

Energy-efficient Multi-objective Flexible Manufacturing Scheduling

Energy-efficient Multi-objective Flexible Manufacturing Scheduling

Sasan Barak^{1*}, Reza Moghdani^{2,3}, Hamidreza Maghsoudlou^{2,4},

1. Department of Decision Analytics and Risk, Southampton Business School, University of Southampton, UK. Email: s.barak@soton.ac.uk

2. Faculty of Economics, VŠB Technical University of Ostrava, Ostrava 70200, Czech Republic.

3. Computational Intelligence and Intelligent Optimization Research Group (CIORG), Persian Gulf University, Bushehr, Iran. Email: reza.moghdani@gmail.com

4. Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran. Email: maghsoudlou@qiau.ac.ir

Abstract

This paper presents a novel scheduling of a resource-constrained Flexible Manufacturing System (FMS) with consideration of the following sub-problems: (i) machine loading and unloading, (ii) manufacturing operation scheduling, (iii) machine assignment, and (iv) Automated Guided Vehicle (AGV) scheduling. In the proposed model, both the AGV and machinery are considered as the required resources. Energy efficiency of AGVs has been studied in order to improve environmental sustainability in terms of a linear function, which is based on load and distance, accordingly. Because of the NP-hard characteristics of the problem, a modified multi-objective particle swarm optimization (MMOPSO) has been developed for

solving the model and compared with the classic version of the multi-objective particle swarm optimization (MOPSO) algorithm in terms of five performance metrics. Finally, the results are evaluated by the application of a multi-criteria decision-making (MCDM) algorithm according to which the MMOPSO outperforms the MOPSO.

Keywords: Flexible Manufacturing Systems (FMS); Automated Guided Vehicle (AGV); Multi-Objective Particle Swarm Optimization (MOPSO); Scheduling.

1. Introduction

A Flexible Manufacturing System (FMS) is a sophisticated automatic production system comprised of a set of computer-controlled unified configurations of multipurpose workstations and Automated Guided Vehicles (AGVs). AGV systems have great potential to eliminate some deficiencies for the current environmental issues (Bechtsis et al. 2017). Technical innovations in these autonomous vehicles are considered by the characteristic of business that endorses reduced emissions, and sustainability (Bechtsis et al. 2018).

Due to the changing products' demands and customers' varying requirements, such a flexible manufacturing environment subject to resource limitations has become an essential parameter for remaining competitive in the current global markets. Flexibility is defined as the capability of applying changes in the manufacturing system and throughout the life cycle of a product without any interruption in the production line (Yadav and Jayswal 2018). The significant differences between FMS and non-FMS are achieved due to the capability to (1) produce various products, (2) add new product designs, (3) schedule changes, and (4) enable breakdown recovery (ElMaraghy and Caggiano 2016).

Furthermore, based on the literature available on FMS research, the flexibility aspects are defined as: a machine, operation, production, process, material handling, product, market flexibility, expansion, earlier routing, and volume (Sethi and Sethi 1990).

Technological development in the area of material handling system has brought some modifications in material flow design. The primary material handling equipment employed in the new design is the AGV. In AGV systems, determining material flow plays a vital role in increasing efficiency. In such a complicated environment, presenting a comprehensive framework to enhance the performance of material flow is a key issue that optimizes the costs. The task is achieved within the application of various modelling techniques in the literature, such as simulation, fuzzy logic, and artificial intelligence approaches. In this paper, we study the problem of material flow handling in production lines in which the vehicles are used as carriers.

Most of the mathematical modelling optimizes the scheduling of manufacturing activities along with transportation and storage tasks to enhance the efficiency of the system, which is also the significant focus of the current paper.

The rest of the paper is organized as follows: the literature review is discussed in section 2, Problem description, including FMS problem, AGV features, and fuel consumption, is described in section 3. The mathematical formulation is in section 4. Section 5 presents the approach to solving the problem. Tuning algorithm parameters are considered in section 6. The computational results are shown in section 7. Finally, the conclusions and recommendations for future research are discussed in section 8.

2. Literature review

In this section, we will concentrate on two main concepts, including FMS scheduling and environmental consideration of FMS, focusing on AGVs.

2.1. Flexible manufacturing systems scheduling

FMS scheduling involves operations, such as machine loading and unloading, production activities programming, machine planning and allocation, and the AGVs schedule. FMS

loading deals with operation allocation to machines in terms of resources and technological constraints, while production activities programming specifies the start, duration, and finishing time of each machine activity. Additionally, AGVs scheduling determines material handling devices (Novas and Henning 2014). Moreover, machine planning specifies the number of machines which should be assigned to special operations.

FMS scheduling is one of the most complicated programming problems which has been extensively noted in the last decades (Novas and Henning 2014). The literature available on FMS offers several competing definitions, such as the possibility of operating more than one machine and also the ability of the production system to utilize multiple routes to produce a given part (Bruna 2011). Due to the complexity of scheduling procedures in FMS, researchers rarely consider all introduced operations at the same time, and they mostly overlook some of these elements for ease of programming.

Blazewicz et al. (1991) used a two-step FMS programming system in machine scheduling and vehicle routing problems. In the first step, operations are allocated to machines in order to obtain a feasible vehicle schedule and then, in the next step, operations are allocated to machines, and vehicle routing is also implemented, simultaneously. The authors applied dynamic programming that was only capable of solving a small-sized problem. The results of this study showed that obtained solutions were quite practical for most real-world applications. Bilge and Ulusoy (1995) formulated mixed-integer non-linear programming for simultaneous scheduling of machinery and transportation system in FMS. Liu and MacCarthy (1997) developed a mixed-integer linear programming model for storage and AGV scheduling. They used two different heuristic algorithms, but because of the model sophistication, even for small-sized problems, the algorithms were not successful in obtaining proper results. Ulusoy, Sivrikaya-Şerifoğlu and Bilge (1997) applied a Genetic Algorithm (GA) to address the FMS

scheduling where the chromosome structure represents both the operation number and AGV assignment. The GA results were promising in comparison with the opponents.

Sabuncuoglu and Karabuk (1998) developed a beam search-based algorithm for the scheduling of machines and AGVs simultaneously. They assumed that the vehicles always come back to the load/unload station after fulfilling their loading operations. The key point of this study indicated the importance of the load/unload station in the FMS problem.

Jerald et al. (2006) proposed a model for concurrent scheduling of production and AGVs and used an adaptive GA-based approach to solve the problem. They classified machines into cells with two AGVs between cells. The approach does not consider the machine loading issue and transportation times as a unique part-time. However, such assumptions affect the applicability of their proposed approach.

This paper formulates comprehensive mathematical modelling of FMS to handle the following five sub-problems simultaneously: (i) machine loading and unloading, (ii) manufacturing operation scheduling, (iii) machine assignment, and (iv) AGVs scheduling. Moreover, the storage cost and machine limitations are considered as the most crucial constraints, and AGVs as the primary limiting resources, for the first time in FMS scheduling.

Liu, Wang and Jin (2008) proposed a modified particle swarm optimization to minimize the make-span in a permutation flow shop scheduling problem regarding limited resources between consecutive machines. They assumed that the preparation time depends on the successive order, while most of the other studies consider this factor as an independent parameter.

AGV scheduling and its limitation in the scheduling problem is a topic of interest for the researchers. Zeng, Tang and Yan (2014) suggested a Blocking Job Shop (BJS)–AGV problem while considering transferring jobs between different machines using a limited number of AGVs. Two integers non-linear programming models and a two-stage heuristic algorithm that

combines an improved timetabling method and a local search are proposed to solve the BJS–AGV problem, and results showed the capability of proposed local search to obtain optimal solution in this case. Karimi et al. (2018) presented a mathematical model, in which AGVs transfer all parts and materials in the manufacturing system. To solve this problem, a combination of a simulation approach based on Weibull distribution and two well-known metaheuristic algorithms, namely the non-dominated sorting cuckoo search and multi-objective teaching-learning-based optimization, were proposed. The results of this study indicated that multi-objective cuckoo search obtained better results in comparison with the multi-objective teaching-learning-based optimization approach. Heger and Voß (2019) focused on a dynamic approach for dispatching autonomous AGVs, considering routing flexibility in uncertain environment conditions by using a simulation approach. Llopis-Albert, Rubio and Valero (2019) embedded a fuzzy approach into a Multi-Criteria Decision Making (MCDM) method, namely qualitative comparative analysis, to design an AGV system, in which various kinds of factors were considered in uncertain conditions.

2.2. Environmental considerations

Generally in FMS scheduling, due to a large amount of transportation with AGV systems, environmental concern needs further discussion. Nowadays, with the promotion of environmental protection awareness, green production has become a new research area (An et al. 2020).

There have been many aspects of environmental sustainability for different levels of decision making: the strategic, tactical, and operational levels.

At the strategic level, the pivotal point in using AGV is energy management strategies (Acciaro, Ghiara and Cusano 2014). For example, Shukla and Karki (2016) presented a review applied in the oil and gas industries considering the latest challenges of using AGVs, which may lead to a potential accident. In another aspect of this level, Fuc et al. (2016) focused on

electric vehicles that using of AGVs are taken into consideration by using some management systems, such as ISO 14044 and the IMPACT 2002.

For the tactical level, the most critical issues are on the refueling and charging method of AGVs. In this regard, Schmidt et al. (2015) provide a comprehensive study to show the economical and ecological aspects of using battery-powered AGVs throughout a real case in Germany. In the same vein, focused on a real case in the Netherlands, a new methodology for analyzing energy consumption has been proposed by Geerlings and van Duin (2011) considering different layout designs to reduce CO₂ emissions. Leriche et al. (2015) proposed a new method using agent-based simulation to show the efficacy of implementing electric-powered vehicles in reducing CO₂ emissions. Other eminent studies at a tactical level have been presented by Schmidt, Eisel and Kolbe (2014) and Hopf and Müller (2015), which focus on new concepts of sustainability related to controlled charging concepts and resource consumption efficiency, respectively. Zhou and Shen (2018) proposed an energy-efficient scheduling method for AGVs to fulfil material follow problem. They used a linear function to estimate energy consumption.

For the operational level, scholars have considered some criteria and policies for different applications of AGVs regarding ecological concerns. Xin, Negenborn and Lodewijks (2014) focused on the sustainability of maritime logistics via reducing energy consumption using a specific policy considering a hierarchical controller. Furthermore, Lee, Low and Kim (2015) studied various strategies to show which one of them was efficient in reducing energy consumption. In the same vein, single and dual operations of several types of AGVs were examined in terms of energy efficacy and operation, respectively.

2.3. The Contribution of Research

By reviewing related papers in the FMS area, we conclude that there have been two main streams: FMS scheduling and environmental considerations that have been mainly neglected in the literature. Generally, there is a lack of sustainable implementation in the FMS scheduling problem (Zhang, Xu and Zhang 2020), and specifically, the combination of fuel consumption approach and FMS for the AGV system has not been considered in the previous research. Also, we propose using a MCDM approach embedded in the multi-objective optimization for the FMS problem. Furthermore, there has not been a comprehensive study that focuses on implementing and analyzing new metaheuristic methods considering parameters tuning in FMS, using the grey Taguchi approach.

Therefore, the main contributions of this paper initially include the novel mathematical modelling represented for the FMS problem which covers five effective and essential sub-problems of real-life systems, i.e. (1) machine loading and unloading, (2) machine assignment, (3) scheduling of manufacturing operations, (4) fuel consumption constraints in FMS (5) scheduling of AGVs. It is worth mentioning that in the proposed model, the preparation time is assumed as a dependent parameter to successive order while most of the current studies consider this feature independent of successive order. The second contribution of this paper concerns a green approach by incorporating the fuel consumption concept into the mathematical model. As the third contribution of the current paper, a Modified Multi-Objective Particle Swarm Optimization (MMOPSO) enhanced with efficient encoding and decoding algorithms has been utilized for the first time to cope with the problem complexities. Fourthly, the grey Taguchi approach is presented for parameters tuning in FMS and finally, to evaluate the results obtained by algorithms, a Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) model is proposed.

3. Problem description

3.1. FMS problem

The main features of our proposed FMS are as follows: (i) one loading and one unloading station (L/U station) in which the AGV should first transfer parts from the loading station to the first process machine and then take them back to the unloading station at the end of the whole process, (ii) multipurpose machines with limited available time and their assignment issues, (iii) different kinds of AGVs with different speeds and transportation costs, (iv) different parts with specific operation sequences.

The presented scheduling problem has a predefined set of parts with particular demands. Each part has a successive operation order, executed on one machine from a set of alternative multipurpose machines, and each machine executes just one operation on a single part at a time.

Some AGVs do parts transferring between stations through direct paths. A predefined time is required for parts loading and unloading from AGV on machines. Waiting parts are picked up by the robot when the station becomes idle. Nowadays, more and more mass customization manufacturers begin to use AGVs in their assembly lines. In such a condition, a lot of different products are assembled on the same assembly line, and required raw-materials for production in each station are continually altering. The proposed model in this study can be used for mass customization manufacturer which produces small products.

3.2. AGV features

AGVs play the most crucial role in FMS scheduling, and they should be strictly considered for obtaining relevant and feasible solutions for real FMSs. Figure 1 illustrates the parts movement pattern utilizing AGVs.

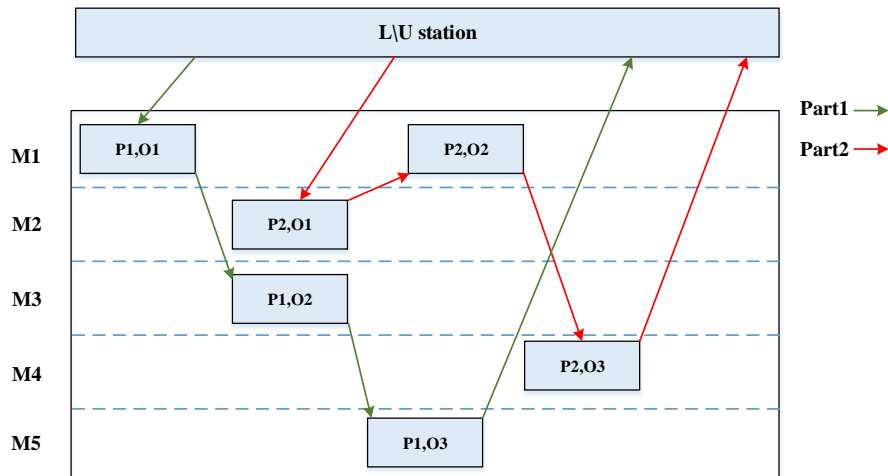


Figure 1. Movement procedure of parts with AGV

As illustrated above, an appropriate FMS scheduling deals with moving parts between machines or transferring them from/to L/U stations with AGVs. However, in this paper, the time required for the AGV depends on five parameters: (1) the distance between machines (or the machine and the L/U station and vice versa) (2) loading and unloading times on each machine (3) pre-processing time (pick up at the loading station and delivery to the machine) that will be executed in the first operation, (4) the last moving time (time for transferring parts from the final operation/machine to the unloading station), and finally (5) the speed of the transferring vehicle which causes a difference in the cost of transportation. In this paper, all of the above issues are considered for AGV scheduling.

Besides, no preemption is allowed for parts transportation by the AGV. When it is assigned to a part, the part must be unloaded before the AGV starts any other operation. Therefore, there will be no conflict between any pair of AGV transferring operations.

Empty AGV trips are those movements that are performed each time the AGV moves unloaded from its last delivery destination to another resource (machine or L/U station), where it is required to pick up a part that needs to be transferred to another FMS resource (Novas and

Henning 2014). The waiting interval to reload the AGV is considered insignificant. The type of AGVs used in production planning has significant effects on material handling problem. In this paper, homogenous AGVs are used. Since only one kind of AGV has been utilized, the preparation time is assumed to be constant in the model. Therefore, using a constant value in the model can be neglected due to its low effect on the computing of cost function.

All in all, the FMS layout and the vehicle attribute are significant issues in determining transport times and have a critical impact on the resulting schedule. All of these are done to minimize the maximum time of operations completion and minimize the cost of processing, transportation, fuel cost, and storage.

3.3. Fuel Consumption Rate (FCR) considered for AGVs

There are many factors that affect the total cost of the routing problem. Generally speaking, it has been recognized that the cost of a vehicle travelling along a route depends on many factors, which have a direct and indirect relation with the en route path. In this regard, the FCR has been considered as an indirect factor measured by distance, fuel price, and so on. Therefore, the first group of factors, especially FCR, can be used as a variable cost. According to Zhang et al. (2014), an acceptable estimation of fuel consumption can be achieved on the basis of two factors, load and distance, in which other factors are assumed to be constant. In this regard, an empty fleet has less cost than a fully loaded one routing in a specific schedule at a particular speed.

With no loss of generality, we conclude that an AGV's gross weight can be divided into two sections in which it carries no load weight (Q_0) for the first section, and a load(Q_1) in the second section. Therefore, a linear function based on load can be given as follows:

$$\theta(Q_1) = \alpha(Q_0 + Q_1) + \beta \quad (1)$$

To obtain an efficient formula for FCR, we define Q for maximum load carried by AGV, and θ_0 and θ_* are used for empty-load and full-load of FCR, respectively. Therefore, the following Eqs. are given based on Eq. (1) as:

$$\theta_0 = \alpha Q_0 + \beta \quad (2)$$

$$\theta_* = \alpha(Q_0 + Q_1) + \beta \quad (3)$$

Considering the above Eqs. the value of α can be computed as:

$$\alpha = \frac{\theta_* - \theta_0}{Q} \quad (4)$$

So, we reformulate $\theta(Q_1)$ as:

$$\theta(Q_1) = \theta_0 + \frac{\theta_* - \theta_0}{Q} Q_1 \quad (5)$$

According to Eq. (5), it can be concluded that the most crucial factor in estimating fuel consumption is load, and there is a linear relationship between FCR and load of AGV, correspondingly.

For a given arc $\{m, \hat{m}\}$, where an AGV moves m^{th} to \hat{m}^{th} machine, the cost function of consumed fuel can be expressed as follows:

$$C_{m\hat{m}}^{fuel} = \theta_{m\hat{m}} d_{m\hat{m}} \kappa \quad (6)$$

where $d_{m\hat{m}}$ is the distance between m^{th} to \hat{m}^{th} and, κ denotes the unit cost of fuel. We simply assume that the load carried between two points can be expressed as $q_{m\hat{m}}$, leads to

$$\theta_{m\hat{m}} = \theta_0 + \frac{\theta_* - \theta_0}{Q} q_{m\hat{m}} = \theta_0 + \alpha q_{m\hat{m}} \quad (7)$$

4. Mathematical modelling

In this section, we first introduce the model, including assumptions, model subscripts, parameters, and related variables. Then, the mathematical model is given in detail. The following notation and assumptions are used throughout this paper:

4.1. Assumptions

- The precedence of each operation is considered.

- The operation time on each part and machine is determined.
- The time of the part loading and unloading on a machine is considered.
- The FMS has several types of AGV, but the type number is limited.
- Each part has a certain demand.
- The availability time of each machine is limited.
- The capacity of the fuel tank for the AGV is limited.
- Time is discrete.
- The preparing time of the empty AGV is insignificant.
- No preemption is allowed in operations.
- The first assigned AGV should take parts from loading to the machine.
- The last assigned AGV should take parts in the machine to unloading.

4.2. Notations

In the following, the notations used in the mathematical model formulation are shown in Table

1.

Table 1. Summary of Notation

Indices		
p	Index of part	$p = 1, 2, \dots, P$
o, o'	Index of operation	$o = 1, 2, \dots, O_p$
m, m'	Index of Machine	$m = 1, 2, \dots, M$
t	Index of time	$t = 1, 2, \dots, T$
k	Index of AGV	$k = 1, 2, \dots, K$
Set		
M_{po}	the set of Machine ability process o^{th} operation on p^{th}	
Parameters		
$T_{pkmm'}^{tr}$	Transportation time of p^{th} part by k^{th} AGV from machine m to m'	
T_{pom}^{pr}	Processing time of o^{th} operation on p^{th} part by machine m	
C_{pom}^{pr}	Processing cost of o^{th} operation on p^{th} part by machine m	
T_m^{at}	Availability time of machine m	
C_{pmmk}^{tr}	The transportation cost of p^{th} part from machine m to m' by k^{th} AGV per time	
C_{plmk}^{tr}	The transportation cost of p^{th} part from the loading station to machine m by k^{th} AGV per time	

C_{pmuk}^{tr}	The transportation cost of p^{th} part from machine m to unloading station by k^{th} AGV per time
A	A very large amount
θ_0^k	Empty-load of the of k^{th} AGV of fuel consumption rate model.
θ_*^k	Full-load of the of k^{th} AGV of fuel consumption rate model.
$d_{mm'}$	Distance between m^{th} machine to m'^{th} machine
κ	Fuel cost unit
q_p	The weight of of p^{th} part
Q^k	the maximal weight the AGV could carry
M^k	Vehicle fuel tank capacity of k^{th} AGV

Continues variables

y_{po}^{ft}	The finishing time of operation o on part p
y_{po}^{ut}	Unloading time of part p in operation o
y_{pom}^{lt}	loading time of part p on machine m before executed operation o
y_{pom}^{rt}	The receive time of part p beside machine m to start operation
y_{km}^f	Fuel level variable of k^{th} AGV specifying the remaining tank fuel level upon arrival to vertex m

Binary variables

x_{kt}^{aa}	If k^{th} AGV be available at time t , it is 1 otherwise 0
x_{pomkt}^{tr}	If the k^{th} AGV transfers p^{th} part from machine m after o^{th} operation at the time t , it is 1 otherwise 0
x_{plmkt}^{tr}	If the k^{th} AGV transfers p^{th} part from the loading station to machine m at the time t , it is 1 otherwise 0
x_{pmukt}^{tr}	If the k^{th} AGV transfers p^{th} part from machine m to unloading station at the time t , it is 1 otherwise 0
$x_{pmm'kt}^{tr}$	If the k^{th} AGV transfers p^{th} part from m^{th} machine to m'^{th} machine at the time t , it is 1 otherwise 0
x_{pomkt}^{or}	I If the k^{th} AGV transfers operation o of the part p from m^{th} at the time t , it is 1 otherwise 0
x_{pomt}^{or}	If operation o of the part p is executed at the time t by machine m , it is 1 otherwise 0.
$x_{po-1mom'}^{or}$	If operation o and $o - 1$ of part p are executed by machine m and m' respectively, it is 1 otherwise 0
x_{pom}^{or}	If operation o of part p is executed by machine m , it is 1 otherwise 0
$x_{mm'kt}^{or}$	If the k^{th} AGV moves from m^{th} machine to m'^{th} machine at the time t with empty-load, it is 1 otherwise 0
x_{p1m}^{or}	If the first operation of part p is executed by machine m , it is 1 otherwise 0

4.3. Mathematical model

In this sub-section, the mathematical model of the problem under study is expressed as follows:

(8)

$$\min Z_1 = \sum_{p=1}^P \sum_{m \in M_{op}} \sum_{k=1}^K \sum_{t=un_{pop}}^{Time} \left(x_{pmm'kt}^{tr} \left(\theta_0^k + \left(\frac{\theta_*^k - \theta_0^k}{Q^k} \right) \cdot q_p \right) \cdot d_{mm'} \right) +$$

$$\sum_{p=1}^P \sum_{m \in M_{op}} \sum_{k=1}^K \sum_{t=un_{pop}}^{Time} \left(x_{mm'kt}^{tr} (\theta_0^k) \cdot d_{mm'} \right)$$

(9)

$$\min Z_2 = \sum_{p=1}^P \sum_{o=1}^{o_p} \sum_{m \in M_{op}} \left(C_{pom}^{pr} \cdot x_{pom}^{or} \cdot T_{pom}^{pr} \right) +$$

$$\sum_{p=1}^P \sum_{o=1}^{o_p} \sum_{m \in M_{op}} \sum_{m' \in M_{op}} \left(C_{pmm'kt}^{tr} \cdot \sum_{t=1}^T x_{pmm'kt}^{tr} \right) + \sum_{p=1}^P \sum_{m \in M_{op}} \sum_{k=1}^K \sum_{t=r_p}^{load_{p1}} \left(C_{plmk}^{tr} \cdot \sum_{t=1}^T x_{plmkt}^{tr} \right) +$$

$$\sum_{p=1}^P \sum_{m \in M_{op}} \sum_{k=1}^K \sum_{t=un_{pop}}^{Time} \left(C_{pmuk}^{tr} \cdot x_{pmukt}^{tr} \right)$$

(10)

$$\sum_{m=1}^M x_{pom}^{or} = 1$$

 $\forall p, o$

(11)

$$\sum_{t=1}^T x_{pomt}^{or} = x_{pom}^{or} \cdot T_{pom}^{pr}$$

 $\forall p, o, m \in M_{op}$

(12)

$$t \cdot x_{pomt}^{or} \cdot t' \cdot x_{po+1m't'}^{or} \leq 0$$

 $\forall p, o, m \in M_{op}, m' \in M_{op},$ $t = 1, \dots, Time,$ $t' = t, \dots, Time$

(13)

$$x_{pomkt}^{tr} \leq x_{kt}^{aa}$$

 $\forall p, o, k, m \in M_{op}, t$ $= 1, \dots, Time$

(14)

$$y_{po}^{ut} = \min_{m \in M_{po}} \left(\min_{t \in C_{po}} \left(t \cdot (1 - x_{pomkt}^{tr}) \cdot A + 1 \right) \right)$$

 $\forall p, o, k, m \in M_{op}, t$ $= 1, \dots, Time$

(15)

$$x_{p1m}^{or} \leq \sum_{t=1}^{Time} \sum_{k=1}^K x_{plmkt}^{tr}$$

 $\forall p, k, m \in M_{op}$

(16)

$$y_{po}^{ft} \geq y_{pom}^{lt} + \left(T_{pom}^{pr} \cdot x_{pom}^{or} \right)$$

 $\forall p, o, m \in M_{op}$

(17)

$$F(k, q_p) = \theta_0^k + \left(\frac{\theta_*^k - \theta_0^k}{Q^k} \right) \cdot q_p$$

$$y_{m'k}^f \leq y_{mk}^f - d_{mm'}(F(k, q_p)) \cdot x_{pmm'kt}^{tr} \quad (18)$$

$$y_{mk}^f = \begin{cases} y_{mk}^f \geq d_{mm'}(F(k, q_p)) \cdot x_{pmm'kt}^{tr} & \rightarrow y_{mk}^f \\ y_{mk}^f < d_{mm'}(F(k, q_p)) \cdot x_{pmm'kt}^{tr} & \rightarrow M^K \end{cases} \quad (19)$$

$$x_{po-1m}^{or} \cdot x_{pom'}^{or} = x_{po-1mom'}^{or} \quad \forall p, o, m \in M_{op}, m' \in M_{op} \quad (20)$$

$$\sum_{t=un_{po-1}}^{Time} x_{po-1mkt}^{or} = \sum_{m' \in M_{op}} T_{pknm'}^{tr} \cdot x_{po-1mom'}^{or} \quad \forall p, k, m \in M_{op}, o = 2, 3, \dots, o_p \quad (21)$$

$$\sum_{m \in M_{op}} \left(\sum_{t=y_{pom}^{rt}}^{Time-1} |x_{pomt}^{or} - x_{pomt+1}^{or}| - 2 + x_{pomt}^{or}(y_{pom}^{rt}) \right) = 0 \quad \forall p, o, m \in M_{op} \quad (22)$$

$$y_{pom}^{st} = y_{pom}^{lt} - y_{pom}^{rt} \quad \forall p, o, m \in M_{op} \quad (23)$$

$$\sum_{p=1}^P \sum_{o=1}^{o_p} (T_{pom}^{pr} \cdot x_{pom}^{pr}) \leq T_m^{at} \quad \forall m \in M_{op} \quad (24)$$

$$Time = \sum_{p=1}^P \sum_{m=1}^M \sum_{o=1}^{o_p} T_{pom}^{pr} \quad (25)$$

$$y_{po}^{ft}, y_{po}^{ut}, y_{pom}^{lt}, y_{pom}^{rt}, y_{km}^f \geq 0 \quad \forall p, o, m \in M_{op} \quad (26)$$

$$x_{kt}^{aa}, x_{pomkt}^{tr}, x_{plmkt}^{tr}, x_{pmukt}^{tr}, x_{pmm'kt}^{tr}, x_{pomkt}^{tr}, x_{pmm'kt}^{tr}, x_{pomt}^{or}, x_{po-1mom'}^{or}, x_{pom}^{or}, x_{mm'kt}^{tr}, x_{p1m}^{or} \in 0, 1 \quad \forall p, o, k, t, m \in M_{op} \quad (27)$$

The first objective in Eq. (8) minimizes the total fuel consumption of loaded and empty machines that is the function of distance traversing among machines or L/U stations, and Eq.(9) minimizes the total cost. The first part of this objective calculates the processing cost of all operations by machines, and the second to the fourth parts measure the transportation cost of traversing among machines and L/U stations, respectively. Eq. (10) ensures that each operation in every part is executed by one machine. Equality constraint (11) ensures that the sum of the time of every operation on a part must be equal to the processing time of that operation on the machine. Inequality constraint (12) ensures the precedence relations.

Inequality constraint (13) indicates that k th AGV must be available at time t to be assigned to an operation/machine. Eq. (14) computes the part unloading time, which indicates the time that operation o of part p (after completion) assigns to AGV. Inequality constraint (15) assures that one AGV should transfer part p for starting the process from the loading station. Inequality constraint (16) computes the part processing completion time. Eq. (17) along with inequality constrain (18) computes the fuel level upon arrival at vertex j based on the distance travelled from vertex i and the vehicle's FCR. Eq. (20) shows the relation between zero-one variables. Eq. (21) ensures that the assigned time for the AGV in operation o of part p is equal to the transportation time of that part for the next operation. Eq. (22) indicates that all the operations are non-preemptive and must be continued to complete the operation. Eq. (23) computes the waiting time of part p in operation o on machine m . Inequality constraint (24) ensures that the time in which a machine is working is less than the availability time of that machine. Eq. (25) defines an upper bound for time. Continuous and binary variables are defined in Eqs. (26) and (27), respectively.

5. Proposed solving algorithm

In this study, multi-objective Particle Swarm Optimization (MOPSO) model is proposed for solving the problem. A general overview about PSO can be found in Appendix A. Since the reciprocal analysis is complicated in MOPSO, Pareto-frontier methodology is used to compare the objective functions. In addition to the MOSPO algorithm, a new MMOPSO algorithm is presented to solve the problem.

Figure 2 illustrates the schematic procedure of MOPSO and MMOPSO for our FMS scheduling algorithm.

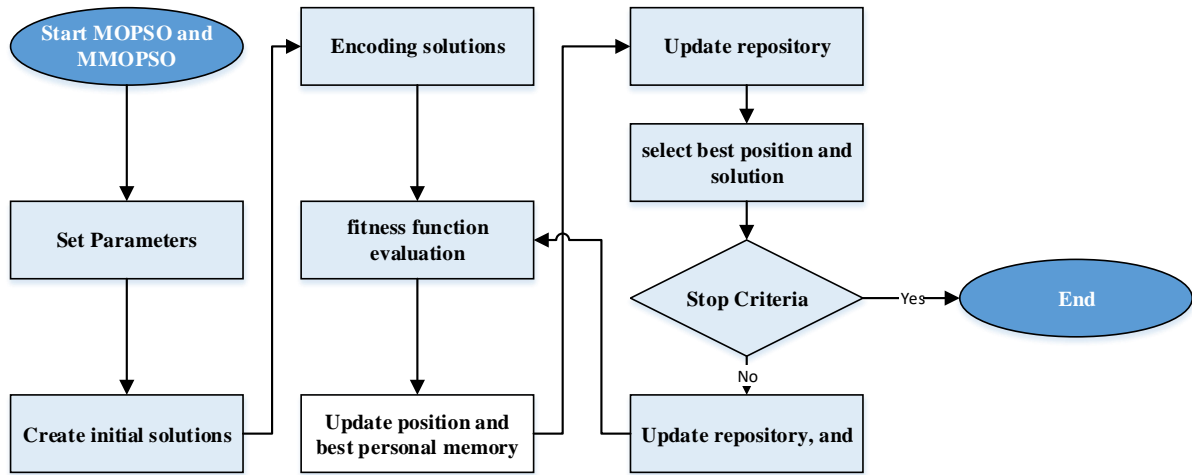


Figure 2. Flowchart of MOPSO & MMOPSO

It is worth mentioning here that the differences between two proposed algorithms have been considered in the updating position and the best personal memory, highlighted in the white rectangle shown in Figure 2.

5.1. Method of encoding and decoding

Three rows represent the solution encoding with the length of the total number of operations. The encoding consists of these rows randomly generated between $[0,1]$. Suppose, the three rows of our example are shown as follows:

Part and operation	0.634	0.525	0861	0797	0.036	0.976	0.080
Machine	0.721	0.815	0.061	0.507	0.476	0.763	0.895
AGV	0.901	0.625	0.231	0.743	0.146	0.963	0.605

Figure 3. An example of an encoding method

The first row represents part assignments and their operation sequence, the second row represents machine assignments, and the third row expresses the AGV assignment. For example, consider we have 6 machines, 3 AGVs, and 4 parts and that the parts have $O_p = [3,2,3,3]$ operations, respectively. The length of the row is equal to 11, which is the summation

of operations($11 = 3 + 2 + 3 + 3$). Therefore, the following code is used to generate the encoding procedure:

Function $CE=CreateEncoding(O_p)$

$$SumO = \sum_{i=1}^P O_p$$

$CE=$ random generation $[3, SumO]$;

End

where CE denotes the initial solutions for encoding the matrix with three rows and $SumO$ columns as shown in Figure 3. Note that we have defined any assignment as a step in which the members of predefined sets are changed throughout the process of solving the problem. At the first step, we have assigned the first row of encoding to compute the part and operation assignment positions, which are projected in the first and second row of decoding, respectively. Let \mathbb{P} be the part index which is selected for the i th number of the first row of encoding, so that, the following formula is given:

$$\mathbb{P} = [CE(1, i) \times np] \quad (28)$$

In the above formula, np is the number of parts that can be assigned in the current step, \mathbb{P} denotes the part index to the i th number for encoding. For a given problem, two different sets, namely $\mathcal{C}\mathcal{P}$ and $\mathcal{C}\mathcal{O}$, are defined, which are used to control the number of parts and operations, respectively. In this regard, we first define the member of set $\mathcal{C}\mathcal{P}$ that shows the number of available parts for the first step and $\mathcal{C}\mathcal{O}$ is defined based on the index members of $\mathcal{C}\mathcal{P}$ that have greater values than zero ($\mathcal{C}\mathcal{O} = \{x \in \mathcal{C}\mathcal{O} | (\mathcal{C}\mathcal{P}(x)) > 0\}$). Consider each assignment as a consecutive as shown in the following formula:

$$\mathcal{C}\mathcal{P}(\mathbb{P}) = \mathcal{C}\mathcal{P}(\mathbb{P}) - 1 \quad (29)$$

and then members of set $\mathcal{C}\mathcal{O}$ are updated. Therefore, the following formula is given for the part assignment:

$$part(i) = \mathcal{C}\mathcal{O}(\mathbb{P}) \quad (30)$$

Another critical aspect of the decoding procedure is related to the operation position, in which the index of operation, as mentioned earlier, can be grasped from the first part of the encoding. Let \mathcal{JO} be a set of operations of part \mathbb{p} which are not assigned. So, we have set the smallest index of operation of set \mathcal{JO} . Therefore, the following formula is given

$$\text{Operation}(i) = \min(\mathcal{JO}\{\mathbb{p}\}) \quad (31)$$

Table 2 describes how to determine the part and operation based on a given solution. As we can see in this paper, each assignment has been defined as a step, so in the first step, the absolute value of 0.634, the first number of the first row of encoding, multiple to the number of member set $\mathcal{CO} = \{1,2,3,4\}$, equal to 4, is regarded as the index of the part in set \mathcal{CO} , and the operation of the step has been determined via Eq.(31). To show how the first and second rows of the decoding procedure are obtained, we have provided Table 2 as shown as follows:

Table 2. The computation of part and operation for an example

Steps	\mathcal{CP}	\mathcal{CO}	\mathbb{p}	$part$	\mathcal{JO}	$operation$
1	{2,1,1,3}	{1,2,3,4}	$\lceil 0.634 \times 4 \rceil = 3$	$\mathcal{CO}(3) = 3$	$\mathcal{JO}_{\{3\}} = (1)$	1
2	{2,1,0,3}	{1,2,4}	$\lceil 0.525 \times 3 \rceil = 2$	$\mathcal{CO}(2) = 2$	$\mathcal{JO}_{\{2\}} = (1)$	1
3	{2,0,0,3}	{1,4}	$\lceil 0.861 \times 2 \rceil = 2$	$\mathcal{CO}(2) = 4$	$\mathcal{JO}_{\{4\}} = (1,2,3)$	1
4	{2,0,0,2}	{1,4}	$\lceil 0.797 \times 2 \rceil = 2$	$\mathcal{CO}(2) = 4$	$\mathcal{JO}_{\{4\}} = (2,3)$	2
5	{2,0,0,1}	{1,4}	$\lceil 0.036 \times 2 \rceil = 1$	$\mathcal{CO}(1) = 1$	$\mathcal{JO}_{\{1\}} = (1,2)$	1
6	{1,0,0,1}	{1,4}	$\lceil 0.976 \times 2 \rceil = 2$	$\mathcal{CO}(2) = 4$	$\mathcal{JO}_{\{4\}} = (3)$	3
7	{1,0,0,0}	{1}	$\lceil 0.080 \times 1 \rceil = 1$	$\mathcal{CO}(1) = 1$	$\mathcal{JO}_{\{1\}} = (2)$	2

The second row of CE, which is related to encoding, is used for machine assignment, in which the \mathbb{M} value is defined to determine the i th number of the second row connected to the index of the machine. Therefore, the following formula is given to decode the second row of CE, which is projected in the third row of decoding:

$$\mathbb{M} = \lceil CE(2, i) \times nm \rceil \quad (32)$$

Where nm is the number of the machine which can be assigned to the considered part and operation. Let $\mathcal{OM}_{(xy)}$ be a limited set that expresses the eligible machine for x th part and y th

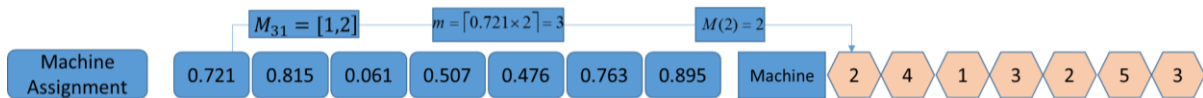
operation; therefore, nm is equal to the number of member $\mathcal{OM}_{(xy)}$, then, the following relation is used to determine machine position

$$Machine(i) = \mathcal{OM}_{(xy)}(\mathbb{M}) \quad (33)$$

To show clearly how this part of the decoding method is implemented, Table 3 are provided in this regard, and the last row graphically shows the computation related to this table.

Table 3. The computation of machine for an example

Step	\mathcal{OM}	\mathbb{M}	$machine$
1	$\mathcal{OM}_{31} = [1,2]$	$\lceil 0.721 \times 2 \rceil = 3$	$\mathcal{OM}_{31}(2) = 2$
2	$\mathcal{OM}_{21} = [3,4]$	$\lceil 0.815 \times 2 \rceil = 2$	$\mathcal{OM}_{21}(2) = 4$
3	$\mathcal{OM}_{41} = [1,2,4,5]$	$\lceil 0.061 \times 4 \rceil = 1$	$\mathcal{OM}_{41}(1) = 1$
4	$\mathcal{OM}_{42} = [1,3,5]$	$\lceil 0.570 \times 3 \rceil = 2$	$\mathcal{OM}_{42}(2) = 3$
5	$\mathcal{OM}_{11} = [2,3]$	$\lceil 0.476 \times 2 \rceil = 1$	$\mathcal{OM}_{11}(1) = 2$
6	$\mathcal{OM}_{43} = [1,4,5]$	$\lceil 0.763 \times 3 \rceil = 2$	$\mathcal{OM}_{43}(3) = 5$
7	$\mathcal{OM}_{12} = [1,2,3]$	$\lceil 0.895 \times 3 \rceil = 3$	$\mathcal{OM}_{12}(3) = 3$



After determining all steps, the indices of the obtained part and operation are applied for the machine assignment considering the second row of encoding input. This procedure will be continued on the third row of encoding, CE , to determine the AGV position on the corresponding solution. Let \mathcal{A} be the AGV index which is selected for the i th number of the third row of CE , so that, the following formula is given:

$$\mathcal{A} = \lceil CE(3, i) \times na \rceil \quad (34)$$

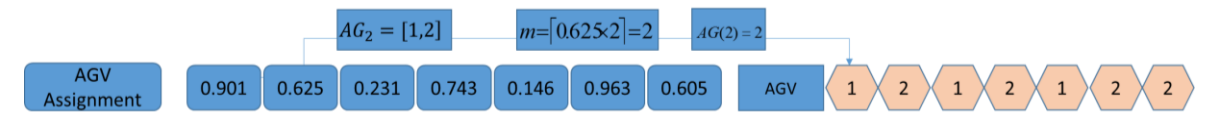
where na is the number of AGVs which can be assigned to the corresponding part. Let \mathcal{AG} be a limited set that expresses the eligible AGV for different parts, and then, the following relation is used to decode the third row of CE :

$$Agv(i) = \mathcal{AG}_{\{part\}}(\mathcal{A}) \quad (35)$$

where Agv denotes the fourth row of the decoding procedure in which the index of AGV is expressed as i th part. Table 4 is provided to clarify more details, and the last row shows the computation of this stage, respectively.

Table 4. The computation of AGV for an example

Step	Part	\mathcal{AG}	\mathcal{A}	\mathcal{Agv}
1	3	$\mathcal{AG}_{\{3\}} = [1]$	$\mathcal{A} = [0.901 \times 1] = 1$	$\mathcal{AG}_{\{3\}}(1) = 1$
2	2	$\mathcal{AG}_{\{2\}} = [1,2]$	$\mathcal{A} = [0.625 \times 2] = 2$	$\mathcal{AG}_{\{2\}}(2) = 2$
3	4	$\mathcal{AG}_{\{4\}} = [1,2]$	$\mathcal{A} = [0.231 \times 2] = 1$	$\mathcal{AG}_{\{4\}}(2) = 2$
4	4	$\mathcal{AG}_{\{4\}} = [1,2]$	$\mathcal{A} = [0.743 \times 2] = 2$	$\mathcal{AG}_{\{4\}}(2) = 2$
5	1	$\mathcal{AG}_{\{1\}} = [1,2]$	$\mathcal{A} = [0.146 \times 2] = 1$	$\mathcal{AG}_{\{1\}}(2) = 2$
6	4	$\mathcal{AG}_{\{4\}} = [1,2]$	$\mathcal{A} = [0.963 \times 2] = 2$	$\mathcal{AG}_{\{4\}}(2) = 2$
7	1	$\mathcal{AG}_{\{1\}} = [1,2]$	$\mathcal{A} = [0.605 \times 2] = 2$	$\mathcal{AG}_{\{1\}}(2) = 2$



Therefore, with respect to the procedure implemented in the decoding process, Figure 4 is obtained, representing the final solution.

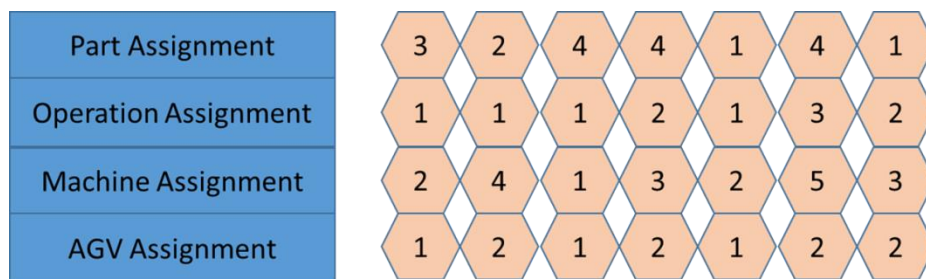


Figure 4. Decoding method for Meta-heuristics algorithms

To illustrate how the decoding method shows its efficacy in our study, Figure 5 is given, where two types of AGVs are distinctive based on two different colors. Also, each AGV can have different trips, including an empty trip and loaded trip. This example involves only two AGVs with different capacity of fuel tanks ($\mathcal{M}^1 = 100$ and $\mathcal{M}^2 = 120$). It is simply assumed that fuel stations are available in all machine stations.

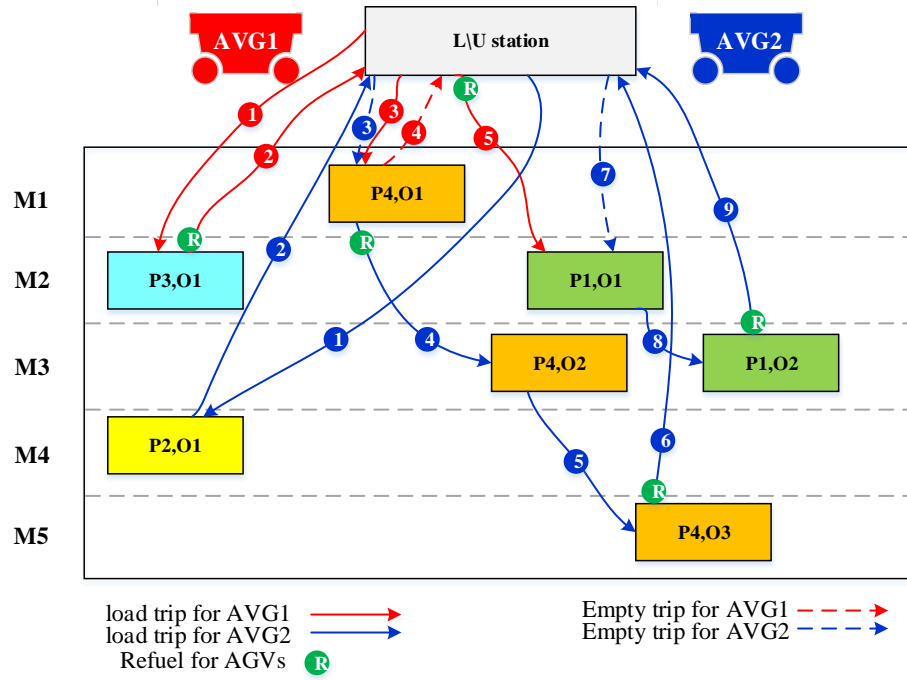


Figure 5. Graphical representation for the current example

For this example, the main parameters of the FCR model are $\theta_*^1 = 10$, $\theta_0^1 = 1$, $\theta_*^2 = 10$ and $\theta_0^2 = 1$. To determine the fuel rate of the AGV, which is based on distance and load, we set the maximal weight ($Q^1 = 20, Q^2 = 25$) that AGVs can carry. We also assume if an AGV transfers a last operation of the part to a machine, it will carry the completed part to the L/U station, correspondingly. Table 5 is provided to show how the weight of load and distance affects the fuel consumption and refueling requests and accumulated through a specific FMS problem.

Table 5. The procedure of refueling for the current solution

Type of AGV	Move number	From	To	Weight of load	Distance	Current Level of Battery	Fuel needed for the move	Refuel
AGV1	1	L/U	M2	10	10	100	55	No
	2	M2	L/U	10	10	45	55	Yes
	3	L/U	M1	8	5	45	23	No
	4	M1	L/U	0	5	32	5	No
	5	L/U	M2	9	10	27	50.5	Yes
AGV2	1	L/U	M4	6	9	120	44.28	No
	2	M4	L/U	6	9	75.72	44.28	No
	3	L/U	M1	0	5	31.44	15	No

Type of AGV	Move number	From	To	Weight of load	Distance	Current Level of Battery	Fuel needed for the move	Refuel
	4	M1	M3	8	8	16.44	44.48	Yes
	5	M3	M5	8	6	75.52	33.36	No
	6	M5	L/U	8	11	42.16	61.16	Yes
	7	L/U	M2	0	10	58.84	30	No
	8	M2	M3	9	4	28.84	23.52	No
	9	M3	L/U	9	7	5.32	41.16	Yes

According to Figure 5 and Table 5, fourteen moves must be implemented to complete the whole process of FMS in this example. For both AGVs, the first move is initiated from the L/U station taking the first operation of a part to a machine. When an AGV starts from the L/U station, the capacity of the fuel tank is set to full level, and before each move, the availability of the fuel is checked with respect to distance and load, making a decision to refuel is taken into consideration; therefore, in this case, refueling has taken place four times for both AGVs.

5.2. Generating the initial solution

In MOPSO, the size of the population is represented by $nParticle$. The initial population is produced randomly and, the target for the initial population is then calculated. A repository is considered as the collection of non-dominant solutions with a minimum capacity of 5, and a maximum capacity of twice the size of the population ($2 * nParticle$). Solutions are copied to sort non-dominant population criteria and crowding distance criteria and kept in their respective repository set.

5.3. Select leader

To move the position of particles, we need a leader which is determined by taking the following steps:

1. Grid all particles in the repository

2. Calculate the grid selection probability for every grid via $P_{grid} = e^{-beta.N_{grid}}$ in which N_{grid} is the number of particles in that grid and $beta$ is the leader selection pressure.
3. Choose one grid based on grid selection probability via the tournament selection algorithm.
4. Randomly choose one particle as a leader between the particles of the selected grid.

5.4. Particle velocity and position update

For the proposed multi-objective FMS scheduling of this paper, the MOPSO algorithm applies traditional operators known in the origin PSO in order to update the particle velocity and position; however, MMOPSO uses the procedure given in Eq. (36) in order to increase the efficiency of the algorithm and prevent it from being trapped in the local optimum.

if $rand \leq 0.5$

$$V_{ij}^{t+1} = w \cdot V_{ij}^t + C_1 \cdot r_1 \cdot (Pbest_{ij}^t - X_{ij}^t) + C_2 \cdot r_2 \cdot (Leader_{ij}^t - X_{ij}^t)$$

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1}$$

(36)

else

$$X_{ij}^{t+1} = r \cdot opt_i^t + (1 - r) \cdot (opt_i^t - X_{ij}^t)$$

$$V_{ij}^{t+1} = X_{ij}^{t+1} - X_{ij}^t$$

End

Where the random number r is $U \sim [0,1]$ and opt^t is the member chosen randomly from the external repository.

5.5. Fitness evaluation particle and update repository members

After updating the position of the particles, the fitness of solutions must be evaluated. Since it is not possible to evaluate the fitness of one solution against others for the multi-objective problems, we must use the terms dominant and non-dominant. To update the repository, all the members of the particle are added to the repository set, and then the non-dominant particle will be sorted, while three possible conditions can happen:

- If the number of non-dominant members of Rank 1 is less than the lower bound of the repository, the remaining numbers are provided from Rank 2. The non-dominant members of Rank 1 and Rank 2 remain in the repository set, and the dominant members are discarded.

- If the number of dominant members of Rank 1 is less than the upper bound of the repository and more than the lower bound, the existing non-dominant members remain in the repository set, and the dominant members are discarded.
- If the number of dominant members of Rank 1 is more than the upper bound of the repository, ($2nParticle$) the superior number remains in the repository set according to crowding distance criteria, and the rest are discarded.

In this step, the repository set is updated to be used in the next iteration. Then, the stop condition is checked. If the algorithm does not meet the stop criteria, the best personal, as well as global position, must be updated for the next iteration.

5.6. Update the best personal position (*Pbest*)

To update *Pbest* of the MOPSO algorithm for solving problems, the procedure as described in (37) is followed.

```

If  $Pbest_i^t > X_i^{t+1}$ 
   $Pbest_i^{t+1} = Pbest_i^t$ 
Elseif  $Pbest_i^t < X_i^{t+1}$ 
   $Pbest_i^{t+1} = X_i^{t+1}$ 
Else
  Choose one of them randomly
End

```

(37)

In multi-objective problems, several objectives must be optimized simultaneously. Therefore, the number of non-dominant solutions on the Pareto front or close to it is more than one. So, every non-dominant solution can be a *leader* and change its position with another particle. By searching the behaviour of the particles, *Pbest* of a particle can be its current position as well. Thus, *Pbest* and *leader* cannot be used for guiding the particles to find new solutions. To solve these issues, the proposed MMOPSO algorithm can be used to update *Pbest* as follows (Qiao 2012):

```

if  $rand_j < 0.5$ 
   $Pbest_{ij}^{t+1} = X_{ij}^t$ 

```

(38)

Else

$$Pbest_{ij}^{t+1} = X_{i1j}^t + r \cdot (X_{i2j}^t - X_{i3j}^t)$$

End

Where the random number r is taken from $U \sim [0,1]$. In the same way, X_{i1j}^t , X_{i2j}^t , and X_{i3j}^t are the members chosen randomly from the evolutionary population. The pseudo code of MOPSO and MMOPSO are presented in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1. The MOPSO

Input: nParticle, C_1 , C_2 , W , Max iteration

Output: The repository front

- 1: Generate the initial particle.
- 2: Evaluate the fitness value of the initial particle.
- 3: Calculate the best personal memory.
- 4: Create the best leader memory.
- 5: Create a grid index for the solution dimension.
- 6: Determine the repository member.
- 7: Determine the grid for repository members.
- 8: **For** it = 1: Max iteration
- 9: **For** j = 1: nParticle
- 10: Update leader memory
- 11: Update particle position by Eqs. (A.1) and (A.2)
- 12: Evaluate the fitness value of the particle.
- 13: Update the best personal memory using Eq. (37)
- 14: **End For**
- 15: Determine repository members.
- 16: Combine new repository members with old repository members $rep = \{rep \cup rep_{new}\}$
- 17: Update repository members by the non-dominance sorting algorithm.
- 18: Update the grid index for the solution dimension.
- 19: Find the grid for repository members.
- 20: Delete extra repository members.
- 21: **End For**
- 22: **Return** the repository front.

Algorithm 2. The MMOPSO**Input:** nParticle, C_1 , C_2 , W, Max iteration.**Output:** The repository front.

```
1:   Generate the initial particle.
2:   Evaluate the fitness value of the initial particle.
3:   Calculate the best personal memory.
4:   Create the best leader memory.
5:   Create a grid index for the solution dimension.
6:   Determine the repository member.
7:   Determine the grid for repository members.
8:   For it = 1: Max iteration
9:       For j = 1: nParticle
10:          Update leader memory
11:          Update particle position by Eq. (36)
12:          Evaluate the fitness value of the particle.
13:          Update the best personal memory using Eq. (38)
14:       End For
15:   Determine repository members.
16:   Combine new repository members with old repository members  $rep = \{rep \cup rep_{new}\}$ 
17:   Update repository members by the non-dominance sorting algorithm.
18:   Update the grid index for the solution dimension.
19:   Find the grid for repository members.
20:   Delete extra repository members.
21:   End For
22:   Return the repository front.
```

The above algorithms will be repeatedly performed until the stop condition is met. Meeting the stop condition, the non-dominant solution of Rank 1 in the repository set will be considered as the final solution.

Since there is no executive library for FMS Scheduling, all data in this paper have been generated randomly and presented in **Appendix B**.

6. Tuning algorithm parameters

Without a doubt, the results of the MOPSO and MMOPSO algorithms significantly depend on their parameters to attain better fitness function value. The main parameters of a PSO algorithm that should be calibrated at their best level are: cognitive factor (C_1), social factor (C_2), swarm size (N-Particle), maximum iteration (Max It), and inertia factor (w). Thus, in this section, the parameters of MOPSO and MMOPSO algorithms are calibrated by using the Taguchi method (Sadeghi and Niaki 2015).

Instead of the Fisher factorial designs, Taguchi developed fractional factorial experiments to reduce the complexity of experiments in full factorial designs (Taguchi, Chowdhury and Wu 2005). The Taguchi method presents two result analyses including: (1) analysis of variance for experiments with a single replicate, (2) the signal to noise ratio (S/N) for experiments with multiple replications when (N) is the noise factor and (S) is a controllable factor. Since the one with multiple replications has better performance, the S/N is applied in this research to analyze the solutions. In this paper, an updated version of the Taguchi method, namely the Taguchi method based on Grey Relational Analysis (GRA) has been employed. In the **Appendix C**, more information regarding the approach is presented.

In this study, Minitab software is applied, and L^{27} orthogonal array design is selected to fulfil the GRA method in order to achieve the optimum levels of the proposed algorithm parameters. Therefore, the optimum levels of parameters based on the GRA method are shown in Table 6. With respect to the results shown in Figure 6, the highest value of S/N is the most favorable for the optimum level of all parameters.

Table 6. Defined level of parameters and obtained optimum value

Algorithm	Number of algorithm parameter	Parameter	Level			Optimal Value
			Level 1	Level 2	Level 3	
MOPSO	Personal Learning Coefficient	C1	1	1.4962	2	1
	Global Learning Coefficient	C2	1	1.4962	2	2
	Inertia Weight	W	0.6	0.7298	0.9	0.7298
	Max Iteration	Max It	5*N	10*N	15*N	15*N
	Number of Particle	N Particle	100	150	200	200
	Number of Repository	N Rep	50	70	100	70
	Number of grid	N Grid	5	8	10	10
	Inflation rate	Alpha	0.05	0.1	0.15	0.1
	Leader selection pressure	Beta	1	2	3	2
	Deletion selection pressure	Gamma	1	2	3	1
MMOPSO	Personal Learning Coefficient	C1	1	1.4962	2	1.4962
	Global Learning Coefficient	C2	1	1.4962	2	2
	Inertia Weight	W	0.6	0.7298	0.9	0.6
	Max Iteration	Max It	5*N	10*N	15*N	15*N
	Number of Particle	N Particle	100	150	200	200
	Number of Repository	N Rep	50	70	100	70
	Number of grid	N Grid	5	8	10	5
	Inflation rate	Alpha	0.05	0.1	0.15	0.15
	Leader selection pressure	Beta	1	2	3	2
	Deletion selection pressure	Gamma	1	2	3	2

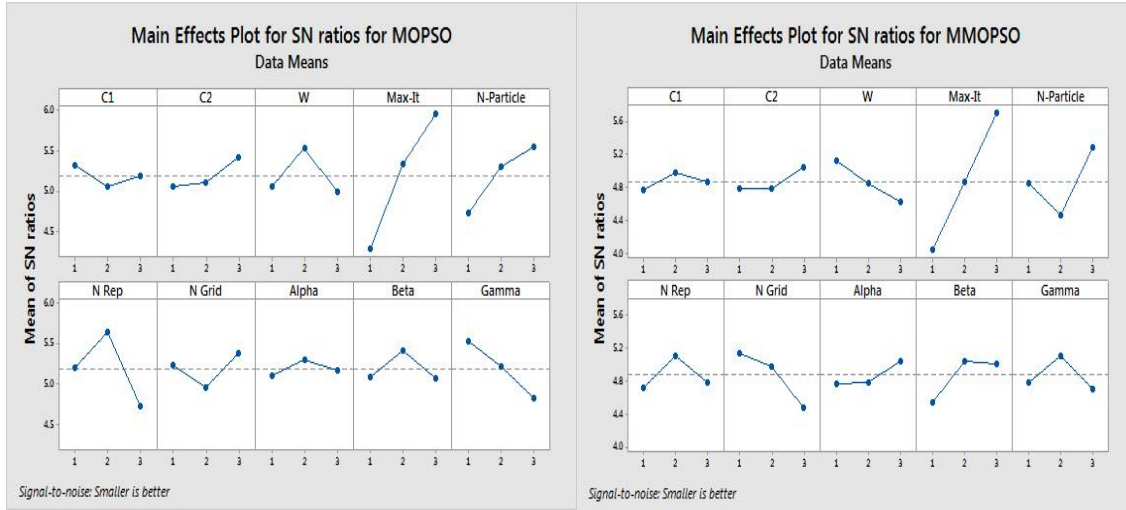


Figure 6. S/N ratio for all parameters of MOPSO (Left) and MMOPSO (Right)

Five different metrics are considered to specify the quality of the solution obtained from Multi Objective Evolutionary Algorithms (MOEAs). They are: (1) mean ideal distance (MID), (2) spacing (SP), (3) spread of non-dominated solutions (SNS), (4) CPU Time (CT) and (5) max spread (MS). Interested readers may refer to Maghsoudlou, Afshar-Nadjafi and Akhavan Niaki (2017) for more details. It should be noted that the considered performance metrics are widely used by several researchers (see, for example, (Wang et al. 2012, Panigrahi et al. 2010). Note that since a minimization problem is involved, MID, SP, and CT with lower values explain better efficiency, while SNS and MS with higher values explain the better efficiency of MOEAs.

7. Computational Results and Discussion

In this section, we discuss the ability of the two proposed algorithms (MOPSO, MMOPSO) on the test problems. The model is implemented in Matlab Software R2018a, on a PC with CPU core i5, 4GB ram, and OS Windows 10. In this step, the results of the algorithms on different problems are obtained and formed in a decision matrix. Table 7 illustrates the results obtained by MOPSO and MMOPSO.

Table 7. Results of metrics for MMOPSO and MOPSO

Number of Problem	MOPSO					MMOPSO				
	MID	SP	SNS	CT	MS	MID	SP	SNS	CT	MS
1	7.29	0.28	742.42	13.31	174.09	5.51	0.18	787.67	10.73	157.06
2	7.72	0.24	1117.53	23.17	375.06	9.16	0.26	1114.81	19.02	386.02
3	7.83	0.21	2084.36	34.66	122.49	11.1	0.16	2170.84	28.68	608
4	6.21	0.56	1911.45	51.56	302.15	12.17	0.13	2323.43	49.02	148.12
5	8.27	0.12	2514.57	74.41	1099.08	9.14	0.1	2391	61.49	332.15
6	12.94	0.02	2905.34	105.51	810.05	11.66	0.08	2753.13	85	517.21
7	14.52	0.12	4074.32	144.36	1223.39	12.98	0.09	3934.98	117.64	1136.18
8	12.8	0.09	4327.27	176.05	1255.11	13.62	0.1	4459.02	142.67	1052.09
9	16.16	0.11	4707.23	207.57	791.02	15.84	0.08	4782.14	164.32	1082.1
10	15.19	0	6235.48	275.52	977.19	13.13	0.09	6559.34	345.38	931.12
11	12.2	0.19	8283.71	420.6	1390.32	13.11	0.15	8598.44	521.56	1079.12
12	11.06	0.2	11012.8	585.14	1018.52	20.33	0.05	10885.9	733.13	1809.23
13	18.17	0.15	12424.31	733.61	943.15	19.02	0.08	11758.34	924.2	1686.3
14	13.86	0.15	12446.94	965.09	1446.58	17.7	0.11	12630.77	1209.26	1563.38
15	17.17	0.07	16305.89	1354.02	1630.55	16.04	0.07	15678.38	1688.48	1527.67
16	14.95	0.07	17751.84	1917.78	2747.19	15.1	0.09	16839.89	1538.13	1527.42
17	21.49	0	21097.36	2532.1	1570.99	31.37	0.08	21346.48	2020.61	3620.13
18	17.53	0.04	19965.89	2539.69	2558.18	26.56	0.14	20248.5	2041.35	3259.04
19	19.18	0.07	21588.75	3174.22	2465.05	15.83	0.05	20854.16	2525.63	2016.97
20	25.72	0.03	24415.59	3867.85	3041.47	13.68	0.04	24472.65	3088.25	1241.8
21	19.49	0.1	26842.81	4488.36	2825.4	18.28	0.21	27757.3	3569.63	2810.37
22	18.68	0.04	29553.92	5009.49	3240.21	32.61	0.02	29384.85	3949.89	3443.34
23	22.09	0.1	30575.83	5609.97	2292.98	11.06	0.2	32097.39	4454.51	1879.16
24	27.03	0.26	36301.57	6781.46	4548.15	30.06	0.13	35021.31	5202.24	2553.27
25	24.75	0.1	39333.53	8064.57	3594.46	32.07	0.02	39600.83	6204.46	2324.28
26	23.38	0.07	41719.3	9481.06	3950.43	24.26	0.12	42725.16	7286.56	4420.63
27	29.4	0.01	50852.3	11752.28	3518.88	37.69	0.02	49990.88	8995.87	4066.44
28	16.83	0.49	54837.91	15859.63	3104.2	33.65	0.09	55318.17	12194.6	4650.11
29	28.7	0.2	58487.96	18615.84	4004.35	27.25	0.19	60026.27	14205.8	2729.51
30	21.71	0.22	67766.03	21513.2	6983.41	33.32	0.15	68683.78	16504.2	3146.23

Figure 7 shows the overall trend of obtained metrics for both proposed MOEAs. It can be concluded that there have been gradual increases in the obtained value of MID, SNS, MS and CT metrics as the problem size increases. In this figure, there is no significant relationship for the values of the SP metric. By reviewing the obtained values of the CT and SNS metrics, it can be concluded that they have kept pace over the problem size, and surprisingly both of them are the same gradient.

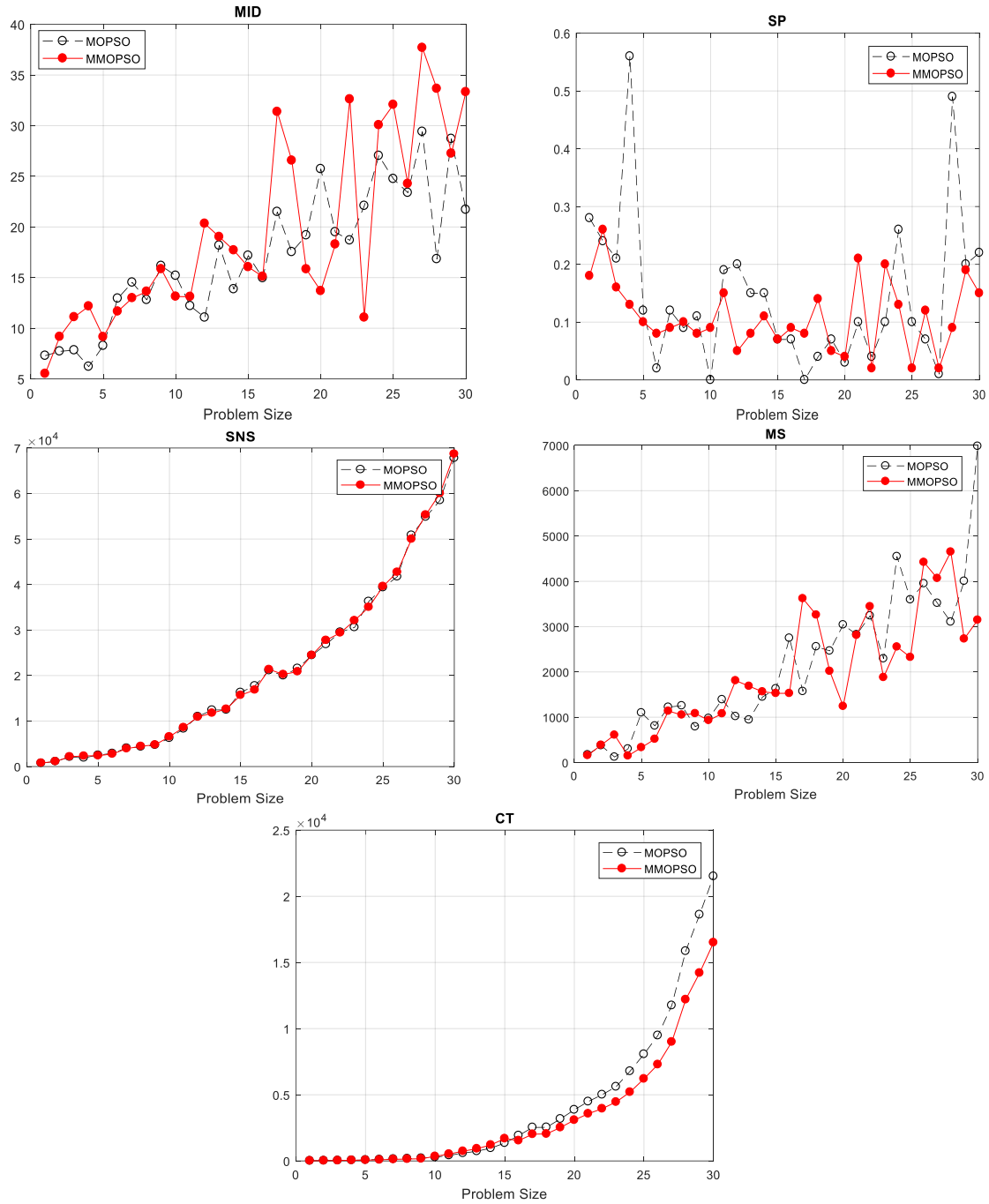


Figure 7. The trends of the metrics over the test problems

To analyze and compare the ability of two algorithms, we use a MCDM method to show the superiority of the proposed approach. In other words, since we use two different multi-objective evolutionary algorithms, it is required to measure the importance of metrics to evaluate the performance of the algorithms, and then applying MCDM models is useful. However, in the literature, most of the typical MCDM methods put the responsibility of criteria importance

weights on decision-makers, while the valuation of criteria sometimes leads to diverse opinions and meanings. We use the Analytic Hierarchy Process (AHP) method to obtain the overall weight of the metrics by using five expert opinions. The overall weight of the metrics (criteria) based on the AHP method for MID, Spacing, SNS, CPU Time, and Max Spread are 0.3, 0.2, 0.1, 0.2, and 0.2, respectively.

After obtaining the weighting of the criteria, we should evaluate and rank the algorithms with the MCDM model. A TOPSIS method, which is presented by Hwang and Yoon (1981) is a well-known MCDM type of modelling (Akhavan et al. 2015). We use this method to prioritize algorithms (MOPSO, MMOPSO) based on weighted criteria resulting from the AHP method. In this model, first of all, the problems are divided into three: small, medium and large groups, and the average of five criteria is obtained then this matrix is normalized and multiplied to the criteria weights obtained from the AHP (see Table 8).

After calculating the normalized weighted decision matrix, the Euclidean distance of alternatives from positive and negative ideal solutions can be calculated using Eqs. (39)-(40). The positive ideal solution is equal to the largest value for positive criteria and the smallest value for negative criteria. In contrast, the negative ideal solution is equal to the smallest value for positive criteria and the largest value for negative criteria.

$$d_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad i = 1, \dots, m \quad (39)$$

$$d_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad i = 1, \dots, m \quad (40)$$

Finally, the relative closeness value for each alternative is calculated using Eq. (41). The alternative with the larger relative closeness value should be selected as the best one.

$$CL_i = \frac{d_i^-}{d_i^- + d_i^+} \quad (41)$$

The average decision matrix, normalized weighted decision matrix, Euclidean distance of alternatives, and relative closeness values of alternatives for MOPSO and MMOPSO are shown in Table 8.

Table 8. The result of the TOPSIS method for problems

Problem Size	MOEAs	Average of decision matrix					Normalized weight of decision matrix					d_i^+	d_i^-	CL_i	Rank
		MID	SP	SNS	CT	MS	MID	SP	SNS	CT	MS				
Small size	MOPSO	0.68	0.83	0.70	0.77	0.75	0.20	0.17	0.07	0.15	0.15	0.06	0.02	0.28	2
	MMOPSO	0.73	0.56	0.71	0.63	0.66	0.22	0.11	0.07	0.13	0.13	0.02	0.06	0.72	1
Medium size	MOPSO	0.66	0.81	0.71	0.62	0.65	0.20	0.16	0.07	0.12	0.13	0.05	0.04	0.45	2
	MMOPSO	0.75	0.59	0.70	0.78	0.76	0.22	0.12	0.07	0.16	0.15	0.04	0.05	0.55	1
Large size	MOPSO	0.65	0.76	0.70	0.79	0.76	0.20	0.15	0.07	0.16	0.15	0.04	0.04	0.47	2
	MMOPSO	0.76	0.65	0.71	0.61	0.65	0.23	0.13	0.07	0.12	0.13	0.04	0.04	0.53	1

8. Conclusion and future research

This section summarizes and discusses the main findings of the work. This paper presents a novel multi-objective mathematical formulation that considers the resource-constrained short-term FMS scheduling problem with several critical features that are present in many FMS environments. To the best of our knowledge, this is the first approach in which several limiting resources in a comprehensive formulation are taken into account. Moreover, the following problems are also covered: (1) machine loading and unloading, (2) manufacturing operation scheduling, (3) machine plant and allocation, and (4) AGV scheduling. On the other hand, the fuel consumption model, considering the sustainability approach, has been embedded in this study as a distinctive objective function minimizing fuel consumption.

This work emphasizes the importance of not only dealing with the machine allocation and operation scheduling problem but also considers the AGV scheduling. These are the results measured using the recommended approaches. To solve the problem, multi-objective algorithms, including MMOPSO and MOPSO, are used, and their parameters are tuned employing the Taguchi method. Thirty sample problems are randomly created and solved by these two multi-objective algorithms.

In order to analyze the Pareto frontier results, this paper used five criteria: MID, SNS, MS, SP, and CT. The TOPSIS model is then used to evaluate and select the best algorithm. After rigorous examination, it was discovered that the MMOPSO algorithm outperforms the MOSPO algorithm in solving the proposed multi-objective FMS problem.

Taking sustainability issues into account when solving the material flow problem will ominously contribute to the reduction of carbon emissions. As growing concerns have been paid to the global warming issues, this research is in line with the trend that energy consumption of the production process should be minimized. The other sustainability issues and applications in this area maybe encompass the development and implementation of closed-loop production fulfilled with AGVs to reuse raw materials, and correspondingly, the adverse effects of disposal materials could be minimized.

Some future research in this area is recommended as follows:

- i. Developing a model with uncertain parameters (stochastic or fuzzy parameters).
- ii. Considering another multi-objective mathematical model that regards the third pillar of sustainability, including social issues.
- iii. Applying response surface methodology for parameter tuning.
- iv. The planning of maintenance of AGVs is a challenging task that should be considered
- v. Form the heuristic viewpoint, the scenario in this paper can also be solved by reinforcement learning to enhance the quality of the obtained solutions.
- vi. Integrating the planning of buffers into FMS.

This work can be extended in terms of mathematical model and dataset. Real conditions for AGV, e.g., planning for maintenance, or consideration of mass production criteria could be potential future research directions. Also, it can be mentioned that there is no relevant study providing sustainability approach concerning fuel consumption modelling. In terms of using real data, lack of standard benchmarks to test the proposed mathematical model forces us to generate some random problems.

Acknowledgement

The research was supported by the Czech Science Foundation (GACR), Project number GA18- 15530S.

References

- Acciaro, M., H. Ghiara & M. I. Cusano (2014) Energy management in seaports: A new role for port authorities. *Energy Policy*, 71, 4-12.
- Akhavan, P., S. Barak, H. Maghsoudlou & J. Antuchevičienė (2015) FQSPM-SWOT for strategic alliance planning and partner selection; case study in a holding car manufacturer company. *Technological and Economic Development of Economy*, 21, 165-185.
- An, Y., X. Chen, J. Zhang & Y. Li (2020) A hybrid multi-objective evolutionary algorithm to integrate optimization of the production scheduling and imperfect cutting tool maintenance considering total energy consumption. *Journal of Cleaner Production*, 268, 121540.
- Bechtsis, D., N. Tsolakis, D. Vlachos & E. Iakovou (2017) Sustainable supply chain management in the digitalisation era: The impact of Automated Guided Vehicles. *Journal of Cleaner Production*, 142, 3970-3984.
- Bechtsis, D., N. Tsolakis, D. Vlachos & J. S. Srai (2018) Intelligent Autonomous Vehicles in digital supply chains: A framework for integrating innovations towards sustainable value networks. *Journal of Cleaner Production*, 181, 60-71.
- Bilge, Ü. & G. Ulusoy (1995) A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS. *Operations Research*, 43, 1058-1070.
- Blazewicz, J., H. Eiselt, G. Finke, G. Laporte & J. Weglarz (1991) Scheduling tasks and vehicles in a flexible manufacturing system. *International Journal of Flexible Manufacturing Systems*, 4, 5-16.
- Bruna, R. G. (2011) Efeitos dos laminados a quente e a morno na microestrutura, textura e propriedades de aços baixo carbono. *Rem: Revista Escola de Minas*, 64, 57-62.
- ElMaraghy, H. & A. Caggiano. 2016. Flexible Manufacturing System. In *CIRP Encyclopedia of Production Engineering*, eds. P. The International Academy for, L. Laperrrière & G. Reinhart, 1-7. Berlin, Heidelberg: Springer Berlin Heidelberg.

- Fuc, P., P. Kurczewski, A. Lewandowska, E. Nowak, J. Selech & A. Ziolkowski (2016) An environmental life cycle assessment of forklift operation: a well-to-wheel analysis. *The International Journal of Life Cycle Assessment*, 21, 1438-1451.
- Geerlings, H. & R. van Duin (2011) A new method for assessing CO₂-emissions from container terminals: a promising approach applied in Rotterdam. *Journal of Cleaner Production*, 19, 657-666.
- Heger, J. & T. Voß (2019) Dynamic priority based dispatching of AGVs in flexible job shops. *Procedia CIRP*, 79, 445-449.
- Hopf, H. & E. Müller (2015) Providing energy data and information for sustainable manufacturing systems by Energy Cards. *Robotics and Computer-Integrated Manufacturing*, 36, 76-83.
- Hwang, C. & K. Yoon (1981) Multiple Attribute Decision Making: Methods and Applications, A State of the Art Survey. 1981. *Springer-Verlag, New York, NY*.
- Jerald, J., P. Asokan, R. Saravanan & A. D. Rani (2006) Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 29, 584-589.
- Karimi, B., S. T. A. Niaki, H. Haleh & B. Naderi (2018) Bi-objective optimization of a job shop with two types of failures for the operating machines that use automated guided vehicles. *Reliability Engineering & System Safety*, 175, 92-104.
- Lee, B. K., J. M. W. Low & K. H. Kim (2015) Comparative evaluation of resource cycle strategies on operating and environmental impact in container terminals. *Transportation Research Part D: Transport and Environment*, 41, 118-135.
- Leriche, D., M. Oudani, A. Cabani, G. Hoblos, J. Mouzna, J. Boukachour & A. E. H. Alaoui (2015) Simulating new logistics system of Le Havre Port. *IFAC-PapersOnLine*, 48, 418-423.
- Liu, B., L. Wang & Y.-H. Jin (2008) An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, 35, 2791-2806.
- Liu, J. & B. L. MacCarthy (1997) A global MILP model for FMS scheduling. *European Journal of Operational Research*, 100, 441-453.
- Llopis-Albert, C., F. Rubio & F. Valero (2019) Fuzzy-set qualitative comparative analysis applied to the design of a network flow of automated guided vehicles for improving business productivity. *Journal of Business Research*, 101, 737-742.
- Maghsoudlou, H., B. Afshar-Nadjafi & S. T. Akhavan Niaki (2017) Multi-skilled project scheduling with level-dependent rework risk; three multi-objective mechanisms based on cuckoo search. *Applied Soft Computing*, 54, 46-61.
- Novas, J. M. & G. P. Henning (2014) Integrated scheduling of resource-constrained flexible manufacturing systems using constraint programming. *Expert Systems with Applications*, 41, 2286-2299.
- Panigrahi, B., V. R. Pandi, S. Das & S. Das (2010) Multiobjective fuzzy dominance based bacterial foraging algorithm to solve economic emission dispatch problem. *Energy*, 35, 4761-4770.
- Qiao, Y. 2012. Modified Multi-Objective Particle Swarm Optimization Algorithm for Multi-objective Optimization Problems. In *Advances in Swarm Intelligence*, eds. Y. Tan, Y. Shi & Z. Ji, 520-527. Springer Berlin Heidelberg.
- Sabuncuoglu, I. & S. Karabuk (1998) A beam search-based algorithm and evaluation of scheduling approaches for flexible manufacturing systems. *IIE Transactions*, 30, 179-191.
- Sadeghi, J. & S. T. A. Niaki (2015) Two parameter tuned multi-objective evolutionary algorithms for a bi-objective vendor managed inventory model with trapezoidal fuzzy demand. *Applied Soft Computing*, 30, 567-576.

- Schmidt, J., M. Eisel & L. M. Kolbe (2014) Assessing the potential of different charging strategies for electric vehicle fleets in closed transport systems. *Energy Policy*, 74, 179-189.
- Schmidt, J., C. Meyer-Barlag, M. Eisel, L. M. Kolbe & H.-J. Appelrath (2015) Using battery-electric AGVs in container terminals — Assessing the potential and optimizing the economic viability. *Research in Transportation Business & Management*, 17, 99-111.
- Sethi, A. K. & S. P. Sethi (1990) Flexibility in manufacturing: A survey. *International Journal of Flexible Manufacturing Systems*, 2, 289-328.
- Shukla, A. & H. Karki (2016) Application of robotics in offshore oil and gas industry— A review Part II. *Robotics and Autonomous Systems*, 75, 508-524.
- Taguchi, G., S. Chowdhury & Y. Wu. 2005. *Taguchi's quality engineering handbook*. Wiley.
- Ulusoy, G., F. Sivrikaya-Şerifoğlu & Ü. Bilge (1997) A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. *Computers & Operations Research*, 24, 335-351.
- Wang, W., L. Chen, J. Jie, Y. Zhao & J. Zhang. 2012. A novel multi-objective particle swarm optimization algorithm for flow shop scheduling problems. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, 24-31. Springer.
- Xin, J., R. R. Negenborn & G. Lodewijks (2014) Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control. *Transportation Research Part C: Emerging Technologies*, 44, 214-230.
- Yadav, A. & S. C. Jayswal (2018) Modelling of flexible manufacturing system: a review. *International Journal of Production Research*, 56, 2464-2487.
- Zeng, C., J. Tang & C. Yan (2014) Scheduling of no buffer job shop cells with blocking constraints and automated guided vehicles. *Applied Soft Computing*.
- Zhang, B., L. Xu & J. Zhang (2020) A multi-objective cellular genetic algorithm for energy-oriented balancing and sequencing problem of mixed-model assembly line. *Journal of Cleaner Production*, 244, 118845.
- Zhang, S., C. K. M. Lee, K. L. Choy, W. Ho & W. H. Ip (2014) Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. *Transportation Research Part D: Transport and Environment*, 31, 85-99.
- Zhou, B.-H. & C.-Y. Shen (2018) Multi-objective optimization of material delivery for mixed model assembly lines with energy consideration. *Journal of Cleaner Production*, 192, 293-305.