Process Integration and Optimization for Sustainability
https://doi.org/10.1007/s41660-026-00705-0

ORIGINAL RESEARCH PAPER

Energy-Aware Flexible Flow-Shop Scheduling for Sustainable Manufacturing: A Multi-Objective Approach

Ali Mokhtari-Moghadam[1] · Trung Thanh Nguyen[1] · Massoud Mohsendokht[1]

Received: 11 September 2025 / Revised: 19 January 2026 / Accepted: 21 January 2026

Abstract

Traditional manufacturing scheduling often focuses on minimizing makespan, overlooking rising energy costs and sustainability pressures. Global energy demand is expected to grow by 11-18% by 2050, with electricity becoming the dominant source due to industrial electrification and Net Zero goals. UK manufacturing electricity prices have increased by up to 87% since 2019, highlighting the need for energy-aware production strategies to reduce costs and environmental impact. This study addresses flexible flow-shop scheduling problems (FFSPs) by developing an energy-aware approach that minimizes both total electricity consumption (TEC) and makespan. A multi-objective mathematical model is proposed, incorporating sequence-dependent setup times to reflect real-world complexity in energy-intensive manufacturing. To efficiently solve large-scale instances of the proposed model, a multi-objective genetic algorithm (MOGA) is developed and evaluated using a diverse set of randomly generated test problems covering different problem sizes and parameter settings. Its performance is compared against two well-known evolutionary algorithms, SPEA2 and PESA2, using three widely adopted multi-objective evaluation metrics. The results show that MOGA achieves more effective trade-offs between energy consumption and production time. By incorporating energy considerations into scheduling decisions, the proposed approach supports cost reduction, environmental sustainability, and improved operational robustness in modern manufacturing systems. Overall, the findings contribute to advancing green production scheduling and energy-aware smart factory development.

Keywords Sustainable manufacturing · Multi-objective optimization · Evolutionary algorithm · Energy-aware scheduling · Sequence-dependent setup times

Introduction

Global population growth and the rise in industrial activities, coupled with worsening environmental conditions, underscore the need for sustainable industrial development (Chamandoust et al. 2020; Mokhtari-Moghadam et al. 2023). By 2050, the world's energy demand is expected to increase by 11-18%, with electricity emerging as the primary energy source (McKinsey , Company 2024), driven by the electrification of industrial processes–an essential solution for achieving Net Zero emissions (Government 2024a). However, electricity prices for the UK manufacturing industry have surged by 87% since 2019 (Government 2024b), making energy efficiency and cost management crucial for long-term industrial sustainability. Among the various approaches to reducing energy consumption in manufacturing, such as investing in energy-efficient machinery, redesigning products, and optimizing production scheduling, an energy-efficient schedule has proven to be a fast and highly cost-effective solution, particularly for medium and small manufacturing companies (Dai et al. 2013).

Flexible flow-shop, also known as hybrid flow-shop, is one of the most common types of manufacturing environments, drawing significant attention from researchers. It combines traditional flow-shop scheduling with parallel

✉ Trung Thanh Nguyen
t.t.nguyen@ljmu.ac.uk

Ali Mokhtari-Moghadam
A.MokhtariMoghadam@2024.ljmu.ac.uk

Massoud Mohsendokht
m.mohsendokhtamlashi@2023.ljmu.ac.uk

[1] Liverpool Logistics, Offshore and Marine (LOOM) Research Institute, Liverpool John Moores University, Liverpool L3 3AF, United Kingdom

Published online: 10 February 2026

machine scheduling challenges, featuring at least two stages where jobs move through successive stages in a consistent direction. Machines within each stage can be identical, uniform, or unrelated (Duan et al. 2024). This type of production system is prevalent in industries such as chemicals, pharmaceuticals, oil, food, logistics, semiconductors, textiles, paper, and metallurgy (Zandieh et al. 2006). The flexible flow-shop system leverages its parallel machines to handle production line interruptions and respond swiftly to fluctuating market demands. An illustrative schematic of the flexible flow-shop configuration is provided in Fig. 1.

Over the past half-century, extensive efforts have been directed toward improving manufacturing shop productivity, with a primary focus on minimizing job completion time, total machine workload, and total tardiness (Mraihi et al. 2024). More recently, the emphasis has shifted toward energy-efficient production, which has emerged as a critical area of research across diverse shop floor environments (Han et al. 2020). Machine setup time is a crucial factor in real-world production environments that directly impacts efficiency. If not properly managed, it can lead to a reduction in machine utilization by up to 20% (Pinedo 2008). Additionally, production mix and sequence have a significant impact on makespan and machine setups, making production scheduling more complex.

The FFSP has been proven to be NP-hard, even in simplified scenarios like a two-stage flow shop with only two parallel machines in one stage (Ribas et al. 2010). This computational complexity renders exact solution methods impractical for addressing large-scale, real-case scenarios efficiently. As a result, there is a critical need for robust approaches that can deliver high-quality approximate solutions-either optimal or near-optimal schedules-within a reasonable computational time-frame. Metaheuristic algorithms are effective frameworks for solving NP-hard optimization problems (Roshanaei et al. 2010).

This paper contributes to research on FFSPs by considering energy efficiency alongside sequence-dependent setup times (SDST) in the scheduling model, with the goal of simultaneously minimizing makespan and TEC. To solve this problem efficiently, a MOGA framework is proposed. The main contributions of this study are:

(1) An energy-aware scheduling approach for FFSPs that balances production efficiency and electricity consumption.
(2) A multi-objective MILP model incorporating SDST to capture real-world operational complexity.
(3) A MOGA solution method that demonstrates superior performance over competitive, widely used multi-objective optimization algorithms in balancing energy consumption and makespan.
(4) Integration of production and energy-related constraints, including machine standby and setup energy.
(5) A framework supporting green production scheduling, cost reduction, and operational resilience in energy-intensive manufacturing.

The remainder of the paper is organized as follows: Section 2 provides a brief review of recent related works, while Section 3 presents the problem description and mathematical model. Section 4 details the development of the MOGA framework, followed by the computational results in Section 5. Finally, Section 6 concludes the paper and outlines future research directions.

Related Work

This section reviews existing studies on flexible and hybrid flow-shop scheduling problems (FFSPs/HFSPs), with particular attention to problem characteristics, optimization techniques, and identified research gaps. It also highlights the limitations of current studies and clarifies the motivation for the present research.

Driven by growing industrial demands, researchers have extended the classical flow shop problem to accommodate diverse shop floor configurations and operational constraints, with a strong emphasis on improving production efficiency. Comprehensive reviews of production efficiency

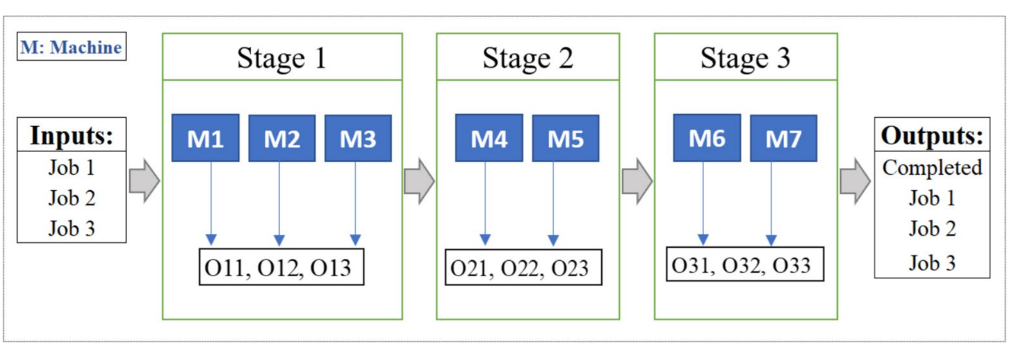


Fig. 1 Illustration of a flexible flow-shop environment

in FFSPs–covering solution approaches, production systems, environmental assumptions, system constraints, and objective functions–can be found in Ruiz and Vázquez-rodríguez (2010); Ribas et al. (2010); Lee and Loong (2019). However, research on energy efficiency in FFSPs and HFSPs, while limited, is gaining increasing attention. From a managerial perspective, optimizing production schedules can significantly reduce energy consumption in manufacturing industries (Dai et al. 2019). Energy-efficient scheduling was first introduced by Mouzon et al. (2007), who aimed to minimize both makespan and energy consumption through the application of various methods, including a machine turn-off and turn-on strategy. Subsequently, Mouzon , Yildirim (2008) expanded this focus to single-machine environments, developing a framework to simultaneously minimize TEC and total tardiness (TT). The research on energy-efficient FFSPs and HFSPs has seen significant advancements over the years.

Early energy-aware manufacturing studies mainly used traditional multi-objective algorithms to balance energy consumption and makespan in shop-floor scheduling. Dai et al. (2013) proposed an improved hybrid genetic algorithm and simulated annealing (GA-SA) framework targeting the minimization of makespan and energy consumption. Similarly, Yan et al. (2016) proposed a genetic algorithm (GA) for minimizing makespan and TEC in FFSP. Tang et al. (2016) addressed dynamic scheduling in a FFSP, focusing on reducing both makespan and energy consumption using an improved particle swarm optimization (PSO) method.

Subsequent research extended the modeling framework by incorporating additional shop-floor configurations–such as human resources alongside machines–thereby providing a more realistic representation of practical manufacturing environments. Building on this perspective, Gong et al. (2020) investigated energy-efficient scheduling in FFSPs with worker flexibility and proposed a hybrid evolutionary algorithm (HEA) to optimize three objectives: makespan, total worker cost (TWC), and green production indicators (GPI).

Furthermore, recognizing that electricity prices fluctuate throughout the day, several studies have integrated Time-of-Use (ToU) electricity pricing into FFSP and HFSP scheduling models. Zhang et al. (2019) addressed a FFSP under ToU pricing by developing an improved Strength Pareto Evolutionary Algorithm II (SPEA2) to simultaneously reduce makespan and total electricity cost (TCE). For a two-stage hybrid flow shop in glass production, Wang et al. (2020) formulated a mixed-integer programming (MIP) model and employed an augmented ϵ-constraint method to analyze the trade-off between makespan and total energy consumption (TEC) under ToU pricing. They further proposed several constructive heuristics and metaheuristic approaches, including tabu search (TS) and ant colony optimization (ACO). Similarly, Mhanna et al. (2024) presented a bi-objective MILP model for an FFSP under ToU pricing, aiming to minimize makespan and TEC, which was solved using the ϵ-constraint method combined with the CPLEX solver. More recently, an integrated approach that simultaneously considers ToU electricity pricing and worker flexibility as dual resources in an HFSP was investigated by Mokhtari-Moghadam et al. (2023). The authors proposed an adjusted multi-objective genetic algorithm (AMOGA) to minimize makespan and TEC.

Other practical constraints and hybrid solution approaches have also been explored to address real-world shop-floor scheduling challenges. A blocking HFSP was studied by Qin et al. (2022), who employed an improved iterated greedy algorithm (IGA) to minimize TEC. In a sheet metal manufacturing context, Ge et al. (2025) applied a teaching-learning-based optimization (TLBO) algorithm to an HFSP with machine preventive maintenance (PM), targeting the minimization of makespan and TEC. Chen et al. (2025) investigated hybrid flow-shop scheduling in printed circuit board assembly and proposed a multi-objective hyper-heuristic based on a two-stage improved spider monkey optimization (TS-ISMO), aiming to optimize makespan, TEC, and total tardiness (TT), while also incorporating lot streaming. Yue et al. (2025) proposed a hybrid variable neighborhood search and Non-dominated genetic algorithm II (VNS-NSGA-II) framework for an HFSP with transportation constraints in panel furniture manufacturing, aiming to minimize makespan and TEC. For an integrated flexible flow shop and parallel batch machine scheduling problem in casting and steelmaking, Wang et al. (2024) developed a feedback-based artificial bee colony (ABC) algorithm that simultaneously minimizes makespan, TEC, and TT. Wu and Cao (2022) proposed a decomposition-based multi-objective evolutionary algorithm (MOEA/D) to address the re-entrant hybrid flow shop scheduling problem for cold-drawn seamless steel pipes. The algorithm integrates batch processing, buffer sizing, and energy considerations, aiming to minimize both makespan and machine energy consumption. Multi-population evolutionary algorithms have been widely applied to energy-efficient HFSPs. hong Jia et al. (2026) addressed a two-stage hybrid flow shop with parallel batch machines under ToU pricing by proposing an adaptive multi-population cooperative evolutionary algorithm (AMCEA) to minimize makespan and TEC. Similarly, Wang et al. (2026) developed a multi-population co-evolutionary algorithm (MOMPCEA) for energy-efficient HFSPs with the same objectives.

Recently, scholars have increasingly incorporated learning-based algorithms to address energy-aware FFSPs and HFSPs. Li et al. (2023) embedded a Q-learning-driven GVNS into an NSGA-II framework to solve HFSPs with the objectives of minimizing total tardiness, total energy cost, and total carbon cost. Zhao et al. (2026) investigated a distributed heterogeneous flexible flow shop problem (DHFSP) with lot streaming, considering release times, SDST, and transport times. They proposed a learning-based co-evolution framework (LBCOF) to minimize makespan and TEC, integrating global–local population search, heuristic initialization, knowledge-driven operators, and a dueling deep Q-network (DDQN) for adaptive operator selection, supported by energy-saving strategies. He et al. (2026) introduced a learning-guided multi-objective evolutionary algorithm (LgMOEA) for HFSPs with limited buffers, aiming to minimize total weighted tardiness and non-processing energy cost. In a related study, Zhang et al. (2026) developed a Q-learning-based multi-objective hyper-heuristic algorithm (QLMHHA) for an energy-efficient integrated DHFSP with preventive maintenance, targeting reductions in makespan and total carbon emissions (TCE). Finally, Yuan et al. (2025) proposed a multi-agent double deep Q-network (MADDQN) for the DHHFSP to minimize TEC and TT, while Wu et al. (2025) presented a multi-objective evolutionary co-learning framework (MOECLF) for an HFSP featuring human–machine collaboration, with objectives of minimizing makespan and TEC.

The review of the existing literature highlights considerable efforts to bridge the gap between theoretical models and real-world manufacturing scenarios by incorporating practical factors to address shop floor challenges. Table 1 summarizes key studies in the field, including the reference, shop layout, problem settings, objectives, and solution approaches, while Table 2 provides a list of abbreviations for the problem settings and energy components. Although energy efficiency has been extensively studied in HFSPs, research on energy-efficient FFSPs remains relatively limited. Moreover, only a few studies have integrated SDST within their models.

Overall, previous studies have made extensive efforts to address manufacturing shop challenges and energy-aware scheduling. However, a clear research gap remains for FFSPs, particularly those incorporating SDST. This study is motivated by this gap and aims to formulate a bi-objective, energy-aware MILP model for FFSPs that considers SDST, and to propose a MOGA framework that balances makespan and energy consumption while integrating key production and energy-related shop constraints.

Problem Statement and Formulation

This section extends the formulation developed by Gong et al. (2020), incorporating key energy efficiency factors and sequence-dependent setup time constraint into the multi-objective FFSP.

Problem Description and Assumptions

The energy-aware FFSP with SDST can be described as follows: A set of n jobs passes sequentially through $s > 1$ stages, where each stage comprises $m > 1$ identical parallel machines with varying power consumption rates. Machines function in three distinct states: processing, setup, and idle, with each state consuming energy at a different rate. An energy-efficient schedule arranges job operations within a given time frame to simultaneously optimize both objectives, ensuring feasibility by adhering to the following constraints:

(1) All jobs are ready for processing at t=0 for the first stage.
(2) Each job can only be processed by a single machine at any given stage.
(3) While operations of the same job follow a precedence order, there are no precedence constraints between operations of different jobs.
(4) A machine can handle only one operation at a time.
(5) Job processing is non-preemptive, meaning interruptions are not allowed once processing begins.
(6) At each stage, all identical machines have uniform processing times for the same operation.
(7) Processing times for operations and sequence-dependent setup times between operations are predefined.
(8) The energy consumption parameters for processing operations, sequence-dependent setup activities, and machine idle states are assumed to be known in advance.

Mathematical Formulation

The mathematical model of energy-efficient scheduling for a FFS with SDST can be described as follows:

Sets and Indices i, j Job indices, $i, j = 1, \ldots, n_b$.
m Machine index, $m = 1, \ldots, n_m$.
s Stage index, $s = 1, \ldots, n_s$.
k Position index of operations on machine m at stage s, $k = 1, \ldots, n_l$.
$\mathcal{M}_s$ Set of available machines at stage s.
Parameters n_b Total number of jobs.
n_m Total number of machines.
n_s Total number of stages.
$P_{j,s,m}$ Processing time of job j on machine m at stage s

Table 1 A summary of related work on energy-efficient FFSP and HFSP

Reference	Shop layout	Problem settings	Energy components	Objectives	Solution approaches
Dai et al. (2013)	FFSP	G	I,P	$C_{\max}$, TEC	Hybrid GA-SA
Yan et al. (2016)	FFSP	G	I,P,ST,T	$C_{\max}$, TEC	GA
Tang et al. (2016)	FFSP	Pspeed, breakdown	I,P,ST	$C_{\max}$, TEC	Improved PSO
Gong et al. (2020)	FFSP	DR	P	$C_{\max}$, GPI,TWC	Hybrid EA
Zhang et al. (2019)	FFSP	ToU	I,P,ST	$C_{\max}$, TCE	Improved SPEA2
Wang et al. (2020)	HFSP	ToU	I,P	$C_{\max}$, TCE	AUGECON / Heuristic / Metaheuristic
Mhanna et al. (2024)	FFSP	ToU	I,P	$C_{\max}$, TEC	ϵ-constraint + CPLEX
Mokhtari-Moghadam et al. (2023)	HFSP	SDST, ToU, DR	I,P,SDST	$C_{\max}$, TEC	AMOGA
Qin et al. (2022)	HFSP	block	I,P,B	TEC	Improved IGA
Ge et al. (2025)	HFSP	Pspeed, pm	I,P	$C_{\max}$, TEC	TLBO
Chen et al. (2025)	HFSP	losm	I,P	$C_{\max}$, TEC,TT	Hyper-heuristic TS-ISMO
Yue et al. (2025)	HFSP	Tr, Pspeed	I,P,T,A,ST	$C_{\max}$, TEC	Hybrid VNS–NSGA-II
Wang et al. (2024)	FFSP + BPMSP	MBatch	P,T	$C_{\max}$, TEC,TT	Feedback-based ABC
Wu and Cao (2022)	HFSP + BPMSP	MBatch, Pretr	I,P	$C_{\max}$, TEC	Improved MOEA/D
hong Jia et al. (2026)	FFSP + BPMSP	MBatch, ToU, SDST	I,P	$C_{\max}$, TEC	AMCEA
Wang et al. (2026)	HFSP	Pspeed	I,P	$C_{\max}$, TEC	MOMP-CEA
Li et al. (2023)	HFSP	ToU	I,P,R	CTC, TECost, TT	QVNS–NSGA-II
Zhao et al. (2026)	DHFFSP	SDST, T, R, losm	I,P,T,R	$C_{\max}$, TEC	LBCOF
He et al. (2026)	HFSP	block	I	TWT, NPE	LgMOEA
Zhang et al. (2026)	DHFSP	Pspeed, pm	I,P	$C_{\max}$, TCE	QLMHHA
Yuan et al. (2025)	DHHFSP	Pspeed	I,P	TT, TCE	MADDQN
Wu et al. (2025)	HFSP	DR	I,P,ST	$C_{\max}$, TEC	MOECLF

Table 2 List of abbreviations related to the problem settings and energy components

Problem settings		Energy components	
G	General constraints	P	Processing energy
Mbatch	Batch scheduling	I	Idle energy
block	Limited buffers	ST	Setup/startup energy
breakdown	Machine breakdowns / availability	SDST	Sequence-dependent setup energy
DR	Dual resources / multi-skilled workers	SD	Shutdown energy
losm	Lot streaming / splitting / sizing	T	Transportation energy
Pretr	Re-entrant processing	R	Release energy
SDST	Sequence-dependent setup times	A	Auxiliary energy
Pspeed	Variable processing speed	B	Breakdown / recovery energy
ST	Setup times		
ToU	Time-of-use pricing		
pm	Preventive maintenance		
Tr	Transportation		

(minutes; convert to hours for energy calculations).

$st_{i,j,s,m}$ Sequence-dependent setup time when switching from job *i* to job *j* on machine *m* at stage *s* (minutes; convert to hours for energy calculations).

$PE_{j,s,m}$ Processing power consumption of machine *m* when processing job *j* at stage *s* (kW).

$SE_{i,j,s,m}$ Power consumption during setup from job *i* to job *j* on machine *m* at stage *s* (kW).

$IE_{s,m}$ Idle power consumption of machine *m* at stage *s* (kW).

M A sufficiently large positive constant.

Variables

$$x_{jsmk} = \begin{cases} 1, & \text{if job} j \text{is processed in the} k^{th} \text{position on machine} m \text{in stage s} \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha_{jsm} = \begin{cases} 1, & \text{if job} j \text{is processed on machine} m \text{in stage s} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{i,j,s,m} = \begin{cases} 1, & \text{if job} j \text{follows job} i \text{on machine} m \text{at stage} s \\ 0, & \text{otherwise} \end{cases}$$

$ST_{j,s}$ Start time of job j at stage s

$C_{j,s}$ Completion time of job j at stage s

$\tau_{i,j,s,m}$ Idle time on machine m between jobs i and j at stage s

$z_{i,j,s,m}$ Continuous auxiliary variable for linearizing $\tau_{i,j,s,m}\, y_{i,j,s,m}$

$C_{\max}$ Makespan of the schedule (completion time of the last job)

TEC Total electricity consumption of the system

Objective Functions

(1) Minimizing Makespan:

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

(2) Minimizing Total Electricity Consumption:

$$\min \quad f_2 = TEC \tag{2}$$

Subject to:

$$C_{\max} \geq C_{j,s} \quad \forall j = 1,\ldots,n_b,\ s = n_s \tag{3}$$

$$\sum_{j=1}^{n_b} x_{j,s,m,k} \leq 1 \qquad \forall s = 1,\ldots,n_s,\ m \in \mathcal{M}_s,\ k = 1,\ldots,n_l \tag{4}$$

$$\sum_{m\in\mathcal{M}_s} \sum_{k=1}^{n_l} x_{j,s,m,k} = 1 \qquad \forall j = 1,\ldots,n_b,\ s = 1,\ldots,n_s \tag{5}$$

$$C_{j,s} = ST_{j,s} + \sum_{m\in\mathcal{M}_s} p_{j,s,m}\,\alpha_{j,s,m} \qquad \forall j = 1,\ldots,n_b,\ s = 1,\ldots,n_s \tag{6}$$

$$ST_{j,s+1} \geq C_{j,s} \qquad \forall j = 1,\ldots,n_b,\ s = 1,\ldots,n_s - 1. \tag{7}$$

$$\begin{aligned} &ST_{j,s} \geq C_{i,s} + st_{i,j,s,m} - M\left(2 - x_{i,s,m,k} - x_{j,s,m,k+1}\right) \\ &\forall i,j = 1,\ldots,n_b,\ i \neq j,\ s,\ m \in \mathcal{M}_s,\ k = 1,\ldots,n_l - 1. \end{aligned} \tag{8}$$

$$TEC = TE_{setup} + TE_{process} + TE_{idle} \tag{9}$$

$$TE_{setup} = \sum_{i=1}^{n_b}\sum_{j=1}^{n_b}\sum_{s=1}^{n_s}\sum_{m\in\mathcal{M}_s} SE_{i,j,s,m}\cdot st_{i,j,s,m}\cdot y_{i,j,s,m}, \tag{10}$$

$$TE_{process} = \sum_{j=1}^{n_b}\sum_{s=1}^{n_s}\sum_{m\in\mathcal{M}_s} PE_{j,s,m}\cdot P_{j,s,m}\cdot \alpha_{j,s,m}, \tag{11}$$

$$TE_{\text{idle}} = \sum_{s=1}^{n_s}\sum_{m\in M_s}\sum_{i=1}^{n_b}\sum_{j=1}^{n_b} IE_{s,m}\,\tau_{i,j,s,m}\,y_{i,j,s,m}. \tag{12}$$

$$\tau_{i,j,s,m} \geq ST_{j,s} - C_{i,s} - st_{i,j,s,m}, \quad \forall i \neq j,\ s \geq 2,\ m \in \mathcal{M}_s. \tag{13}$$

$$\begin{aligned} x_{j,s,m,k},\ \alpha_{j,s,m},\ y_{i,j,s,m} \in \{0,1\} \\ \forall i,j = 1,\ldots,n_b,\ s = 1,\ldots,n_s,\ m \in \mathcal{M}_s,\ k = 1,\ldots,n_l. \end{aligned} \tag{14}$$

$$\begin{aligned} \tau_{i,j,s,m},\ z_{i,j,s,m} \geq 0 \\ \forall i,j = 1,\ldots,n_b,\ s = 1,\ldots,n_s,\ m \in \mathcal{M}_s,\ k = 1,\ldots,n_l. \end{aligned} \tag{15}$$

$$ST_{j,s},\ C_{j,s} \geq 0 \quad \forall j = 1,\ldots,n_b,\ s = 1,\ldots,n_s \tag{16}$$

$$C_{\max},\ TEC \geq 0 \tag{17}$$

The objective function (1) minimizes the makespan, i.e., the completion time of all jobs. The second objective function, (2), minimizes the TEC. Constraint (3) ensures that $C_{\max}$ is no smaller than the completion time of any job at the final stage. Constraint (4) imposes the capacity restriction that at most one job can be assigned to each position k of machine m at stage s. Constraint (5) guarantees that every job is processed exactly once at each stage by enforcing that each job is assigned to exactly one machine and one position per stage. Equation 6 defines the completion time of each job at each stage as the sum of its start time and its processing time on the assigned machine. Constraint (7) enforces technological precedence between stages by ensuring that a job cannot start stage $s+1$ before it finishes stage s. Constraint (8) incorporates sequence-dependent setup times. If job i precedes job j on the same machine in consecutive positions, the start time of job j must be at least the completion time of job i plus the required setup time from i to j. The big-M term deactivates this constraint when the two jobs are not adjacent on the same machine. Constraint (9) defines the TEC for the production system consists of three components: energy during machine setup, energy during job processing, and energy consumed while machines are idle. The first term of Constraint (9) represents the total energy consumed for setup, i.e., the energy required to transition machines from processing one job to another, as defined in Eq. 10. The second term of Constraint (9) accounts for the total energy used during the processing of jobs on machines, as given in Eq. 11. The third term of Constraint (9) calculates the total energy consumed by machines while idle, as detailed in Eq. 12. In Eq. 13, the idle time $\tau_{i,j,s,m}$ is defined as the difference between the available time gap,i.e., the time between the start of job j on machine m in stage $s \geq 2$ and the finish time of the previous job i on that machine, and the setup time required between job i and j. This represents the period during which the machine is idle, waiting for the job in the previous stage to complete. This ensures that $\tau_{i,j,s,m}$ captures the true idle periods of the machine, which contribute to the total idle energy consumption in Eq. 12. The domains of the variables are defined in constraints (14) - (17).

In Eq. 12 the term $\tau_{i,j,s,m}\,y_{i,j,s,m}$ is non-linear (continuous × binary). To obtain a linear MILP formulation, we introduce the auxiliary variable

$$z_{i,j,s,m} = \tau_{i,j,s,m}\,y_{i,j,s,m}, \qquad \forall i \neq j,\ s = 2,\ldots,n_s,\ m \in \mathcal{M}_s,$$

and enforce its equivalence with the following linear constraints (18) - (20)., where M is a valid upper bound of $\tau_{i,j,s,m}$ and $z_{i,j,s,m}$:

$$z_{i,j,s,m} \leq M.y_{i,j,s,m}, \quad \forall i \neq j,\ s \geq 2,\ m \in \mathcal{M}_s. \tag{18}$$

$$z_{i,j,s,m} \leq \tau_{i,j,s,m}, \quad \forall i \neq j,\ s \geq 2,\ m \in \mathcal{M}_s. \tag{19}$$

$$\tau_{i,j,s,m} \leq z_{i,j,s,m} + M.(1 - y_{i,j,s,m}), \quad \forall i \neq j,\ s \geq 2,\ m \in \mathcal{M}_s. \tag{20}$$

By substituting $\tau_{i,j,s,m}, y_{i,j,s,m}$ with $z_{i,j,s,m}$ in Eq. 12, constraint (21) represents the total idle energy.

$$TE_{\text{idle}} = \sum_{s=1}^{n_s} \sum_{m \in M_s} \sum_{i=1}^{n_b} \sum_{j=1}^{n_b} IE_{s,m}\, z_{i,j,s,m}. \tag{21}$$

Solution Approach

This section introduces a MOGA designed to solve the FFSP model with SDST. The pseudocode for the proposed MOGA is outlined in Algorithm 1, accompanied by a detailed discussion of its primary computational components. The explanation begins with the encoding and decoding scheme for solutions, followed by an in-depth description of the key steps involved in the algorithm.

Algorithm 1 Pseudo-code of MOGA

Input: Problem instance size, MOGA parameters
Output: A set of Pareto-optimal solutions
Begin
Initialization:
Set MOGA parameters: population size (n_{pop}), maximum iterations (`MaxIt`), crossover probability (`Pc`), and mutation probability (`Pm`).
Initialize the population of size n_{pop} and evaluate their fitness values.
Evaluate the fitness of the population using non-dominated sorting and crowding distance techniques.
Sort the population in ascending order based on ranks.
while the termination condition is not met **do**
  Crossover:
  if random probability $\mathtt{p} < \mathtt{Pc}$ **then**
    Select two parent individuals using the Roulette-wheel selection method.
    Apply the POX operator to generate offspring for the job sequence vector.
    Optimize machine selection using the machine selection algorithm (MSA).
    Evaluate the fitness of the offspring.
  end if
  Mutation:
  if random probability $\mathtt{p} < \mathtt{Pm}$ **then**
    Select one parent individual using the Roulette-wheel selection method.
    Apply a randomly selected mutation operator (e.g., swapping or insertion) to the job sequence vector.
    Optimize machine selection using the MSA algorithm.
    Evaluate the fitness of the mutated solution.
  end if
  Merge the initial population, offspring, and mutated solutions into a single group.
  Remove duplicate solutions with identical fitness values.
  Evaluate the fitness of the population using non-dominated sorting and crowding distance techniques.
  Sort the population in ascending order based on ranks.
  Select the top-ranked individuals using truncation selection to form the next generation.
  Store the Pareto-front solutions.
end while
Terminate

Algorithm 2 Machine Selection Algorithm (MSA)

Input: Job sequence vector, available machines in stage s, model parameters
Output: Index of the selected machine
Begin
Create a two-level array to record the $C_{j,s}$ and TEC for each machine in stage s.
for each available machine in stage s **do**
  Calculate $C_{j,s}$ using inequality (7).
  Calculate TEC using Eq. 2.
end for
Normalize the recorded $C_{j,s}$ and TEC.
Compute the Euclidean distance of each point ($C_{j,s}$, TEC) in the phenotype space from the ideal point (0, 0).
Identify the machine with the minimum distance.
Return the index of the selected machine.
Terminate

Encoding and Decoding of Solutions

For the energy-efficient FFSPs with SDST, a solution consists three-level chromosome with the length of $n_j \times n_s$. The first level represents the sequence of jobs for visiting multi stages, the second level is the machine assignment, and the third level is defined as the finish time of the job in the corresponding stage ($C_{j,s}$). Figure 2 presents the structure of three-level chromosome encoding for six jobs and three stages. The encoding and decoding procedure in this paper is as follows:

- **Step 1:** Initialize an empty three-level chromosome of length $n_b \times n_s$.
- **Step 2:** For the first stage, define the job sequence by permuting the job indices.
- **Step 3:** Starting with the first job in the sequence, select a machine from the available machines in stage s to process the job. The machine is chosen to minimize total energy consumption and makespan. Record the index of the selected machine using Algorithm 2.
- **Step 4:** Decode the partial encoding into a partial schedule, recording the completion time of the job and the index of selected machine.
- **Step 5:** Repeat Steps 2 and 3 for the remaining jobs in the current stage.
- **Step 6:** Sort jobs in ascending order of their finish time to define the sequence for the next stage.
- **Step 7:** Repeat Steps 3 to 6 for the remaining stages.
- **Step 8:** Decode the chromosome to obtain the final energy-efficient schedule.

Figure 3 depicts the optimal (or near-optimal) Gantt chart, representing the chromosome decoding of the

	Stage 1						Stage 2						Stage 3					
Job Sequence	2	3	4	5	1	6	3	2	4	5	1	6	3	2	4	5	1	6
Machine Selection	1	3	2	3	1	1	5	4	5	4	5	4	8	7	7	8	6	7
Finish Time	59	35	102	129	145	234	125	151	205	221	269	269	216	197	280	302	324	322

Fig. 2 Structure of three-level chromosome encoding

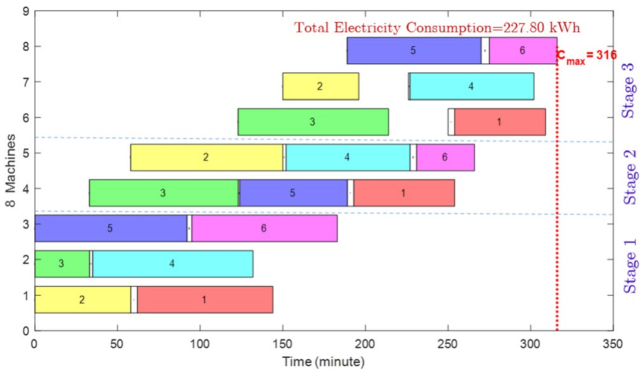


Fig. 3 Gantt chart (decoded) of energy-efficient schedule

energy-efficient schedule for the chromosome shown in Fig. 2. The job identifiers are indicated by the numbers within the bars.

Population Initialization and Fitness Function Evaluation

To ensure solution diversity, the job sequence for the first stage is randomly determined by permuting the jobs, followed by the application of the encoding-decoding approach to generate a complete solution. In this study, the non-dominated sorting (NDS) method, as proposed by Deb et al. (2002), is employed to evaluate each solution and assign a rank based on the dominance criterion. Solutions that are not dominated by others form the Pareto front.

In addition to the dominance-based technique that guides the solution set toward the Pareto-optimal front, ensuring a uniform distribution of solutions is crucial for any evolutionary algorithm. To achieve this, the crowding distance (CD) for all solution members within each rank is calculated. After classifying solutions into different fronts and computing their crowding distances, the solutions are sorted, and those with higher ranks (in this case, equivalent to the population size n_{pop} are selected to survive into the next generation of the algorithm. For a detailed description of the NDS and CD methods, refer to Mokhtari-Moghadam et al. (2023).

MOGA Operators

- Selection operator: The Roulette-wheel selection method (Holland 1975) is employed to choose individuals for the reproduction phase of the MOGA algorithm. This probabilistic selection mechanism ensures a balance between exploration (encouraging diversity) and exploitation (favoring high-quality solutions), guiding the evolution of the population towards optimal solutions.
- Crossover operator: In each generation, to generate offspring from parent solutions, the Preceding Order-Based Crossover (POX) method (Lee et al. 1998) is used. This operator preserves both the order of jobs and facilitates information sharing between parents. The procedure is as follows: first, two parents are selected through Roulette-wheel selection, designated as Parent 1 (P1) and Parent 2 (P2). A sub-job, referred to as sj_1, is randomly generated, while the remaining jobs form sj_2. Next, the jobs in sj_1 are copied from P1 to offspring 1 and from P2 to offspring 2. Finally, the jobs in sj_2 are copied from P2 to offspring 1 and from P1 to offspring 2. The application of the POX operator is depicted in Fig. 4.
- Mutation operator: To enhance population diversity and prevent premature convergence, a mutation operator is employed to generate offspring (Moghadam et al. 2014). In this study, two mutation operators, swap and insertion, are applied to the job sequence in the first stage. The process begins by selecting an individual from the population through Roulette-wheel selection, designated as Parent 1. One of the mutation operators is then randomly chosen with equal probability to produce an offspring, as shown in Fig. 5.

Population Update and Termination Criteria

In evolutionary algorithms, it is common to encounter identical solutions with the same fitness, which reduces diversity within the population and may lead to premature convergence. To address this, after combining the current

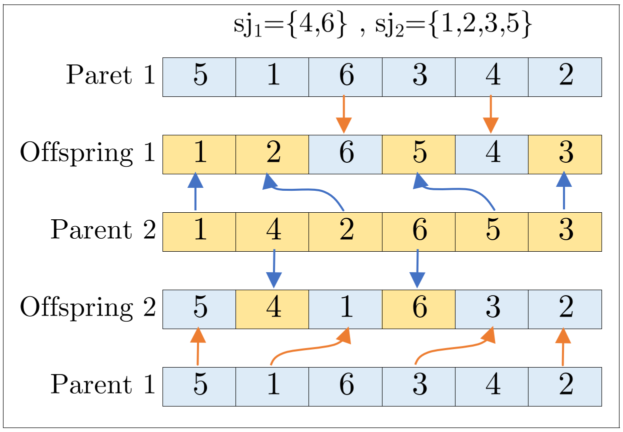


Fig. 4 POX crossover operator

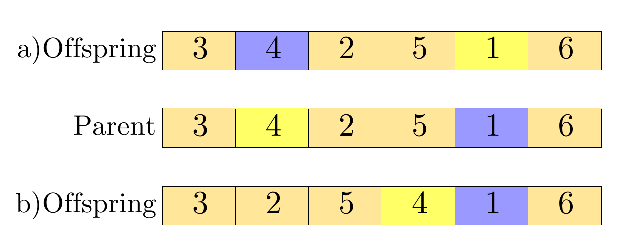


Fig. 5 Mutation operators: **a**)Swap and **b**)Insertion

population with new solutions generated from the crossover and mutation phases, individuals with identical fitness values (in terms of makespan and total energy consumption) are eliminated. Subsequently, the remaining solutions are ranked using non-dominated sorting and crowding distance techniques. The top-ranked individuals, up to a size of n_{pop}, are selected to form the new population for the next generation. The algorithm terminates once the maximum number of iterations (*nIter*) is reached.

Computational Experiments

Description on Test Algorithms

The performance of the proposed MOGA is evaluated by comparing the obtained solutions with those of the two state-of-the-art multi-objective evolutionary algorithms namely strength Pareto evolutionary algorithm (SPEA2) (Zitzler et al. 2001) and Pareto envelop-based selection algorithm (PESA2) (Corne et al. 2001). SPEA2 and PESA2 are selected for comparison as they have been widely used by researchers and also they use different approach towards Pareto front members. In comparison with MOGA that uses individual-base approach, SPEA2 and PESA2 use K-nearest neighbor(s) and region-based approaches respectively. The encoding-decoding procedure of the solutions in PSEA2 and PESA2 are the same as the proposed MOGA and also we used the same genetic operators for all three competing algorithms. It is notable that we only adjusted SPEA2 and PESA2 to address discrete problems and the procedures and operators followed what were implemented by Zitzler et al. (2001) and Corne et al. (2001). All the algorithms are coded in MATLAB R2020a and run on a personal computer with AMD Rayzen 3, 2.60 GHz CPU and 12 GB main memory.

Test Data Description

In this study, the test data used to evaluate the performance of the three competing algorithms are generated randomly and are available online (see Data Availability). This approach is adopted due to the challenge of obtaining real-world data that reflects the varying configurations in terms of both scale and structure across manufacturing shops. The factors influencing algorithm performance, along with their respective levels and ranges, are detailed in Table 3.

The test problems are presented in the format "number of jobs - number of stages - total number of machines across all stages." For example, a flexible flow shop (FFS) problem with 40 jobs, 5 stages, and 12 machines is denoted as "40-5-12."

Evaluation Metrics for Algorithm Comparison

To assess the performance of the proposed MOGA, a comparison is made with the SPEA2 and PESA2 algorithms using three key evaluation metrics: the Spacing Metric (SM), Inverted Generational Distance (IGD), and the number of Non-dominated Solutions (nF1). The definitions of these metrics are as follows:

- **SM:** This metric evaluates the uniformity of the distribution of non-dominated solutions obtained by each algorithm. It is calculated using the formula in Eq. 22. A smaller SM value indicates a more evenly distributed set of solutions.

$$SM = \frac{\sum_{i=1}^{n-1} |\bar{d} - d_i|}{(n-1) \times \bar{d}} \quad (22)$$

where d_i represents the Euclidean distance between two consecutive non-dominated solutions, and $\bar{d}$ is the average Euclidean distance. A lower value of this metric is preferred, as it indicates better solution spread (Piroozfard et al. 2018).

- **IGD:** This metric combines both convergence and diversity by measuring the average Euclidean distance between the solutions on the Pareto front obtained by each

Table 3 Summary of test data

Parameters	Factors	Levels	Number of levels
n_b	Number of jobs	10, 15, 20, 40, and 60	5
n_s	Number of stages	2, 3, and 5	3
$\|M_s\|$	Number of machines per stage	2, 3, 4, and 5	4
$P_{i,s,m}$	Processing time for each operation	Discrete uniform[30,100]	1
$st_{i,j,s,m}$	Sequence-dependent setup times	Discrete uniform[1,5]	1
$PE_{j,s,m}$	Machine power in process state	Discrete uniform[8,10]	1
$SE_{i,j,s,m}$	Machine power in SDST state	Discrete uniform[5,7]	1
$IE_{s,m}$	Machine power in idle state	Discrete uniform[3,5]	1

algorithm and the reference Pareto front. It is computed using Eq. 23, where a lower value is considered better (Coello Coello , Reyes Sierra 2004).

$$IGD = \frac{\sqrt{\sum_{i=1}^{n^*} (d_i^*)^2}}{n^*} \tag{23}$$

In this formula, d_i^* denotes the Euclidean distance between each reference Pareto front solution and the closest solution obtained by the algorithm, while n^* represents the number of non-dominated solutions. In cases where the reference Pareto front is unknown, it can be defined from the integrated non-dominated solutions produced by all competing algorithms (Coello Coello , Reyes Sierra 2004).

- **Number of nF1:** This metric counts the number of non-dominated solutions obtained by each algorithm. A higher number of non-dominated solutions suggests better performance in terms of exploring the solution space.

Parameter Settings

The performance of algorithms is clearly influenced by the values of their parameters. To determine the most effective parameter settings, trial experiments were conducted to test various potential values for each parameter. For the proposed MOGA, key parameters that impact the search behavior include population size, number of iterations, crossover probability, and mutation probability. In the case of SPEA2, the parameters to consider are population size, number of iterations, archive size, crossover probability, and mutation probability. PESA2 requires the same parameters as SPEA2, but also includes the grid number as an additional parameter. To ensure a fair comparison when evaluating the performance of all three algorithms, the same population size and number of iterations were used across all experiments. Table 4 provides a summary of the parameters for each algorithm along with their selected values.

Table 4 Parameter settings values of MOGA, SPEA2, and PESA2

MOGA	SPEA2	PESA2
n_{pop} = 100	n_{pop} = 100	n_{pop} = 100
nIter = 5000	nIter = 5000	nIter = 5000
Pc = 0.9	Pc = 0.7	Pc = 0.8
Pm = 0.2	Pm = 0.3	Pm = 0.2
	nArchive = 80	nArchive = 80
		nGrid = 40

Computational Results and Discussion

In this section, we compare the results of our proposed MOGA across 15 test instances with those of SPEA2 and PESA2. For each instance we run each algorithm five times. Table 5 presents the comparison based on three performance metrics: SM, IGD, and nF1, reporting the best, average, and standard deviation over the five runs. The first two columns list the problem name and its scale, while the remaining columns show the results achieved by the three algorithms according to the performance criteria. The best results for each criterion are highlighted in bold. It is worth noting that, for the IGD metric, we follow the procedure described in Coello Coello , Reyes Sierra (2004) to obtain the reference Pareto front. Specifically, for each test problem, we first aggregate all Pareto-optimal solutions obtained from the five runs of each of the three algorithms (15 runs in total). This combined set is then used as the reference Pareto front for computing the IGD value of each individual run.

Overall, it can be observed that the MOGA outperforms both SPEA2 and PESA2, particularly in terms of the IGD and nF1 indicators.

Regarding the first criterion, SM, SPEA2 clearly outperforms both MOGA and PESA2 on 13 and 10 of the 15 test problems, respectively, based on the best and average results over five runs. This can be attributed to SPEA2's strength-based fitness assignment and density estimation, which penalize crowded regions and promote a more evenly distributed set of solutions. In contrast, PESA2's grid-based diversity mechanism and MOGA's convergence-oriented strategy do not explicitly enforce uniform point-to-point spacing, leading to more irregular distributions and higher SM values. However, as the problem size increases, PESA2 shows improved performance, likely due to the effectiveness of its grid-based diversity mechanism in larger search spaces. Although MOGA does not achieve the best SM values on average, it often displays more consistent behavior

Table 5 Performance comparison of MOGA, SPEA2, and PESA2 on FFSP test problems across three metrics

Problem	Size	MOGA							SPEA2							PESA2						
		SM			IGD			nF1	SM			IGD			nF1	SM			IGD			nF1
		best	avg	sd	best	avg	sd	avg	best	avg	sd	best	avg	sd	avg	best	avg	sd	best	avg	sd	avg
FFSP01	10 – 2 – 7	**0.47**	**0.60**	0.15	**0.05**	**0.73**	**1.16**	**13.20**	0.54	0.78	0.22	4.31	10.08	4.14	6.80	0.57	0.77	**0.11**	2.14	4.48	2.24	12.00
FFSP02	10 – 3 – 12	0.65	0.71	**0.05**	**0.11**	**0.42**	**0.35**	**15.40**	**0.42**	**0.56**	0.10	1.60	2.20	0.53	8.60	0.64	0.78	0.10	0.77	1.37	0.63	11.80
FFSP03	10 – 5 – 22	0.59	0.71	0.13	**0.00**	**0.29**	0.29	**13.80**	**0.34**	**0.51**	0.13	3.95	5.06	0.73	7.40	0.83	0.84	**0.02**	0.78	0.90	**0.11**	10.60
FFSP04	15 – 2 – 6	**0.00**	0.49	**0.35**	**1.29**	**8.07**	3.92	**4.00**	**0.00**	**0.18**	0.41	6.82	11.07	**2.66**	2.60	**0.00**	0.42	0.49	2.28	9.19	4.02	3.60
FFSP05	15 – 3 – 11	0.59	0.66	**0.08**	**2.65**	**3.27**	**0.56**	**15.60**	**0.00**	**0.49**	0.31	9.27	12.95	2.80	10.40	0.48	0.70	0.22	3.84	6.15	2.05	12.80
FFSP06	15 – 5 – 17	0.58	0.69	**0.09**	**2.65**	**3.71**	**0.93**	**22.80**	**0.25**	**0.60**	0.31	9.05	12.93	2.82	14.60	0.32	0.61	0.20	5.23	6.77	1.26	15.80
FFSP07	20 – 2 – 5	0.71	0.87	0.16	**3.00**	**3.93**	**1.12**	**10.20**	**0.50**	**0.66**	0.14	6.29	10.33	4.27	6.60	0.65	0.74	**0.06**	4.53	7.35	3.77	9.40
FFSP08	20 – 3 – 10	0.63	0.90	0.21	**1.90**	**3.58**	1.25	**7.80**	**0.60**	**0.84**	**0.18**	3.48	5.37	1.30	6.20	0.76	0.93	0.19	3.10	4.44	**1.08**	6.80
FFSP09	20 – 5 – 14	0.50	0.89	0.28	**3.49**	**48.97**	40.65	**21.40**	**0.45**	**0.72**	**0.24**	28.49	72.69	27.92	14.00	0.53	0.83	0.29	43.64	72.66	**25.39**	13.60
FFSP10	40 – 2 – 8	0.76	0.82	**0.04**	**4.18**	**4.99**	**0.67**	**30.60**	0.78	0.93	0.10	9.38	20.35	15.37	16.80	**0.57**	**0.81**	0.20	6.78	25.99	15.96	16.20
FFSP11	40 – 3 – 11	0.54	0.70	**0.13**	**2.43**	**8.81**	5.50	**16.00**	0.52	0.76	0.19	6.68	12.78	5.76	11.20	**0.43**	**0.59**	0.15	9.92	13.55	**3.35**	11.00
FFSP12	40 – 5 – 17	0.57	0.70	**0.10**	**3.10**	**4.95**	1.99	**12.80**	**0.51**	**0.69**	0.17	12.27	18.52	5.91	10.00	0.59	0.82	0.19	11.94	14.98	**1.93**	12.00
FFSP13	60 – 2 – 6	1.01	1.08	**0.09**	**9.13**	**17.86**	**8.60**	**25.00**	0.56	0.83	0.21	60.21	80.51	15.49	17.20	**0.49**	**0.77**	0.16	13.43	66.25	31.89	14.00
FFSP14	60 – 3 – 9	0.67	0.78	0.12	**0.70**	**13.68**	7.46	**22.40**	**0.35**	0.79	0.30	19.82	33.87	15.58	16.80	0.51	**0.64**	**0.08**	18.00	20.94	**3.87**	12.80
FFSP15	60 – 5 – 20	0.76	1.06	0.27	**8.65**	50.15	71.58	**40.60**	0.80	0.99	0.30	25.52	94.98	68.16	21.00	**0.73**	**0.97**	**0.22**	29.10	**33.01**	**5.46**	21.20

across runs, resulting in a lower standard deviation in more than half of the test problems.

For the IGD metric, MOGA outperforms both SPEA2 and PESA2 in nearly all cases in terms of both the best and average results. The only exceptions are the FFSP04 and FFSP15 test cases, where SPEA2 and PESA2 achieve a better average over the five runs, respectively. Interestingly, PESA2 demonstrates greater consistency as problem size increases, resulting in lower standard deviations in six of the test problems. It is important to note that the IGD indicator holds greater significance than the other two metrics. As discussed, IGD simultaneously accounts for both convergence and diversity, which are essential for any multi-objective evolutionary algorithm. This allows the algorithm to move toward the Pareto-optimal set while maintaining a widely spread distribution of solutions, as highlighted by Laumanns et al. (2002).

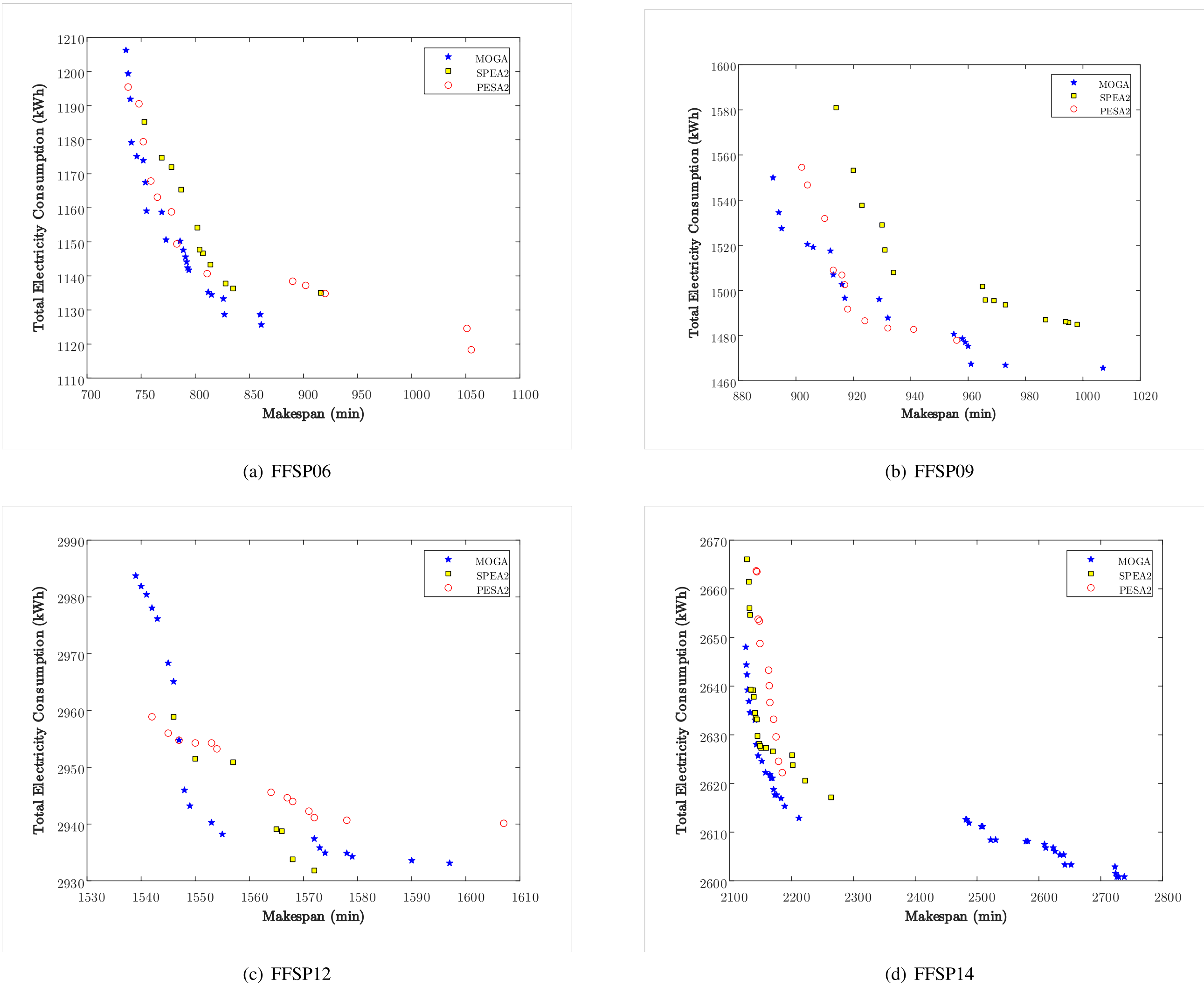
For the third performance indicator, nF1, the proposed MOGA consistently outperforms both SPEA2 and PESA2 across all test problems based on the reported average results.

To illustrate the performance of the competing algorithms, four test problems of varying scales (FFSP06, FFSP09, FFSP12, and FFSP15) were selected for a graphical representation of the Pareto-front solutions obtained from a single run of each algorithm, as shown in Fig. 6. It is evident that the proposed MOGA generates more and superior solutions compared to the other algorithms, indicating that the schedules produced by MOGA not only reduce energy consumption in the manufacturing process but also shorten the overall production time.

To monitor the progress of the competing algorithms toward the Pareto-optimal set while considering solution diversity, the IGD values after the second iteration for each algorithm are plotted in Fig. 7 for the selected problems: FFSP01, FFSP09, FFSP12, and FFSP14. It is important to note that for calculating the IGD values, the newly obtained Pareto-front set from the current iteration is used as the reference points. The IGD value for each iteration (starting from the second iteration to the maximum number of iterations) is then computed based on the Pareto-front solutions found in the previous iteration, as defined in Eq. 23. Additionally, the Gantt charts of a non-dominated solution for two selected test problems, FFSP11 and FFSP15, are shown in Figs. 8 and 9.

Statistical Analysis

In this study, to determine whether the performance differences among the benchmark algorithms are statistically significant, the Wilcoxon signed-rank test was applied to the average results obtained by the competing algorithms.



(a) FFSP06

(b) FFSP09

(c) FFSP12

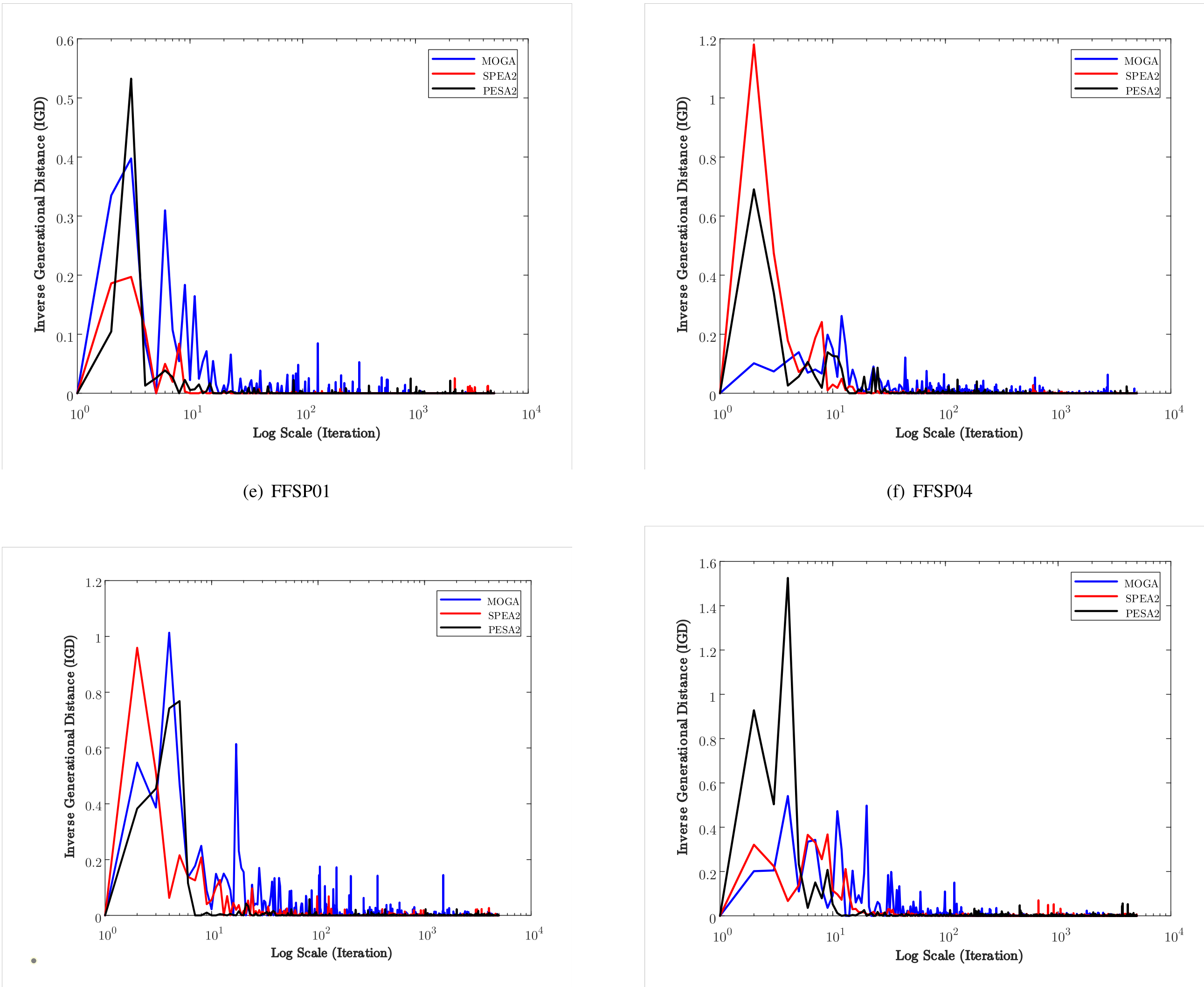
(d) FFSP14

Fig. 6 Non-dominated solutions achieved by MOGA, SPEA2, and PESA2 across selected FFSP problems

Table 6 presents the results of the pairwise comparisons among the three algorithms across all 15 test instances using the Wilcoxon signed-rank test. Considering the first metric, SM, a statistically significant difference is observed only between MOGA and SPEA2, with a p-value of 0.03549, indicating that SPEA2 achieves better diversity preservation, as discussed earlier. For the IGD metric, MOGA demonstrates statistically significant superiority over both SPEA2 and PESA2, with p-values of 0.00006 and 0.00427, respectively. In addition, SPEA2 outperforms PESA2, indicating that both MOGA and SPEA2 produce solution sets closer to the true Pareto front. Regarding the nF1 metric, the results show that MOGA outperforms both SPEA2 and PESA2 with small p-values of 0.00006. However, no statistically significant difference is found between SPEA2 and PESA2 for this metric. In summary, the statistical analysis using the Wilcoxon signed-rank test reinforces the performance trends observed among the competing algorithms, demonstrating the superior performance of MOGA over the other two multi-objective algorithms in terms of the IGD and nF1 metrics. However, SPEA2 shows better solution distribution based on the SM metric.

Conclusion

This research highlights the importance of energy efficiency in manufacturing processes, driven by continuous rise in global energy demand and increase international strict regulations aimed at achieving Net Zero emissions targets, which mandate the electrification of industries. In the light of rising electricity price, particularly in the UK, manufacturers face growing challenges in managing energy consumption while maintaining productivity to meet customer's demand in a highly competitive environment. To address this challenge effectively, this study proposed a multi-objective genetic



(e) FFSP01

(f) FFSP04

(g) FFSP09

(h) FFSP14

Fig. 7 IGD convergence trends for MOGA, SPEA2, and PESA2 on selected FFSP instances

algorithm for the flexible flow-shop scheduling problem, considering SDST to simultaneously minimize both makespan and total energy consumption. Results across various problem scales show that the proposed algorithm effectively balance production time and energy usage, outperforming benchmark multi-objective competing algorithms, SPEA2 and PESA2, and offering viable solution for enhancing energy efficiency in manufacturing environment.

This study provides valuable insights for manufacturing managers by highlighting the importance of energy efficiency and cost reduction in the context of rising energy prices. Managers can achieve considerable cost savings by applying optimized scheduling on the shop floor to minimize both production time and energy consumption, particularly in small and medium-sized enterprises. The proposed MOGA serves as an effective tool for achieving these objectives while supporting sustainability goals. Adopting energy-aware scheduling can enhance machine utilization, improve productivity, and reduce setup times. In addition, metaheuristic algorithms have proven effective for addressing large-scale and complex scheduling problems. The proposed approach fosters sustainability, enhances competitive advantage, and supports corporate social responsibility. Further optimization of energy use can be achieved through real-time adaptive scheduling systems, positioning manufacturers for success in an energy-conscious market.

There are several research avenues for exploring and enhancing energy-aware scheduling in manufacturing floor environments. One promising direction is the use of real-time data to develop dynamic scheduling approaches that can effectively respond to changing shop-floor conditions. Another extension of this research is the integration of optimization and simulation to reduce the complexity of mathematical models and to visualize and validate schedules

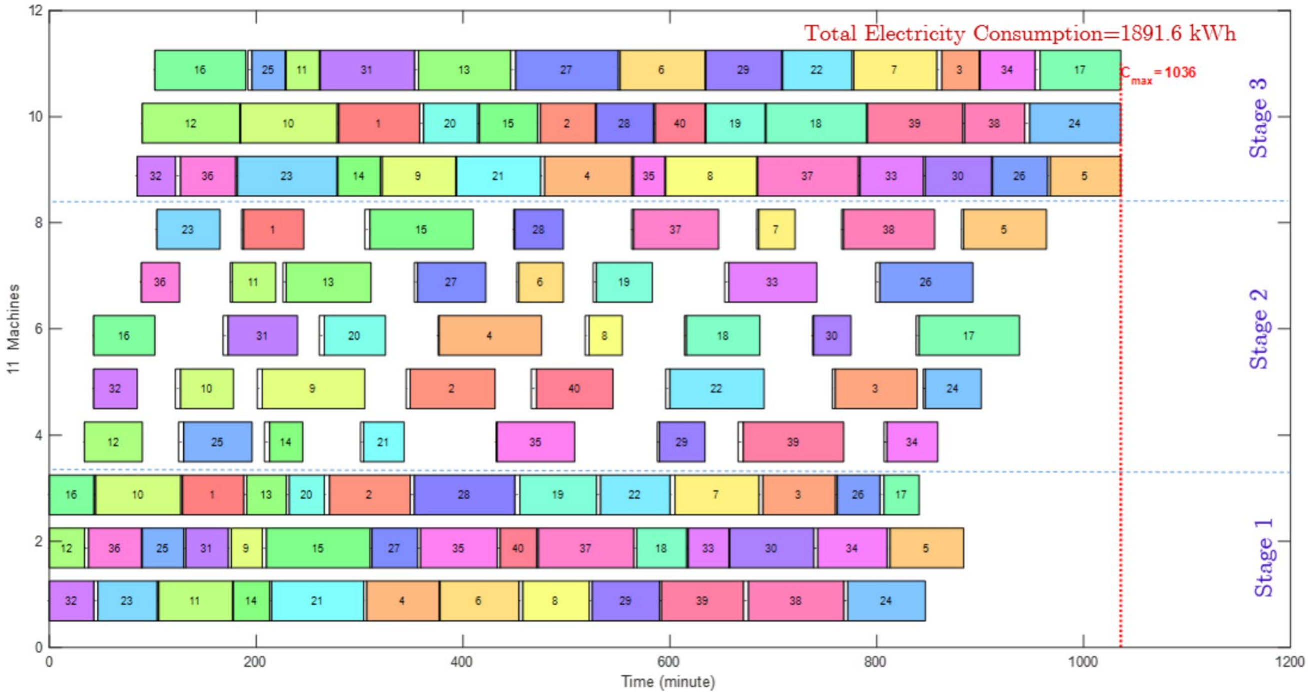


Fig. 8 Gantt charts of a non-dominated solution for FFSP11 obtained by MOGA

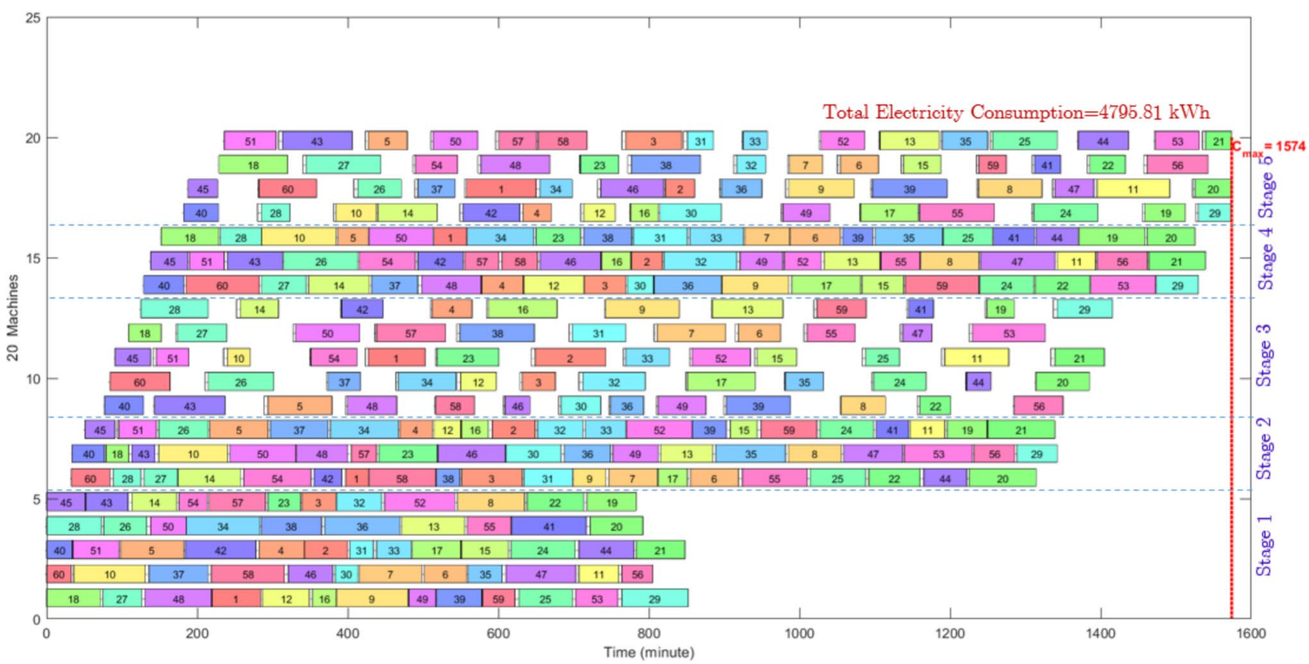


Fig. 9 Gantt charts of a non-dominated solution for FFSP15 obtained by MOGA

in a virtual environment. Additionally, advanced machine learning techniques, such as reinforcement learning, could lead to adaptive scheduling methods that evolve over time and can be efficiently applied to manufacturing systems to support informed decision-making under uncertain machine performance and other critical factors. Finally, broader sustainability factors, such as carbon emissions, repair and recycling, and waste reduction, should be incorporated into scheduling models to promote a more holistic approach to green manufacturing.

Table 6 Wilcoxon signed-rank test results for pairwise comparisons of MOGA, SPEA2, and PESA2 across all metrics ($\alpha = 0.05$). Significant p-values are highlighted in bold

Metric	Comparison	Statistic	p-value
SM	MOGA vs SPEA2	23.0000	**0.03549**
	MOGA vs PESA2	46.5000	0.44314
	SPEA2 vs PESA2	37.5000	0.20119
IGD	MOGA vs SPEA2	0.0000	**0.00006**
	MOGA vs PESA2	12.0000	**0.00427**
	SPEA2 vs PESA2	12.0000	**0.00427**
nF1	MOGA vs SPEA2	0.0000	**0.00006**
	MOGA vs PESA2	0.0000	**0.00006**
	SPEA2 vs PESA2	35.0000	0.15547

Acknowledgements We would like to express our sincere gratitude to the editor and anonymous reviewers for their constructive and insightful comments, which greatly improved the quality of this manuscript.

Author Contributions Ali Mokhtari-Moghadam: conceptualization, Methodology, Formal analysis and investigation, Writing - original draft preparation, and numerical data analysis. Trung Thanh Nguyen: Funding acquisition, Resources, Supervision, Writing - review and editing. Massoud Mohsendokht: Formal analysis and investigation, Writing - review and editing.

Funding This work was supported by Liverpool John Moores University (LJMU) through the Freeport and Net Zero Transport Thematic Doctoral Pathways (FTDP) Scholarship.

Data Availability The data used in this study are available at: https://github.com/AliMokhtari58/Datasets/blob/main/FFSP_dataset.zip.

Code Availability The code used for solution decoding in this study is available at: https://github.com/AliMokhtari58/Datasets/blob/main/ParseSolutionCode.m.

Declarations

Ethical Approval This article does not involve any studies with animals or human participants conducted by the authors.

Competing Interests The authors declare that they have no competing interests.

References

Chamandoust H, Derakhshan G, Bahramara S (2020) Multi-objective performance of smart hybrid energy system with Multi-optimal participation of customers in day-ahead energy market. Energy and Buildings 216:109964

Chen Y, Du J, Mumtaz J, Zhong J, Rauf M (2025) An efficient Q-learning integrated multi-objective hyper-heuristic approach for hybrid flow shop scheduling problems with lot streaming. Expert Syst Appl 262:125616

Coello Coello, Carlos A., & Reyes Sierra, Margarita. 2004. A Study of the Parallelization of a Coevolutionary Multi-objective Evolutionary Algorithm. *Pages 688–697 of:* Monroy, Raúl, Arroyo-Figueroa, Gustavo, Sucar, Luis Enrique, & Sossa, Humberto (eds), *MICAI 2004: Advances in Artificial Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg

Corne, D., Jerram, Nick R., Knowles, Joshua D., & Oates, M. 2001. PESA-II: region-based selection in evolutionary multiobjective optimization

Dai M, Tang D, Giret A, Salido MA, Li WD (2013) Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. Robotics and Computer-Integrated Manufacturing 29(5):418–429

Dai M, Tang D, Giret A, Salido MA (2019) Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. Robotics and Computer-Integrated Manufacturing 59:143–157

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Duan J, Liu F, Zhang Q, Qin J, Zhou Y (2024) Genetic programming hyper-heuristic-based solution for dynamic energy-efficient scheduling of hybrid flow shop scheduling with machine breakdowns and random job arrivals. Expert Syst Appl 254:124375

Ge Y, Ding H, Wang A, Yang H, Wang Y (2025) Scheduling for hybrid flow shop with energy-efficiency and machine preventive maintenance in sheet metal manufacturing system. Computers & Industrial Engineering 204:111050

Gong G, Chiong R, Deng Q, Han W, Zhang L, Lin W, Li K (2020) Energy-efficient flexible flow shop scheduling with worker flexibility. Expert Syst Appl 141:112902

Government, UK. 2024a. *Future Opportunities for Electrification to Decarbonise UK Industry*. Accessed January 17, 2025

Government, UK. 2024b. *Quarterly Energy Prices - December 2024*. Accessed January 17, 2025

Han Y, Li J, Sang H, Liu Y, Gao K, Pan Q (2020) Discrete evolutionary multi-objective optimization for energy-efficient blocking flow shop scheduling with setup time. Appl Soft Comput 93:106343

He L, Jiang S-L, Sun L, Chiong R (2026) A learning-guided multi-objective approach for energy-oriented hybrid flow shop scheduling with limited buffers. Inf Sci 731:122884

Holland JH (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press

hong Jia, Zhao, Wu, Tianfu, Zhang, Han, Liu, Chuang, & Li, Kai. (2026) An energy-efficient scheduling approach for a two-stage hybrid flow shop with parallel batch machines. Eur J Oper Res 328(3):762–784

Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. Evol Comput 10(3):263–282

Lee, K.-M., Yamakawa, T., & Lee, Keon-Myung. 1998. A genetic algorithm for general machine scheduling problems. *Pages*

60–66 vol.2 of: 1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No.98EX111), vol. 2

Lee T-S, Loong Y (2019) A review of scheduling problem and resolution methods in flexible flow shop. Int J Ind Eng Comput 10:67–88

Li P, Xue Q, Zhang Z, Chen J, Zhou D (2023) Multi-objective energy-efficient hybrid flow shop scheduling using Q-learning and GVNS driven NSGA-II. Computers & Operations Research 159:106360

McKinsey & Company. 2024. *Global Energy Perspective*. Accessed 17 Jan 2025

Mhanna, Joyce, Nouinou, Hajar, Caillard, Simon, & Baudry, David. 2024. Energy-Efficient Flexible Flow Shop Scheduling Under Time-Of-Use Rates with Renewable Energy Sources. *IFAC-PapersOnLine*, **58**(19), 319–324. 18th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2024

Moghadam, Ali Mokhtari, Wong, Kuan Yew, & Piroozfard, Hamed. 2014. An efficient genetic algorithm for flexible job-shop scheduling problem. *Pages 1409–1413 of: 2014 IEEE International Conference on Industrial Engineering and Engineering Management*

Mokhtari-Moghadam A, Pourhejazy P, Gupta D (2023) Integrating sustainability into production scheduling in hybrid flow-shop environments. Environ Sci Pollut Res 30(11):31132–31147

Mouzon G, Yildirim MB (2008) A framework to minimise total energy consumption and total tardiness on a single machine. Int J Sustain Eng 1(2):105–116

Mouzon G, Yildirim MB, Twomey J (2007) Operational methods for minimization of energy consumption of manufacturing equipment. Int J Prod Res 45(18–19):4247–4271

Mraihi T, Driss OB, EL-Haouzi, Hind Bril. (2024) Distributed Permutation Flow Shop Scheduling Problem with Worker flexibility: Review, trends and model proposition. Expert Syst Appl 238:121947

Pinedo ML (2008) Scheduling: Theory, Algorithms, and Systems, 3rd edn. Springer Publishing Company, Incorporated

Piroozfard H, Wong KY, Wong WP (2018) Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. Resour Conserv Recycl 128:267–283

Qin H-X, Han Y-Y, Zhang B, Meng L-L, Liu Y-P, Pan Q-K, Gong D-W (2022) An improved iterated greedy algorithm for the energy-efficient blocking hybrid flow shop scheduling problem. Swarm Evol Comput 69:100992

Ribas, Imma, Leisten, Rainer, & Framiñan, Jose M. 2010. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers and Operations Research*, **37**(8), 1439–1454. Operations Research and Data Mining in Biological Systems

Roshanaei V, Seyyed Esfehani MM, Zandieh M (2010) Integrating non-preemptive open shops scheduling with sequence-dependent setup times using advanced metaheuristics. Expert Syst Appl 37(1):259–266

Ruiz, Rubén, & Vázquez-rodríguez, José Antonio. 2010. The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 1–18

Tang, Dunbing, Dai, Min, Salido, Miguel A., & Giret, Adriana. 2016. Energy-Efficient Dynamic Scheduling for a Flexible Flow Shop Using an Improved Particle Swarm Optimization. *Comput. Ind.*, **81**(C), 82–95

Wang C, Yao Y, Li W, Li X (2026) A multi-population co-evolutionary algorithm for solving energy-efficient hybrid flow shop scheduling problem. Expert Syst Appl 297:129536

Wang J, Tang H, Lei D (2024) A feedback-based artificial bee colony algorithm for energy-efficient flexible flow shop scheduling problem with batch processing machines. Appl Soft Comput 153:111254

Wang S, Wang X, Chu F, Yu J (2020) An energy-efficient two-stage hybrid flow shop scheduling problem in a glass production. Int J Prod Res 58(8):2283–2314

Wu J, Liu Y, Zhang Y (2025) Multi-objective evolutionary co-learning framework for energy-efficient hybrid flow-shop scheduling problem with human-machine collaboration. Swarm Evol Comput 95:101932

Wu X, Cao Z (2022) An improved multi-objective evolutionary algorithm based on decomposition for solving re-entrant hybrid flow shop scheduling problem with batch processing machines. Computers & Industrial Engineering 169:108236

Yan J, Li L, Zhao F, Zhang F, Zhao Q (2016) A multi-level optimization approach for energy-efficient flexible flow shop scheduling. J Clean Prod 137:1543–1552

Yuan M, Ye Y, Huang H, Zhang Z, Pei F, Gu W (2025) Multi-objective energy-efficient scheduling of distributed heterogeneous hybrid flow shops via multi-agent double deep Q-Network. Swarm Evol Comput 98:102076

Yue X, Xiong X, Zhang M, Xu X, Yang L (2025) Multi-objective optimization for energy-efficient hybrid flow shop scheduling problem in panel furniture intelligent manufacturing with transportation constraints. Expert Syst Appl 274:126830

Zandieh M, Fatemi Ghomi SMT, Moattar Husseini SM (2006) An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. Appl Math Comput 180(1):111–127

Zhang, Mingyang, Yan, Jihong, Zhang, Yanling, & Yan, Shenyi. 2019. Optimization for energy-efficient flexible flow shop scheduling under time of use electricity tariffs. *Procedia CIRP*, **80**, 251–256. 26th CIRP Conference on Life Cycle Engineering (LCE) Purdue University, West Lafayette, IN, USA May 7-9, 2019

Zhang Z-Q, Wu X-Y, Qian B, Hu R, Yang J-B (2026) A Q-learning-based multi-objective hyper-heuristic algorithm for energy-efficient integrated distributed hybrid flow-shop scheduling with preventive maintenance. Computers & Operations Research 185:107267

Zhao F, Yin F, Zhang J, Xu TP (2026) A learning-based co-evolution optimization framework for energy-aware distributed heterogeneous flexible flow shop lot-streaming scheduling problem. Expert Syst Appl 296:128986

Zitzler, Eckart, Laumanns, Marco, & Thiele, Lothar. 2001. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Tech. rept