scientific reports

Check for updates

OPEN

Energy-efficient scheduling of AGV-assisted robotic flexible flowshops under learning and processing time uncertainty

Saeed Dehnavi✉, Hadi Mokhtari & Mohammad Taghi Rezvan

This study addresses an energy-efficient flexible flow shop scheduling problem (EEFFSP) that integrates automated guided vehicles (AGVs), sequence-dependent setup times, and the learning effect under fuzzy uncertainty in processing times and learning coefficients. The energy consumption is related to both machines and AGVs in such a way that higher speeds result in higher energy consumption. A mixed-integer programming model is developed to simultaneously minimize makespan and total energy consumption, considering the operation of both machines and AGVs. To handle uncertainty, a fuzzy programming approach based on Jiménez's ranking method is employed. The problem's multi-objective nature is tackled using approaches: the classical AUGMECON method, NSGA, and the NSGA-II algorithms. The proposed hybrid optimization framework effectively captures realistic production features while ensuring computational efficiency. Extensive experiments on benchmark instances demonstrate that the fuzzy-based NSGA-II provides high-quality Pareto solutions with a balanced trade-off between energy efficiency and production performance. Compared with other methods, NSGA-II achieves superior solution diversity and robustness while maintaining acceptable computational time. In this study, an integration of fuzziness, learning effects, and AGV scheduling within an EEFFSP context is provided, a combination not previously addressed in the literature, which provides managerial insights for sustainable manufacturing system design.

Keywords Energy-efficient scheduling, Flexible flowshop, Automated guided vehicle (AGV), Learning effect, NSGAII, NSGA

Energy consumption is rapidly increasing due to population growth and the global economy. In today's world, about 50% of the total energy consumption is related to the industrial sector[1]. In this regard, due to government legislation, customers' environmental concerns, and the continuous increase in energy costs, reducing energy consumption appears crucial for every industrial manufacturer. The issue of managing energy consumption has attracted the attention of government officials, industrial managers, and many researchers worldwide. To manage energy consumption, manufacturing companies can adopt strategies such as improving machinery technology, product redesign, or process optimization; however, such strategies often require high capital investment. On the other hand, strategies like optimizing production scheduling can effectively reduce energy consumption without significant additional costs[2].

Energy efficiency has become a critical concern in modern manufacturing, not only to reduce costs but also to minimize environmental impacts and achieve sustainable production. Industrial production consumes energy not only during active processing but also in idle, setup, and standby states. Material handling and internal transportation, particularly via automated guided vehicles (AGVs), further contribute to overall energy use. Inefficient coordination of production and transportation can therefore result in excessive energy consumption, higher operational costs, and increased environmental footprint. Reducing energy usage while maintaining high production throughput and product quality is a central challenge for operational managers, process engineers, and researchers. This challenge necessitates strategies that integrate production planning, machine scheduling, and transportation management to achieve energy savings without compromising operational efficiency.

Department of Industrial Engineering, Faculty of Engineering, University of Kashan, Kashan, Iran. ✉email: dehnavi@kashanu.ac.ir

Flexible flowshop systems (FFS) are widely applied across industries such as electronics, textiles, glass, and chemical manufacturing, due to their ability to process multiple jobs sequentially while allowing flexibility in machine allocation. Despite their widespread use, coordinating job sequencing, machine operations, and material transportation efficiently remains complex, especially when multiple objectives, such as minimizing makespan and energy consumption, must be optimized simultaneously. The complexity is further increased by sequence-dependent setup times, learning effects that alter processing times depending on job sequences, and uncertainties in machine performance and processing durations. Efficiently addressing these factors is essential for operational efficiency and sustainable energy use, highlighting the need for sophisticated mathematical modeling and advanced multi-objective optimization techniques. Developing such approaches not only reduces energy costs and environmental impacts but also enables manufacturers to achieve robust and adaptable production schedules in increasingly dynamic industrial settings.

Literature review

Energy-efficient scheduling as a key decision-making process in manufacturing systems, in addition to minimizing production costs, minimizes the total energy consumption of the system. In the literature, there are different types of production energy-efficient scheduling that have been presented in various environments such as energy-efficient flowshop, job shop, open shop, flexible flowshop and flexible job shop problems. EEFFSP which is also known as EEHFS (Energy-efficient hybrid flowshop scheduling) can be one of the most important issues, especially for industrial environments such as electronics, paper, textile, petrochemical, airplane engine and semiconductor[3]. In recent years, many researchers have focused on EEFFSP due to the growing concerns over energy efficiency and environmental sustainability. Various studies have explored ways to balance production performance with energy savings through bi-objective or multi-objective optimization approaches. For instance, Li et al.[4] and Jiang et al.[5] formulated mathematical models that jointly minimized makespan and energy consumption, considering different machine operating states such as processing, setup, and idle conditions. To solve these complex problems, both studies employed advanced evolutionary algorithms; Li et al.[4] proposed an energy-aware multi-objective optimization algorithm (EA-MOA), while Jiang et al.[5] utilized a multi-objective evolutionary algorithm based on decomposition (MOEA/D). These efforts collectively highlight the growing research attention toward integrating energy consumption modeling with efficient scheduling strategies in flexible flow shop systems. Dai et al.[6] analyzed various operational modes of machines—such as running, production, idle, and start-up/shutdown—and emphasized the need to jointly minimize production time and energy use. Tang et al.[7] extended this line of research by considering dynamic production environments with job arrivals and machine breakdowns, highlighting how energy consumption patterns vary across start-up, readiness, and machining states. Building on these ideas, Zeng et al.[8] incorporated group technology principles into a two-stage EEFFSP framework, showing that accounting for both processing and setup energy can further enhance overall system efficiency. Together, these studies demonstrate the evolving complexity of EEFFSP modeling, moving from static machine-level energy considerations toward more dynamic and system-level optimization approaches. Liu et al.[2] converted the model with two objective functions into single ones by the sum weighted method and solved it by an evolutionary algorithm (EV). The energy consumption occurred in three modes of operating, stand-by and heating furnace for machines. Meng et al.[9] developed a single-objective model to reduce both total energy consumption and makespan, emphasizing the importance of incorporating processing, idle, and auxiliary energy into scheduling decisions. Similarly, Wang et al.[10] addressed a two-stage energy-efficient flexible flowshop in glass production, demonstrating how integrating energy considerations into scheduling can enhance overall system performance. These studies further highlight the value of designing scheduling models that balance production efficiency and energy consumption across different industrial settings. Han et al.[11] developed a single-objective EEFFSP model aimed at minimizing makespan, incorporating both limited and public buffers, and accounting for energy consumption through constraint-based functions. Building on this, Han et al.[12] and Wang et al.[13] extended EEFFSP models to consider blocking effects and sequence-dependent setup times, integrating energy consumption in different operational modes. These studies demonstrate the increasing attention toward modeling both production performance and energy efficiency in flexible flowshop systems under realistic operational constraints. Other objective functions besides the ones mentioned above can be seen in the different papers. For example, Luo et al.[14] studied a dynamic EEFFSP where new jobs could enter the system after the original schedule had started, integrating makespan and total tardiness into a single objective while including energy constraints. Similarly, Bruzzone et al.[15] developed a model combining makespan and total tardiness with energy limits, and Jiang and Zhang[16] extended EEFFSP to minimize weighted tardiness and non-processing energy under finite intermediate buffers. Gong et al.[17] incorporated worker flexibility into a bi-objective EEFFSP, aiming to reduce both makespan and overall energy, noise, and recycling impact. Hasani and Hosseini[3] further advanced the field by considering multiple multifunctional machines, sequence-dependent setup times, and dual objectives of energy consumption and production cost. Collectively, these studies illustrate the growing complexity of EEFFSP models, highlighting the integration of operational flexibility, energy management, and realistic industrial constraints.

In some papers, more than two objective functions are also observed. Li et al.[18] developed a three-objective EEFFSP model targeting minimization of makespan, total tardiness, and energy consumption, while Zeng et al.[19] extended the EEFFSP framework by also considering material wastage along with makespan and total electricity use. Both studies integrated multiple machine operating states, such as processing, setup, standby, and auxiliary modes, to more accurately capture energy consumption, and employed multi-objective optimization to identify Pareto-efficient solutions.

Some researchers have also included the issue of carbon emissions in their developed models. Liu et al.[20] proposed an EEFFSP model with two objectives: minimizing makespan and total carbon footprint during processing and idle states. Similarly, Yin et al.[21] incorporated carbon emissions across multiple machine states,

including processing, tool switching, idle, and transportation. Wu et al.[22] considered the impact of renewable and non-renewable energy on processing times, modeling EEFFSP with variable energy availability. These studies collectively highlight the increasing attention toward integrating environmental considerations, particularly carbon footprint and renewable energy, into energy-efficient flexible flowshop scheduling.

Yan et al.[23] proposed a two-level EEFFSP model, where the first level focused on optimizing cutting parameters at the machine-tool level, and the second level addressed scheduling at the shop-floor level. Energy consumption was considered across multiple machine states, including processing, setup, standby, and overhead, highlighting the integration of machine-level and system-level energy considerations in flexible flowshop scheduling.

Some researchers have focused on the problem assuming the uncertainty of the parameters. For instance, Zhou et al.[24] addressed EEFFSP under uncertainty in processing times using interval numbers, formulating a bi-objective model to minimize interval makespan and energy consumption. They solved the problem using an imperialist competitive algorithm with empire grouping (ICAEG), demonstrating its efficiency over alternative methods. In a related study, Zhou and Liu[25] considered a bi-objective EEFFSP with fuzzy processing times and due dates, incorporating sequence-dependent setup times. An energy-efficient bi-objective differential evolution algorithm (BODEA) was employed to obtain the Pareto-optimal solutions.

Zabihzadeh and Rezaeian[26] studied a flexible flowshop scheduling (FFSS) problem with different release times, robotic transportation, and blocking, aiming to minimize makespan,both ACO and GA were employed as solution methods. Hidri et al.[27] investigated a two-center FFSS with job-dependent transportation times and solved it using a heuristic and branch-and-bound (B&B) approach. Zhang et al.[28] considered FFSS with random and state-dependent batch transport, formulating an open queuing network model solved via decomposition (DM). Lei et al.[29] addressed FFSS with dynamic transport waiting times and limited buffers, utilizing AGVs between stages and solving the problem with a memetic algorithm (MA). Liou and Hsieh[30] studied FFSS with sequence-dependent setup and transportation times, solved using a hybrid PSO-GA. Li et al.[31] proposed a hybrid GA and tabu search (TS) for FFSS with transportation considerations. Gheisariha et al.[32] incorporated inspection procedures, sequence-dependent setup times, and transportation times in FFSS, solving the problem with a heuristic and an enhanced multi-objective harmony search (MOHS). Hidri and Elsherbeeny[33] focused on a two-stage FFSS with removal and transportation times, solved through a two-phase heuristic combined with B&B. Biskup[34] considered the concept of position-based worker learning effect on processing time in scheduling problems for the first time.

Many researchers have investigated this concept in different production systems. Regarding FFSS with learning effects, Seidgar et al.[35] proposed a mathematical model incorporating learning and forgetting effects, solved using Lingo software. Kazemi Esfeh et al.[36] studied FFSS with transporters and intermediate limited buffers under budget constraints and learning effects, solving small instances with GAMS and medium-to-large instances with MOSA and MOEA/D. Pargar and Zandieh[37] also addressed FFSS with learning effects, employing a water flow-like (WFL) algorithm as the solution approach. While several studies have applied multi-objective optimization techniques such as MOEA/D[5,16], NSGAII[3], and hybrid algorithms (e.g., ACO-GA by Zabihzadeh and Rezaeian, 2015; PSO-GA by[30],GA-TS by[31], ACO by[38]), these approaches generally do not address uncertainties in both processing/setup times nor incorporate AGV-based job transportation. Our study distinguishes itself by integrating fuzzy modeling, worker learning effects, and automated guided vehicle (AGV) routing within a bi-objective EEFFSP framework, offering a more holistic and realistic scheduling approach for modern production systems.

Recent research trends also highlight increasing attention to energy-efficient production and intelligent scheduling approaches. Maghami et al.[39] optimized electric vehicle charging infrastructure to improve grid performance, emphasizing the relevance of energy-aware decision-making in modern transportation and power systems. Similarly, Geng et al.[40] introduced a reinforcement-learning-based memetic algorithm for energy-efficient distributed flexible job-shop scheduling, demonstrating the potential of hybrid intelligent optimization techniques in addressing complex multi-objective scheduling challenges under energy considerations. Hu et al.[41] applied a constructive-destructive neighborhood search within an artificial bee colony algorithm for variable-speed green hybrid flowshop scheduling, illustrating the importance of speed adjustment mechanisms in energy-efficient production. In addition, Wang et al.[42] formulated a mixed-integer model for flexible flowshop scheduling with AGV integration, focusing primarily on synchronization between machines and material-handling systems, though without considering fuzzy parameters or learning effects. Tang et al.[43] developed a deep-reinforcement-learning-based scheduler for multi-objective AGV-enabled hybrid flowshops, further confirming the emerging role of learning-based strategies in complex industrial scheduling environments. Moreover, Yang et al.[44] proposed a multi-objective MILP model for flexible job-shop scheduling with assembly and AGV transportation, highlighting the importance of AGV coordination in multi-stage systems,however, their model does not incorporate fuzzy uncertainty or learning behavior, which distinguishes the contributions of the present study.

As seen in Table 1, in this study, we develop a bi-objective mathematical model for EEFFSP so that the first objective function is to minimize makespan and the second one is total energy consumption. In here, the learning effect on setup times by workers and existence of one AGV between any two consecutive stages that are responsible for transferring the jobs is considered. Moreover, the sequence-dependent setup times are taken to account. The processing time because of lack of knowledge and decision maker's experience and judgment in measurements as well as the learning effect because it is not equal in different iterations, are considered as fuzzy number.

Despite the extensive research on energy-efficient scheduling in flexible flowshop systems, several key practical aspects remain largely unexplored. Existing studies typically assume deterministic processing conditions, overlook the learning effect in setup operations, and ignore the role of AGV-based material handling, which is a critical energy-consuming component in modern automated manufacturing systems. Moreover, only a limited

No	References	Energy consumption	Learning Effect	Setup time	Transportation time	Parameter		No. of objective functions			Solution approach
						Deterministic	Non deterministic	Multi-objective	Bi-objective	Single-objective	
1	Bruzzone et al.[15]	✓	✗	✗	✗	✓	✗	✗	✗	✓	RNS
2	Pargar and Zandieh[37]	✗	✓	✓	✗	✓	✗	✗	✗	✓	WFL
3	Dai et al.[6]	✓	✗	✗	✗	✓	✗	✗	✓	✗	IGSA
4	Seidgar et al.[35]	✗	✓	✓	✗	✓	✗	✗	✗	✓	Lingo
5	Zabihzadeh and Rezaeian (2015)	✗	✗	✗	✓	✓	✗	✗	✗	✓	ACO + GA
6	Liou and Hsieh[30]	✗	✗	✓	✓	✓	✗	✗	✗	✓	Hybrid PSO + GA
7	Tang et al.[7]	✓	✗	✗	✗	✓	✗	✗	✓	✗	IPSO
8	Li et al.[4]	✓	✗	✗	✗	✓	✗	✗	✓	✗	EA-MOA
9	Hidri et al.[27]	✗	✗	✗	✓	✓	✗	✗	✗	✓	Heuristic + B&B
10	Wu et al.[22]	✓	✗	✗	✗	✓	✗	✗	✓	✗	HNSGAII
11	Luo et al.[14]	✓	✗	✗	✗	✓	✗	✗	✓	✗	PBHPGA
12	Li et al.[18]	✓	✗	✗	✗	✓	✗	✓	✗	✗	TICA
13	Wang et al. (2019)	✓	✗	✗	✗	✓	✗	✗	✓	✗	BOTS + BOACO
14	Zhou and Liu[25]	✓	✗	✓	✗	✗	✓	✗	✓	✗	BODEA
15	Jiang and Zhang[16]	✓	✗	✗	✗	✓	✗	✗	✓	✗	MOEA/D
16	Zhou et al.[24]	✓	✗	✗	✗	✗	✓	✗	✓	✗	ICAEG
17	Gong et al.[17]	✓	✗	✗	✗	✓	✗	✗	✓	✗	HEA
18	Lei et al.[29]	✗	✗	✗	✓	✓	✗	✗	✗	✓	MA
19	Gong et al.[17]	✓	✗	✗	✗	✓	✗	✗	✓	✗	HEA
20	Han et al.[12]	✓	✗	✓	✗	✓	✗	✗	✓	✗	DEMO
21	Hasani and Hosseini[3]	✓	✗	✓	✗	✓	✗	✗	✓	✗	NSGAII
22	Zhang et al.[28]	✗	✗	✗	✓	✓	✗	✗	✗	✗	DM
23	Jiang et al.[5]	✓	✗	✗	✗	✓	✗	✗	✓	✗	MOEA/D
24	Yin et al.[21]	✓	✗	✗	✗	✓	✗	✗	✗	✓	IGA
25	Gheisariha et al.[32]	✗	✗	✓	✓	✓	✗	✗	✓	✗	Heuristic + MOHS
26	Hidri and Elsherbeeny[33]	✗	✗	✗	✓	✓	✗	✗	✗	✓	Heuristic + B&B
27	Li et al.[31]	✗	✗	✗	✓	✓	✗	✗	✗	✓	Hybrid GA + TS
28	Kazemi Esfeh et al.[36]	✗	✗	✗	✓	✓	✗	✗	✗	✓	GAMS + MOSA + MOEA/D
29	Geng et al.[40]	✓	✗	✗	✗	✓	✗	✓	✗	✗	RL + Memetic Algorithm
30	Wang et al.[42]	✓	✗	✓	✓	✓	✗	✗	✓	✗	MILP
31	Tang et al.[43]	✗	✗	✓	✓	✓	✗	✓	✗	✗	DRL
32	Yang et al.[44]	✗	✗	✓	✓	✓	✗	✗	✓	✗	MILP
33	Maghami et al.[39]	✓	✗	✗	✗	✓	✗	✓	✗	✗	PSO
34	Hu et al.[41]	✓	✗	✗	✗	✓	✗	✓	✗	✗	ABC + Constructive-Destructive Neighbor Search
	Our paper	✓	✓	✓	✓	✗	✓	✗	✓	✗	AUGMECON + NSGAII + NSGA

Table 1. Summary of the literature.

number of works consider parameter uncertainty, and none simultaneously incorporate fuzzy processing times, fuzzy learning coefficients, and AGV-assisted job transfer within a multi-objective EEFFSP framework. These gaps highlight the need for a more realistic and comprehensive model that jointly addresses uncertainty, learning behavior, and transportation decisions to minimize both makespan and energy consumption. Motivated by these limitations, this study develops a fuzzy bi-objective EEFFSP model with sequence-dependent setup times and single-AGV transportation between stages, and employs both AUGMECON and NSGA-II to efficiently obtain high-quality Pareto solutions. Thus, our contribution can be as follows:

- Integrated Modeling: A unified fuzzy bi-objective model for EEFFSP that simultaneously considers fuzzy processing times, learning effects, and AGV-based transportation, a combination not previously addressed in the literature.
- Energy Formulation: Inclusion of energy components (processing, setup, idle, loaded-AGV, and no-loaded-AGV) for a more realistic representation of total energy consumption.
- Hybrid Solution Framework: Development of a two-stage optimization approach combining AUGMECON (for small instances) and an improved NSGA-II (for large-scale problems) to achieve superior Pareto diversity and convergence.
- Validated Performance: Experimental results show that the proposed NSGA-II outperforms classical NSGA and AUGMECON in solution quality and computational efficiency, offering practical insights for energy-efficient and sustainable production scheduling.

The remainder of this paper is organized as follow. "Literature review" section, includes the definition of the problem and then the considered problem is formulated. "Problem statement and formulation" section, describes proposed NSGAII for solving this problem and in "Fuzzy programming" section, several test problems are generated to determine model and algorithm's efficiency. The Conclusion and future research for this study is provided in "Solution methodology" section.

Problem statement and formulation

Problem description

In this study, a flexible flowshop system with J jobs that should be processed in S stages is considered. Each stage includes M_S unrelated parallel machines (for at least one stage M_S >1), and there are input and output buffers corresponding to each machine with infinite capacity. Each job has the same production flow as the others: stage 1, stage 2, …, stage S and can be processed by any machines at the stage. Between two consecutive stages, there is only one AGV that is responsible for transferring the jobs from stage s to $s+1$ ($s < S$). Jobs enter the production system in the same order. Each job is processed on one machine at each stage and then transported to the next stage by an AGV. The goal is to find a job sequence and parallel machine allocation so that both makespan and total energy consumption are minimized. In this system, both machines and AGVs can be set to speeds from a finite discrete set of $V = (v1, v2, \ldots, vd)$ which consumes more energy for higher speeds. A schematic view of the environment considered in our model is given in Fig. 1.

In real conditions, not all parameters such as processing time due to lack of knowledge and the decision maker's experience and judgment or learning effect due to not being the same in every iteration may not be accurately measured. For this reason, the processing time and learning effect are considered fuzzy and designed in the form of triangular fuzzy numbers in this paper. A triangular fuzzy number $\widetilde{A}$ is presented such as $\widetilde{A} = \left(A^L, A^C, A^R\right)$ where A^L, A^C and A^R are optimistic, most likely and pessimistic value, respectively. Membership function can be expressed as follows. Besides, a triangular distribution is depicted in Fig. 2.

$$\mu_{\widetilde{A}}^{(x)} = \begin{cases} \frac{x-A^L}{A^C-A^L} if A^L \le x \le A^C \\ \frac{A^R-x}{A^R-A^C} if A^C \le x \le A^R \\ 0 otherwise \end{cases}$$

In this paper, the learning effect is intended to reduce set-up time if two consecutive jobs are placed on the same machine at subsequent stations. For example, if job.2 is placed right after job.1 on machine.1 at the station.1, and again job.2 is placed on machine 1 at station.2 right after job.1, the required setup time job.2 after job.1 on machine.1 in stage.2 will be reduced from its standard value because the skill of the worker is increased in repetition.

Problem assumptions

In addition to the above, there are also the following assumptions:

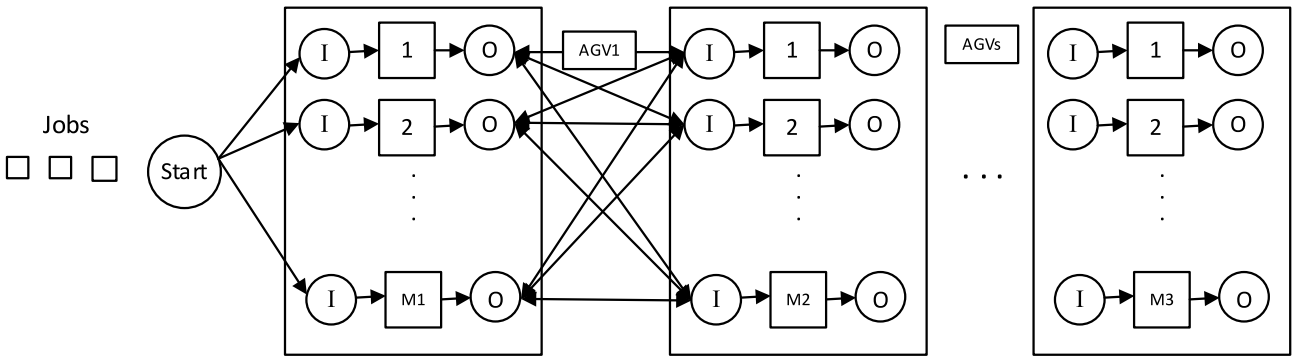

Fig. 1. A schematic view of the environment considered.

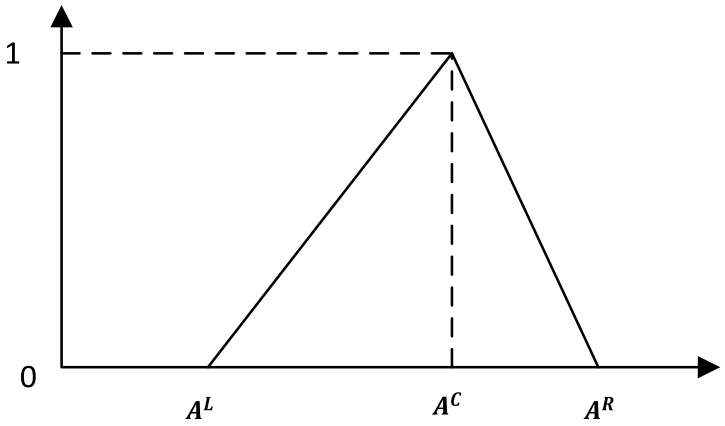

Fig. 2. A membership function of triangular Fuzzy number.

(1) All jobs, AGVs and machines are available at the beginning of the planning horizon.
(2) The machines and AGVs do not break down.
(3) By choosing different parallel machines in each stage, the processing time of job may be different.
(4) There is only one AGV between two adjacent operations. Each AGV only transfers the jobs between adjacent operations.
(5) The capacity of each AGV is only one job per transfer.
(6) Each job in each stage has to be processed by one specific parallel machine.
(7) Once AGV transfers and delivers job, it is released and back to the machine corresponding to next transportation in advance.
(8) Each machine can process at most one job at a time without interruption and each job can be processed on only one machine at a time.
(9) Each job can be transported by only one AGV at a time and each AGV can transport only one job at a time.
(10) The speed of each machine/AGV is chosen from a discrete set of numbers.
(11) The speed of each machine/AGV cannot be changed once the machine/AGV begins to process/transfer a job.
(12) Setup times are sequence-dependent and known.
(13) Processing times and learning effect coefficients are in the form of triangular fuzzy numbers.
(14) Higher machine/AGV speeds consume more energy although they contribute to faster processing of jobs and shorter transfer times.
(15) Only one worker is responsible for setting up the machines and it is assumed that this worker is always available and learning occurs for him/her.
(16) The AGV speed is selected only after the job is delivered and is unchangeable until the next job is delivered. In other words, the speed of each AGV is determined after receiving the job, and this AGV maintains this specified speed until the next job is delivered by same AGV.

Notations

The following abbreviations table are presented to describe the above-mentioned problem:

Indices:		
j, i, h	The index of jobs ($0 \leq j \leq J$)	
s, $s\prime$	The index of stages ($1 \leq s \leq S$)	
$m, m\prime, m''$	The index of machines	
p, $p\prime$	The index of position of job in the job sequence	
	The index of machine speeds	
u	The index of AGV speeds	
Parameters:		
J	The number of jobs	
S	The number of stages	
M_s	The number of machines of stage s	
L	The number of machine speeds	
U	The number of AGV speeds	
$\widetilde{Pr}_{jmsl}$	The fuzzy processing time of job j on machine m of stage s when machine selects th speed	
$D_{msm\prime(s+1)}$	The distance between machine m of stage s and machine $m\prime$ of stage $s+1$	
Se_{jims}	The basic sequence-dependent setup time of machine m of stage s for switching from job j to job	

Indices:		
PEC_{msl}	The energy consumption per unit by machine m of stage s when it selects th speed in processing mode	
SEC_{ms}	The energy consumption per unit by machine m of stage s in setup mode	
IEC_{ms}	The energy consumption per unit by machine m of stage s when it selects th speed in idle mode	
LEC_{jsu}	The energy consumption per unit by AGV loaded when its u th speed is selected for transferring of job j from stage s to next stage	
UEC_{su}	The energy consumption per unit by AGV no-loaded when its u th speed is selected in stage s	
$\widetilde{a}_{jim}$	The fuzzy learning coefficient of switching from job j to job on machine m	
BM	A big positive number	
Dependent variables:		
v_{js}	The speed of AGV when it transfers job j at stage s	
$t_{jmsm\prime(s+1)}$	The transportation time between machine m of stage s and machine $m\prime$ of stage	$s+1$ for job j
Ns_{jims}	The number of setup on machine m for switching from job j to job that are done before stage s	
RSe_{jims}	The revised sequence-dependent setup time of machine m of stage s for switching from job j to job after learning effects	
St_{jms}	The start time of job j on machine m of stage s	
C_{jms}	The completion time of job j on machine m of stage s	
Hr_{jms}	The start time of transportation of job j to the next stage after its process on machine m of stage s is finished	
Ar_{jms}	The arrival time of transportation of job j to the next stage after its process on machine m of stage s is finished	
C_{max}	The makespan (maximum completion time of jobs)	
Decision variables:		
x_{jims}	$\begin{cases} 1; if job i is processed directly after job j on machine m at stages \\ 0; otherwise \end{cases}$	
y_{jps}	$\begin{cases} 1; if job j is at the pth position of j job sequence at stages \\ 0; otherwise \end{cases}$	
z_{jmsl}	$\begin{cases} 1; if job j on machine m selects l th speed at stages \\ 0; otherwise \end{cases}$	
w_{jus}	$\begin{cases} 1; if AGV selects u th speed for transferring of job j at stages \\ 0; otherwise \end{cases}$	

Objective functions

The first objective function is related to makespan minimization as Eq. (1):

$$\min f_1 = C_{max} \tag{1}$$

The second objective is to minimize total energy consumption. In this paper, five types of energy consumption including the following are considered 1) Energy consumption for processing mode of machines (ECPM), 2) Energy consumption for setup mode of machines (ECSM), 3) Energy consumption for idle mode of machines (ECIM), 4) Energy consumption for loaded mode of AGVs (ECLM) and 5) 1)Energy consumption for no-loaded mode of AGVs (ECNM).

(ECPM): part of the energy is consumed in the processing of jobs by machines. This part depends on the selected speed of the processing by machines. Therefore, this part can be obtained as Eq. (2).

$$EC^{(1)} = \sum_{j=1}^{J} \sum_{m=1}^{M_s} \sum_{l=1}^{L} \sum_{s=1}^{S} \widetilde{Pr}_{jmsl}.PEC_{msl}.z_{jmsl} \tag{2}$$

(ECSM): this energy of setup is consumed in preparing of activities such as clamping, loading, unloading, changing tools, switching and positioning. It can be calculated by Eq. (3):

$$EC^{(2)} = \sum_{j=1}^{J} \sum_{\substack{i=1 \\ i \neq j}}^{J} \sum_{m=1}^{M_s} \sum_{s=1}^{S} RSe_{jims}.SEC_{ms}.x_{jims} \tag{3}$$

(ECIM): machines are in three states: in process, setting up and idle state. As a result, the energy consumption in idle mode is equal to the energy that is spent during the times except in process and setup modes. The Eq. (4) calculates the energy consumption for idle mode as follows:

$$EC^{(3)} = \sum_{m=1}^{M_s}\sum_{s=1}^{S}\left(\left[\begin{array}{l}\max\limits_{j}\left(C_{jms}.x_{jims}\right) - \sum\limits_{h=1}^{J}\sum\limits_{l=1}^{L}\widetilde{Pr}_{hmsl}.z_{hmsl} \\ -\sum\limits_{h=1}^{J}\sum\limits_{i=1}^{J}RSe_{hims}.x_{hims}\end{array}\right].IEC_{ms}\right) \tag{4}$$

(ECLM) and (ECNM): In performing their operations, AGVs are either loaded or no-loaded. From the moment of receiving the job until the moment of delivering it to the next input buffer or machine, the AGV is loaded, and from the moment of delivery to the buffer until returning to the previous output buffer or machine, the AGV is no-loaded. Equation (5) calculates the energy consumption for loaded AGV in and Eq. (6) calculates the energy consumption for no-loaded AGV.

$$EC^{(4)} = \sum_{j=1}^{J}\sum_{m\prime=1}^{M_{(s+1)}}\sum_{m=1}^{M_s}\sum_{i=1}^{J}\sum_{h=1}^{J}\sum_{s=1}^{S-1}\sum_{u=1}^{U}\left(t_{jmsm\prime(s+1)}.x_{jims}.x_{jhm\prime(s+1)}.LEC_{jsu}.w_{jus}\right) \tag{5}$$

$$EC^{(5)} = \sum_{j=1}^{J}\sum_{i=0}^{J}\sum_{p=1}^{J}\sum_{m\prime=1}^{M_{s+1}}\sum_{m=1}^{M_s}\sum_{s=1}^{S}\sum_{u=1}^{U}\left(\left[Hr_{jms} - Ar_{im\prime s}\right].\sum_{i=0}^{J}x_{jims}.\sum_{h=0}^{J}x_{ihm\prime s}.y_{jp}.y_{i(p+1)}.UEC_{su}.w_{jus}\right) \tag{6}$$

In summary, the second objective function is as Eq. (7):

$$\min f_2 = EC^{(1)} + EC^{(2)} + EC^{(3)} + EC^{(4)} + EC^{(5)} \tag{7}$$

Constraints

$$\sum_{\substack{j=0\\ j\neq i}}^{J}\sum_{m}^{M_s}x_{jims} = 1\forall i = 1,\ldots,J; s = 1,2,\ldots,S \tag{8}$$

$$\sum_{\substack{i=0\\ i\neq j}}^{J}x_{jims} \leq 1\forall j = 0,1,\ldots,J; m = 1,2,\ldots,M_s; s = 1,2,\ldots,S \tag{9}$$

$$\sum_{\substack{j=0\\ j\neq h}}^{J}x_{jhms} - \sum_{\substack{i=0\\ h\neq i}}^{J}x_{hims} = 0\forall h = 1,2,\ldots,J; m = 1,2,\ldots,M_s; s = 1,2,\ldots,S \tag{10}$$

$$\sum_{p=1}^{J}y_{jps} = 1\forall j = 1,2,\ldots,J; s = 1,2,\ldots,S \tag{11}$$

$$\sum_{j=1}^{J}y_{jps} = 1\forall p = 1,2,\ldots,J; s = 1,2,\ldots,S \tag{12}$$

$$\sum_{l=1}^{L}z_{jmsl} = \sum_{i=0}^{J}x_{jims}\forall j = 1,\ldots,J; m = 1,2,\ldots,M_s; s = 1,2,\ldots,S \tag{13}$$

$$\sum_{u=1}^{U}w_{jus} = 1\forall j = 1,\ldots,J; s = 1,2,\ldots,S \tag{14}$$

$$\sum_{u=1}^{U}v_u.w_{jus} = v_{js}\forall j = 1,\ldots,J; s = 1,2,\ldots,S \tag{15}$$

$$t_{jmsm'(s+1)} = \frac{D_{msm'(s+1)}}{v_{js}}\forall j = 1,\ldots,J; s = 1,2,\ldots,S \tag{16}$$

$$\begin{array}{l}RSe_{jims} = Se_{jims}.\left(Ns_{jims}+1\right)^{\tilde{a}_{jim}} \\ \forall i,j = 1,\ldots,J;\ m = 1,2,\ldots,M_s;\ s = 2,\ldots,S\end{array} \tag{17}$$

$$\begin{aligned} &\mathrm{Re}_{jims} = \left(\sum_{s'=1}^{s-1} x_{jims'}\right).x_{jims} \\ &\forall i, j = 1, \ldots, J;\ m = 1, 2, \ldots, M_s;\ s = 2, \ldots, S \end{aligned} \tag{18}$$

Constraint (8) guarantees that each job is processed once on only one machine as successor of another job at each stage. Constraint (9) states that each job has maximum one successor job at each stage on each machine. Constraint (10) ensures that if a job as successor of other job is processed on each machine at each stage, that job also will be as a predecessor for other job on same machine at same stage. Constraints (11) and (12) ensures that each job is assigned to only one position of job sequence and each position will accommodate only one job, respectively. Constraint (13) states that if each machine at each stage is selected for processing of job, that machine selects only one speed. Constraint (14) guarantees that each AGV in order to transfer of each job at each stage selects only one speed. Constraint (15) calculates the selected speed of AGV for transferring of each job at each stage. Constraint (16) computes the transportation time equals travel distance between two machines divides speed of AGV. Constraints (17) and (18) compute the revised sequence-dependent setup time of machine with respect to learning effect. As shown in these two constraints, the revised sequence-dependent setup time of machine *m* at stage *s* for switching from job *j* to job *i* is adjusted by considering the learning phenomenon. This means that as operators or machines repeatedly perform similar setup operations, their efficiency increases and the required setup time gradually decreases. Consequently, the model assumes that future setup times become shorter due to the accumulated learning experience. This consideration enables the proposed model to more accurately represent real production environments, where the setup process is continuously improved through. The constraints (19–27) are added to the above equations:

$$\begin{aligned} &C_{jms} = St_{jms} + \sum_{l=1}^{L} \widetilde{Pr}_{jmsl}.z_{jmsl} \\ &\forall j = 1, \ldots, J;\ m = 1, 2, \ldots, M_s;\ s = 1, 2, \ldots, S \end{aligned} \tag{19}$$

$$\begin{aligned} &St_{jms} \geq \sum_{m'=1}^{M_{(s-1)}} Ar_{jm'(s-1)}.\sum_{i=0}^{J} x_{jims} \\ &\forall j = 1, \ldots, J;\ m = 1, 2, \ldots, M_s;\ s = 2, 3, \ldots, S \end{aligned} \tag{20}$$

$$\begin{aligned} &St_{jms} \geq \left[\left(C_{ims}.\sum_{h=0}^{J} x_{ihms}\right) + \sum_{l=1}^{J} RSe_{ljms}.x_{ljms}\right] \\ &\forall i, j = 1, \ldots, J; i \neq j; m = 1, 2, \ldots, M_s; s = 1, 2, \ldots, S \end{aligned} \tag{21}$$

$$\begin{aligned} &Ar_{jms} = Hr_{jms}\left(\sum_{i=0}^{J} x_{jims}\right) + \sum_{m}^{M_s}\sum_{m'}^{M_{(s+1)}} t_{jmsm'(s+1)}.\left(\sum_{i=0}^{J} x_{jims}\right).\left(\sum_{i=0}^{J} x_{jim'(s+1)}\right) \\ &\forall j = 1, \ldots, J; m = 1, 2, \ldots, M_s; s = 1, 2, \ldots, S-1 \end{aligned} \tag{22}$$

$$H_{jms} \geq C_{jms} \forall j = 1, \ldots, J; m = 1,2, \ldots, M_s; s = 1,2, \ldots, S \tag{23}$$

$$\begin{aligned} H_{jms} \geq &\sum_{m'=1}^{M_s} Ar_{im's}.\left(\sum_{h=0}^{J} x_{jhms}\right) + \sum_{m''}^{M_{(s+1)}} \frac{D_{msm''(s+1)}}{v_{is}}.\left(\sum_{h=0}^{J} x_{jhms}\right).\left(\sum_{h=0}^{J} x_{ihm''(s+1)}\right) \\ &- BM\left(2 - y_{jps} - y_{i(p-1)s}\right) \\ \forall i, j = 1, \ldots, J; i \neq j; s = 1, 2, \ldots, S-1; p = 1, 2, \ldots, P & \end{aligned} \tag{24}$$

Constraint (19) calculates the completion time of each job on each machine at each stage. The completion time is equal to sum of start time and process time. Constraints (20) and (21) determine the start time of each job on each machine at each stage. For stages 2,3, ..., *S*, the start of the job processing can occur when both the AGV of the previous stage has delivered the job to the current stage and the higher priority jobs of that job on the machine have been completed. Constraint (22) calculates the arrival time of transportation of each job to the next stage after its process is finished. This value is equal to the AGV start time of transportation plus transportation time. Constraints (23) and (24) state that the start time of transportation for AGV is maximum between the completion time of job on machine and arrival time of higher priority job to next stage by AGV plus transportation time. Eventually, constraints (25) specify the domains and kinds of decision variables.

$$\begin{aligned} &t_{jmsm'(s+1)},\ RSe_{jims},\ St_{jms},\ C_{jms},\ Ns_{jims},\ H_{jms},\ Ar_{jms},\ C_{max} \geq 0 \\ &x_{jims},\ y_{jp},\ z_{jmsl},\ w_{jus} \in \{0, 1\} \end{aligned} \tag{25}$$

In summary, the mathematical model of EEFFSP is represented as follows:

Model:

First objective function = Eq. (1);

Second objective function = Eq. (7);

Constraints = Eqs. (8–25);
Where:

- *Constraints (8–10) define logical sequencing of jobs on machines,*
- *Constraints (11–14) manage assignment and speed decisions for both machines and AGVs,*
- *Constraints (15–24) link machine processing and AGV transport timings using energy- efficient speed selection and fuzzy learning effects,*
- *Constraint (25) defines domains of decision variable.*

Fuzzy programming

As seen in the developed model, two key parameters including processing time and learning coefficient are in the form of fuzzy numbers. It is assumed that both of them have triangular numbers as $\left(Pr_{jmsl}{}^{L}, Pr_{jmsl}{}^{C}, Pr_{jmsl}{}^{R}\right)$ and $\left(a_{jim}{}^{L}, a_{jim}{}^{C}, a_{jim}{}^{R}\right)$. In this paper, in order to convert the fuzzy mathematical model to its equivalent crisp formulations, the ranking technique proposed by Jiménez et al. [45] is used. This method ranks fuzzy numbers based on their expected value and the degree of satisfaction, transforming fuzzy constraints into deterministic equivalents through a linear combination of their membership functions. The main advantage of the Jiménez approach lies in its computational simplicity and its ability to preserve the balance between the optimistic and pessimistic evaluations of fuzzy parameters. Owing to these features, it has been widely used in fuzzy optimization and scheduling problems, and it is therefore adopted in this study to handle the fuzziness associated with processing times and learning coefficients effectively. Therefore, Eqs. (2), (4), (17) and (19) as fuzzy formulations in model are converted as follows:

Equation (2) is changed to Eqs. (26) and (27):

$$EC^{(1)} \geq \sum_{j=1}^{J}\sum_{m=1}^{M_s}\sum_{l=1}^{L}\sum_{s=1}^{S}\left[\left(\frac{\rho}{2}\right)\frac{Pr_{jmsl}{}^{C}+Pr_{jmsl}{}^{R}}{2}+\left(1-\frac{\rho}{2}\right)\frac{Pr_{jmsl}{}^{L}+Pr_{jmsl}{}^{C}}{2}\right].PEC_{msl}.z_{jmsl} \quad (26)$$

$$EC^{(1)} \leq \sum_{j=1}^{J}\sum_{m=1}^{M_s}\sum_{l=1}^{L}\sum_{s=1}^{S}\left[\left(1-\frac{\rho}{2}\right)\frac{Pr_{jmsl}{}^{C}+Pr_{jmsl}{}^{R}}{2}+\left(\frac{\rho}{2}\right)\frac{Pr_{jmsl}{}^{L}+Pr_{jmsl}{}^{C}}{2}\right].PEC_{msl}.z_{jmsl} \quad (27)$$

Equation (4) is changed to Eqs. (28) and (29):

$$EC^{(3)} \geq \sum_{m=1}^{M_s}\sum_{s=1}^{S}\left(\left[\max_j\left(C_{jms}.x_{jims}\right)-\sum_{h=1}^{J}\sum_{l=1}^{L}\left[\left(\frac{\rho}{2}\right)\frac{Pr_{hmsl}{}^{C}+Pr_{hmsl}{}^{R}}{2}+\left(1-\frac{\rho}{2}\right)\frac{Pr_{hmsl}{}^{L}+Pr_{hmsl}{}^{C}}{2}\right].z_{hmsl}-\sum_{h=1}^{J}\sum_{i=1}^{J}RSe_{hims}.x_{hims}\right].IEC_{ms}\right) \quad (28)$$

$$EC^{(3)} \leq \sum_{m=1}^{M_s}\sum_{s=1}^{S}\left(\left[\max_j\left(C_{jms}.x_{jims}\right)-\sum_{h=1}^{J}\sum_{l=1}^{L}\left[\left(1-\frac{\rho}{2}\right)\frac{Pr_{hmsl}{}^{C}+Pr_{hmsl}{}^{R}}{2}+\left(\frac{\rho}{2}\right)\frac{Pr_{hmsl}{}^{L}+Pr_{hmsl}{}^{C}}{2}\right].z_{hmsl}-\sum_{h=1}^{J}\sum_{i=1}^{J}RSe_{hims}.x_{hims}\right].IEC_{ms}\right) \quad (29)$$

Equation (17) is changed to Eqs. (30) and (31):

$$RSe_{jims} \geq Se_{jims}.\left(Ns_{jims}+1\right)^{\left[\left(\frac{\rho}{2}\right)\frac{a_{jim}^{C}+a_{jim}^{R}}{2}+\left(1-\frac{\rho}{2}\right)\frac{a_{jim}^{L}+a_{jim}^{C}}{2}\right]} \quad \forall i,j=1,\ldots,J;\ m=1,2,\ldots,M_s;\ s=2,\ldots,S \quad (30)$$

$$RSe_{jims} \leq Se_{jims}.\left(Ns_{jims}+1\right)^{\left[\left(1-\frac{\rho}{2}\right)\frac{a_{jim}^{C}+a_{jim}^{R}}{2}+\left(\frac{\rho}{2}\right)\frac{a_{jim}^{L}+a_{jim}^{C}}{2}\right]} \quad \forall i,j=1,\ldots,J;\ m=1,2,\ldots,M_s;\ s=2,\ldots,S \quad (31)$$

Equation (19) is changed to Eqs. (32) and (33):

$$C_{jms} \geq St_{jms}+\sum_{l=1}^{L}\left[\begin{array}{l}\left(\frac{\rho}{2}\right)\frac{Pr_{jmsl}^{C}+Pr_{jmsl}^{R}}{2}\\+\left(1-\frac{\rho}{2}\right)\frac{Pr_{jmsl}^{L}+Pr_{jmsl}^{C}}{2}\end{array}\right].z_{jmsl} \quad \forall j=1,\ldots,J;\ m=1,2,\ldots,M_s;\ s=1,2,\ldots,S \quad (32)$$

$$C_{jms} \leq St_{jms}+\sum_{l=1}^{L}\left[\begin{array}{l}\left(1-\frac{\rho}{2}\right)\frac{Pr_{jmsl}^{C}+Pr_{jmsl}^{R}}{2}\\+\left(\frac{\rho}{2}\right)\frac{Pr_{jmsl}^{L}+Pr_{jmsl}^{C}}{2}\end{array}\right].z_{jmsl} \quad \forall j=1,\ldots,J;\ m=1,2,\ldots,M_s;\ s=1,2,\ldots,S \quad (33)$$

where $\rho\epsilon\,[0,1]$ is as feasibility degree of constraints determined by decision maker to introduce a threshold of risk acceptance. In this study, ρ is set to 0.7 to achieve a balanced compromise between solution robustness and adaptability. This selection aligns with previous works adopting similar fuzzy optimization frameworks, ensuring both methodological consistency and reliable decision-making performance under uncertainty.

Briefly, in the proposed approach, the fuzzy model initially contains uncertain parameters expressed as triangular fuzzy numbers, including processing times and learning coefficients. To obtain a solvable crisp model, the ranking method of Jiménez et al. [45] is applied. This technique computes the expected value and satisfaction degree for each fuzzy parameter according to a predefined feasibility degree ($\rho = 0.7$). The fuzzy constraints are then converted into their deterministic equivalents by substituting the fuzzy values with their corresponding crisp representations derived from the ranking process. Consequently, the fuzzy model is transformed into a deterministic mixed-integer programming formulation, which can be solved using the optimization procedure. This process ensures that the uncertainty in the original model is appropriately reflected while maintaining computational tractability.

Solution methodology

Because in multi-objective mathematical programming (MMP) there is more than one objective function therefore there is no single optimal solution that optimizes all of them simultaneously. In such cases, the decision makers select the most preferred solution. As a result, the concept of efficiency or Pareto optimality is raised in MMP. A Pareto optimal (non-dominated) solution cannot improve one objective without worsening at least one other objective. In literature, there are various categories of techniques to solve MMP problems. In one of these categories, the techniques are divided into two categories: classical and evolutionary algorithms. Classical algorithms are usually suitable for small to medium-sized problems because they have a high computational time while, evolutionary algorithms are utilized due to their lower computational time and the quality of optimal or near-optimal solutions. In this paper, a classical method as augmented ε-constraint method (AUGMECON) to achieve the Pareto optimal solutions for small-sized examples as well as a non-dominated sorting genetic algorithm II (NSGAII) to obtain the Pareto optimal or near-optimal solution are developed.

The augmented ε-constraint method (AUGMECON)

AUGMECON method is introduced by Mavrotas [46] for the first time. This method enhances the conventional ε -constraint method for generating the Pareto optimal solutions. Indeed, AUGMECON method fixes some weak points of conventional ε-constraint method such as the guarantee of Pareto optimality of the obtained solution in the payoff table and generation process. In order to introduce AUGMECON method, assume the following MMP problem:

$$\begin{aligned} &\min\left(f_1(x), f_2(x), \ldots, f_p(x)\right) \\ &\text{St}: \quad X \in S; \end{aligned} \tag{34}$$

where X is the vector of decision variables, $f_1(x), f_2(x), \ldots, f_p(x)$ are the p objective functions and S is the feasible region.

In this method, one of the objective functions is optimized, while the remaining objectives are converted into constraints with allowable ε-levels. To avoid weakly efficient solutions, an augmented term consisting of the weighted sum of normalized slack variables is added to the main objective function. The method divides the range between the utopia and pseudo-nadir points of each objective function into equal intervals using parameters q_i and n_i producing multiple ε-level combinations. For each combination, the corresponding mixed-integer program is solved, and the non-dominated solutions form the Pareto frontier. This approach allows systematic exploration of the trade-offs among conflicting objectives while maintaining numerical stability and ensuring the true Pareto optimality of the obtained solutions. Therefore, the above model is converted into the following one by applying the AUGMECON method:

$$\min(f_1(x) - r_1.(s_2/r_2 + s_3/r_3 +, \ldots, s_p/r_p)) \tag{35}$$

$$\text{St}: f_2(x) + s_2 = e_2 \tag{36}$$

$$f_3(x) + s_3 = e_3 \tag{37}$$

$$f_p(x) + s_p = e_p \tag{38}$$

$$e_2 = f_2^{SN} + \left(\frac{f_2^U - f_2^{SN}}{q_2}\right).n_2 \forall n_2 = 0,1,\ldots,q_2 \tag{39}$$

$$e_3 = f_3^{SN} + \left(\frac{f_3^U - f_3^{SN}}{q_3}\right).n_3 \forall n_3 = 0,1,\ldots,q_3 \tag{40}$$

$$e_p = f_p^{SN} + \left(\frac{f_p^U - f_p^{SN}}{q_p}\right).n_p \forall n_p = 0,1,\ldots,q_p \tag{41}$$

$$X \epsilon S \tag{42}$$

where $s_2, \ldots, s_p$ are slack variables for the constraints. Also, f_i^U is as a utopia point of i th objective function where this objective function has is at its best possible value. f_i^{SN} is as a pseudo nadir point of i th objective function so that $f_i^{SN} = \max\left\{f_i(x_1^*), \ldots, f_i(x_i^*), \ldots, f_i\left(x_p^*\right)\right\}$ where $f_i(x_i^*)$ the optimum value of i th objective function in x_i^* as vector of decision variables which optimizes the i th objective function. $r_1, r_2, \ldots, r_p$

		Machine 1			Machine 2		
		Speed 1	Speed 2	Speed 3	Speed 1	Speed 2	Speed 3
Job 1	Stage 1	(8,11,14)	(5,9,11)	(2,5,9)	(10,14,17)	(7,12,13)	(4,8,10)
	Stage 2	(8,12,14)	(6,9,12)	(4,7,8)	(5,7,10)	(3,5,8)	(1,3,6)
	Stage 3	(12,15,20)	(8,12,16)	(5,9,10)	(8,12,15)	(6,10,13)	(3,8,9)
Job 2	Stage 1	(4,6,8)	(3,5,6)	(2,4,5)	(6,9,13)	(4,7,10)	(2,6,9)
	Stage 2	(9,13,18)	(6,10,12)	(3,7,8)	(11,15,16)	(9,11,12)	(6,8,9)
	Stage 3	(6,9,10)	(4,7,9)	(2,4,5)	(7,8,10)	(5,6,8)	(3,4,5)
Job 3	Stage 1	(8,10,11)	(6,9,10)	(3,5,8)	(4,7,9)	(3,5,8)	(1,3,6)
	Stage 2	(5,10,15)	(3,7,10)	(2,4,6)	(9,13,15)	(8,11,12)	(5,9,10)
	Stage 3	(7,8,10)	(5,6,9)	(4,5,6)	(7,9,12)	(6,8,9)	(5,7,8)

Table 2. The processing time of jobs on machines (minute).

		Machine 1			Machine 2		
		Stage 1	Stage 2	Stage 3	Stage 1	Stage 2	Stage 3
Machine 1	Stage 1	0	18	24	5	14	35
	Stage 2	18	0	17	20	3	25
	Stage 3	24	17	0	28	15	8
Machine 2	Stage 1	5	20	28	0	18	10
	Stage 2	14	3	15	18	0	15
	Stage 3	35	25	8	10	15	0

Table 3. The distances between the machines (meter).

		Machine 1				Machine 2			
		Stage 1	Stage 2	Stage 3	fuzzy learning coefficient	Stage 1	Stage 2	Stage 3	fuzzy learning coefficient
Job 1	Job 2	1.5	0.75	1.5	(-0.7,-0.5,-0.3)	2	1.5	1	(-3.5,-2,-1)
	Job 3	1.25	1.75	1.5	(-0.6,-0.5,-0.4)	2.5	1.25	1.75	(-3,-2,-1)
Job 2	Job 1	1	1	1.75	(-0.6,-0.5,-0.4)	1.5	1.5	1.75	(-3,-2.5,-1)
	Job 3	2	2.5	3	(-0.7,-0.5,-0.3)	1.25	1.5	1.75	(-3,-2,-1)
Job 3	Job 1	2	1.75	2	(-0.6,-0.5,-0.4)	1	1.75	1.25	(-3,-2,-1)
	Job 2	0.75	2	1.75	(-0.7,-0.5,-0.3)	0.5	1.5	1.5	(-3,-2,-1.5)

Table 4. The sequence-dependent setup times (minute) and fuzzy learning coefficient.

are the range of i th objective function that $r_i = f_i^{SN} - f_i^U$. q_i and n_i are used to divide the range of objective functions to equal intervals.

Example 1

Suppose there is a manufacturing system including three jobs, three stages, two machines at each stage and two same AGVs. Also, there are three types of speeds of machines and AGVs. The speed of AGVs can be selected between *AGV.speed 1* = *1*, *AGV.speed 2* = *2* and *AGV.speed 3* = *3* m per minute. The processing time of jobs on machines at stages for each type of machine speed are triangular fuzzy numbers and have been given in Table 2. As you can see, as the speed of the machines increase, the processing time decrease. The distance between machines and the basic sequence-dependent times together with the fuzzy learning coefficients are shown in Table 3 and Table 4. In addition, the value of energy consumption factors PEC_{msl}, SEC_{ms} and IEC_{ms} are shown in Table 5. It is clear that with increasing the speed of machines, more energy will be consumed by them. Finally, two parameters LEC_{jsu} and UEC_{su} are shown in Table 6. It should be noted there is in the number of stages minus one AGV in considered system.

It is important to note that, in a practical manufacturing environment, the two fuzzy parameters used in this study, namely, the processing times and the learning coefficients, were calibrated using a combination of historical production data and expert judgment. Specifically, the lower and upper bounds of each triangular fuzzy number represent the best-case (optimistic) and worst-case (pessimistic) observed values in the production environment data, while the most likely value corresponds to the expert-estimated typical or average performance. Triangular fuzzy numbers were selected because of their simplicity, interpretability, and ability to model uncertainty when only limited statistical information is available. This structure allows imprecise and subjective information to be represented with low computational complexity while maintaining a realistic reflection of operational variability.

		PEC_{msl}			SEC_{ms}	IEC_{ms}
		Speed 1	Speed 2	Speed 3		
Machine 1	Stage 1	100	300	500	60	50
	Stage 2	40	160	340	70	70
	Stage 3	80	290	440	45	40
Machine 2	Stage 1	70	250	530	30	35
	Stage 2	50	200	400	90	50
	Stage 3	90	410	745	60	45

Table 5. The energy parameters related to the machines (Joule).

		Stage 1			Stage 2		
		AGV.speed 1	*AGV.speed 2*	*AGV.speed 3*	*AGV.speed 1*	*AGV.speed 2*	*AGV.speed 3*
Jobs	Job 1	100	150	200	40	60	100
	Job 2	20	50	60	100	150	210
	Job 3	70	95	135	30	70	100
UEC_{su}		20	40	50	40	70	40

Table 6. The energy parameters related to the AGVs (Joule).

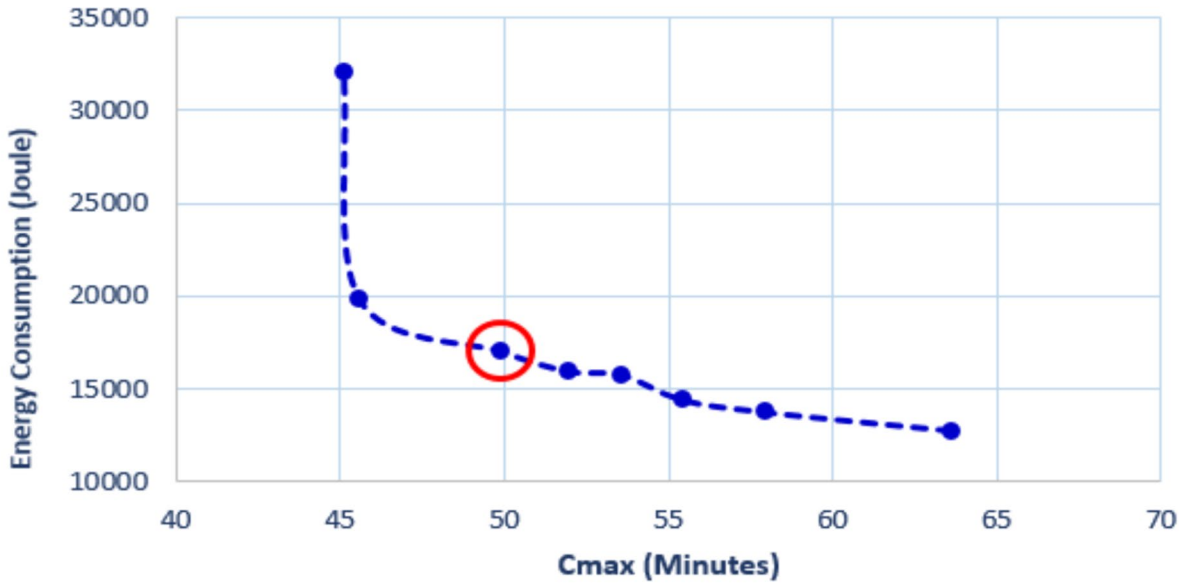

Fig. 3. The Pareto solution frontier for presented example.

In this research, the fuzzy parameters were defined as follows:

- Optimistic value (a): The lower bound, representing the minimum observed value from historical production data, reflecting the best operational conditions;
- Most likely value (b): The most probable value, derived from a combination of past data and expert estimation, representing typical performance;
- Pessimistic value (c): The upper bound, representing the maximum observed value from historical data, reflecting the worst-case conditions including delays or inefficiencies

The obtained solutions for example 1

The above numerical example is solved by AUGMECON method coded in optimization software GAMS 23.5 and the Pareto solution frontier was obtained according to Fig. 3.

In order to check the details of the results more closely, one of the Pareto frontier points was selected which has $z_1^* = 49.942$ and $z_2^* = 16972$ (equivalent to the point inside the red circle on Fig. 2). The results of obtained optimal EFFSS are shown in Gantt chart according Fig. 4. As can be seen in this figure, the white boxes are jobs

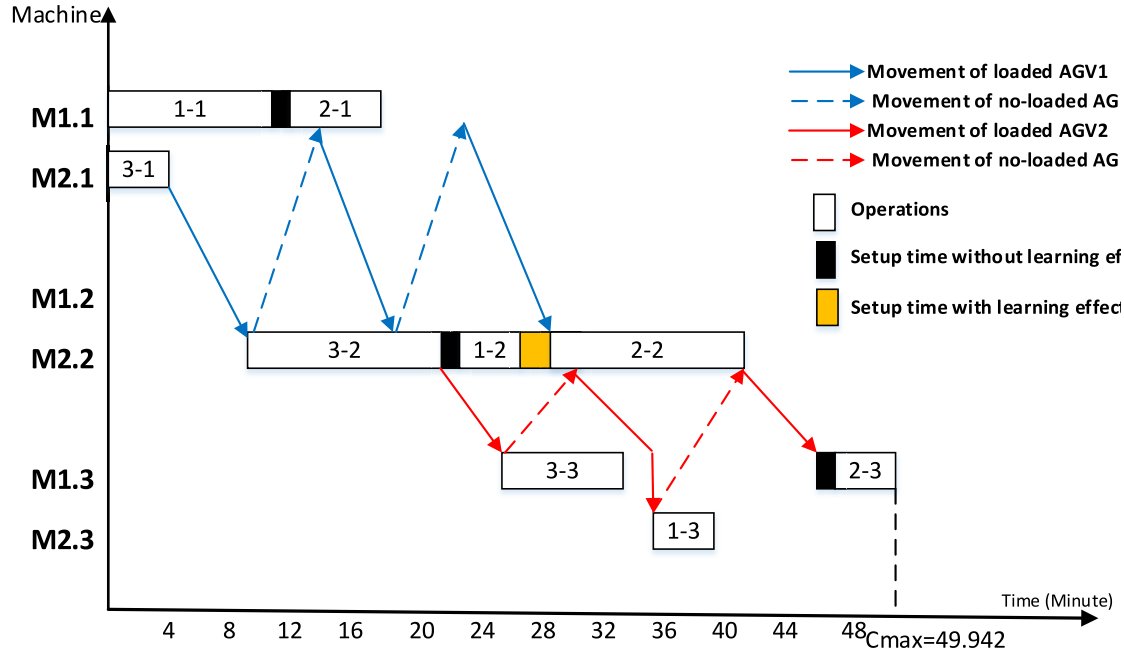

Fig. 4. The obtained Gantt chart of one of points of optimal solution frontier in example.

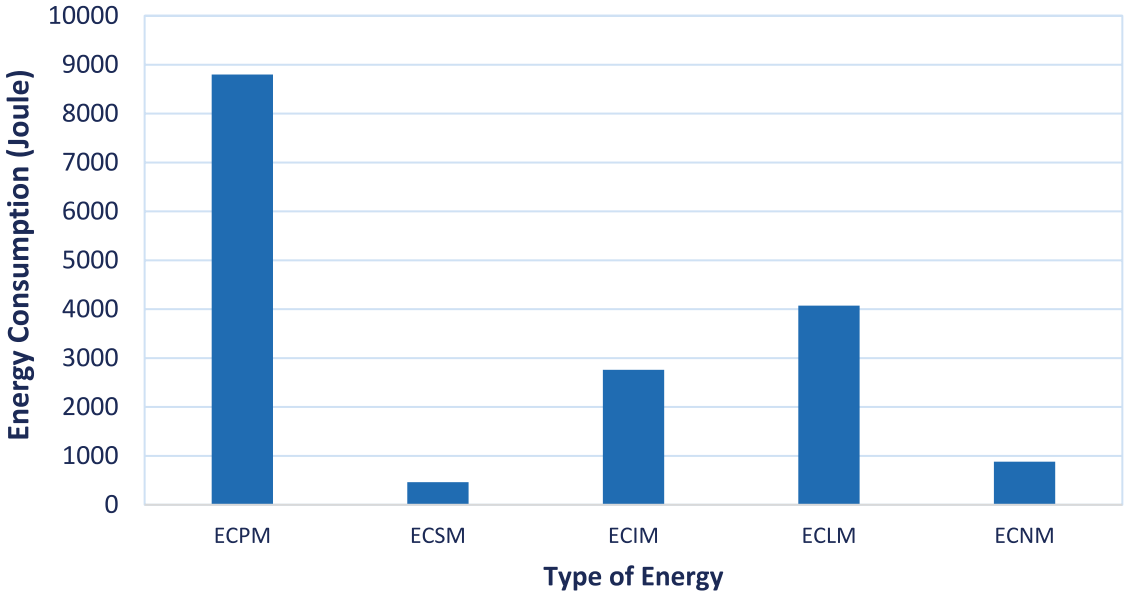

Fig. 5. The kind of consumed energy in related to developed example.

processing, and black boxes are sequence-dependent setups time between jobs (of course, it should be noted that the yellow box is sequence-dependent setups time that is together with learning effect). The blue directional arrows show the AGV between the first and second stage and the red one corresponds to the AGV between the second and third stage. Also, the solid and dash lines are loaded and no-loaded AGVs' movements respectively. The amount of energy consumption introduced in the paper for the selected point on Pareto frontier is according to Fig. 5. Note that in our model, energy consumption is driven by both machine-related decisions and AGV-related operations. Scheduling decisions such as machine speed selection have a direct impact on ECPM, higher processing speeds increase energy usage but may reduce completion times. Similarly, the job sequencing influence ECPM, ECSM and ECIM especially under the learning effect, where repeated setups for similar jobs reduce energy-intensive preparation phases. Also, speed decisions of AGVs directly affect ECLM and ECNM. The impact of scheduling decisions on energy consumption can be represented in Table 7.

Scheduling Decision	ECPM	ECSM	ECIM	ECLM	ECNM
Machine speed selection	✓				
Job sequencing	✓	✓	✓		
Setup learning effect		✓			
AGV speed selection				✓	✓

Table 7. The Impact of scheduling on kinds of energy consumption.

Case	Total setup time (minute)	ECSM	Total energy consumption (Joule)	Makespan (minute)
A: 20% reduction in lower, middle and upper values	2.75	243	16,831	47.323
B: 10% reduction in lower, middle and upper values	3.25	276	16,864	48.726
C: no change	4.5	384	16,972	49.942
D: 10% increase in lower, middle and upper values	4.75	411	16,999	50.124
E: 20% increase in lower, middle and upper values	5.25	431	17,020	50.136

Table 8. The sensitivity analysis on learning coefficient.

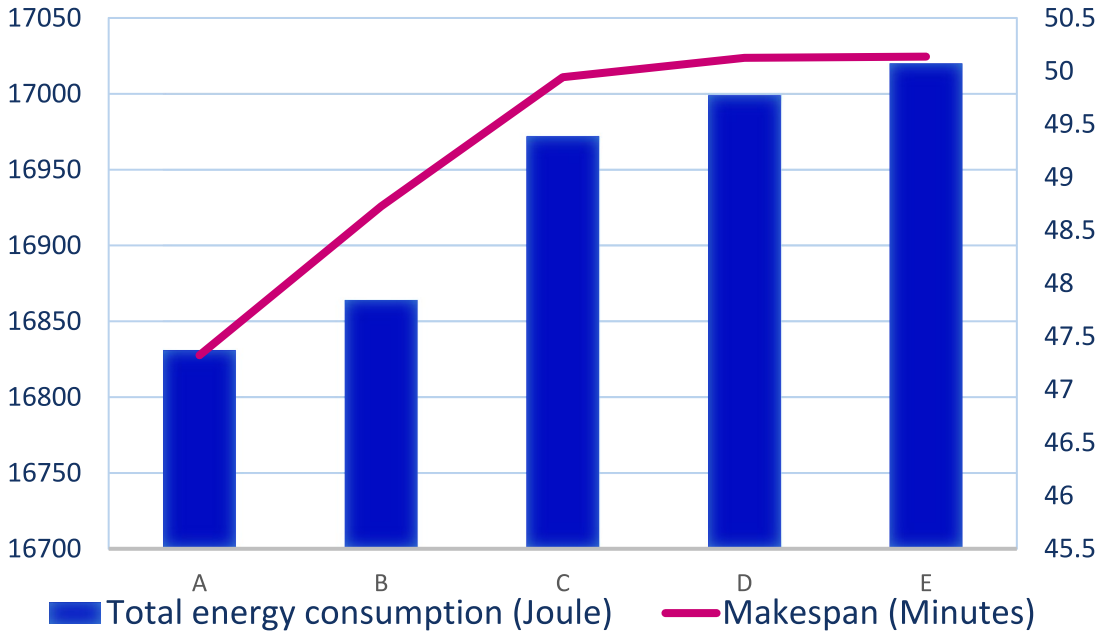

Fig. 6. The sensitivity analysis of total energy consumption and makespan on learning coefficient.

The sensitivity analysis on learning coefficient

To evaluate the impact of the fuzzy learning effect on system performance, a sensitivity analysis was conducted by varying the triangular fuzzy learning coefficient used in setup time calculations. This coefficient captures the degree of setup time reduction based on accumulated learning and is modeled using triangular fuzzy numbers to reflect its uncertainty. For each case, we measured. The results are summarized as Table 8.

- Total Setup Time
- Energy Consumption in Setup Mode (ECSM)
- Total Energy Consumption
- Makespan

As expected, a stronger learning coefficient (i.e., more negative α values) results in shorter setup times. The ECSM, total energy consumption and makespan improve with stronger learning coefficient. These findings highlight the critical influence of learning behavior on scheduling performance and validate the model's responsiveness to this fuzzy parameter. A visual summary is provided in Fig. 6, which plots the relationship between the learning coefficient and both total energy consumption and makespan. It is observed that as the learning coefficient increases, both the total energy consumption and the makespan also increase; however, the rate of increase

occurs with a diminishing slope, indicating a nonlinear and gradually stabilizing effect of the learning curve on system performance.

The computation time of this small-sized example on the GAMS was about 2 h. Certainly, in order to solve the examples with higher dimensions, using this software will be ineffective. Therefore, a Non-dominated sorting genetic algorithm II (NAGA-II) is developed to solve medium/large-sized examples.

NSGAII algorithm

In order to solve the multi-objective problems, two general categories of methods have been developed in the literature. The first category, which reaches to an optimal or near-optimal result is named classic methods. They define an integrated objective function based on all objective functions (e.g. goal programming, weighted sum method and goal attainment). The second category is evolutionary methods. The aim of this category is to find a set of solutions named Pareto solutions. In a Pareto set, there are a number of solutions so that none of which can completely dominate other Pareto solutions. One of the most widely used methods of the evolutionary methods is Non-dominated sorting genetic algorithm II (NSGAII) to solve the problems with more than one goal. The NSGAII proposed by Deb [47] works like genetic algorithm (GA) in terms of mechanism for generation and changing of the population. The flowchart of proposed NSGAII is depicted in Fig. 7.

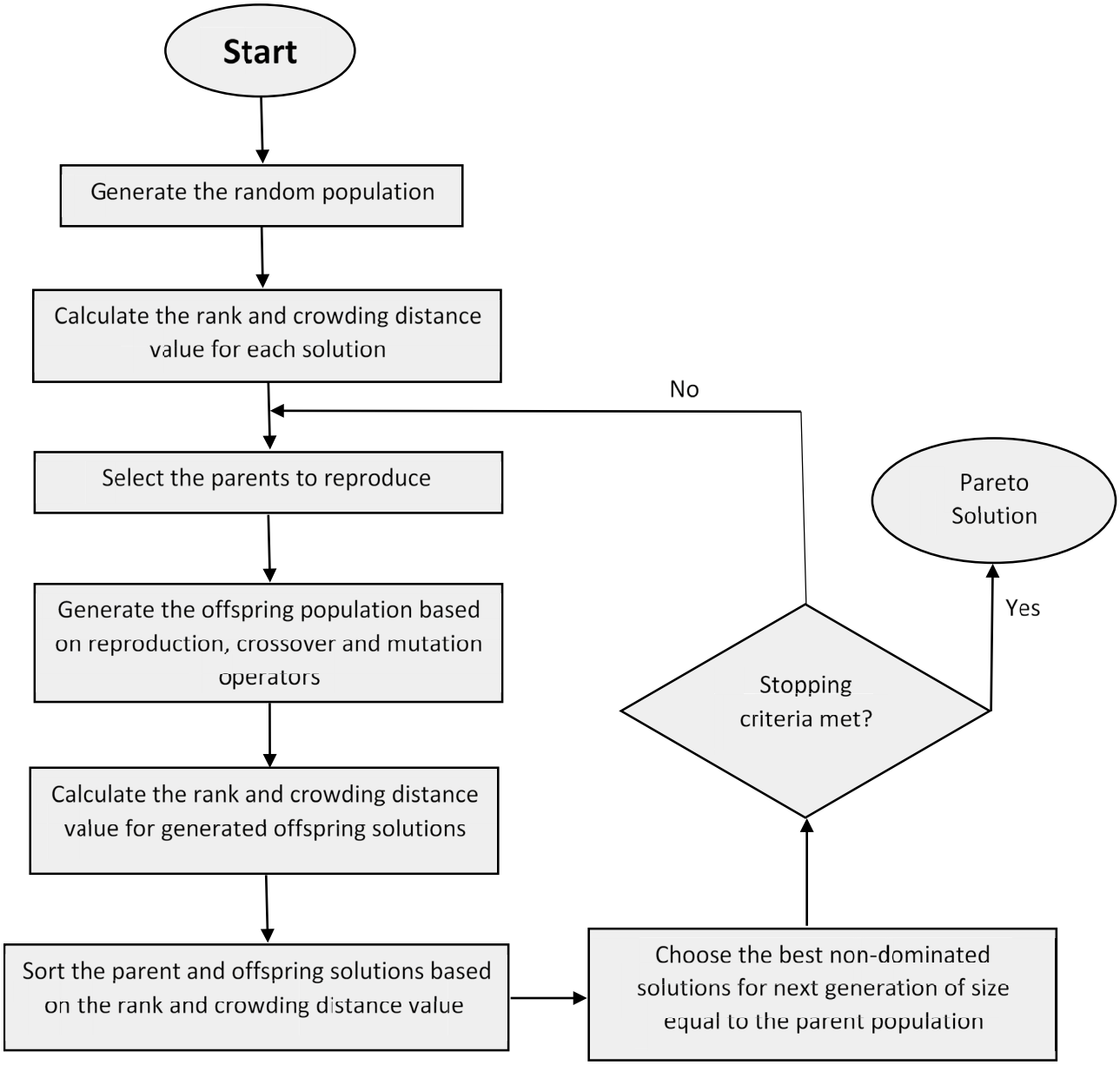

Fig. 7. The flowchart of developed NSGAII.

Solution coding (chromosome structure)

The efficiency of the algorithm is highly dependent on the coding and representation of the solutions. A feasible solution (chromosome) representation of the described model is shown in Fig. 8. This chromosome is related to a problem with 7 jobs, 3 stages, 3 machines and AGV speeds; and 3 machines in stage 1, 2 machines in stage 2 and 3 machines in stage 3. Jobs 5, 4 and 1 are processed on machine 1 in stage 1. Also, jobs 6 and 2 are processed on machine 2 in stage 1; and jobs 3 and 7 are processed on machine 3 in stage 1. Numbers 8 and 9 are used as machine separators in stage 1. In stage 2, because there are two machines, a separator number was needed. Therefore, the number 8 is repeated twice, and of course, compulsorily one of them must be placed in the last column of the corresponding row. In stage 2, jobs 4, 1 and 2 are located on machine 1; and jobs 5, 6, 7 and 3 are processed on machine 2. Generally, chromosome related to each problem with *J* jobs, *S* stages, *L* speeds for machines and *U* speeds for AGVs and M_s machines in stage *s*th is a *3S-1* by *J+max*(M_s)*-1* matrix.

In the proposed algorithm, each chromosome represents a complete production schedule that simultaneously encodes the job sequence, machine assignment, and AGV speed levels across all stages. The chromosome structure is designed as a matrix, where each row corresponds to a production stage and each column denotes the processing order of jobs on the available machines. Separator numbers are inserted to distinguish between different machines within a stage. The encoding process generates feasible chromosomes by randomly sequencing jobs while respecting machine capacities and stage precedence constraints. During the decoding process, the chromosome is sequentially interpreted to reconstruct an executable schedule: jobs are assigned to machines according to their encoded order, setup and transportation times are calculated based on AGV speed levels, and start/finish times are determined while ensuring all precedence and resource constraints are satisfied.

To better justify the proposed chromosome encoding and improve clarity, a visual example of decoding the chromosome into a feasible schedule has been added. Specifically, a Gantt chart representation is provided in Fig. 9, which illustrates how the encoded job sequences, machine assignments, and speed selections are translated into an actual production schedule. This decoding example visually demonstrates the logical flow from the chromosome structure to operational scheduling decisions, helping to validate the complexity of the encoding approach. The added figure clearly shows job processing orders, machine allocations, and AGV movements across stages, providing an intuitive understanding of the proposed representation.

Generation of initial population

In this paper, the initial population is obtained by a random process. In other words, it is created by random generation of above-mentioned chromosomes in number of population size. The first *S* rows (job sequence in stages) are filled by numbers $1, 2, \ldots, J$ and separator numbers. The next *S* rows (machine speeds in stages) are filled by numbers $1, 2, \ldots, L$ and the final $S-1$ rows are filled by numbers $1, 2, \ldots, U$.

Fitness Value

The fitness function is used to determine the quality of a chromosome or feasible solution. In here, quality of chromosome depends on two objective function values. In proposed NSGAII, the fitness is calculated based on both ranking and crowding distance. Each solution has a non-domination level. The solutions with rank level 1 are not dominated, and solutions with rank level $R+1$ are only dominated by solutions in rank level R. Solutions of a same rank level cannot dominate each other. Such solutions are compared by calculating their crowding distance. The crowding distance computes the absolute distance between the normalized objective functions of the adjacent elements. Equation (43) calculates the crowding distance d_i of solution i where f_i^r indicates the r th objective function value of the i th solution, and f_{min}^r and f_{max}^r are minimum and maximum values of the r th objective function. A high value for crowding distance is preferred. Moreover, the crowding distance of the first and last points are infinite and therefore these two points are always selected. Briefly, the

	Job list of machine 1 at stage 1			Job list of machine 2 at stage 1			Job list of machine 3 at stage 1		
Job sequence at stage 1	5	4	1	8	6	2	9	3	7
Job sequence at stage 2	4	1	2	8	5	6	7	3	8
Job sequence at stage 3	1	6	4	5	8	3	9	7	2
Machine speeds at stage 1	2	1	1	-	1	2	-	2	1
Machine speeds at stage 2	3	3	1	-	1	2	3	3	-
Machine speeds at stage 3	2	3	3	2	-	1	-	2	2
AGV speeds between stages 1 and 2	1	1	1	-	3	2	-	2	1
AGV speeds between stages 2 and 3	3	3	2	-	1	2	2	1	-

Fig. 8. An example of solution representation (chromosome).

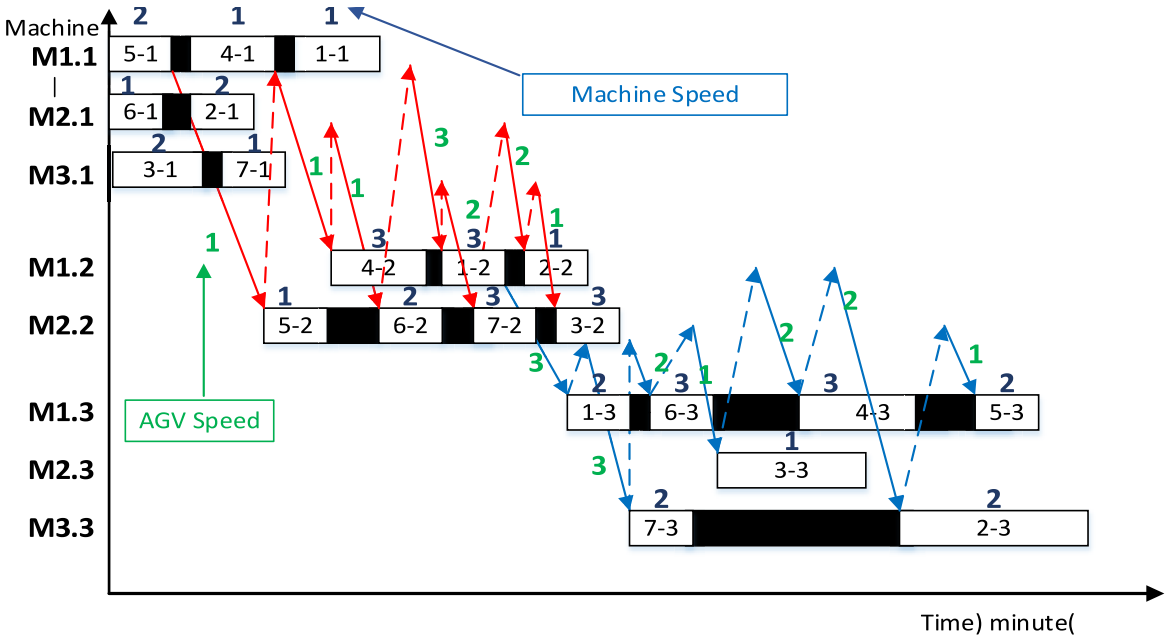

Fig. 9. Gantt chart designed for chromosome corresponding to Fig. 8

solutions with a higher rank level have higher fitness and if two solutions have same rank level, solution with high crowding distance has high fitness.

$$d_i = \sum_r \frac{|f^r_{i+1} - f^r_{i-1}|}{f^r_{max} - f^r_{min}} \tag{43}$$

Crossover and mutation operator

In this paper, at first, two parents are selected by the binary tournament selection method and then the one-point crossover is used to generate the offspring. A sample of used crossover operator is presented in Fig. 10.

For the considered problem of this paper, three mutation operators including swap, reversion and insertion are applied randomly in each iteration. For instance, three mutation operators are shown in Fig. 11. The stop criterion for NSGAII algorithm is to meet the maximum iteration.

With the chromosome structure defined above as the foundation, we now turn our attention to strategies for effectively navigating the high-dimensional solution space. The representation of chromosomes directly influences how candidate solutions are generated, manipulated, and evaluated, which is critical in a complex problem such as EEFFSP with AGVs, sequence-dependent setup times, and fuzzy processing times. By carefully designing the chromosome structure, we ensure that each potential solution accurately reflects the multiple dimensions of the scheduling problem, providing a solid basis for the search process. Building on this foundation, the strategies for managing high-dimensional solution spaces are essential to maintaining both diversity and convergence within NSGA-II. High-dimensional spaces often contain many local optima, which can mislead the search process or cause premature convergence. By employing specific mechanisms—such as adaptive crowding distance calculation, elite preservation, and problem-tailored genetic operators—NSGA-II is guided to explore the solution space more effectively, avoiding stagnation and promoting a well-distributed set of non-dominated solutions along the Pareto frontier. These strategies collectively ensure that the algorithm can handle the complexity of the EEFFSP while balancing multiple objectives. The chromosome design and high-dimensional management are thus intrinsically linked: the former defines what is being searched, and the latter governs how the search proceeds efficiently. This integrated approach allows NSGA-II to produce high-quality, diverse solutions that capture the trade-offs between makespan, energy consumption, and other relevant performance criteria in our proposed scheduling model.

Strategies for managing high-dimensional solution space

In developed NSGAII, the high-dimensional solution space was managed through several strategies for large-scale problem instances:

- Penalty functions for infeasible solutions guide the population toward feasible regions.
- The compact chromosome design ensures efficient encoding of multiple decision layers.
- NSGA-II's ranking and crowding distance maintain solution diversity and prevent premature convergence.
- Simple genetic operators (binary tournament selection and one-point crossover) balance exploration and exploitation without adding computational burden.

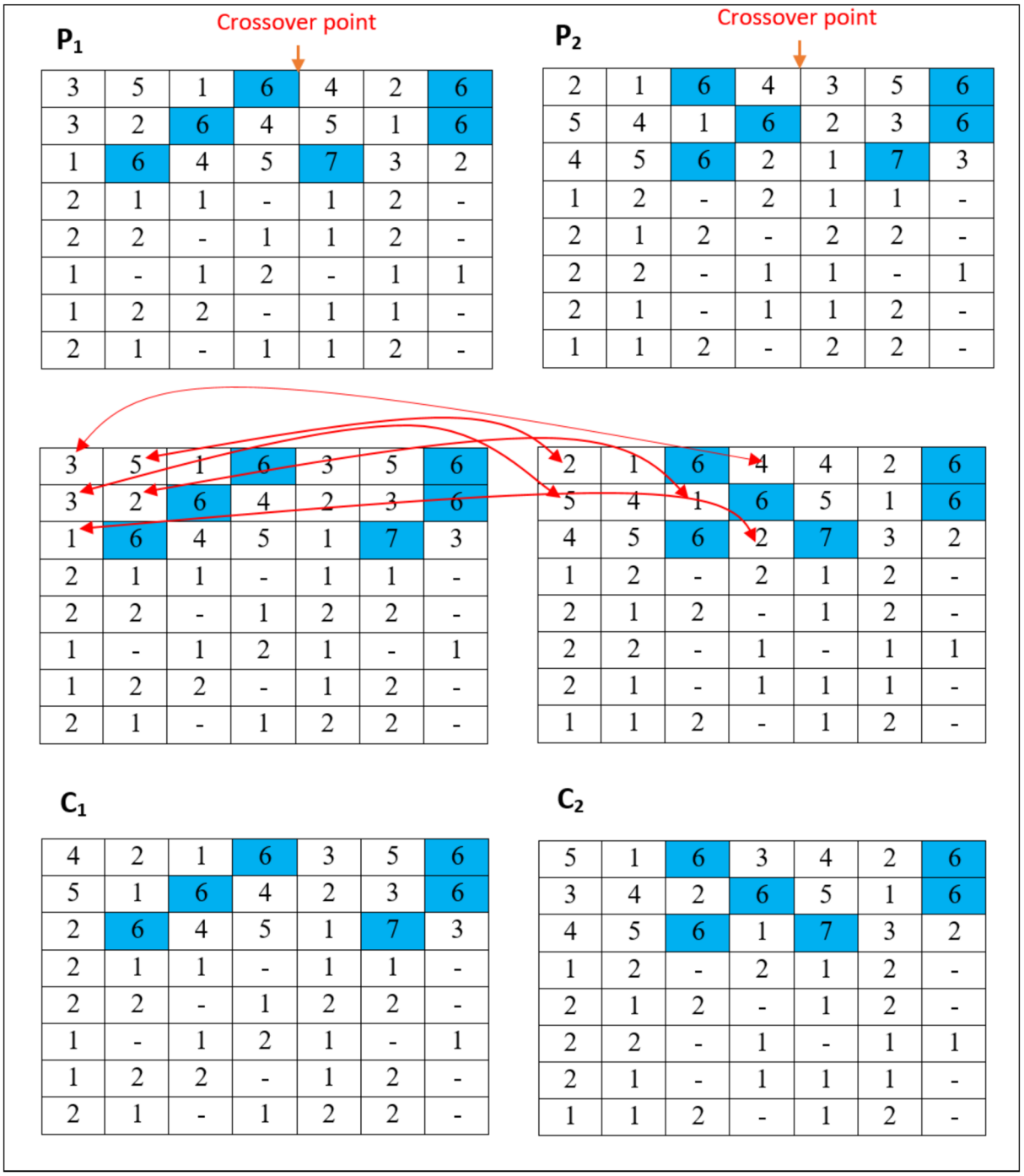

Fig. 10. The crossover of NSGAII.

These strategies together help maintain efficiency even as the problem size increases.

Parameter setting
NSGAII is strongly dependent on its parameters which these parameters are:

- Number of members in a population (*NPop*)
- Number of iterations to find best results (*MaxIt*)
- Crossover rate (*CrR*)
- Mutation rate (*MuR*)

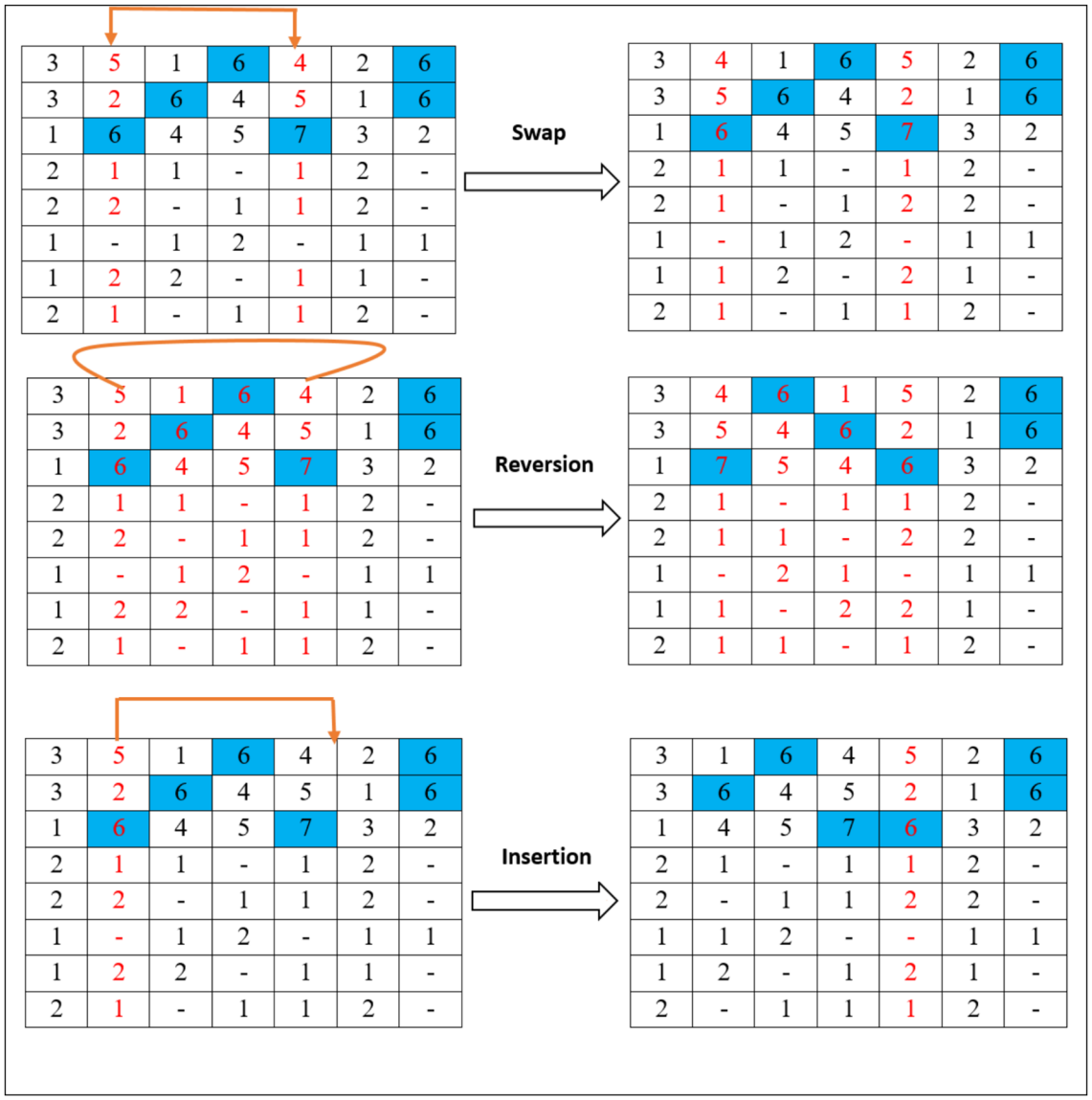

Fig. 11. The mutation operators (swap, reversion and insertion) of NSGAII.

Parameters	Level 1	Level 2	Level 3	Final value
NPop	50	75	100	75
MaxIt	150	250	400	250
CrR	0.7	0.8	0.9	0.7
MuR	0.1	0.2	0.3	0.3

Table 9. The suitable levels for the parameters of NSGAII.

In here, a Taguchi method [44] considering the plan of L_{27} is used to obtain the suitable levels for the parameters. The defined parameter levels and the final values for parameters with respect to mean ideal distance (MID) index are represented in Table 9. The MID index is calculated as Eq. (44):

$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \tag{44}$$

where $c_i = \sqrt{f_{i1}^2 + f_{i2}^2}, \forall i = 1,2,\ldots,n$. f_{i1}^2 and f_{i2}^2 are the first and second objective function value for each solution.

Problem	Dimensions				
	J	S	M_s	L	U
PN1	3	3	2	2	2
PN2	5	3	2	3	3
PN3	7	3	3	3	3
PN4	10	3	3	4	4
PN5	15	4	4	5	5
PN6	18	5	4	5	5
PN7	20	6	5	5	5
PN8	20	6	5	7	7

Table 10. The dimension of 8 test problems.

Parameter	Description	Numerical value (uniform distribution function)	Unit
AGV speeds	Speed of AGVs used for material handling	$\sim U$ [1,5]	meter/minute
$\widetilde{Pr}_{jmsl}$	Fuzzy triangular processing time	$\sim$(U [1,5] , U [6,9] , U [10,15])	minute
$D_{msm\prime(s+1)}$	Travel distance	$\sim U$ [5,50]	meter
Se_{jims}	Sequence-dependent setup time	$\sim U$ [0.25,5]	minute
$\widetilde{a}_{jim}$	Fuzzy learning coefficient	$\sim U$ $[-5, -1]$	dimensionless
PEC_{msl}	Processing energy consumption cost	$\sim U$ [10,1000]	dollar
SEC_{ms}	Setup energy consumption cost	$\sim U$ [10,200]	dollar
IEC_{ms}	Idle energy consumption cost	$\sim U$ [10,80]	dollar
LEC_{jsu}	Loaded AGV energy consumption cost	$\sim U$ [10,500]	dollar
UEC_{su}	Un-loaded AGV energy consumption cost	$\sim U$ [10,200]	dollar

Table 11. The parameter values for test problems.

Computational experiments

In this section, several test problems and the result analysis for solving of these test problems using mathematical model, NSGAII and NSGA are presented. The mathematical model is solved by the coded AUGMECON method in optimization software GAMS 23.5; NSGAII and NSGA is coded by MATLAB R2012a software on a personal computer with Intel core i7 CPU, 2.2 GHz and 6 Gb RAM. All computational experiments, including AUGMECON, NSGA, and NSGA-II, were performed on the same personal computer with Intel Core i7 CPU (2.2 GHz) and 6 GB RAM. Notably, the NSGAII and NSGA were run for 3 times and their average was used. In here, 8 test problems are developed with dimensions and parameter value according to Table 10 and Table 11, respectively. To compare AUGMECON, NSGAII and NSGA; and assess these methods in efficiency of their solution, six indices including NPF (number of solutions in the Pareto front), which represents the total number of non-dominated solutions, MID (mean ideal distance), which measures the average distance of the obtained solutions from the ideal point, SM (spacing metric), which evaluates the uniformity of solutions along the Pareto front, MSI (maximum spread index), which quantifies the extent of spread of solutions across the objective space, HV (hypervolume), which indicates the volume of the objective space dominated by the Pareto front, and CPU-time, which denotes the computational time in seconds required to obtain the solutions are represented in Table 12. The lower values for MID, SM and CPU-time; and the bigger values for MSI, HV and NPF are preferred. Overall, NSGAII demonstrated superior performance compared to NSGA in most evaluation metrics. It outperformed NSGA in terms of NPF, SM, MSI, and HV in the majority of examples, indicating better solution quality, distribution uniformity, and spread. Both algorithms showed comparable results for MID, while NSGA was faster in CPU-time, demonstrating its advantage in computational efficiency. These results confirm NSGAII as the more robust and reliable optimization approach for the proposed EFFSS problem.

Comparing the obtained indices, it can be concluded that the NSGAII and NSGA have gained the efficient solutions within shorter computational time. To provide a clearer comparison between the two algorithms, a summary table as Table 13 has been depicted, presenting the numerical results of NSGA-II versus NSGA along with the percentage improvement for each performance metric. As shown in this table, NSGA-II achieves better overall results in terms of solution quality and Pareto front diversity. Specifically, it provides improvements of approximately 4% in the number of Pareto front solutions (NPF), 12% in hypervolume (HV), and 4% in spacing metric (SM), along with a slightly better mean ideal distance (MID). Although NSGA-II requires about 14% more computational time, its superior convergence and distribution of solutions indicate a more effective search capability compared to NSGA. These findings confirm that NSGA-II provides a more robust balance between exploration and exploitation compared to NSGA, especially for medium and large problem instances.

Method	Problem name	Index					
		NPF	MID	SM	MSI	HV	CPU-time (s)
AUGMECON	PN1	4	12,465	11.24	86.72	99,678	2820
	PN2	9	46,789	14.17	169.8	803,456	5622
	PN3	12	73,800	23.36	189.14	2,857,218	12,418
	PN4	–	–	–	–	–	> 14,400
	PN5	–	–	–	–	–	> 14,400
	PN6	–	–	–	–	–	> 14,400
	PN7	–	–	–	–	–	> 14,400
	PN8	–	–	–	–	–	> 14,400
NSGAII	PN1	4	13,106*	11.78*	84.37*	98,943*	432
	PN2	10*	48,119*	14.96	168.5*	751,230*	479
	PN3	15*	73,812	27.13*	202.68	2,567,819	873
	PN4	15	187,469	27.56*	293.81*	7,150,451*	924*
	PN5	19*	240,650*	30.04*	410.72	12,453,733	930*
	PN6	20*	287,904	30.45	453.29*	17,903,461*	1212
	PN7	20*	355,923	35.62*	481.55	24,871,645*	2620
	PN8	22	376,812*	37.11*	549.34*	32,678,902*	3340
NSGA	PN1	4	13,843	11.79	79.57	97,231	209*
	PN2	9	51,380	14.95*	168	714,560	389*
	PN3	12	70,189*	28.93	205.11*	2,690,159*	812*
	PN4	16*	187,426*	30.14	290.18	6,589,302	1080
	PN5	16	254,382	30.08	415.33*	14,678,939*	1020
	PN6	19	287,616*	30.41*	448.17	15,892,456	1119*
	PN7	19	353,248*	38.61	482.01*	20,568,112	1912*
	PN8	25*	378,015	38.87	545.22	26,451,743	2918*

Table 12. The comparison of AUGMECON, NSGAII and NSGA based on NPF, SM, MSI, HV, CPU-time. *The superior solution results.

Method	NPF	MID	SM	MSI	HV	CPU-time (s)
NSGAII	15.625	197,974	26	330	12,309,523	1351
NSGA	15	199,512	27	329	10,960,312	1182
Deviation	0.625	− 1538	− 1	1	1,349,211	169
Percentage deviation	4%	− 0.7%	− 4%	0.4%	12%	14%

Table 13. The percentage improvement for each performance metric for NSGAII versus NSGA.

It should be noted that for PN4-PN8 problems, the AUGMECON method does not reach a feasible solution in a reasonable time. Therefore, it is necessary to use the NSGAII or NSGA method for problems with larger dimensions. The Pareto frontier for PN1 by AUGMECON, NSGAII and NSGA is depicted in Fig. 12. Also, the Pareto frontier for PN4 which has been solved by NSGAII and NSGA is presented in Fig. 13. Figure 14 illustrates the differences in CPU-time among AUGMECON, NSGA, and NSGA-II. Both NSGA and NSGA-II exhibit significantly lower CPU-time compared to AUGMECON. Furthermore, NSGA demonstrates a lower average CPU-time than NSGA-II, indicating its superiority in terms of computational efficiency for this metric. Finally, the Gantt chart of jobs on machines for a point on the Pareto frontier of PN4 (Inside the red circle in Fig. 13) has been depicted in Fig. 15.

Also, to investigate the effect of learning on the objective function values and to analyze both the means and standard deviations, Table 14 has been created. As shown, the objective function values without learning are generally higher than those with learning. The table compares the performance of the algorithms both with and without the learning effect, highlighting the improvements in solution quality due to the learning effect, while also considering the variability in results across multiple runs, as indicated by the standard deviations. It should be noted that GAMS was unable to obtain feasible optimal solutions for problems PN4–PN8 within the computational limit of 14,400 s; therefore, these instances are omitted from Table 14 for the AUGMECON method. Furthermore, since GAMS was executed only once for each problem, the standard deviation for the objective values in PN1–PN3 is zero. Briefly, Table 14 presents a detailed comparison of the objective function values in two scenarios: with and without the learning effect. The results clearly show that the "without learning" values for both makespan and total energy consumption are consistently higher across all problem sizes. This confirms that incorporating the learning effect leads to a measurable reduction in processing times and energy

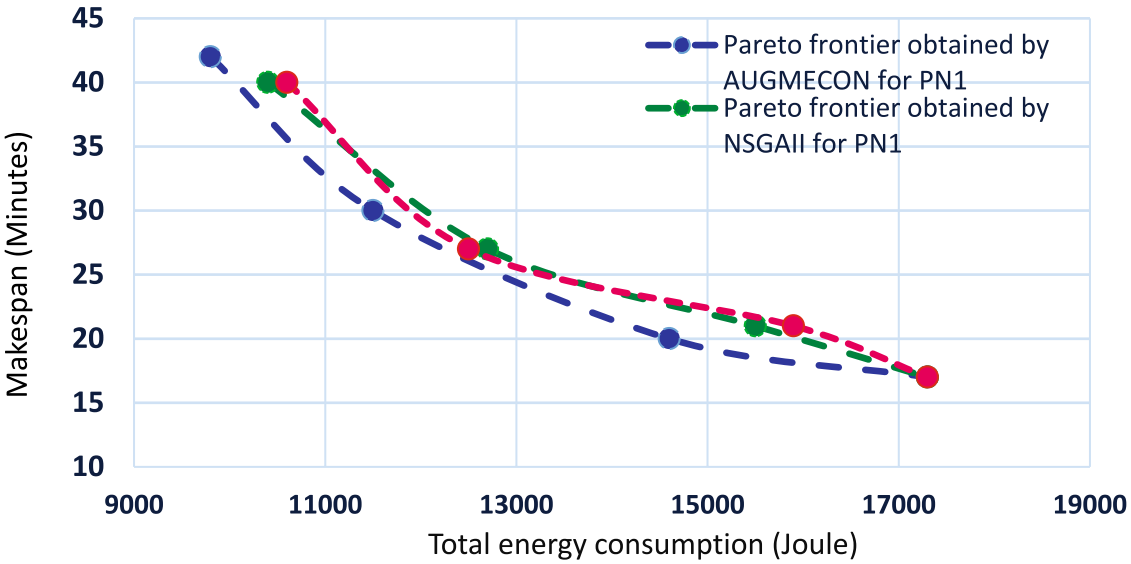

Fig. 12. Pareto frontiers obtained by AUGMECON, NSGA-II, and NSGA for test problem PN1. The figure compares the nondominated solutions in terms of makespan and total energy consumption, illustrating how each algorithm explores the trade-off between the two objectives. AUGMECON provides a limited but accurate set of points, whereas NSGA-II shows better diversity and coverage of the search space compared to NSGA.

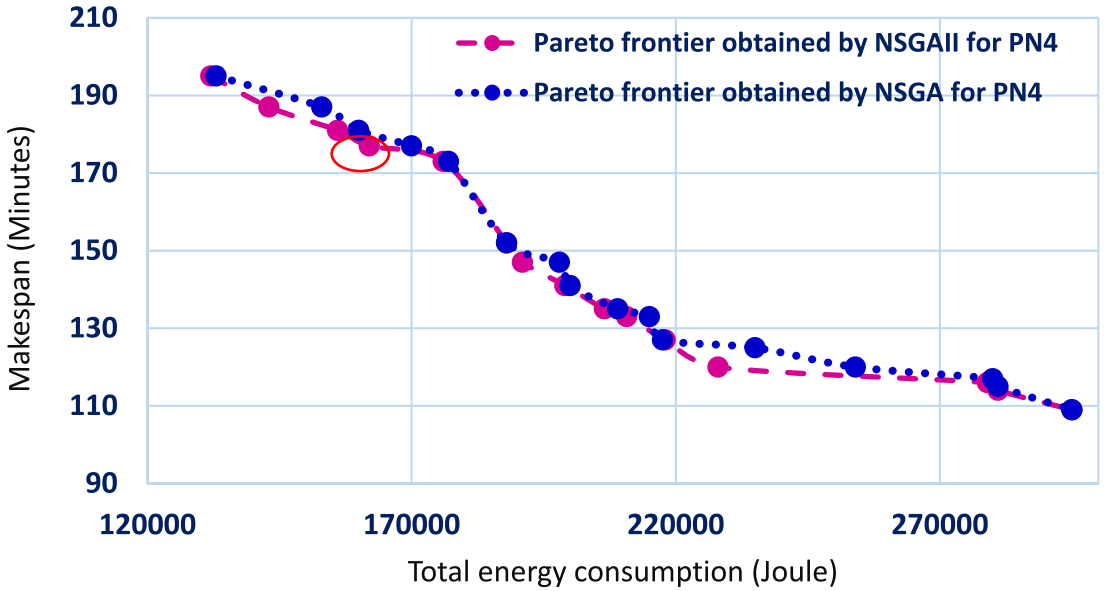

Fig. 13. Pareto frontiers generated by NSGA-II and NSGA for the large-sized instance PN4. As AUGMECON fails to produce feasible solutions for this case, only the evolutionary algorithms are compared. The figure highlights the difference in the distribution, spread, and convergence of the nondominated solutions, with NSGA-II demonstrating superior diversity and a broader exploration of the objective space.

usage, producing more efficient schedules. Additionally, the reported standard deviation values for NSGA-II and NSGA provide insight into the stability of these metaheuristic algorithms. Lower deviations indicate more consistent solutions over multiple runs, while higher deviations reveal increased variability. These observations reinforce the importance of both learning considerations and multi-run analysis in evaluating algorithm performance.

Conclusions

Todays, a practical production scheduling problem is to consider the scheduling and transportation simultaneously in flowshops. Furthermore, due to the importance of the energy issue, determining the minimum energy

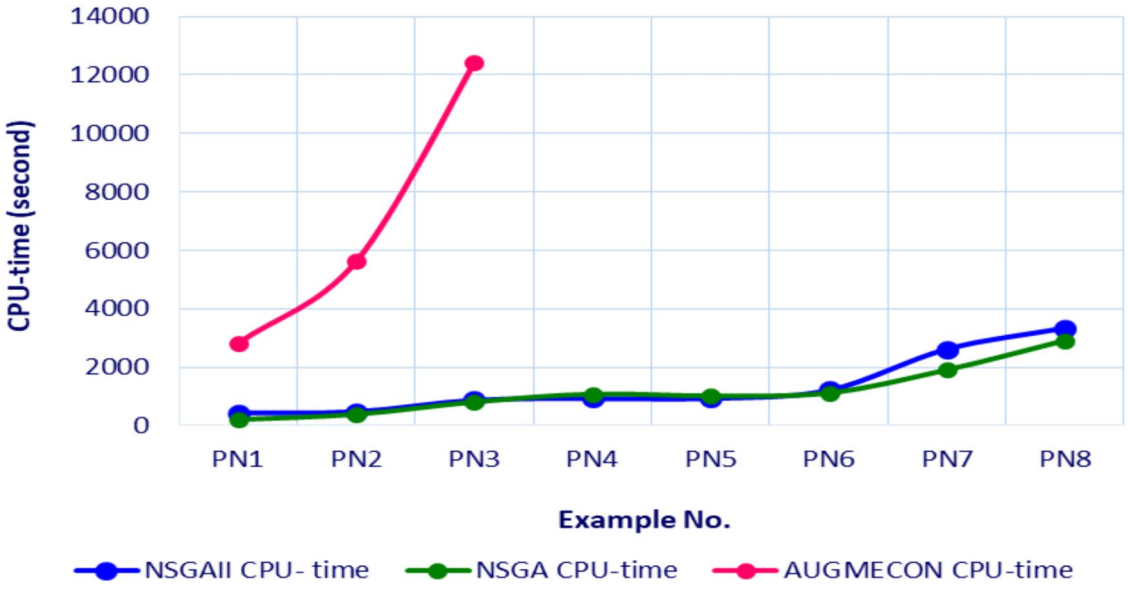

Fig. 14. CPU-time performance of AUGMECON, NSGA-II, and NSGA across all numerical test problems. The figure shows that AUGMECON requires substantially higher computation time, especially for medium and large problems, due to the complexity of solving the exact multi-objective model. In contrast, NSGA-II and NSGA offer much faster execution, making them more practical for larger instances.

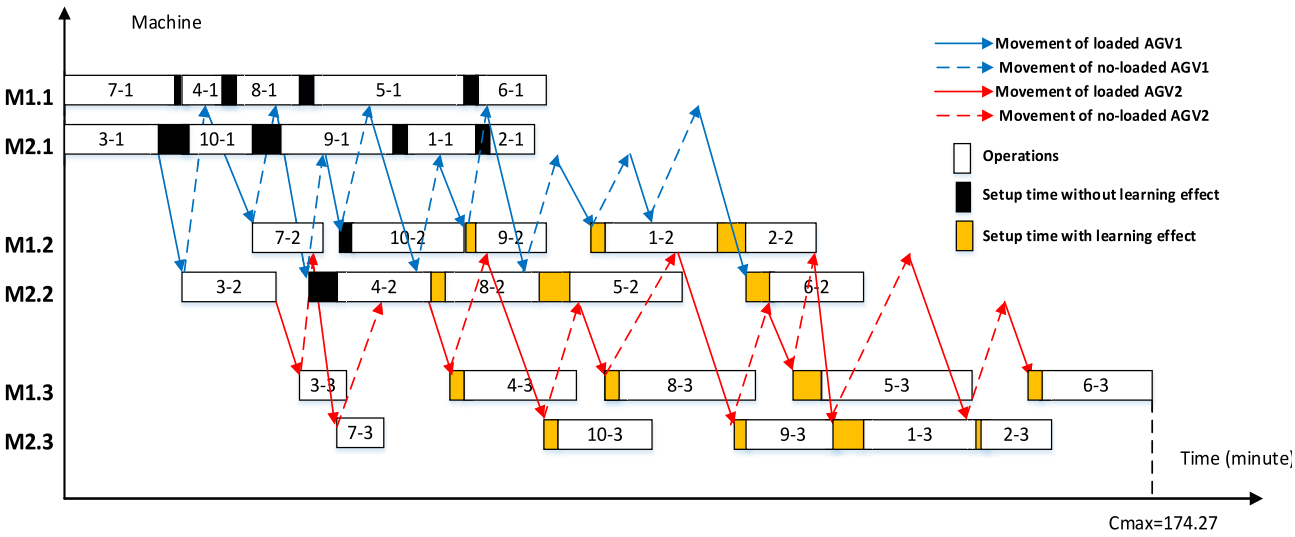

Fig. 15. The obtained Gantt chart of one of points of optimal solution frontier for PN4.

consumption in scheduling has also become a challenge problem. For these reasons, we addressed an energy-efficient scheduling in a flexible flowshop with AGV transportation constraints. A bi-objective optimization model is formulated to minimize both makespan and total energy consumption by machines and AGVs. Because the real environment is uncertain, parameters such as the processing time as well as the learning effect coefficient have been considered as triangular fuzzy numbers. To solve the proposed model, an AUGMECON for reaching to Pareto optimal solution in small-sized examples and a NSGAII to reach the Pareto frontier in a more reasonable time and for larger examples (with higher dimensions) are developed. The computational results clearly demonstrate the superiority of the proposed NSGA-II algorithm over the classical NSGA and AUGMECON methods across most performance metrics. While the AUGMECON approach was able to generate high-quality Pareto solutions for small-sized problems, it failed to obtain feasible solutions within a reasonable computation time for medium and large instances. In contrast, NSGA-II consistently achieved better solution diversity, wider Pareto spread, and higher hypervolume values, indicating a more comprehensive approximation of the Pareto front. Moreover, NSGA-II outperformed NSGA in terms of the number of nondominated solutions and spacing metrics in most cases, confirming its robustness and effectiveness in handling large-scale fuzzy

Method	Problem name	With learning effect				Without learning effect			
		Average total energy (Joule)	Standard deviation for total energy (Joule)	Average makespan (Minute)	Standard deviation for makespan (Minute)	Average total energy (Joule)	Standard deviation for total energy (Joule)	Average makespan (Minute)	Standard deviation for makespan (Minute)
AUGMECON	PN1	16,972	0	49.94	0	16,981	0	51.73	0
	PN2	24,541	0	76.34	0	25,717	0	79.14	0
	PN3	28,390	0	84.05	0	33,711	0	89.56	0
NSGAII	PN1	16,992	1.8	50.41	0.2	16,996	2.9	53.42	0.4
	PN2	25,124	3.7	79.23	1.1	25,863	5.1	84.18	1.2
	PN3	28,563	7.8	88.00	1.4	34,023	11.2	92.12	0.9
	PN4	43,696	13.5	120.11	0.8	47,551	21.5	127.45	1.7
	PN5	85,329	15.8	194.92	2.3	91,529	10.5	201.17	0.5
	PN6	105,812	0.2	196.89	0.9	119,234	14.3	200	0.3
	PN7	120,400	21.3	207.14	0.5	127,348	6.4	215.55	1.1
	PN8	123,018	12.9	210.89	1.3	132,840	19.3	215.78	0.8
NSGA	PN1	16,997	0.89	50.5	0.11	17,001	1.5	53.35	0.85
	PN2	25,456	2.5	77.12	1.4	27,523	2.4	84.18	1.0
	PN3	28,940	5.5	87.18	0.5	34,212	13.9	93.12	1.2
	PN4	44,080	14.2	122.56	0.2	47,000	8.5	130.18	2.5
	PN5	85,700	5.5	195.00	1.6	92,900	2.1	203.11	1.1
	PN6	105,815	0.4	196.90	1.0	120,459	6.5	199.17	0.2
	PN7	121,205	16.6	210.06	0.9	127,418	8.8	215.00	1.3
	PN8	123,309	7.9	212.55	1.0	134,592	12.7	214.12	0.2

Table 14. The average and standard deviation of two objective function values in two scenarios (with learning effect and without learning effect) for EUGMECON run in GAMS for one time, and NSGAII and NSGA run in MATLAB with three times.

EEFFSP with AGV constraints and learning effects. These findings validate that NSGA-II is a more reliable and scalable solution approach for complex production scheduling scenarios where both energy efficiency and makespan must be optimized simultaneously.

The contributions of this paper are 1) Developing the simultaneous scheduling model together with various types of energy in process and transportation mode 2) Considering the learning effect on sequence-dependent setup time in this problem 3) Considering the two parameters of processing time and learning effect coefficient in a fuzzy number shape 4) Developing AUGMECON and NSGAII in order to solve the proposed model.

From a managerial standpoint, the proposed model provides several actionable insights for energy-aware and efficient shopfloor operations. First, it enables production planners to make informed decisions by simultaneously considering job sequencing and transportation under realistic constraints, thereby reducing both makespan and energy usage. Second, by incorporating fuzzy processing times and learning effects, managers can develop more resilient schedules that adapt to operational uncertainties. Third, the energy breakdown allows managers to identify which components contribute most to consumption and where to target energy-saving strategies. Finally, by integrating learning effects into setup times, factories can take advantage of worker or machine experience over time, leading to gradual performance improvement and reduced setup energy.\

For future research, the proposed model can be extended by considering limited or zero buffer capacities to better reflect practical production constraints. Moreover, exploring environments with multiple AGVs between stages would enhance the realism of the transportation system and further challenge coordination and scheduling mechanisms. We assume a single AGV to keep the model manageable and focus on energy and learning effects. Multi-AGV extensions are left for future research. Finally, validating the model using real production data and advanced digital technologies, such as digital twins or simulation-based platforms, would strengthen its industrial relevance and practical applicability.

Data availability

Data is available on request from the authors.

Received: 14 August 2025; Accepted: 10 December 2025
Published online: 15 December 2025

References

1. Liu, Y., Dong, H., Lohse, N., Petrovic, S. & Gindy, N. An investigation into minimising total energy consumption and total weighted tardiness in job shops. *J. Clean. Prod.* **65**, 87–96 (2014).
2. Liu, Z. et al. The mixed production mode considering continuous and intermittent processing for an energy-efficient hybrid flowshop scheduling. *J. Clean. Prod.* **246**, 119071 (2020).

3. Hasani, A. & Hosseini, S. M. H. A bi-objective flexible flowshop scheduling problem with machine-dependent processing stages: Trade-off between production costs and energy consumption. *Appl. Math. Comput.* **386**, 125533 (2020).
4. Li, J.-Q., Sang, H.-Y., Han, Y.-Y., Wang, C.-G. & Gao, K.-Z. Efficient multi-objective optimization algorithm for hybrid flowshop scheduling problems with setup energy consumptions. *J. Clean. Prod.* **181**, 584–598 (2018).
5. Jiang, E., Wang, L. & Wang, J. Decomposition-based multi-objective optimization for energy-aware distributed hybrid flowshop scheduling with multiprocessor tasks. *Tsinghua Sci. Technol.* **26**(5), 646–663 (2021).
6. Dai, M., Tang, D., Giret, A., Salido, M. A. & Li, W. D. Energy-efficient scheduling for a flexible flowshop using an improved genetic-simulated annealing algorithm. *Robot. Comput. Integr. Manuf.* **29**(5), 418–429 (2013).
7. Tang, D., Dai, M., Salido, M. A. & Giret, A. Energy-efficient dynamic scheduling for a flexible flowshop using an improved particle swarm optimization. *Comput. Ind.* **81**, 82–95 (2016).
8. Zeng, Z., Gao, Le. & Hong, Z. Energy-aware scheduling of two-stage flexible flowshop based on group technology: consideration of the inconsistency of process differences. *Int. J. Comput. Intell. Syst.* **15**(1), 60 (2022).
9. Meng, L., Zhang, C., Shao, X., Ren, Y. & Ren, C. Mathematical modelling and optimisation of energy-conscious hybrid flowshop scheduling problem with unrelated parallel machines. *Int. J. Prod. Res.* **57**(4), 1119–1145 (2019).
10. Wang, S., Wang, X., Chu, F. & Jianbo, Yu. An energy-efficient two-stage hybrid flowshop scheduling problem in a glass production. *Int. J. Prod. Res.* **58**(8), 2283–2314 (2020).
11. Han, Z., Han, C., Lin, S., Dong, X. & Shi, H. Flexible flowshop scheduling method with public buffer. *Processes* **7**(10), 681 (2019).
12. Han, Y. et al. Discrete evolutionary multi-objective optimization for energy-efficient blocking flowshop scheduling with setup time. *Appl. Soft Comput.* **93**, 106343 (2020).
13. Wang, Y., Jia, Z.-H. & Zhang, X.-y. A hybrid meta-heuristic for the flexible flowshop scheduling with blocking. *Swarm Evol. Comput.* **75**, 101195 (2022).
14. Luo, J., Fujimura, S., El Baz, D. & Plazolles, B. GPU based parallel genetic algorithm for solving an energy efficient dynamic flexible flowshop scheduling problem. *J. Parallel Distrib. Comput.* **133**, 244–257 (2019).
15. Bruzzone, A. A. G., Anghinolfi, D., Paolucci, M. & Tonelli, F. Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flowshops. *CIRP Ann.* **61**(1), 459–462 (2012).
16. Jiang, S.-L. & Zhang, L. Energy-oriented scheduling for hybrid flowshop with limited buffers through efficient multi-objective optimization. *IEEE Access* **7**, 34477–34487 (2019).
17. Gong, G. et al. Energy-efficient flexible flowshop scheduling with worker flexibility. *Expert Syst. Appl.* **141**, 112902 (2020).
18. Li, M., Lei, D. & Cai, J. Two-level imperialist competitive algorithm for energy-efficient hybrid flowshop scheduling problem with relative importance of objectives. *Swarm Evol. Comput.* **49**, 34–43 (2019).
19. Zeng, Z. et al. Multi-object optimization of flexible flowshop scheduling with batch process—consideration total electricity consumption and material wastage. *J. Clean. Prod.* **183**, 925–939 (2018).
20. Liu, Y., Farnsworth, M. & Tiwari, A. Energy-efficient scheduling of flexible flowshop of composite recycling. *Int. J. Adv. Manuf. Technol.* **97**, 117–127 (2018).
21. Yin, R., Chaowen, L., Xuqing, F., & Qingbin, Y. Flexible Flowshop Scheduling and Energy Saving Optimization Strategy under Low Carbon Target. P. 12095 in *Journal of Physics: Conference Series*. Vol. 1939. IOP Publishing (2021).
22. Wu, X., Shen, X. & Cui, Qi. Multi-objective flexible flowshop scheduling problem considering variable processing time due to renewable energy. *Sustainability* **10**(3), 841 (2018).
23. Yan, J., Lin Li, Fu., Zhao, F. Z. & Zhao, Q. A multi-level optimization approach for energy-efficient flexible flowshop scheduling. *J. Clean. Prod.* **137**, 1543–1552 (2016).
24. Zhou, R. U. I., Lei, D. & Zhou, X. Multi-objective energy-efficient interval scheduling in hybrid flowshop using imperialist competitive algorithm. *IEEE Access* **7**, 85029–85041 (2019).
25. Zhou, B. & Liu, W. Energy-efficient multi-objective scheduling algorithm for hybrid flowshop with fuzzy processing time. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **233**(10), 1282–1297 (2019).
26. Zabihzadeh, S. S. & Rezaeian, J. Two meta-heuristic algorithms for flexible flowshop scheduling problem with robotic transportation and release time. *Appl. Soft Comput.* **40**, 319–330 (2016).
27. Hidri, L., Elkosantini, S. & Mabkhot, M. M. Exact and heuristic procedures for the two-center hybrid flowshop scheduling problem with transportation times. *IEEE Access* **6**, 21788–21801 (2018).
28. Zhang, H.-Y. et al. Performance analysis of a flexible flowshop with random and state-dependent batch transport. *Int. J. Prod. Res.* **59**(4), 982–1002 (2021).
29. Lei, C., Zhao, N., Ye, S. & Xiuli, Wu. Memetic algorithm for solving flexible flow-shop scheduling problems with dynamic transport waiting times. *Comput. Ind. Eng.* **139**, 105984 (2020).
30. Liou, C.-D. & Hsieh, Y.-C. A hybrid algorithm for the multi-stage flowshop group scheduling with sequence-dependent setup and transportation times. *Int. J. Prod. Econ.* **170**, 258–267 (2015).
31. Li, W., Han, D., Gao, L., Li, X. & Li, Y. Integrated production and transportation scheduling method in hybrid flowshop. *Chin. J. Mech. Eng.* **35**(1), 1–20 (2022).
32. Gheisariha, E., Tavana, M., Jolai, F. & Rabiee, M. A simulation-optimization model for solving flexible flowshop scheduling problems with rework and transportation. *Math. Comput. Simul.* **180**, 152–178 (2021).
33. Hidri, L. & Elsherbeeny, A. M. Optimal solution to the two-stage hybrid flowshop scheduling problem with removal and transportation times. *Symmetry* **14**(7), 1424 (2022).
34. Biskup, D. A state-of-the-art review on scheduling with learning effects. *Eur. J. Oper. Res.* **188**(2), 315–329 (2008).
35. Seidgar, H., Kiani, M., Abedi, M. & Fazlollahtabar, H. An efficient imperialist competitive algorithm for scheduling in the two-stage assembly flowshop problem. *Int. J. Prod. Res.* **52**(4), 1240–1256 (2014).
36. Esfeh, K., Meysam, A. A., Shojaie, H. J. & Khalili-Damghani, K. Flexible flowshop scheduling problem with reliable transporters and intermediate limited buffers via considering learning effects and budget constraint. *Complexity* **2022**, 1253336 (2022).
37. Pargar, F. & Zandieh, M. Bi-criteria SDST hybrid flowshop scheduling with learning effect of setup times: Water flow-like algorithm approach. *Int. J. Prod. Res.* **50**(10), 2609–2623 (2012).
38. Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F. & Mahmoodian, V. An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Comput. Ind. Eng.* **86**, 2–13 (2015).
39. Maghami, M. R. et al. Optimized planning of electric vehicle charging infrastructure for grid performance improvement. *Discover Sustain.* **6**(1), 706 (2025).
40. Geng, K., Liu, L. & Wu, S. A reinforcement learning based memetic algorithm for energy-efficient distributed two-stage flexible job shop scheduling problem. *Sci. Rep.* **14**(1), 30816 (2024).
41. Hu, D., Wu, Y., Qiu, L., Mi, J. & Yang, Y. Constructive-destructive neighbor search drives artificial bee colony algorithm for variable speed green hybrid flowshop scheduling problem. *Sci. Rep.* **15**(1), 9671 (2025).
42. Wang, D. et al. A mixed-integer linear programming model for addressing efficient flexible flow shop scheduling problem with automatic guided vehicles consideration. *Appl. Sci.* **15**(6), 2076–3417 (2025).
43. Tang, H., Huang, J., Ren, C., Shao, Y. & Lu, J. Integrated scheduling of multi-objective lot-streaming hybrid flowshop with AGV based on deep reinforcement learning. *Int. J. Prod. Res.* **63**(4), 1275–1303 (2025).
44. Yang, S., Meng, L., Ullah, S., Zhang, B., Sang, H. & Duan, P. MILP Modeling and Optimization of Multi-Objective Three-Stage Flexible Job Shop Scheduling Problem With Assembly and AGV Transportation. *IEEE Access* (2025).

45. Jiménez, M., Arenas, M., Bilbao, A. & Victoria Rodrı, M. Linear programming with fuzzy parameters: An interactive method resolution. *Eur. J. Oper. Res.* **177**(3), 1599–1609 (2007).
46. Mavrotas, G. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Appl. Math. Comput.* **213**(2), 455–465 (2009).
47. Deb, K. NSGA II paper by Kalyanmoy Deb. *IEEE Trans. Evoluti. Comput.* **6**(2), 182–197. https://doi.org/10.1109/4235.996017 (2002).

Author contributions
Saeed Dehnavi: conceptualization, visualization, software, methodology, analysis, writing, and editing. Hadi Mokhtari : conceptualization, supervision, software, visualization, analysis, and editing. Mohammad Taghi Rezvan : conceptualization, supervision, and writing.

Funding
The authors did not receive support from any organization for the submitted work.

Declarations

Competing interests
The authors declare no competing interests.

Human and animal rights
This article does not contain any studies with human participants or animals performed by any of the authors.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-32390-3.

Correspondence and requests for materials should be addressed to S.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.