and repair times. To do so, the initial data of the previous periods are given to MATLAB software to estimate the machines' breaking times in the middle of executing solution algorithms. As an obvious fact, a time interval will be spent for repairing the machines which are broken in the middle of performing operations. So, artificial neural network algorithm will be used to estimate the broken machines' repairing time rather than computing their breaking time. This procedure will be implemented by an artificial neural network algorithm on MATLAB software to estimate broken machines' next breaking and repairing times. The flowchart is shown in Fig. 1.

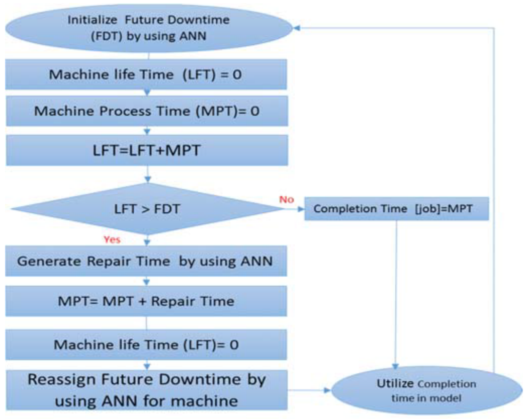

Fig. 1 *Flowchart of machine downtime and repair time assignment.*

3.1 Assumptions

The main assumptions of the proposed mixed integer non-linear model are presented as follows: Jobs' operation can use a set of special tools; each job can be transported using a set of AGVs. A soft due date interval time and earliness and lateness times are considered in the proposed model. The jobs cannot be interrupted. The time required for transporting jobs between various machines is not the same. Different operations of each job can use a special predefined set of AGVs and the time of transportation between machines depends on the type of the job. Machines' failure and repair time is considered in the model.

3.2 Indices

j index of operations	$j = 1, 2, \ldots, n$
i index of jobs	$i = 1, \ldots, m$
k index of machines	$k = 1, 2, \ldots, K$
v index of AGVs	$v = 1, 2, \ldots, V$
l index of tools	$l = 1, 2, \ldots, L$
s index of failures	$s = 1, 2, \ldots, S$

3.3 Parameters

PT_{ijkl}	Processing time of operation j of job i on machine k by tool l
PC_{ijkl}	Processing cost of operation j of job i on machine k by tool l
LD_i and UD_i	Due date's upper bound and lower bound job i
HC_v	Handling cost by AGV v
R_{ij}	Is 1 when job i requires operation j; otherwise *0*
A_{ijk}	Is 1 when an operation j job i can be processed on machine k; otherwise *0*
B_{ijl}	Is 1 if tool l can perform operation j job i; otherwise *0*
$t_{kk'v}$	AGV_V transportation time between two machines k and k'
t_{L_Okv}	AGV_V transportation time between loading area and machines k
t_{U_nkv}	AGV_V transportation time between unloading area and machines k
$E(MTTR)_{sk}$	Expected value mean time to repair failure s of machine k
$E(MTBF)_{sk}$	Expected value mean time to repair failure s of machine k
W	large number

3.4 Decision variables

Tr_i and Er_i	Earliness and tardiness
S_{ij}	Operation's start time
St_{ij}	Time of beginning transportation between operations j and $j-1$
C_{ij}	Operation's finishing time
Ct_{ij}	Finishing transportation time between operations j and $j-1$
CP_i	Finishing time of job i's last operation
PO_{iv}	Is 1 if AGV v can transport job i; otherwise *0*
Y_{ijkl}	Is 1 if tool l can process operation j job i on machine k; otherwise *0*
x_{ijv}	Is 1 if AGV v transport a predefined job between operations j and $j-1$; otherwise *0*
Mk_s	Breakdown time downtime s of machine k
Z_{ijks}	Is 1 if beak s is occurred in machine k in the middle of performing operation j of job i; 0 otherwise

EP_i Is 1 if job i is performed earlier than its predefined due date; 0 otherwise

TP_i Is 1 if job i is performed later than its predefined due date; 0 otherwise

3.5 Mathematical model

3.5.1 Objective functions

The model presented in this paper includes two different conflicting objectives. The model's first objective which is mainly designed to minimize manufacturing system's total costs includes two different component defined as:

$$\text{Process Cost} = \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{l=1}^{l}\sum_{j=1}^{O_j} Y_{ijkl} \cdot P_{ijkl} \cdot PC_{ijkl}, \tag{1}$$

$$\begin{aligned}\text{Travel Cost} \quad = \quad & \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{l=1}^{l}\sum_{l'=1}^{l}\sum_{v=1}^{v}\sum_{j=2}^{o_j} HC_v \cdot Y_{ijk'l'} \cdot Y_{iO_{j-1}kl} \cdot t_{k'kv} \cdot x_{ijv} + \\ & + \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{l=1}^{l}\sum_{v=1}^{v} HC_v \cdot Y_{i1kl} \cdot t_{L_okv} \cdot x_{i1v} + \sum_{i=1}^{n}\sum_{k=1}^{k}\sum_{l=1}^{l}\sum_{v=1}^{v} HC_v \cdot \\ & \cdot Y_{iO_{j+1}kl} \cdot t_{U_nkv} \cdot x_{iO_{j+1}v},\end{aligned} \tag{2}$$

$$\text{Erliness Cost} = \sum_{i=1}^{i} Er_i \cdot CE_i, \tag{3}$$

$$\text{Tardiness Cost} = \sum_{i=1}^{i} Tr_i \cdot CT_i. \tag{4}$$

Total cost will be obtained by adding up following components. So, we have:

$$\text{Cost} = \text{Travel Cost} + \text{Process Cost} + \text{Earliness Cost} + \text{Tardiness Cost}.$$

The second objective which is aimed to minimize total completion time is calculated as:

$$\text{Total completion time (makespan)} = C_{Max} \geq C_{ij} \qquad \forall ij. \tag{5}$$

3.5.2 Constraints

The model's constraints are presented as:

$$\sum_{v=1}^{V} x_{ijv} = 1 \qquad \forall i = 1, \ldots, m \qquad j = 1, \ldots, n+1, \tag{6}$$

$$C_{ij} = S_{ij} + \sum_{k=1}^{K}\sum_{l=1}^{L} P_{ijkl} \cdot Y_{ijkl} + \sum_{s=1}^{s'}\sum_{k=1}^{K} E(MTTR)_{sk} \cdot Z_{ijsk} \cdot \sum_{l=1}^{L} Y_{ijkl} \qquad \forall i = 1, \ldots, m \qquad j = 1, \ldots, n, \tag{7}$$

$$St_{ij} \geq 0 \qquad \forall i = 1, \ldots, m, \qquad j = 1, \tag{8}$$

$$St_{ij} \geq C_{i(j-1)} \qquad \forall i = 1, \ldots, m, \qquad j = 2, \ldots, n+1, \tag{9}$$

$$S_{ij} \geq St_{ij} + \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{v=1}^{V} Y_{ijkl} \cdot t_{L_O kv} \cdot x_{ijv} \qquad \forall i = 1, \ldots, m, \qquad j = 1, \tag{10}$$

$$S_{ij} \geq St_{ij} + \sum_{k=1}^{K} \sum_{k'=1}^{K} \sum_{l=1}^{L} \sum_{l'=1}^{L} \sum_{v=1}^{V} Y_{i(j-1)k'l'} \cdot Y_{ijkl} \cdot t_{kk'v} \cdot x_{ijv} \quad \forall i = 1, \ldots, m, \qquad j = 2, \ldots, n, \tag{11}$$

$$Ct_{ij} = St_{ij} + \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{v=1}^{V} Y_{ijkl} \cdot t_{L_O kv} \cdot x_{ijv} \quad \forall i = 1, \ldots, m, \qquad j = 1, \tag{12}$$

$$Ct_{ij} = St_{ij} + \sum_{k=1}^{K} \sum_{k'=1}^{K} \sum_{l=1}^{L} \sum_{l'=1}^{L} \sum_{v=1}^{V} Y_{i(j-1)k'l'} Y_{ijkl} \cdot t_{kk'v} \cdot x_{ijv} \quad \forall i = 1, \ldots, m, \qquad j = 2, \ldots, n, \tag{13}$$

$$Ct_{ij} = St_{ij} + \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{v=1}^{V} Y_{i(j-1)kl} \cdot t_{kU_o v} . x_{ijv} \quad \forall i = 1, \ldots, m, \qquad j = n+1, \tag{14}$$

$$x_{ijv} \cdot x_{i'j'v} \cdot \left(St_{ij} - S_{i'j'} + \sum_{k'=1}^{K} \sum_{l'=1}^{L} Y_{i'j'k'l'} \cdot t_{L_O k'v} \right) \cdot \left(St_{i'j'} - S_{ij} + \sum_{k=1}^{K} \sum_{l=1}^{L} Y_{ijkl} \cdot t_{L_O kv} \right) \leq 0$$
$$\forall i, i' = 1, \ldots, m, \quad i \neq i', \quad v = 1, \ldots, V, \quad j = 1, j' = 1, \tag{15}$$

$$x_{ijv} \cdot x_{i'j'v} \cdot \left(St_{ij} - S_{i'j'} - \sum_{k'=1}^{K} \sum_{l'=1}^{L} Y_{i'j'k'l'} \cdot t_{L_O k'v} \right) \cdot \left(St_{i'j'} - S_{ij} + \sum_{k=1}^{K} \sum_{k'=1}^{K} \sum_{l=1}^{L} \sum_{l'=1}^{L} Y_{ijkl} \cdot Y_{i'(j'-1)k'l'} \cdot t_{kk'v} \right) \leq 0$$
$$\forall i, i' = 1, \ldots, m, i \neq i', \; v = 1, \ldots, V, \; j = 1, j' = 2, \ldots, n, \tag{16}$$

$$x_{ijv} \cdot x_{i'j'v} \cdot \left(St_{ij} - St_{i'j'} - \sum_{k'=1}^{K}\sum_{l'=1}^{L} Y_{i'j'k'l'} \cdot t_{k'U_n v}\right) \cdot$$
$$\cdot \left(St_{i'j'} - S_{ij} - \sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{l=1}^{L}\sum_{l'=1}^{L} Y_{ijkl} \cdot Y_{i'(j'-1)k'l'} \cdot t_{kk'v}\right) \leq 0$$
$$\forall i, i' = 1, \ldots, m, i \neq i', v = 1, \ldots, V, \qquad j = 1, j' = n+1, \quad (17)$$

$$x_{ijv} \cdot x_{i'j'v} \cdot \left(St_{ij} - S_{i'j'} - \sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{l=1}^{L}\sum_{l'=1}^{L} Y_{i'j'k'l'} \cdot Y_{i(j-1)kl} \cdot t_{k'kv}\right) \cdot$$
$$\cdot \left(St_{i'j'} - S_{ij} + \sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{l=1}^{L}\sum_{l'=1}^{L} Y_{ijkl} \cdot Y_{i'(j'-1)k'l'} \cdot t_{kk'v}\right) \leq 0$$
$$\forall i, i' = 1, \ldots, m, i \neq i', \quad j = 2, \ldots, n, \quad j' = 2, \ldots, n, \qquad (18)$$

$$x_{ijv} \cdot x_{i'j'v} \cdot \left(St_{ij} - St_{i'j'} - \sum_{k'=1}^{K}\sum_{l'=1}^{L} Y_{i'(j'-1)k'l'} \cdot t_{L_O k'v} - \sum_{k=1}^{K}\sum_{l=1}^{L} Y_{i(j-1)kl} \cdot t_{L_O kv}\right) \cdot$$
$$\cdot \left(St_{i'j'} - S_{ij} - \sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{l=1}^{L}\sum_{l'=1}^{L} Y_{ijkl} \cdot Y_{i'(j'-1)k'l'} \cdot t_{kk'v}\right) \leq 0$$
$$\forall i, i' = 1, \ldots, m, i \neq i', \quad j = 2, \ldots, n, \quad j' = n+1, \qquad (19)$$

$$x_{ijv} \cdot x_{i'j'v} \cdot \left(St_{ij} - St_{i'j'} + \sum_{k'=1}^{K}\sum_{l'=1}^{L} Y_{i'(j'-1)k'l'} \cdot t_{L_O k'v} - \sum_{k=1}^{q}\sum_{l=1}^{q} Y_{i(j-1)kl} \cdot t_{L_O kv}\right) \cdot$$
$$\cdot \left(St_{i'j'} - St_{ij} + \sum_{k=1}^{K}\sum_{l=1}^{L} Y_{i(j-1)kl} - \sum_{k'=1}^{K}\sum_{l'=1}^{L} Y_{i'(j'-1)k'l'} \cdot t_{L_O k'v}\right) \leq 0$$
$$\forall i, i' = 1, \ldots, m, i \neq i', \quad j = n+1, j' = n+1, \qquad (20)$$

$$Y_{ijkl} \cdot Y_{i'j'k'l'} \cdot \left(S_{ij} - St_{i'(j'+1)}\right)\left(S_{i'j'} - St_{i(j+1)}\right) \leq 0 \quad \forall i, i' = 1, \ldots, m, i \neq i',$$
$$k, k' = 1, \ldots, K, l, l' = 1, \ldots, L, \quad j = 1, \ldots, n, j' = 1, \ldots, n, \qquad (21)$$

$$Y_{ijkl} Y_{i'j'k'l'} \left(S_{ij} - St_{i'(j'+1)}\right)\left(S_{i'j'} - St_{i(j+1)}\right) \leq 0 \quad \forall i, i' = 1, \ldots, m, i \neq i',$$
$$k, k' = 1, \ldots, K, \quad l, l' = 1, \ldots, L, \quad j = 1, \ldots n-1, j' = 1, \ldots n-1, \qquad (22)$$

$$Y_{ijkl}Y_{i'j'k'l'}\left(S_{ij}-C_{i'j'}\right)\left(S_{i'j'}-St_{i(j+1)}\right)\leq 0\ quad\forall i,i'=1,\ldots,m,i\neq i',$$
$$k,k'=1,\ldots,K,\quad l,l'=1,\ldots,L,\quad j=1,\ldots n-1,j'=n, \tag{23}$$

$$Y_{ijkl}\cdot Y_{i'j'k'l'}\cdot\left(S_{ij}-C_{i'j'}\right)\left(S_{i'j'}-C_{ij}\right)\leq 0\quad\forall i,i'=1,\ldots,m,i\neq i',$$
$$k,k'=1,\ldots,K,\quad l,l'=1,\ldots,L,\quad j=n,j'=n, \tag{24}$$

$$CP_i\geq Ct_{ij}\quad\forall i=1,\ldots,m,\quad j=1,\ldots,n, \tag{25}$$

$$x_{ijv}\leq PO_{iv}\quad\forall i=1,\ldots,m,\quad v=1,\ldots,V, \tag{26}$$

$$\sum_{l=1}^{L}Y_{ijkl}\leq A_{ijk}\quad\forall i=1,\ldots,m,\quad j=1,\ldots,n,\quad k=1,\ldots,K, \tag{27}$$

$$\sum_{k=1}^{K}Y_{ijkl}\leq B_{ijl}\quad\forall i=1,\ldots,m,\quad j=1,\ldots,n,\quad l=1,\ldots,L, \tag{28}$$

$$Er_i\geq LD_i-CP_i\quad\forall i=1,\ldots,m, \tag{29}$$

$$Tr_i\geq CP_i-UD_i\quad\forall i=1,\ldots,m, \tag{30}$$

$$Er_i\leq EP_i\times W\quad\forall i=1,\ldots,m, \tag{31}$$

$$Tr_i\leq TP_i\times W\quad\forall i=1,\ldots,m, \tag{32}$$

$$Lf_{1k}=MTBF_{1k}\left(1-\sum_{i=1}^{n}\sum_{j=1}^{Oj}\sum_{l=1}^{L}Z_{ijk1}\times Y_{ijkl}\right), \tag{33}$$

$$Lf_{s+1k}\left(1-\sum_{i=1}^{n}\sum_{j=1}^{Oj}Z_{ijks+1}\right)=\sum_{i=1}^{n}\sum_{j=1}^{Oj}Z_{ijks+1}\times$$
$$\times MTBF_{s+1k}\left(1-\sum_{i=1}^{n}\sum_{j=1}^{Oj}Z_{ijks}\right)\forall k,s, \tag{34}$$

$$Lf_{1k}-\sum_{l=1}^{L}P_{ij}\cdot Y_{ijkl}\geq -Z_{ijks}\times W\quad\forall i,j,k,s, \tag{35}$$

$$Lf_{sk}-\sum_{i=1}^{n}\sum_{j=1}^{Oj}\sum_{l=1}^{L}P_{ij}\cdot Y_{ijkl}\times Z_{ijks-1}=LF_{sk}\quad\forall k,s, \tag{36}$$

$$\sum_{k=1}^{K}\sum_{l=1}^{L}Y_{ijkl}\leq R_{ij}\quad\forall i=1,\ldots,m,j=1,\ldots,n, \tag{37}$$

$$\sum_{k=1}^{K}\sum_{l=1}^{L} Y_{ijkl} \leq \sum_{v=1}^{V} x_{ijv} \quad \forall i = 1, \ldots, m, j = 1, \ldots, n. \tag{38}$$

Eq. 6 ensures that a transportation operation cannot be performed by more than one AGV. Eq. 7 calculates completion times of each operation by considering repair time. Eq. 8 refers to the transfer start time from the loading station to the first machine. Eq. 42 refers to the AGV's starting transporting time when the machine has completed the job. Eqs. 43 ensures that the first activity's processing procedure should be started when it is completely transported from loading station. Eqs. 44 refers to the relations between the start time for the transportation operation and the processing start time for operation j of job i, Eqs. 45–14 compute finishing time of transporting operation. Eqs. 15–20 are imposed to the model to avoid performing two different operations by a single AGV at the same time. Eqs. 21–24 enforce the model to prevent performing two various operations by a unique machine at the same time. Eq. 25 compute finishing time of the last operation performed on job i. Eqs. 26 assigning a transportable AGV to job i. Eqs. 27 assign a useable tool for job i. Eqs. 28 assign a machine useable for job i. Eqs. 29–32 calculate the earliness and tardiness time. Eq. 33–36 are used to compute machines' failure times. Eq. 37 assigns job's operations requirement. Eq. 38 assigns AGV's operations for transportation.

4. The proposed multi-objective solving methods

4.1 Developed MOIWO algorithm

The multi-objective version of the population-based invasive weeds optimization algorithm was developed by [20]. In this algorithm, a collection of weeds are considered as initial population. The behavior of the weed for occupying soil and generating new colony is based on their pruning system, they first try to obtain appropriate farmlands and employ pruning system to produce new colonies. This action is continued to solve optimization problems. Conforming to the procedure, the seeds are scattered in specified pasture and are turned into weeds. New colony of weeds are generated around their parent weeds. The weeds which are grown in a more arable region will have superior competency and more luck to survive. Therefore, higher breeding will be done in vicinity of these potent weeds. It is important that, by enhancing iteration number, the distance of newly generated weeds from the parent weeds lessen systematically. At the beginning of the algorithm, a far distance allows diversification of generated solutions all over the search space, so long that by progressing the algorithm, intensification will be dominated by dynamically lessening distance from the parent weeds. Standard deviation of the weeds that are recently produced in previous generations can be employed as a basis for confining the greedy nature of the algorithm. In addition, the following consecutive steps can be used in the algorithm in order to:

- Produce initial solutions and asses objective function.
- Employ fuzzy ranking procedure to sort population.

- Permit every member of the population to make several seeds with preferable population members. The seeds will be produced according to Eq. 39

$$\text{seeds}_i = \text{floor}\left(S_{\min} + (S_{\max} - S_{\min}) * \left(\frac{np - rank_i}{np}\right)\right). \tag{39}$$

  In which, $S_{\min}$ and $S_{\max}$ are minimum and maximum numbers of seeds produced by each weed. rank_i is the rank of the i-th population member and seeds_i is the number of seeds produced by it. np is the number of population.

- Breed according to merit and update standard deviation. The formula of calculating standard deviation is shown in Eq. 40.

$$\sigma_{iter} = (iter_{\max} - iter)^n * \frac{\sigma_{\text{initial}} - \sigma_{\text{final}}}{(iter_{\max})^n} + \sigma_{\text{final}}, \tag{40}$$

  where $iter_{\max}$ and $iter$ are maximum and current iteration, σ_{initial} and σ_{final} are initial and final $Sigma$. n is a nonlinear multiplier whose value, during the execution of the algorithm, is changed from the initial value to the final value.

The following steps can be used to identify the discounting measure of the proposed FMOIWO algorithm:

1. Initialize a ratio with members and evaluate them
2. Employ a fuzzy ranking method to rank population members
3. Let the members of the population to make multiple seeds. Make a larger number of seeds with more proportionate members of population by Eq. 39.
4. Scatter the constructed seeds on the search space, applying the numbers that are normally distributed with mean equivalent to zero and standard deviation, calculated by Eq. 40.
5. As the population of the weeds gets more than the upper limit (using fuzzy ranking and retain the best number of members).
6. Repeat the process until the discontinuing criterion is met.

4.1.1 Fuzzy dominance-based sorting

Computing fuzzy dominance of solutions in the population is considered as a primary phase of sorting on the basis of fuzzy dominance. Afterwards, solutions are sorted in an ascending order, using fuzzy dominance. Next, according to the importance of the largest crowding distance, solutions are sorted, descending concerning diversity. Calculating the largest n-dimensional hypercube over objective space per solution can be used as a basis for enhancing hyper cube's capability of finding analogous solutions. Eq. (8) was used by [20] as a basis for obtaining the periphery of each solution's folded cube.

$$I(\vec{v}) = \sum_{i=1}^{n} (y_i(\vec{u}) - y_i(\vec{w}))/(\max(y_i) - \min(y_i)), \tag{41}$$

where $\vec{u}$ and $\vec{w}$ are two adjacent solutions to the answer $\vec{v}$. The population is arranged according to the i-th objective function and in an ascending order. Obviously, the greater the dispersion of the member $I(\vec{v})$, the more priority is given. And this dispersion is calculated for the unsatisfactory answers in the archive. The parameters $\min(yi)$ and $\max(yi)$ are the minimum and maximum values of the i-th objective function. Eq. (41), contributes to normalization; boundary solutions contain the extreme of the assigned value. Aggregation of both solutions that were placed in sparse region and the solutions which have not been dominated by other existing solutions are placed in archive. The main procedure of employing fuzzy dominance mechanism on proposed FMOIWO algorithm for obtaining optimal solutions is schematically shown in Alg. 1, where $\mu(i)$ is the degree of fuzzy i-dominance (dom), $(\vec{x_i})$ is called the decision parameter vector, y is the objective vector, $\prec_i^F$ sign of dominate, $k = 1 : n$ is objective functions index, $j = 1 : m$ is decision variables (parameters) index.

Algorithm 1 Fuzzy dominance sorting.

for $k = 1 : n$ **do** `// calculate fuzzy J-dominance per solution in population`
  $\mu(k) = 0;$
  for $i = 1 : n$ **do**
    $\mu(i) = 1;$
    for $j = 1 : m$ **do** `// compute Fuzzy J-dominance per solution`
      if $y_j(\vec{x_i}) - y_j(\vec{x_k}) < 0$ **then**
        $\mu_i^{dom}\left(\vec{x_k} \prec_i^F \vec{x_i}\right) = 0$
      else if $y_j(\vec{x_i}) - y_j(\vec{x_k}) < p_j$ **then**
        $\mu_i^{dom}\left(\vec{x_k} \prec_i^F \vec{x_i}\right) = (y_j(\vec{x_i}) - y_j(\vec{x_k}))/p_j$
      else
        $\mu_i^{dom}\left(\vec{x_k} \prec_i^F \vec{x_i}\right) = 1$
        $\mu^{dom}\left(\vec{x_k} \prec^F \vec{x_i}\right) = \mu^{dom}\left(\vec{x_k} \prec^F \vec{x_i}\right) * \mu_i^{dom}\left(\vec{x_k} \prec_i^F \vec{x_i}\right)$
      end if
    end for
$\mu(k) = \mu(k) + \mu^{dom}\left(\vec{x_k} \prec^F \vec{x_i}\right) - \mu(k) * \mu^{dom}\left(\vec{x_k} \prec^F \vec{x_i}\right)$
  end for
end for

4.2 Developed MOCS algorithm

The cuckoo search algorithm was initially proposed by [21]. The main concept of this algorithm was inspired from the daily behaviour of cuckoo birds and the cuckoos' habit of putting their eggs on the nests of other types of birds. So, two different possible adventures can be assumed for these eggs. As a first adventure, the bird will recognize cuckoo's eggs and will eradicate it or will leave the nest. As a second adventure, the eggs will not be recognized due to their being similar to other eggs and the hatched cuckoo chick will eradicate the other eggs. Cuckoo birds will distribute their eggs on different regions. Moreover, they would like to put their eggs on the nests which have a lower cuckoo egg eradication rate. As a result, oncoming generations' cuckoo birds will be interested in putting their eggs in the nests with a lower eradication rate. This mechanism will ensure algorithm's convergence. So, cuckoo's movements in each iteration can be divided into two main parts including levy flight movement and a special pattern based on random movement.

4.2.1 Levy flight movement

In cuckoo search algorithm, each solution will move toward a region which includes leader solution and will search around the region. This movement will be performed based on the following equation:

$$\text{nest}_i^{t+1} = \text{nest}_i^t + \alpha r S \left(\text{nest}_i^t - \text{Leader}_i^t\right). \tag{42}$$

In the above equation, nest_i^t, nest_i^{t+1}, Leader_i^t and α are respectively the position of i-th cuckoo bird on iterations t and $t+1$, the position of the i-th solution up to iteration t and step size. S is a number that is calculated according to $S = u \Big/ |\nu|^{1/\beta}$ formula. β is movement radius and νsr are respectively random normal numbers with zero mean and their variance respectively equals to $\sigma_{V'}^2 \sigma_{u'}^2 \sigma^2$ where $\sigma_\nu = 1$ and σ_u is computed according to the following equation:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right]\cdot\beta\cdot 2^{(\beta-1)/2}} \right\}^{1/\beta}. \tag{43}$$

4.2.2 Especial pattern based random movement

Each metaheuristic algorithm should employ a random search method aiming to discover new solution regions and move from local solution toward the solutions with higher quality. Cuckoo search algorithm includes a special pattern based on random movement mechanism that is employed to search new areas and find solutions with higher quality. According to this mechanism, two different solutions are chosen randomly, and then the solution will move with λ probability, and the length will be equal to maximum distance between two selected solutions toward a better position. The random search mechanism will be performed according to the following equations:

$$x = rand\left(\text{nest}_j^t - \text{nest}_k^t\right), \tag{44}$$

$$\text{nest}_i^{t+1} = \text{nest}_i^t + P \cdot x, \tag{45}$$

$$P_{ij} = \begin{cases} 1 & \text{if} \quad rand < \lambda \\ 0 & \text{if} \quad rand \geq \lambda \end{cases}. \tag{46}$$

In the above equations, nest_j^t and nest_k^t which will be chosen randomly are respectively positions of solutions j and k.

4.2.3 Fuzzy multi-objective cuckoo search (FMOCS)

Fuzzy multi-objective cuckoo search is developed in this paper; the solutions obtain based on the fuzzy ranking method. First, the initial population is generated randomly and evaluated. Then, rank them by using fuzzy dominance-based sorting. Computing fuzzy dominance of solutions in the population is considered as a primary phase of sorting on the basis of fuzzy dominance (according to Fig. 2 fuzzy dominance sorting). The levy flight movement and the random movement, are employed to produce the members of the next generations. The levy flight movement is applied to select the leader solution of the randomly generated non-dominated members. The framework of developed FMOCS is shown in Fig. 2.

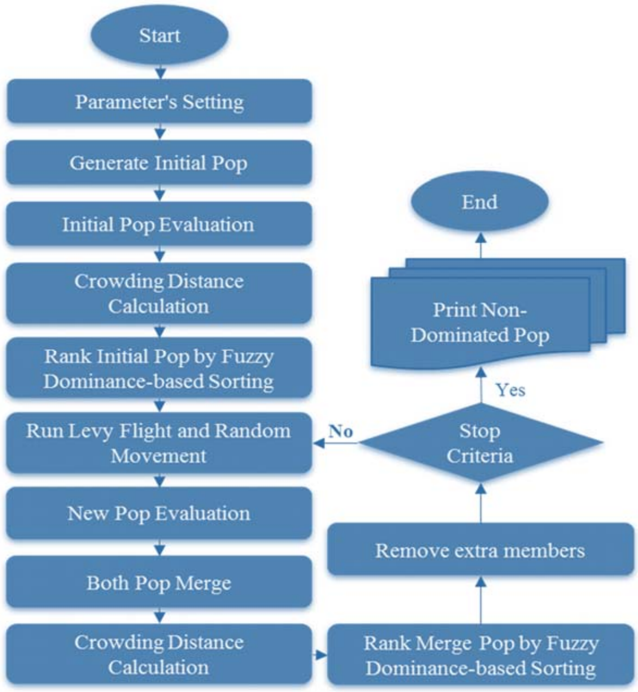

Fig. 2 *The frame work of developed FMOCS.*

4.3 Alternative approaches

As there is no benchmark available to validate the performance of the Cuckoo algorithm, two other meta-heuristic algorithms named NSGA-II (non-dominated sorting genetic algorithm) and MOTLBO (multi-objective teaching learning based optimization algorithms), MOPSO (Multi-objective particle swarm optimization algorithm) were used, with the same coding method as Cuckoo for comparison in the two subsequent sections.

NSGA-II was introduced by [22]. In this algorithm, non-dominance and crowding distance concepts are employed to sort population members. In this article, uniform and single point crossover also swap mutation operators used as intensification and diversification mechanisms for generating the solutions of the generation. Then, the members of the current and the previous generations are mixed. At the end, two popular techniques of the multi-objective genetic algorithms named non-dominance and crowding distance techniques are employed to find the most appropriate solutions obtained in each iteration.

MOTLBO was developed by [23] for solving the multi-objective optimization of heat exchangers. The main mechanism of this algorithm was inspired from teachers and students' behaviour. This algorithm is performed in two different phases called 'teaching phase' and 'learning phase' to elaborate the quality of the solutions generated on each solution, and to seek other solution spaces for the purpose of finding solutions with higher quality. So, teaching and learning phases are used respectively as diversification and intensification mechanisms on each iteration to generate better solutions.

Multi-objective particle swarm optimization algorithm was presented for solving multi-objective problems by [24]. At first some solutions are generated randomly, afterwards update position is used for improving generated solutions. Non-dominated solutions are stored in a bounded archive called repository. The new generated solutions are added to repository. The solutions which are dominated due to non-dominance technique are removed from repository.

4.4 Artificial neural network

In this paper, artificial neural networks are used to estimate machine downtime and machine repair time. The neural networks used in this paper are of a multilayer perceptron type. In many of the complex mathematical problems that lead to the solution of nonlinear complexity problems, a multilayer perceptron network can simply be used to define weights and functions. Different functions are used for the problem model in neurons. In these types of networks, an input layer is used to apply the problem inputs of a hidden layer and an output layer that ultimately provides problem responses. The nodes in the input layer, the sensory neurons, and the nodes of the output layer are the neurons of the responder. In the hidden layer, there are hidden neurons. The training of such networks is usually done by the method of post-propagation error [25].

To predict the time between two failures and repair times, first, 80 % of the previous data are used to train the neural network; and then 20 % of the previous data is used to test the neural network. Kumar proposed that in the general problem, one hidden layer is used for forecasting tasks, so we considered one hidden layer [26]. To specify the number of hidden nodes, Zhang believed that for better result forecasting, the number of the hidden nodes must be equal to the number of input (denoted by "n") [27]; but Hecht-Nielsen proposed $2n + 1$ [28]. Therefore, in this article, because we have one input ($n = 1$), for minimizing mean square error (MSE), the program was run ten times by considering the number of hidden nodes between 1 and 3; and it was realized that the best result is two. In determining the network structure for the problem, a combination of a trial and error test method and a Bayesian method was used. It is worth mentioning that the steps were taken using the 2011 version of the MATLAB software.

4.5 Performance metrics

Four different metrics are considered for performance evaluation of the developed meta-heuristic algorithms [29]. These metrics are determined as follows:

- *Mean Ideal Distance (MID):* This metric is mainly used to calculate closeness distance of obtained non-dominated solutions from ideal solution. The solution with lower distance will have more quality. The value of this metric can be calculated by equation (47)

$$MID = \frac{\sum_{i=1}^{n'} \sqrt{\left(\frac{f1_i - f1_{\text{best}}}{f1_{\text{total}}^{\max} - f1_{\text{total}}^{\min}}\right)^2 + \left(\frac{f2_i - f2_{\text{best}}}{f2_{\text{total}}^{\max} - f2_{total}^{\min}}\right)^2}}{n}. \tag{47}$$

  where the number of non-dominated solutions, and the smallest and largest values of non-dominated solutions are respectively shown by n, $f2_{\text{total}}^{\min}$ and $f1_{\text{total}}^{\max}$. The solution algorithm with lower values of MID metric will be able to generate solutions with lower distance from ideal solutions. So, the algorithm with lower value of this metric can generate better solutions.

- *Spacing metric (SM):* This metric is mainly used to calculate uniformity of the resulted non-dominated solutions. The algorithm with a lower value of this metric can generate better solutions. the following formula can be used for calculating the values of this metric:

$$SM = \frac{\sum_{i=1}^{n'} \left|\overline{d} - d_i\right|}{(n'-1)\,\overline{d}}, \tag{48}$$

  where the number of non-dominated solutions, lowest Euclidean distance of solution i with other existing non-dominated solutions and the average of this distance are respectively shown by n, d_i and $\overline{d}$.

- *Diversification Metric (DM):* The following formula can be used for calculating the values of this metric:

$$DM = \sqrt{\sum_{i=1}^{I} (\min f_i - \max f_i)^2}, \tag{49}$$

  where maximum and minimum values of the first and second objectives are respectively shown by $f_{1\,\text{total}}^{max}$, $f_{1\,\text{total}}^{\min}$, $f_{2\,\text{total}}^{\max}$ and $f_{2\,\text{total}}^{\min}$.

- *Spread of non-dominance solution (SNS):* This metric is mainly known as diversity metric. The algorithm with a higher value of this metric is able to obtain better solutions. The following formula can be used for calculating the value of this metric:

$$SNS = \sqrt{\frac{\sum_{i=1}^{n'} (MID - C_i)^2}{n'-1}}, \tag{50}$$

where $C_i = \sqrt{f_{1i}^2 + f_{2i}^2}$ and the values of the first and second objectives are respectively shown by f_{1i}, f_{2i}.

- *CPU time:* returns the total CPU time (in seconds) used by MATLAB soft wear from the time it was started.

4.6 Solution coding method

Each solution structure has four rows and a number of columns (the number of columns is in accordance with the numbers of jobs and jobs' operations). The chromosome structure contains random key real numbers between 0 and 1. The decoding procedure assures the feasibility of the obtained solution. Due to the assumptions in the proposed model, the solution contains several assignments, therefore the following substantial points were considered in presenting solution's representation: jobs, machines, tools and AGVs assignment. For decoding chromosome, first q-column is separated (q = number of job's operations); this way, jobs, machines, tools and AGVs are assigned. Afterwards, the remaining p columns are used for AGV allocation for transportation to the loading and unloading areas (p = number of job). The procedure of finding the positions of the jobs in the presented small example is demonstrated in Fig. 3.

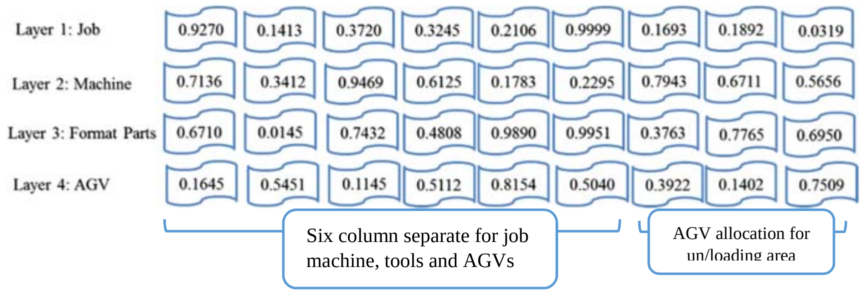

Fig. 3 *Representation of chromosome structure.*

Using the example, suppose that we have three jobs and that each job has two operations; so nine columns with four rows are generated. The six first columns are separated. The first row is used for jobs assignment. In the first row, the two first small numbers are 0.1413 and 0.2106; so, job 1 is assigned to the numbers. The smallest number is considered for the first operation, and the next smallest number is considered for the second operation of job 1. The same procedure is continued for the rest to assign all jobs. The results are demonstrated in Fig. 4.

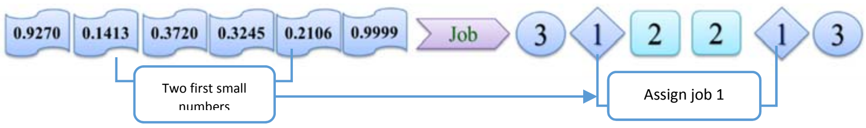

Fig. 4 *Representation of jobs and operations assignment.*

Then, the second row is used for machines' assignment. For instance, the first element equals to 0.7136. Because it is related to job 3 and operation 2. This operation can be performed by machines 1, 2 and 3. So, 0.7136 multiplied by $3(0.7136 \times 3 = 2.1408)$ and round it up $= 3$. Thus, the third machine is selected. In the same way, the rest of the machines are assigned. For tools allocation, the third row is used with same machine's procedure assignment and the fourth row is used for AGVs allocation as well. The results are demonstrated in Fig. 5.

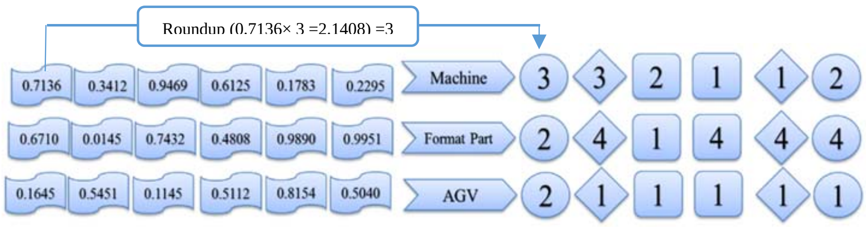

Fig. 5 *Representation of machines, tools and AGVs assignment.*

The last three columns with the same procedures are considered for AGV allocation for transportation to loading and unloading areas.

5. Parameters setting

Parameter setting is categorized into two main parts; first setting the tuning parameters of the algorithms and second setting model parameters.

5.1 Problem parameters setting

To evaluate the algorithms, 30 problems are generated randomly. The problems are classified based on the number of jobs, the number of AGVs, the number of machines, the number of operations, and the number of tools. The problems are solved three times and the average of the solutions is used to evaluate the algorithms. In addition, Moreover, the duration of each process in each operation follows a uniform distribution in the interval $[2, 7]$; the time required for AGV transportation between machines follows a uniform distribution in the interval $[1, 3]$; previous mean time to repair and previous mean time between failures follow a uniform distribution in the interval $[3, 7]$ and $[25, 50]$ for importing data into ANN training; penalty for earliness and tardiness time follows a uniform distribution in the interval $[10, 20]$; processing cost follows a uniform distribution in the interval $[5, 7]$; and handling cost follows a uniform distribution in the interval $[5, 10]$.

5.2 Tuning algorithm parameters

Taguchi method which is applied to calibrate the parameters of the algorithms was designed based on orthogonal arrays. It can be used efficiently as an alternative for full factorial experimental design in order to consider a group of factors. These factors are separated into two groups consisting of controllable noise factors and noise factors. The method's purpose is to select the best level of the factors so that the effect of noise factors is minimized and controllable factors are maximized. Therefore, a measure named signal-to-noise ratio is used to determine the performance of the algorithms. The value of S/N ratio is calculated by using Eq. 51 [30].

$$S/N = -10 \times \log \left(\frac{S(Y^2)}{n} \right), \tag{51}$$

where n and Y are the numbers of response values and orthogonal arrays. Since multi-objective metrics are a basis for evaluating the performance of multi-objective algorithms, the metric was developed by [31]. The main advantage of using this index as a response of Taguchi method is to consider both diversification and intensification features of metaheuristic algorithms. Using Eq. 52 to compute the response.

$$MOCV = MID/MS, \tag{52}$$

where *MID* is considered as convergence while *MS* is considered as diversity. Three levels of the parameters are displayed in Tab. II for each factor Minitab software is used to obtain the optimal values of S/N ratios. L^9 design is selected for NSGA-II and MOTLBO algorithms and L^{27} design is selected for tuning parameters of FMOIWO, MOPOS and FMOCS algorithms. The most appropriate levels of algorithms' parameters are shown Fig. 6.

The ratios of the FMOIWO, FMOCS, MOPSO, NSGA-II, and MOTLBO algorithms are presented in Fig. 6.

A level of the parameter is selected, which has the highest signal-to-noise index. So, in accordance with Fig. 6, it is clear that the FMOCS levels of the algorithms' parameters include: max iteration, number of Cuckoo, step size set in first level, and motion radius parameter set in third level. For MOPSO levels of the algorithms' personal learning coefficient and number of particle set in third level. Global learning coefficient, inertia weight, number of particle, max iteration, maximum number of repository and number of grid in a dimension set in second level. It can be seen that the MOTLBO levels of the algorithms' parameters include: max iteration set in second level, number of population in third level, and percentage of mutation set in first level. For NSGA-II the percentage of crossover and the number of population must be set in the third level, the percentage of mutation must be set in the second level, and max iteration must be set in the first level. For MOIWO Max iteration and initial value of standard deviation set in second level. Number of initial weeds, Final value of standard deviation, minimum number of seeds, maximum number of seeds and maximum number of archive set in first level. Non-linear coefficient and fuzzy dominate pressure set in second level.

6. Computational results

The experimental outputs of the five meta-heuristic algorithms for 30 test problems were classified into three groups: small, medium and large sizes. These problems were coded in MATLAB software and were performed on a PC with 8 GB RAM and Dual 2-GHz CPU. The performance of the algorithms are compared with each other based on the metrics presented in Section 4.5. The experimental outputs of

Algorithm	Parameter	Symbol	Parameters Level		
			Level 1	Level 2	Level 3
NSGA2	Percentage of crossover	Pc	0.7	0.8	0.9
	Percentage of mutation	Pm	0.2	0.15	0.1
	Max iteration	Max iteration	5 ∗ N	10 ∗ N	15 ∗ N
	Population	N-Pop	100	150	200
MOPSO	Personal learning coefficient	C1	1	1.4962	2
	Global learning coefficient	C2	1	1.4962	2
	Inertia weight	W	0.6	0.7298	0.9
	Max iteration	Max iteration	5 ∗ N	10 ∗ N	15 ∗ N
	Number of particle	N-Pop	100	150	200
	Maximum number of repository	N-Rep	50	75	100
	Number of grid in a dimension	N-Grid	5	8	10
FMOIWO	Max iteration	Max iteration	5 ∗ N	10 ∗ N	15 ∗ N
	Number of initial Weeds	N-Pop	100	150	200
	Initial value of standard deviation	Initial Sigma	0.5	0.4	0.3
	Final value of standard deviation	Final Sigma	0.01	0.03	0.05
	Minimum number of seeds	S-Min	1	2	3
	Maximum number of seeds	S-Max	5	8	10
	Non-linear coefficient	n	2	3	4
	Fuzzy dominate pressure	KF	1	2	3
	Maximum number of archive	N-Archive	100	150	200
FMOCS	*Max iteration*	*MaxIt*	5 ∗ N	10 ∗ N	15 ∗ N
	Number of Cuckoo	*NCukoo*	100	150	200
	Motion radius	*BETA*	1	1.5	2
	Step size	*ALPHA*	0.005	0.01	0.015
MOTLBO	*Max iteration*	*MaxIt*	5 ∗ N	10 ∗ N	15 ∗ N
	Number of Population	*Npop*	100	150	200
	Percent of Mutation.	*Pm*	0.1	0.15	0.2

Tab. II *Algorithms parameters ranges along with their levels.*

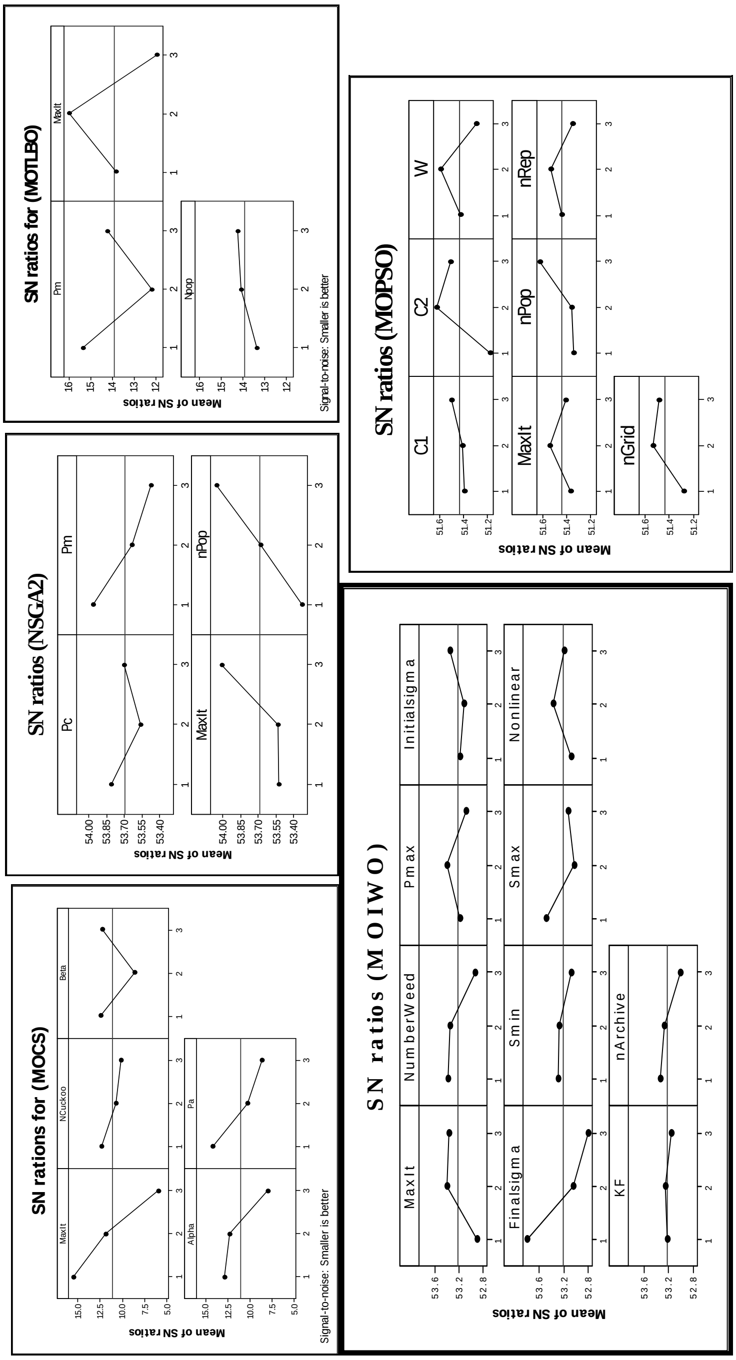

Fig. 6 *S/N ratio plots of the FMOIWO, FMOCS, MOPSO, NSGA-II, and MOTLBO.*

the five algorithms are presented in Tab. III. The trends of the test problems are illustrated in Fig. 7, in terms of *MID, SM, SNS, DM* and *CPU time* metrics.

The result of trend metrics over the test problems is shown in Fig. 7. Fig. 7(a) shows that NSGA-II and MOPSO are dominated by MOTLBO, FMOIWO and FMOCS for all test problems in terms of SM metric. Fig. 7(b) reveals that there is a competition between MOPSO and MOIWO for all test problems in terms of the MID metric where a close competition between MOTLBO and MOCS is obvious. Fig. 7(c) confirms the better performance of MOIWO for all test problems in terms of the DM metric where there is a competition between FMOCS and MOIWO. Fig. 7(d) shows the best performance of FMOIWO for all test problems in terms of the SNS metric, and that there is a competition between MOTLBO and FMOCS again. Fig. 7(e) confirms the better performance of FMOIWO for all test problems in terms of the CPU time metric.

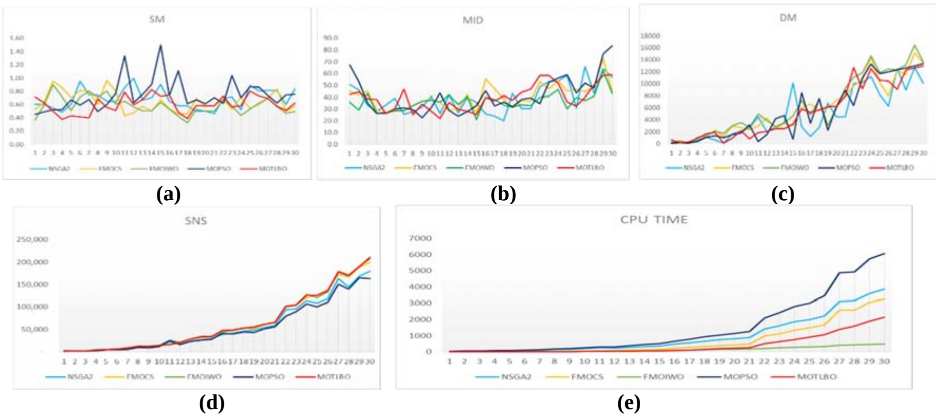

Fig. 7 *The trends of the test problems, in terms of MID, SM, SNS, DM and CPU time metrics.*

To evaluate the performance of the algorithms, a hybrid multi-attribute decision-making method named AHP-TOPSIS was applied in order to prioritize algorithms, considering the metrics. This method was applied to select an algorithm with the highest priority. AHP, firstly introduced by [32], was used to identify the weighs of the criteria. After calculating normalized weights, the TOPSIS method, firstly presented by [33], is used to determine which algorithm has a better performance for solving large, medium and small size problems. The following steps are applied to prioritize alternatives:

1. Decision matrix is created.
2. Normalized decision matrix is calculated
3. Calculate weighted normalized decision matrix.
4. Negative and positive ideal solutions are calculated. The positive ideal solution is determined as the greatest value of positive criteria and the smallest value of negative criteria while the negative ideal solution is determined as

	NSGA2					FMOCS					FMOIWO					MOPSO					MOTLBO				
# problem	SM	MID	DM	SNS	CPU Time	SM	MID	DM	SNS	CPU Time	SM	MID	DM	SNS	CPU Time	SM	MID	DM	SNS	CPU Time	SM	MID	DM	SNS	CPU Time
1	0.60	50.4	43	1,988	37	0.53	44.96	257	1,939	2	0.37	35.89	225	828	3	0.45	67.3	84	2,395	36	0.72	42.18	629	2,206	0
2	0.60	46.1	188	2,130	46	0.63	41.90	288	2,755	3	0.59	29.05	476	2,303	3	0.49	53.2	146	2,128	47	0.63	44.31	147	2,149	1
3	0.55	33.3	95	1,616	48	0.96	45.07	36	2,001	3	0.90	42.95	124	2,153	5	0.52	35.4	91	1,795	57	0.50	38.52	267	1,879	1
4	0.48	28.3	305	2,927	59	0.86	29.08	476	3,810	5	0.72	26.49	1071	4,256	7	0.52	26.3	470	2,899	73	0.38	38.34	901	4,092	1
5	0.60	33.3	1089	4,806	69	0.71	26.82	1262	5,047	7	0.51	26.16	1362	5,044	12	0.67	26.7	1068	4,430	85	0.43	26.12	1627	5,479	2
6	0.96	39.1	630	4,995	84	0.81	28.05	2249	6,313	10	0.71	28.78	2113	6,095	14	0.59	30.0	1301	4,769	102	0.41	30.39	1887	6,555	2
7	0.75	25.5	116	6,373	102	0.80	28.50	1192	7,664	14	0.80	28.87	1723	7,337	19	0.68	31.2	996	5,576	125	0.40	46.49	127	9,173	4
8	0.75	27.9	1683	10,427	142	0.60	32.91	2937	12,685	26	0.71	32.98	2992	12,494	24	0.50	29.6	1532	9,785	179	0.69	25.37	961	12,915	8
9	0.65	31.9	2119	9,118	149	0.96	37.02	2687	13,406	28	0.81	36.45	3519	12,436	27	0.60	22.8	1782	8,708	193	0.56	34.35	2177	12,082	9
10	0.63	40.7	2644	11,683	191	0.81	38.33	2303	14,455	41	0.61	37.49	2749	14,892	36	0.77	31.6	3135	11,665	244	0.51	28.24	842	14,654	15
11	0.86	26.2	4521	22,638	242	0.43	36.09	2931	17,263	45	0.65	35.94	4959	26,101	54	1.33	43.5	382	24,881	316	0.79	22.17	1867	16,108	18
12	1.00	42.2	1968	17,580	236	0.47	28.98	4326	19,905	65	0.56	41.87	3826	21,765	52	0.63	28.8	1554	16,506	313	0.60	35.61	2059	20,820	23
13	0.66	26.8	2789	24,432	282	0.57	30.40	2465	27,776	96	0.51	34.07	2878	28,762	47	0.76	24.0	4152	23,289	379	0.68	29.84	2505	29,247	32
14	0.71	39.8	3516	27,383	335	0.50	41.98	3396	31,766	119	0.51	40.35	3473	32,291	61	0.91	27.9	4712	25,923	446	0.82	28.26	2514	34,188	45
15	0.91	35.7	10123	29,569	366	0.67	26.33	3044	33,073	142	0.64	21.12	4643	33,849	70	1.50	33.1	776	28,010	497	0.73	25.42	3253	34,287	55
16	0.65	26.1	2831	38,937	474	0.54	55.23	6007	46,121	220	0.53	39.88	7249	44,696	94	0.74	45.2	8519	41,740	653	0.74	39.69	5794	47,661	77
17	0.58	24.1	1175	41,275	549	0.46	46.96	6613	46,988	273	0.43	38.32	4800	47,447	105	1.11	32.8	3372	39,938	790	0.48	36.77	5193	48,536	111
18	0.58	20.0	2680	46,487	651	0.46	38.30	5695	52,775	333	0.32	33.52	7357	52,121	133	0.61	35.9	7514	43,924	942	0.39	41.25	5595	53,236	156
19	0.50	43.1	6749	48,714	753	0.69	31.21	5956	52,827	374	0.53	31.69	3079	56,182	145	0.67	32.1	2228	43,569	1034	0.58	37.68	6203	54,983	213
20	0.50	30.5	4446	54,223	806	0.58	37.38	7254	60,453	434	0.50	33.62	6001	61,552	161	0.61	38.4	6242	51,526	1142	0.58	44.01	6246	61,652	227
21	0.46	30.4	4491	58,607	885	0.58	36.63	7865	65,175	488	0.50	33.14	9306	66,392	178	0.70	39.7	8912	55,673	1258	0.58	46.63	7957	66,272	261
22	0.69	48.7	10045	93,981	1415	0.68	53.00	9681	100,500	977	0.67	39.94	11128	102,437	201	0.62	34.7	6360	79,937	2104	0.73	58.34	12730	101,270	505
23	0.72	53.1	10221	96,105	1625	0.53	46.94	10566	104,798	1,117	0.58	40.97	11842	105,034	258	1.04	52.5	10995	89,431	2417	0.56	58.42	9156	104,307	630
24	0.52	53.5	11232	113,794	1861	0.71	53.87	14436	120,987	1,347	0.43	46.25	14639	128,676	281	0.70	56.2	13330	106,670	2816	0.58	51.96	12541	126,583	770
25	0.89	58.3	8160	108,772	2005	0.54	44.95	10187	126,100	1,483	0.53	30.15	11857	120,642	315	0.87	59.0	11665	100,415	3017	0.80	35.86	10580	125,536	919
26	0.81	30.6	6253	118,679	2228	0.60	46.59	8017	138,597	1,682	0.62	39.76	12451	133,751	327	0.87	43.3	12000	109,815	3463	0.73	32.67	10485	136,193	1052
27	0.82	65.9	12508	163,861	3121	0.70	44.46	12293	172,750	2,573	0.69	37.23	12302	179,959	411	0.74	51.7	12335	151,135	4875	0.69	39.97	9091	178,274	1381
28	0.81	43.1	8999	144,349	3159	0.84	49.38	11559	167,427	2,588	0.57	40.89	13143	172,022	431	0.62	47.5	12670	139,671	4928	0.58	50.15	12328	169,633	1599
29	0.60	64.2	12745	168,878	3600	0.49	72.39	15200	189,243	3,070	0.47	62.45	16499	189,999	466	0.75	76.6	13005	165,397	5734	0.51	58.28	12654	190,404	1904
30	0.84	56.6	10046	179,431	3863	0.58	44.84	13419	199,760	3,259	0.49	42.72	13609	212,165	484	0.76	83.5	13340	163,236	6060	0.62	58.93	12980	208,921	2157

Tab. III *The experimental outputs of the FMOIWO, FMOCS, MOPSO, NSGA-II, and MOTLBO.*

the greatest value of positive criteria and the smallest value of the negative criteria.

5. Euclidean distances of alternatives from positive and negative ideal solutions are calculated

6. Relative closeness of each alternative to ideal solution is calculated

An alternative with the highest closing rate value is selected as the best one.

The decision matrix, normalized decision matrix, weighted normalized decision matrix, Euclidean distances of alternatives and relative closeness of alternatives for large and small scale problems are presented in Tabs. IV–VI. The results show that FMOIWO algorithm has the best rank performance in solving large, medium and small scale problems and the FMOCS ranked second level in solving small and medium size problems.

7. Conclusion

A new two-objective mixed integer programming formulation was presented in this paper for modelling machines and AGVs' simultaneous scheduling problem, considering machines' breakdown possibility. The model aimed to minimize total costs including processing costs and minimize total completion time. An artificial neural network approach was employed in this paper to estimate the model's two different and significant parameters including the time spent between the machine's two consecutive breakdowns and the machine's maintenance time. Since the model was strictly NP-hard and because the exact algorithms were not able to find its global optimum solutions in a reasonable time, two metaheuristic algorithm called FMOIWO and FMOCS algorithms were developed to solve the model's various test problems. In addition, a novel chromosome structure was presented to satisfy the model's constraints and ensure the feasibility of the solutions generated in different iterations. Since there was no benchmark available in literature to validate the performance of FMOIWO and FMOCS search algorithm, three other solution algorithms called MOPSO, NSGA-II, and MOTLBO were developed to validate the performance of FMOIWO and FMOCS search algorithms. A Taguchi method was used to calibrate the parameters of the presented algorithms and to enhance the performance of the developed algorithms. All the solutions obtained using developed algorithms were presented in the form of five various metrics called *MID, DM, SNS, CPU time* and *SM*. In addition, solutions are classified into three main groups including large, medium and small scale solutions. Finally, an AHP-TOPSIS method was used to identify an algorithm which has a better overall performance in solving different test problems. The results showed that developed FMOIWO search algorithm had a better performance in solving large, medium and small scale problems and FMOCS ranked second. Adding AGV routing problem to this model can be future work of this study.

Algorithm	Normalize decision matrix					Weighted normalized decision matrix					d+	d-	cl	Rank
	SM	MID	DM	SNS	Time	SM	MID	DM	SNS	Time				
NSGA-II	0.858	1.000	0.545	0.788	0.812	0.145	0.254	0.166	0.133	0.083	0.168	0.031	0.156	5
FMOCS	1.000	0.989	0.837	0.984	0.122	0.169	0.251	0.255	0.167	0.012	0.077	0.132	0.633	2
FMOIWO	0.878	0.912	1.000	0.953	0.131	0.149	0.232	0.305	0.162	0.013	0.036	0.171	0.828	1
MOPSO	0.754	0.993	0.649	0.761	1.000	0.128	0.252	0.198	0.129	0.102	0.153	0.052	0.256	4
MOTLBO	0.681	0.994	0.585	1.000	0.038	0.115	0.253	0.178	0.169	0.004	0.128	0.120	0.482	3

Tab. IV *The results of the TOPSIS method for small-size problems.*

Algorithm	Normalize decision matrix					Weighted normalized decision matrix					d+	d-	cl	Rank
	SM	MID	DM	SNS	Time	SM	MID	DM	SNS	Time				
NSGA-II	0.78	0.84	0.85	0.87	0.72	0.133	0.21	0.26	0.15	0.07	0.09	0.06	0.42	4
FMOCS	0.61	1.00	0.99	0.96	0.32	0.103	0.25	0.30	0.16	0.03	0.04	0.11	0.71	2
FMOIWO	0.58	0.94	1.00	1.00	0.14	0.099	0.24	0.31	0.17	0.01	0.02	0.13	0.84	1
MOPSO	1.00	0.92	0.82	0.84	1.00	0.169	0.23	0.25	0.14	0.10	0.13	0.02	0.14	5
MOTLBO	0.72	0.91	0.85	0.99	0.15	0.122	0.23	0.26	0.17	0.01	0.05	0.10	0.66	3

Tab. V *The results of the TOPSIS method for medium-size problems.*

Algorithm	Normalize decision matrix					Weighted normalized decision matrix					d+	d-	cl	Rank
	SM	MID	DM	SNS	Time	SM	MID	DM	SNS	Time				
NSGA-II	0.935	0.926	0.747	0.883	0.648	0.159	0.235	0.228	0.150	0.066	0.112	0.043	0.278	4
FMOCS	0.817	0.905	0.893	0.982	0.507	0.138	0.230	0.272	0.166	0.052	0.067	0.082	0.552	3
FMOIWO	0.726	0.759	1.000	1.000	0.091	0.123	0.193	0.305	0.169	0.009	0.000	0.146	1.000	1
MOPSO	1.000	1.000	0.904	0.823	1.000	0.169	0.254	0.276	0.139	0.102	0.127	0.048	0.274	5
MOTLBO	0.835	0.902	0.872	0.997	0.305	0.142	0.229	0.266	0.169	0.031	0.060	0.093	0.607	2

Tab. VI *The results of the TOPSIS method for large-size problems.*

References

[1] ABDELMAGUID T.F., NASSEF A.O., KAMAL B.A., HASSAN M.F. A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles. International journal of production research, 2004. pp. 267-81, doi: 10.1080/0020754032000123579.

[2] REDDY B.S.P., RAO C.S.P. A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. The International Journal of Advanced Manufacturing Technology. 2006, 31, pp. 602–13, doi: 10.1007/s00170-005-0223-6.

[3] CORRÉA A.I., LANGEVIN A., ROUSSEAU L.-M. Scheduling and routing of automated guided vehicles: A hybrid approach. Computers & operations research. 2007, 34, pp. 1688–707, doi: 10.1016/j.cor.2005.07.004.

[4] KESEN S.E., BAYKOÇ ÖF. Simulation of automated guided vehicle (AGV) systems based on just-in-time (JIT) philosophy in a job-shop environment. Simulation Modelling Practice and Theory. 2007, 15, pp. 272–84, doi: 10.1016/j.simpat.2006.11.002.

[5] LAU H.Y.K, ZHAO Y. Integrated scheduling of handling equipment at automated container terminals. International Journal of Production Economics. 2008, 112, pp. 665–82, doi: 10.1016/j.ijpe.2007.05.015.

[6] DEROUSSI L., GOURGAND M., TCHERNEV N. A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles. International Journal of Production Research. 2008, 46, pp. 2143–64, doi: 10.1080/00207540600818286.

[7] BRAUNER N., FINKE G., LEHOUX-LEBACQUE V., POTTS C., WHITEHEAD J. Scheduling of coupled tasks and one-machine no-wait robotic cells. Computers & Operations Research. 2009, 36, pp. 301–7, doi: 10.1016/j.cor.2007.10.003.

[8] GNANAVEL BABU A., JERALD J., NOORUL HAQ A., MUTHU LUXMI V., VIGNESWARALU T.P. Scheduling of machines and automated guided vehicles in FMS using differential evolution. International Journal of Production Research. 2010, 48, pp. 4683–99, doi: 10.1080/00207540903049407.

[9] CHE A., HU H., CHABROL M., GOURGAND M. A polynomial algorithm for multi-robot 2-cyclic scheduling in a no-wait robotic cell. Computers & Operations Research. 2011, 38, pp. 1275–85, doi: 10.1016/j.cor.2010.11.008.

[10] Chaudhry I, Mahmood S, Shami M. Simultaneous scheduling of machines and automated guided vehicles in flexible manufacturing systems using genetic algorithms. Journal of Central South University of Technology. 2011, 18, pp. 1473–86, doi: 10.1007/s11771-011-0863-7.

[11] BRUCKER P., BURKE E.K., GROENEMEYER S. A mixed integer programming model for the cyclic job-shop problem with transportation. Discrete Applied Mathematics. 2012, 160, pp. 1924–35, doi: 10.1016/j.dam.2012.04.001.

[12] BATUR G.D., KARASAN O.E., AKTURK M.S. Multiple part-type scheduling in flexible robotic cells. International Journal of Production Economics. 2012, 135, pp. 726–40, doi: 10.1016/j.ijpe.2011.10.006.

[13] LACOMME P., LARABI M., TCHERNEV N. Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles. International Journal of Production Economics. 2013, 143, pp. 24–34, doi: 10.1016/j.ijpe.2010.07.012.

[14] ZENG C., TANG J., YAN C. Scheduling of no buffer job shop cells with blocking constraints and automated guided vehicles. Applied Soft Computing. 2014, 24, pp. 1033–46, doi: 10.1016/j.asoc.2014.08.028.

[15] Narendranath S, Ramesh MR, Chakradhar D, Doddamani M, Bontha S, Nageswararao M, et al. International Conference on Advances in Manufacturing and Materials Engineering, ICAMME 2014Simultaneous Scheduling of Machines and AGVs in Flexible Manufacturing System with Minimization of Tardiness Criterion. Procedia Materials Science. 2014, 5, pp. 1492–501, doi: 10.1016/j.mspro.2014.07.336.

[16] ZHENG Y., XIAO Y., SEO Y. A tabu search algorithm for simultaneous machine/AGV scheduling problem. International Journal of Production Research. 2014, 52, pp. 5748–63, doi: 10.1080/00207543.2014.910628.

[17] UMAR U.A., ARIFFIN M.K.A., ISMAIL N., TANG S.H. Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment. The International Journal of Advanced Manufacturing Technology. 2015, 81, pp. 2123–41, doi: 10.1007/s00170-015-7329-2.

[18] NOURI H.E., DRISS O..B, GHÉDIRA K. Hybrid metaheuristics for scheduling of machines and transport robots in job shop environment. Applied Intelligence. 2016, pp. 1–21, doi: 10.1007/s10489-016-0786-y.

[19] DANG Q.-V., NGUYEN L. A Heuristic Approach to Schedule Mobile Robots in Flexible Manufacturing Environments. Procedia CIRP. 2016, 40, pp. 390–5, doi: 10.1016/j.procir.2016.01.073.

[20] KUNDU D., SURESH K., GHOSH S., DAS S., PANIGRAHI B.K., DAS S. Multi-objective optimization with artificial weed colonies. Information Sciences. 2011, 181, pp. 2441–54, doi: 10.1016/j.ins.2010.09.026.

[21] YANG X.-S., DEB S. Cuckoo search: recent advances and applications. Neural Computing and Applications. 2014, 24, pp. 169–74, doi: 10.1007/s00521-013-1367-1.

[22] DEB K., PRATAP A., AGARWAL S., MEYARIVAN T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation. 2002, 6, pp. 182–97, doi: 10.1109/4235.996017.

[23] ZOU F., WANG L., HEI X., CHEN D., WANG B. Multi-objective optimization using teaching-learning-based optimization algorithm. Engineering Applications of Artificial Intelligence. 2013, 26, pp. 1291–300.

[24] COELLO C.A.C, PULIDO G.T., LECHUGA M.S. Handling multiple objectives with particle swarm optimization. Evolutionary Computation, IEEE Transactions on. 2004, 8, pp. 256–79, doi: 10.1109/TEVC.2004.826067.

[25] AKBARI M., SAEDODIN S., PANJEHPOUR A., HASSANI M., AFRAND M., TORKAMANY M.J. Numerical simulation and designing artificial neural network for estimating melt pool geometry and temperature distribution in laser welding of Ti6Al4V alloy. Optik - International Journal for Light and Electron Optics. 2016, 127, pp. 11161–72, doi: 10.1016/j.ijleo.2016.09.042.

[26] KUMAR S. Neural networks: a classroom approach: Tata McGraw-Hill Education, 2004.

[27] ZHANG G., PATUWO B.E., HU M.Y. Forecasting with artificial neural networks: The state of the art. International journal of forecasting. 1998, 14, pp. 35–62, doi: 10.1016/S0169-2070(97)00044-7.

[28] HECHT-NIELSEN R. Kolmogorov's mapping neural network existence theorem. Proceedings of the international conference on Neural Networks: IEEE Press, 1987, pp. 11–4.

[29] OKABE T., JIN Y., SENDHOFF B. A critical survey of performance indices for multi-objective optimisation. Evolutionary Computation, 2003 CEC '03 The 2003 Congress on 2003, Vol. 2., pp. 878–85, doi: 10.1109/CEC.2003.1299759.

[30] TAGUCHI G. Introduction to quality engineering: designing quality into products and processes, 1986.

[31] RAHMATI S.H.A, HAJIPOUR V., NIAKI S.T.A. A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem. Applied Soft Computing. 2013, 13, pp. 1728–40, doi: 10.1016/j.asoc.2012.12.016.

[32] SAATY R.W. The analytic hierarchy process—what it is and how it is used. Mathematical Modelling. 1987, 9, pp. 161–76, doi: 10.1016/j.asoc.2012.12.016.

[33] HWANG C.-L., YOON K. Lecture Notes in Economics and Mathematical Systems: Multiple Attribute Decision Making: Methods and Appllication: Springer Verlag, 1981, doi: 10.1016/0270-0255(87)90473-8.