Environmental Science and Pollution Research
https://doi.org/10.1007/s11356-023-26986-3

OPPORTUNITIES AND CHALLENGES OF INDUSTRIAL SUSTAINABLE DEVELOPMENT IN THE POST-PANDEMIC WORLD

Integrating sustainability into production scheduling in hybrid flow-shop environments

Ali Mokhtari-Moghadam[1] · Pourya Pourhejazy[2] · Deepak Gupta[3]

Received: 29 November 2022 / Accepted: 9 April 2023

Abstract

Global energy consumption is projected to grow by nearly 50% as of 2018, reaching a peak of 910.7 quadrillion BTU in 2050. The industrial sector accounts for the largest share of the energy consumed, making energy awareness on the shop floors imperative for promoting industrial sustainable development. Considering a growing awareness of the importance of sustainability, production planning and control require the incorporation of time-of-use electricity pricing models into scheduling problems for well-informed energy-saving decisions. Besides, modern manufacturing emphasizes the role of human factors in production processes. This study proposes a new approach for optimizing the hybrid flow-shop scheduling problems (HFSP) considering time-of-use electricity pricing, workers' flexibility, and sequence-dependent setup time (SDST). Novelties of this study are twofold: to extend a new mathematical formulation and to develop an improved multi-objective optimization algorithm. Extensive numerical experiments are conducted to evaluate the performance of the developed solution method, the adjusted multi-objective genetic algorithm (AMOGA), comparing it with the state-of-the-art, i.e., strength Pareto evolutionary algorithm (SPEA2), and Pareto envelop-based selection algorithm (PESA2). It is shown that AMOGA performs better than the benchmarks considering the mean ideal distance, inverted generational distance, diversification, and quality metrics, providing more versatile and better solutions for production and energy efficiency.

Keywords Production management · Energy consumption · Time-of-use electricity rates · Workers' flexibility · Multi-objective optimization

Introduction

The global population growth, which has resulted in an exponential increase in industrial activities, and the worsening environmental situation over the past few decades highlight the need for sustainable industrial development. Global energy consumption is projected to grow by nearly 50% between 2018 and 2050, which is expected to reach a high of 910.7 quadrillion BTU. The industrial sector, including mining, manufacturing, agriculture, and construction, accounts for the largest share of global energy consumption.[1]

To respond to the growing energy demands and alleviate the grid's burden during peak hours, many countries introduced time-dependent energy prices that can vary during the day and across seasons. As a prime example, manufacturing enterprises are asked to shift all or parts of their activities to the off-peak periods to reduce the load on the system to manage the

[1] https://www.eia.gov/outlooks/aeo

Responsible Editor: Philippe Garrigues

✉ Pourya Pourhejazy
pourya.pourhejazy@uit.no

Ali Mokhtari-Moghadam
am22638@essex.ac.uk

Deepak Gupta
deepak.gupta@wichita.edu

[1] Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, UK

[2] Department of Industrial Engineering, UiT – The Arctic University of Norway, Narvik 8514, Norway

[3] Industrial, Systems, and Manufacturing Engineering, College of Engineering, Wichita State University, Wichita, KS 67260, USA

Published online: 24 April 2023

energy crisis. Purchasing energy-efficient machines, updating product designs, and process optimization are some of the other measures to tackle the energy consumption problem. Production scheduling is a relatively affordable and effective solution to help manage operations in the quest for cleaner production (Dai et al. 2013).

As a generalization of the classic flow-shop and parallel-machine scheduling, the hybrid flow-shop scheduling problem (HFSP) consists of at minimum two processing stages with at least one of them having two or more parallel machines (Ruiz and Vázquez-rodríguez 2010). Machines in the production stages can be identical, uniform, unrelated, or a combination, and the jobs flow through the shop floor in the same direction (Linn and Zhang 1999). The wide industrial needs have motivated production researchers to extend the HFSP for various shop floor configurations. Salvador (1973) for the first time applied the HFSP in practical settings, i.e., nylon polymerization production context for minimizing the maximum completion time—makespan. In the packaging industry context, Adler et al. (1993) developed a robust scheduling system that meets the due date and maximizes plants' throughput. Minimizing makespan and buffer queueing in semiconductor manufacturing, Wittrock (1988) applied an adaptable scheduling algorithm to handle the dynamics of the shop floor. Jin et al. (2002) developed a scheduling approach for three-stage printed circuit board production line optimization, aiming to minimize the total completion time of all jobs over a finite planning horizon. Other practical applications of HFSP include the iron and steel industry (Tang and Wang 2011), ion plating cell production, container handling systems in a maritime terminal (Chen et al. 2007), and textile, glass, paper, furniture, and plastic-making industries (Lin et al. 2021).

Incorporating the technical and operational requirements into the basic HFSP, like setup time (Zandieh et al. 2006), machine eligibility (Ruiz and Maroto 2006), the machine turn-off-turn-on decisions (Mouzon et al. 2007), the transfer time of jobs (Pan et al. 2013), a limited buffer between production stages (Wardono and Fathi 2004), and work-center space constraints (Moghadam et al. 2018), has facilitated its industry reach. Most of the existing scheduling problems overlooked human considerations. Considering the impact of workers' capabilities on energy efficiency (Zhang and Dornfeld 2007), simultaneously incorporating these factors helps to account for the possible interactions and improves the optimization outcomes. Inspired by this practical need, the main contribution of this study is to incorporate an energy consumption model into the HFSP formulation while considering workers' performance and sequence dependent setup time (SDST) between the processing of consecutive jobs. Such an extension increases the scheduling complexity and calls for effective solution approaches. A multi-objective metaheuristic algorithm is extended and tested to contribute to the scheduling literature.

The rest of this manuscript is organized into four sections. The "Research gap" section reviews the relevant literature and highlights the research gap. The "Mathematical formulation" section is devoted to problem description and mathematical model formulation. The adjusted multi-objective genetic algorithm (AMOGA) optimization algorithm for solving the model is presented in the "Solution method" section. The "Computational experiments" section presents the computational experiments and analyzes the results considering different instances. Finally, the study is concluded and summarized in the "Conclusions" section.

Research gap

Maximizing the shop floor efficiency has been the central optimization objective in the production scheduling literature (Ruiz and Vázquez-rodríguez 2010; Ribas et al. 2010; Lee and Loong 2019). To tackle up-to-date manufacturing concerns for environmental sustainability and energy security, the energy consumed on the shop floor should also be taken into consideration. A growing number of studies are investigating this paradigm shift. Early studies on energy-efficient scheduling were limited to optimizing energy consumption using execution time variables (Dai et al. 2013; Tang et al. 2016; Yan et al. 2016; Li et al. 2018). More recent studies considered energy consumption models under time-of-use electricity tariffs. Such energy models are more practical and can more effectively address energy efficiency.

Luo et al. (2013) investigated scheduling under time-of-use electricity tariffs while minimizing makespan and electric power costs. Applying the right-shift procedure to reduce the estimated production cost, they found that as the length of each electricity price period increased, the electric power cost decreased. They suggested that it is more energy efficient to consider a mix of high-power and low-power machines rather than having the same number of machines with middle power. In a closely relevant study, Zhang et al. (2019) developed a bi-objective optimization model with the presence of time-of-use electricity tariffs for scheduling flexible flow-shops consisting of machines with heterogeneous energy consumption but did not account for the workers' capabilities and setup times despite their relevance to the energy consumption context. They used the strength Pareto evolutionary algorithm (SPEA2) for solving the multi-objective optimization problem. Wang et al. (2020) proposed augmented epsilon-constraint and evolutionary algorithms for solving various variants of two-stage flow-shops under time-of-use energy tariffs, minimizing makespan and total energy consumption considering a case from the glassmaking

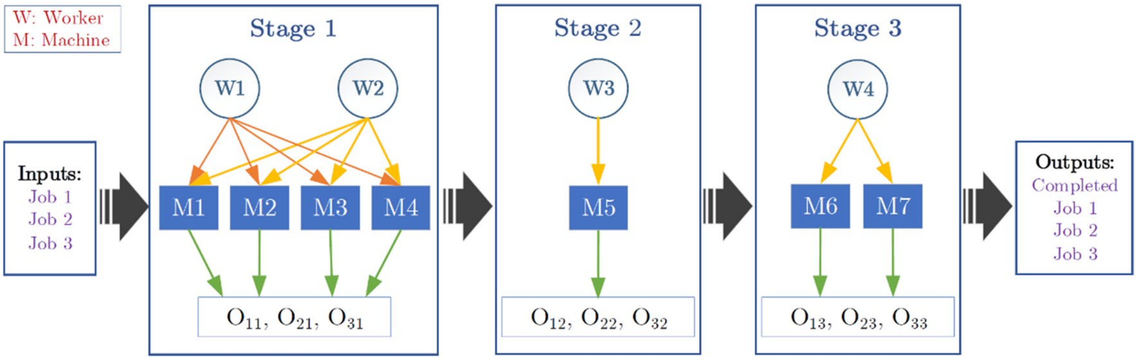


Fig. 1 The real-world illustration of a hybrid flow-shop with worker flexibility

industry. Most recently, Xue and Wang (2023) developed a multi-objective discrete differential evolution algorithm for scheduling under time-of-use electricity tariffs in two-stage flow shop environments.

To improve the practicability of the scheduling problems, direct human involvement should also be considered along with machine-related resource constraints. Industry 5.0 calls for this addition to the production planning and control models. Dual resource constraints have seen some developments in the job shop scheduling literature. Zheng and Wang (2016) proposed a knowledge-guided fruit fly optimization algorithm to solve dual resource-constrained flexible job shops with a single objective of minimizing the makespan. A flexible job shop with dual-resource constrained was studied by Yazdani et al. (2019) where two evolutionary algorithms were developed to minimize the makespan, critical machine workload, and total workload of machines altogether.

As a relevant variant of dual-resource constrained feature, workers' flexibility considers the number of machines a worker can operate. An illustrative example of this scheduling variant is presented in Fig. 1, where identical machines with different power consumption operate with the help of different skilled workers. In this example, worker flexibility is considered in the first stage, meaning that any of worker 1 or worker 2 can be assigned to perform the setup operations.

Gong et al. (2018a) investigated a multi-objective variant with flexible workers aiming to minimize total worker cost and makespan while maximizing green production indicators using an improved genetic algorithm. Gong et al. (2018b) developed a memetic algorithm to solve multi-objective flexible job shops with flexible workers, aiming at the minimization of makespan, maximum machines workload, and total workload of all machines. Gong et al. (2020) addressed flexible flow-shops with worker flexibility. They proposed a hybrid evolutionary algorithm to minimize three objectives of makespan, total worker cost, and green production indicators. They also developed a multi-objective memetic algorithm aiming to minimize the makespan, and total tardiness while accounting for a balanced workers' workload in an HFSP with dual resource constraints. In their proposed model, operations are processed by both machines and workers with predefined processing times. Gong et al. (2021) developed a non-dominated ensemble fitness ranking algorithm for multi-objective flexible job-shop scheduling problem considering worker flexibility. Most recently, Luo et al. (2023) developed a Pareto-based two-stage evolutionary algorithm considering workers' flexibility in flexible job shop production environments.

There are limited scheduling models that consider the workers' flexibility and energy efficiency. Meng et al. (2019) proposed two mixed-integer linear programming models, an effective variable neighborhood search algorithm is utilized, and two energy-saving strategies, namely postponing strategy and turning off/on strategy to minimize total energy consumption in HJSP with worker flexibility. Meng et al. (2023) developed a variable neighborhood search algorithm for the optimization of energy-conscious flexible job shop scheduling problem with workers' flexibility. The flow shop studies that consider dual resource conditions and workers' flexibility are quite limited.

The research on HFSP predominantly considered machines as a sole resource. On the shop floor, different resources are required to complete the products. The most important resources are (1) machines (and robots), which are used to process the planned operations and can be one resource type like in a spool fabrication shop (Moghadam et al. 2014) or a combination of all types of resources like in car manufacturing industries; (2) energy, which is available in forms of electricity, gasoline, and coal, to provide power for process machines, handling systems, auxiliary equipment, lighting, and so forth; and (3) workers that operate machines and/or adjust tools for undertaking different jobs. Depending on the machine shop, a worker may be assigned to only one machine at a time to act as an operator that remains busy until the operation of a specific job is over. Otherwise, they may work on more than one identical or unrelated machine that should be adjusted and/or prepared for starting the next operation. In the latter case, the worker/operator remains busy for the duration of the setup operations. Given the worker's role in these operations, human-related considerations, in particular their capabilities, should be incorporated into production planning to account for the social aspect of sustainability.

The next section extends the formulation developed by Zhang et al. (2019) to integrate the major social and environmental aspects of sustainability into the multi-objective HFSP, that is, simultaneously accounting for the energy consumption model as well as the human-related and setup constraints to account for sustainability in the scheduling problem.

Mathematical formulation

To provide a formal description of the problem, let us assume that n jobs should flow through s successive production stages in the same direction. In each stage, there are one or more identical machines featured with different power consumption and workers' assignment when machines are in the setup state. Workers are required to perform preparational activities, like tools adjustment and cleaning, while the number of workers is limited, and the workers have different skill sets. In this situation, the objective is to arrange the operations in a time interval considering both objective functions. A schedule is feasible only if (a) every job is processed by only one machine and in only one stage; (b) each machine processes only one operation at a given time; (c) each worker can operate only one machine at a time and when the machine is in its setup state; (d) jobs are independent of each other; (e) machines are independent of each other; (f) workers are independent of each other; (g) operations are not interrupted after started; (h) every operation is processed only after its preceding operations are completed. It is also assumed that the moving time between the operations of machines is negligible, and the processing times corresponding to the operations and the SDST between different operations are predefined and deterministic.

The following indices, sets, parameters, and decision variables are used in the mathematical formulation of the problem.

Indices	$j, \mathcal{I}$	Job index, where $j = 1, 2, \ldots, n_j$
	m	Machine tag, where $m = 1, 2, \ldots, n_m$
	s	Index of stages, where $s = 1, 2, \ldots, n_s$
	w	Workers' index, where $w = 1, 2, \ldots, n_w$
	k	Sequential index of operations processed on the same machine,$k = 1, 2, \ldots, n_l$
	q	Sequential index of operations processed by the same worker,$q = 1, 2, \ldots, n_l$

Parameters	n_j	Number of jobs
	n_m	Number of machines
	n_s	Number of stages
	n_w	Number of workers
	AM_s	The set of available machines in stage s
	AW_s	Available worker set in stage s
	C_{ms}	Availability time of machine m in stage s
	C_{ws}	Availability time of worker w in stage s
	$ActW$	Active worker
	nm_s	Number of machines in stage s
	nw_s	Number of workers in stage s
	O_{js}	Operation of job j in stage s
	P_{jsm}	Processing time of O_{js} processed on machine m
	C_{js}	Completion time of O_{js}
	SC_{js}	Sorted set of O_{js} in stage s
	C_j	Completion time of job j
	ST_{jsm}	Start time of processing job j in stage s on machine m
	$st_{j\mathcal{I}s}$	Setup time between jobs j and its predecessor in stage s
	pf_w	Performance factor of worker w
	S_w	Base salary of worker w per minute
	Pc_{jsm}	Processing cost of O_{js} on machine m
	TEC_{jsm}	Energy consumption when O_{js} is processed on machine m
	L	A large positive number
Variables	X_{jsmk}	Binary variable; = 1, if O_{js} is processed in position k on machine m; = 0, otherwise
	Q_{jswq}	Binary variable; = 1, if setup of O_{js} is processed in position q by worker w; = 0, otherwise
	$y_{jismw}(t)$	Binary variable; = 1, if there is an adjustment from job j to job i in stage s on machine m by worker w at moment t; = 0, otherwise
	$x_{j,s,m}(t)$	Binary variable; = 1, if job j in stage s is processed on machine m at moment t; = 0, otherwise
	$z_m(t)$	Binary variable; = 1, if machine m is idle at moment t; = 0, otherwise
	$\rho(t)$	Power cost counter, $= \begin{cases} cp1, t1 \leq t < t2 \\ cp2, t2 \leq t < t3 \\ \ldots \\ cpp, tp \leq t < tp+1 \end{cases}$

Objective functions

The first objective is to minimize the makespan, C_{max}: this refers to the completion time of the last job, to be computed using Eq. (1).

$$minC_{max} = maxC_j \ : \ \forall j = 1, 2, \ldots, n_j; \tag{1}$$

The second objective is to minimize the total cost, TC: this consists of total electricity consumption and workers' costs as shown in Eq. (2).

$$minTC = TEC_{cost} + TW_{cost} \tag{2}$$

These objectives are conflicting in nature; that is, minimizing one objective may increase the magnitude of the other objective. The objective function is subject to the following constraints.

Constraints

In the first stage, when jobs are not pre-processed, and there are no predecessor jobs on a machine, the completion time can be regulated using Eq. (3).

$$C_{\mathcal{I}1m} = max\left(C_j, C_{m1}, C_{w1}\right) + \left(st_{\mathcal{I},\mathcal{I}1} \times pf_w\right) + \left(P_{\mathcal{I}1m}.x_{\mathcal{I}1m}\right) \\ \forall j = 1, 2, \ldots, n_j; m \in AM_1; w \in AW_1 \tag{3}$$

When jobs are not pre-processed, but preparations are required on a machine, the completion time can be described using Eq. (4).

$$C_{\mathcal{I}1m} = max\left(C_j, C_{m1}, C_{w1}\right) + \left(st_{j,\mathcal{I}1} \times pf_w\right) + \left(P_{\mathcal{I}1m}.x_{\mathcal{I}1m}\right) \\ \forall j = 1, 2, \ldots, n_j; m \in AM_1; w \in AW_1 \tag{4}$$

In other stages, when jobs are pre-processed, but there are no antecedents on the machine, Constraint (5) will be the basis for calculating the completion time using Eq. (6). Constraint (5) indicates that the difference between the completion times must be larger than or equal to the required setups.

$$C_{j(s-1)} - C_{ws} \geq st_{\mathcal{I},\mathcal{I}s} \times pf_w \tag{5}$$

$$C_{\mathcal{I}sm} = C_{j(s-1)} + P_{\mathcal{I}sm} \times x_{\mathcal{I}sm} \tag{6}$$

where $\forall j = 1, 2, \ldots, n_j; m \in AM_s; w \in AW_s$. Otherwise, if Constraint (7) applies, Eq. (8) calculates the completion time. Constraint (7) requires that the time difference between completing consecutive tasks is less than the associated setups.

$$C_{j(s-1)} - C_{ws} < st_{\mathcal{I},\mathcal{I}s} \times pf_w \tag{7}$$

$$C_{\mathcal{I}sm} = C_{ws} + st_{\mathcal{I},\mathcal{I}s} \times pf_w + P_{\mathcal{I}sm} \times x_{\mathcal{I}sm}, \\ \forall j = 1,2, \ldots, n_j; m \in AM_s; w \in AW_s \tag{8}$$

Alternatively, when jobs are pre-processed, there are predecessor jobs on the machine, and conditions in Constraints (9) and (10) are both true, the completion time can be calculated using Eq. (11).

$$C_{j(s-1)} > C_{ws} > C_{ms} \tag{9}$$

$$C_{j(s-1)} - C_{ws} \leq st_{j,\mathcal{I}s} \times pf_w \tag{10}$$

$$C_{\mathcal{I}sm} = C_{ws} + st_{j,\mathcal{I}s} \times pf_w + P_{\mathcal{I}sm} \times x_{\mathcal{I}sm}, \forall j = 1,2, \ldots, n_j; m \in AM_s; w \in AW_s \tag{11}$$

In other cases, when jobs are pre-processed and there are antecedents on the machine, the completion time when Constraints (9) and (12) are both true is defined as Eq. (13).

$$C_{j(s-1)} - C_{ws} > st_{j,\mathcal{I}s} \times pf_w \tag{12}$$

$$C_{\mathcal{I}sm} = C_{j(s-1)} + P_{\mathcal{I}sm} \times x_{\mathcal{I}sm}, \forall j = 1, 2, \ldots, n_j; m \in AM_s; w \in AW_s \tag{13}$$

wherever jobs are pre-processed and there are preceding jobs on the machine, the completion time when Constraints (14) and (15) are true should be computed using Eq. (16). Constraint (14) requires that the completion time of job $O_{j(s-1)}$ is greater than the availability time of the assigned machine and worker. According to Constraint (15), the completion time of job $s - 1$ is less than or equal to the summation of setups and availability time of the assigned machine.

$$C_{j(s-1)} > C_{ms} > C_{ws} \tag{14}$$

$$C_{j(s-1)} - C_{ms} \leq st_{j,\mathcal{I}s} \times pf_w \tag{15}$$

$$C_{\mathcal{I}sm} = C_{ms} + st_{j,\mathcal{I}s} \times pf_w + P_{\mathcal{I}sm} \times x_{\mathcal{I}sm}, \\ \forall j = 1,2, \ldots, n_j; m \in AM_s; w \in AW_s \tag{16}$$

In the alternative situation, i.e., when jobs are pre-processed and preceding jobs have been completed on the machine, the completion time can also be defined using Eq. (13) if and only if conditions stated in Constraints (14) and (17) are true.

$$C_{j(s-1)} - C_{ms} > st_{j,\mathcal{I}s} \times pf_w \tag{17}$$

In situations when jobs are pre-processed, there were early jobs on the machine, and Constraint (18) applies, Eq. (16) defines the completion time. Constraint (18) implies that the availability time of machines should be greater than the maximum of the completion time of the previous job and the worker's availability.

$$C_{ms} > max(C_{j(s-1)}, C_{ws}) \tag{18}$$

Finally, when jobs are processed already, there are preceding jobs on the machine, and Constraint (19) is true, and the completion time can be calculated using Eq. (11). Constraint (19) indicates that the worker's availability is greater than the maximum between the completion time of the preceding job and the machine availability for the current job.

$$C_{ws} > max(C_{j(s-1)}, C_{ms}) \tag{19}$$

The remainder of the constraints are as follows.

$$C_{j(s+1)} - C_{js} \geq P_{j(s+1)m} \times x_{j(s+1)m} + st_{j,\mathcal{I}(s+1)} \times pf_w \\ \forall j = 1, 2, \ldots, n_j; s = 1, 2, \ldots, n_s - 1; m = 1, 2, \ldots, n_m; w = 1, 2, \ldots, n_w \tag{20}$$

$$\sum_{k}\sum_{m} x_{jsmk} \leq 1, \forall j = 1, 2, \ldots, n_j; s = 1, 2, \ldots, n_s \tag{21}$$

$$\sum_{s}\sum_{j} x_{jsmk} \leq 1, \forall m = 1, 2, \ldots, n_m; k = 1, 2, \ldots, n_l \tag{22}$$

$$\sum_{q}\sum_{w} Q_{jsmq} = 1, \forall j = 1, 2, \ldots, n_j; s = 1, 2, \ldots, n_s \tag{23}$$

$$\sum_{s}\sum_{j} Q_{jsmk} \leq 1, \forall w = 1, 2, \ldots, n_w; q = 1, 2, \ldots, n_l \tag{24}$$

Constraint (20) ensure that the operation sequence is respected, and there is enough time for processing and setups of consecutive jobs considering the performance of the assigned workers. Equation (21) guarantees that each job can be processed by at most one machine at a given time. Constraint (22) is defined to make sure that each machine can process one job at a time. Equation (23) ensures that when setup is required, it must be done by one worker. Constraint (24) ensures that each worker can process a maximum of one operation at a time.

Finally, Eqs. (25)–(31) are used for calculating the cost elements of the objective function. The total electricity consumption (TEC) is incorporated using Eq. (25); this includes the total electricity consumed during the setups, processing, and idle states of machines. The cost terms are defined by Eqs. (26)–(28), respectively; which are adopted from Zhang et al. (2019).

$$TEC = TE_{setup} + TE_{process} + TE_{idle} \tag{25}$$

$$TE_{setup} = \sum_{j}\sum_{\mathcal{I}}\sum_{w}\sum_{s}\sum_{m}\left\{OE_{setup}^{j\mathcal{I}sm} \times pf_{w} \times \left[\int_{t=0}^{t=C_{max}} y_{j\mathcal{I}smw}(t).\rho(t).d(t)\right]\right\} \tag{26}$$

$$TE_{process} = \sum_{j}\sum_{s}\sum_{m}\left\{E_{process}^{jsm} \times \left[\int_{t=0}^{t=C_{max}} x_{jsm}(t).\rho(t).d(t)\right]\right\} \tag{27}$$

$$TE_{idle} = \sum_{m}\left\{E_{idle}^{i} \times \left[\int_{t=0}^{t=C_{max}} z_{m}(t).\rho(t).d(t)\right]\right\} \tag{28}$$

$OE_{setup}^{j\mathcal{I}sm}$ is the basis for calculating energy consumption per time unit for adjusting from job j to job $\mathcal{I}$ in stage s on machine m; pf_w denotes the performance factor of worker w assigned to undertake the adjustment; $t = 0$ and $t = C_{max}$ are the starting and finishing times of the scheduled jobs; and $\rho(t)$ counts power cost at time t. In Eq. (27), $E_{process}^{jsm}$ is the energy consumption per unit time for processing job j in stage s on machine m. In Eq. (28), E_{idle}^{i} is the energy consumption per unit time when machine m is in the idle state.

The total cost of workers (TW_{cost}) is defined using Eq. (29), which includes base salary and extra payment per busy time that is calculated by Eqs. (30)–(31), respectively.

$$TW_{cost} = TW_{base} + TW_{extra} \tag{29}$$

$$TW_{base} = \sum_{w} C_{max} \times S_w \times ActW_w \tag{30}$$

$$TW_{extra} = \sum_{j}\sum_{\mathcal{I}}\sum_{s}\sum_{w}\sum_{q} S_w \times st_{j\mathcal{I}s} \times pf_w \times \mathrm{Q}_{\mathcal{I}\mathrm{smq}} \tag{31}$$

Solution method

This section elaborates on the improved multi-objective optimization algorithm, AMOGA for solving the HFSPs with the time-of-use electricity price model, workforce flexibility, and SDST. The pseudocode of AMOGA is provided in Fig. 2, followed by a detailed explanation of the major computational elements.

Encoding, decoding, and initialization

A new scheme using a multi-layer chromosome of length $n_j \times n_s$ is proposed to encode the HFSPs with a flexible workforce. As shown in Fig. 3, a vector consisting of job sequence, machine selection, worker assignment, and finish time of job vectors represents a complete solution. In the illustrative example, the number of machines and workers in each stage is (3,1,2) and (2,1,1), respectively. The method introduced by Luo et al. (2013) is applied for decoding, where the optimal selection of resources is made considering both the makespan and total cost computed for different combinations of the available resources. Considering that the precise computation of electricity consumption cost is essential in this application area, the electricity cost of the machine in the idle state is considered in addition to the electricity cost in setup and processing states, as presented in Eq. (25). This technique not only reduces the complexity of the decoding process but also facilitates the generation of high-quality solutions.

The step-by-step implementation procedure of the proposed scheme is explained below.

Step 1. Create a four-level chromosome with empty values.
Step 2. Create a job sequence in the first stage using simple permutation.
Step 3. Select a prior job in the sequence.
Step 4. Assign machine and worker using the resource selection algorithm (RSA) shown in Fig. 4.

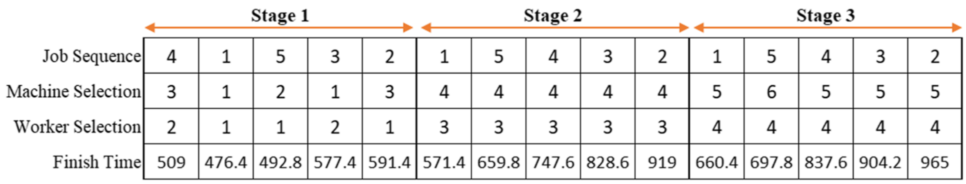
Fig. 2 Pseudocode of the developed algorithm, AMOGA

```
Input: Different instance size, AMOGA parameters
Output: A set of pareto-optimal solutions
Begin
Initialization
        1. Initialize the AMOGA parameters: population size (nPop), number of iterations (MaxIt), crossover probability (Pc), mutation probability (Pm);
        2. Initialize population with size nPop and calculate their fitness value;
        Assign fitness value using non-dominated sorting and crowding distance methods;
        Sort the population ascendingly considering ranks;
        While (not termination condition) do
Crossover
            If p < Pc then
                Select two parent individuals using Tournament selection method;
               I. Apply the PMX and Uniform crossover on the job sequence vector;
              II. Apply the RSA algorithm for resource selection;
               Calculate the fitness value;
            End
Mutation
            If p < Pm then
                Select one parent individual using Tournament selection method;
              I.  Apply the Mutation operator on the job sequence vector;
             II.  Apply the RSA algorithm for resource selectin;
               Calculate the fitness value;
            End
            Merge the solution groups (initial, recombined, and mutated solutions);
            Remove solutions with the same fitness (objective function) value;
            Assign fitness value considering non-dominated sorting and crowding distance;
            Sort population ascendingly considering the ranks;
            Select best-ranked individuals from the pool using Truncation selection;
            Assign fitness value using non-dominated sorting and crowding distance;
            Sort population ascendingly considering their ranks;
            Store the Pareto-front solution;
        End
Terminate
```

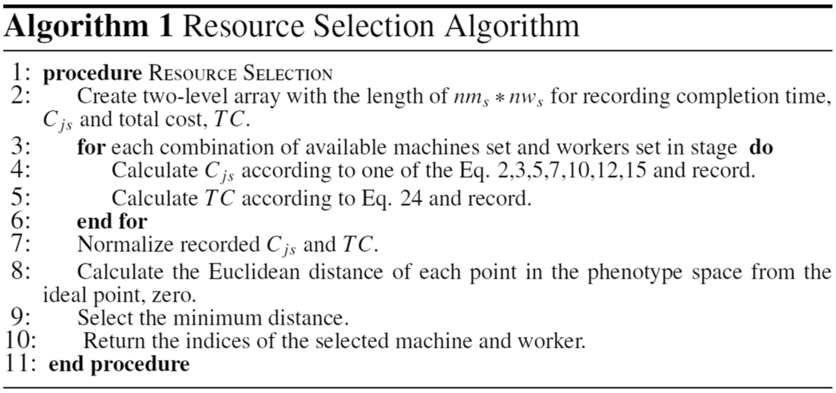
Step 5. Decode a partial encoding to a partial schedule and record the completion time of the job and indices of selected resources.
Step 6. Repeat Steps 3 to 5 for the remaining jobs in this stage.
Step 7. Arrange jobs in ascending order of their finish time for processing in the next stage.
Step 8. Repeat steps 3 to 7 for the remaining stages.
Step 9. Calculate and record start times, as well as the start and finish times of SDST for all jobs/machines in each stage.
Step 10. Decode the chromosome to a complete time–cost-efficient schedule and record the objective function values.

To provide a diverse and high-quality solution population, the sequence of jobs in the first stage is defined by a simple job permutation for every solution followed by the

Fig. 3 Structure of the four-layer chromosome encoding



	Stage 1					Stage 2					Stage 3				
Job Sequence	4	1	5	3	2	1	5	4	3	2	1	5	4	3	2
Machine Selection	3	1	2	1	3	4	4	4	4	4	5	6	5	5	5
Worker Selection	2	1	1	2	1	3	3	3	3	3	4	4	4	4	4
Finish Time	509	476.4	492.8	577.4	591.4	571.4	659.8	747.6	828.6	919	660.4	697.8	837.6	904.2	965



Algorithm 1 Resource Selection Algorithm

```
procedure Resource Selection
    Create two-level array with the length of $nm_s * nw_s$ for recording completion time, $C_{js}$ and total cost, $TC$.
    for each combination of available machines set and workers set in stage do
        Calculate $C_{js}$ according to one of the Eq. 2,3,5,7,10,12,15 and record.
        Calculate $TC$ according to Eq. 24 and record.
    end for
    Normalize recorded $C_{js}$ and $TC$.
    Calculate the Euclidean distance of each point in the phenotype space from the ideal point, zero.
    Select the minimum distance.
    Return the indices of the selected machine and worker.
end procedure
```

Fig. 4 Pseudocode of the resource selection algorithm

implementation of the RSA algorithm and rearrangement of the jobs in the following stages.

Fitness calculation and Pareto-front generation

In the developed algorithm, individuals with the best fitness value have a greater chance of being selected for reproduction and survival into the next generation. To assign a fitness value, the non-dominated sorting method introduced by Deb et al. (2002) is employed, which exploits the concept of Pareto dominance. This technique consists of pairwise comparisons of all the population members to form the fronts. The solutions that are not dominated by any other solutions belong to the first front (rank one) and represent the (near-) Pareto-optimal solutions. AMOGA tries to find a better set of Pareto-optimal solutions and finds solutions closer to the front (the so-called convergence) across computational generations. In addition to the convergence criterion, the crowding distance of individuals in their rank should be computed, using Eq. (32). It is worthwhile noting that the objective values need to be sorted in ascending order before computing the crowding distance of the solutions.

$$CD_r = \sum_{j=1}^{nOb} \frac{\left(f_k^{r+1} - f_k^{r-1}\right)}{\left(f_k^{max} - f_k^{min}\right)} \tag{32}$$

In this formulation, CD_r represents the crowding distance of r^{th} individual on the frontier, nOb is the number of the objectives, f_k^{r+1},f_k^{r-1} are the k^{th} objective values of the adjacent solutions, f_k^{max},f_k^{min} are the maximum and minimum values of the k^{th} objective. For the boundary solutions in each front, the crowding distance amounts to an infinite value. The solutions can be sorted by classifying them on different fronts and calculating the respective crowding distances. In this definition, the solutions with lower domination rank and higher crowding distance values are preferred. Solutions are first sorted in descending order of their crowding distances and then in ascending order of their front values.

Selection methods

The tournament selection method is used to select parents for reproduction. Choosing the size of the tournament determines whether elitism or diversity received more weight. The larger tournaments reflect the elite-selection policy and the fastest algorithm convergence. However, in smaller tournaments, the diverse-selection policy prevails. In this study, the size of the tournament is $T_{size} = 2$ individuals from which the one with the lowest rank is selected.

To select the best-ranked individuals from the pool, which contains the old population, pop_{old}, and the new individuals through the evolutionary phase, pop_{new}, the truncation function is applied. The population should be updated at the end of every generation to select the best individuals for the next generation. For this purpose, the existing population and the newly generated population from the recombination and mutation phases are first integrated. Then, individuals with the same fitness value (makespan and total cost) are removed. After that, the individuals' rank and the crowding distance are considered to select *nPop* top individuals, which will make it to the next generation(s).

Crossover operators

In the evolutionary phase of AMOGA, crossover operators are used for reproducing new (offspring) solutions from the selected parent solutions. This research considers two crossover operators with equal probability, namely partially mapped crossover (PMX) and uniform crossover. These operators are applied only to the part of the chromosome where the sequence of jobs in the first stage is presented. The machine and worker selection sections are then completed by the RSA algorithm which selects the best resources for the chosen job sequence respecting both objectives. The PMX operator is employed in such a way as to respect both preserves of ordering and information sharing from parents. For this purpose, first, a pair of parents is chosen by

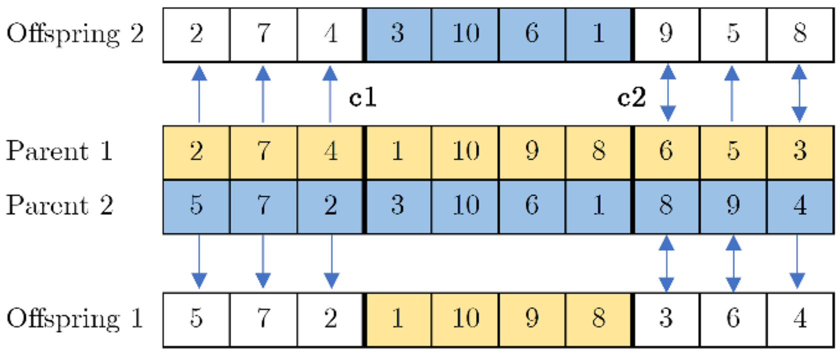


Fig. 5 Illustrative example of the partially mapped crossover (PMX)

tournament selection from the existing population, and two random cut points are selected uniformly. Then the mapping relationship of genes between the cut point is defined, and they are exchanged to form a partial offspring. Eventually, the remaining empty positions are filled by the original parents. In cases of having the same genes, they will be replaced based on the mapping relationship. The illustrative example in Fig. 5 clarifies the PMX procedure.

The uniform crossover begins with selecting two parents using tournament selection. A uniform binary vector, called a mask, with a length equal to the number of jobs (n_j) is then created. Finally, genes are copied from parents to the offspring as shown in Fig. 6.

Mutation operators

The mutation improves search diversity and helps escape premature convergence. This research applies three mutation operators named swap, conversion, and insertion on the job sequence section of the chromosomes for generating a new offspring. For this purpose, an individual is picked from the existing population by tournament selection, and then one of the mutation operators is selected randomly (with an equal probability). The procedure is illustrated in Fig. 7.

Overall, the RSA algorithm and a module for discarding duplicate solutions are the major points differentiating AMOGA from the basic multi-objective genetic algorithm. Besides, the encoding, initialization, and genetic operators are tailored for solving HFSPs.

Computational experiments

Benchmark description

To evaluate the performance of the developed algorithm, the state-of-the-art multi-objective evolutionary algorithms are considered benchmarks: SPEA2 (Zitzler et al. 2001) and Pareto envelop-based selection algorithm (PESA2; (Corne et al. 2001)). SPEA2 and PESA2 are selected because they use different approaches for managing the Pareto-front members in comparison with AMOGA; while AMOGA takes advantage of a single-based solution approach, SPEA2 uses *K*-nearest neighbor(s), and PESA2 uses a region-based approach. These algorithms are most-widely used for solving different classes of multi-objective scheduling problems.

The same solution encoding/decoding and genetic operators are used in all the benchmark algorithms to ensure a fair comparison. Besides, all the algorithms are adjusted for effective handling of discrete problems following (Corne et al. 2001; Zitzler et al. 2001). All the algorithms are coded and compiled on MATLAB R2020a and run on a personal computer with AMD Rayzen 3, 2.60 GHz CPU, and 12 GB Ram.

Test instances

Test data are generated randomly in a way that they can be representative of the shop floor. Table 1 shows the levels and ranges of the factors determining the configuration of the test instances. The test problems are denoted by "number of jobs–number of stages–number of machines in all stages-number of workers in all stages." For example, a problem with 20 jobs, 3 stages, 10 machines, and 5 workers can be identified by "20–3-10–5."

The time-of-use plan rates for small and medium-sized enterprises (SMEs) are used in this study, which is illustrated in Fig. 8. According to the figure, the peak electricity demand increases during the summer months (i.e., June to September), as well as the afternoon time (i.e., 4 to 9 pm); hence, the associated energy cost is higher. On this basis, five time-of-use periods are considered for energy cost calculation.where the following rates apply in power cost calculations:

- $\rho(t) = 0.10497 \text{ if } 24n \le t < 24n + 14$
- $\rho(t) = 0.12476\text{if}24n + 14 \le t < 24n + 16$
- $\rho(t) = 0.15311\text{if}24n + 16 \le t < 24n + 21$
- $\rho(t) = 0.12476\text{if}24n + 21 \le t < 24n + 23$
- $\rho(t) = 0.10497\text{if}24n + 23 \le t < 24n + 24$

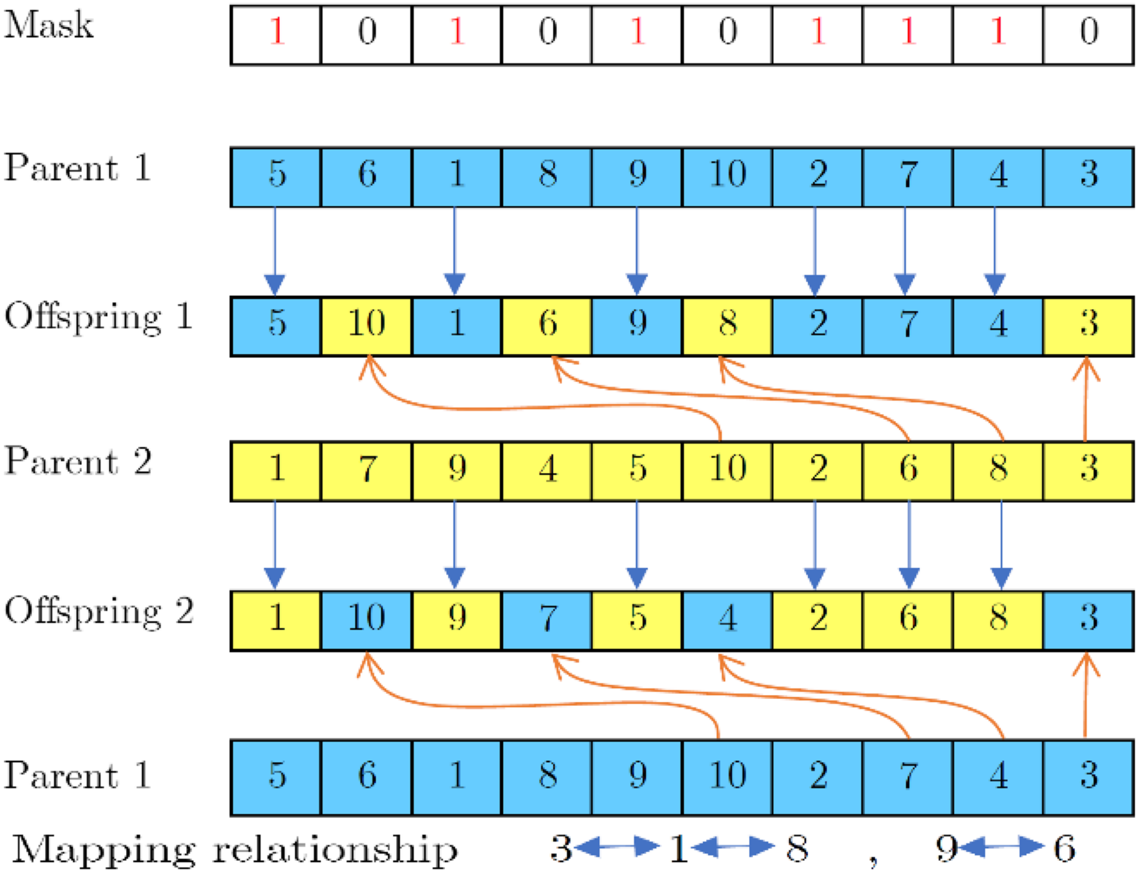


Fig. 6 Illustrative example of the uniform crossover

Comparison metrics for algorithm evaluation

To compare the performance of the multi-objective evolutionary algorithms, various metrics should be considered. In this study, five metrics, namely spacing metric (*SM*), mean ideal distance (*MID*), diversification metric (*DM*), quality metric (*QM*), and inverted generational distance (*IGD*), are used, as considered in earlier studies (Cheng et al. 2020) to benchmark the performance of AMOGA against SPEA2 and PESA2. These metrics are detailed below.

SM measures the uniformity of the spread of non-dominated schedules, which can be computed using Eq. (33).

$$SM = \frac{\sum_{i=1}^{n-1}\left|\overline{d} - d_i\right|}{(n-1)\times\overline{d}} \tag{33}$$

Fig. 7 Illustration of the mutation operators: (**a**) swap, (**b**) insertion, and (**c**) reversion

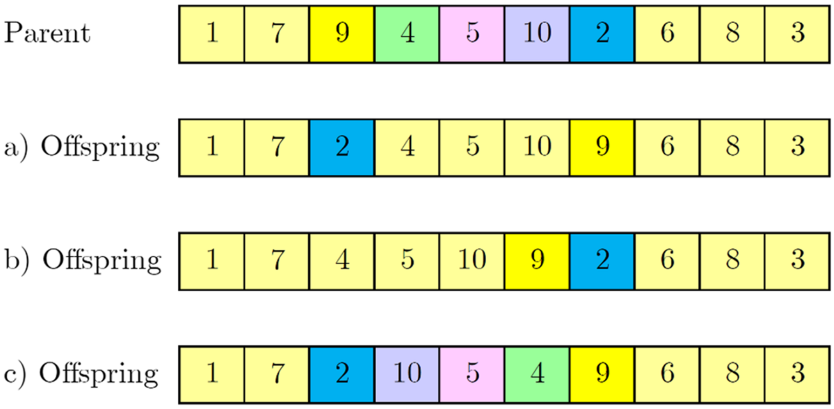


Table 1 Summary of the test data

Factors	Levels	Number of levels
Number of jobs	10, 20, 40, 60, 80, and 100	6
Number of stages	2, 3, 5, and 10	4
Number of machines at each stage	1, 2, 3, 4, and 5	5
Number of workers at each stage	Floor ((no of machine + 1)/2)	1
Processing units of each operation	Discrete uniform [30,100]	1
Sequence-dependent setup time	Discrete uniform [1,5]	1
Performance factor of workers	0.8, 1.0, 1.2	3
Worker cost ($/min)	0.2, 0.17, 0.15	3
Power of machine in process state	Discrete uniform [8,10]	1
Power of machine in setup state	Discrete uniform [5,7]	1
Power of machine in idle state	Discrete uniform [3,5]	1

In this equation, d_i is the Euclidean distance between two successive non-dominated schedules in the solution set and $\overline{d}$ represents the average of Euclidean distances. The smaller value of this metric is preferred (Piroozfard et al. 2018).

MID measures the proximity of the non-dominated solutions $\left(f_{1,i},f_{2,i}\right)$ from the ideal point $\left(f_1^{best},f_2^{best}\right)$ and is calculated by Eq. (34).

$$MID = \frac{\sum_{i=1}^{n}\sqrt{\left(f_{1,i}+f_1^{best}\right)^2+\left(f_{2,i}+f_2^{best}\right)^2}}{n} \tag{34}$$

where $f_{1,i}$ and $f_{2,i}$ denote the fitness function of the *ith* non-dominated solution found by each algorithm (first and second objectives); f_1^{best},f_2^{best} indicate the best fitness function of the first and second objectives among all solutions of the competing algorithms, and n refers to the total number of Pareto front solution find by each algorithm. A smaller value of MID is desired (Nabipoor Afruzi et al. 2013).

DM determines the extent of heterogeneity in the set of non-dominated schedules for each algorithm, which can be calculated using Eq. (35); higher values of this metric is advantageous (Tavakkoli-Moghaddam et al. 2011).

$$DM = \sqrt{\sum_{i=1}^{nOb}\left(max(f_i)+min(f_i)\right)^2} \tag{35}$$

QM shows the percentage of unique non-dominated schedules obtained by each algorithm. To calculate this, a union of obtained Pareto-front solutions of all competing algorithms is first determined to form a new set of non-dominated solutions. Then, the number of solutions obtained by each algorithm in each new Pareto-front is counted and divided by the total number of solutions in the union set. The higher the value of this metric is, the more unique solutions are obtained by the respective algorithm (Moradi et al. 2011).

IGD simultaneously considers both convergence and diversity aspects to measure how far the Pareto-front obtained by an algorithm is from the reference Pareto-front

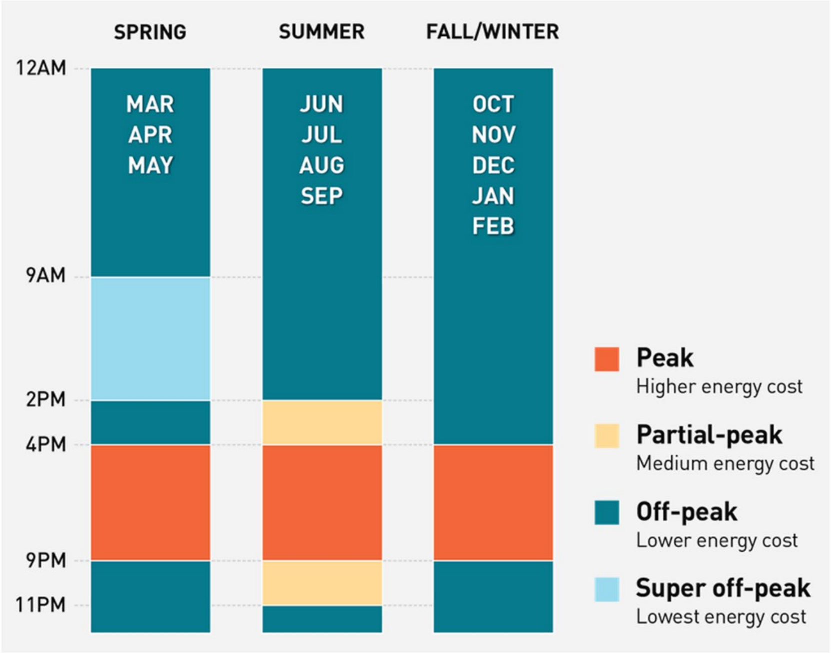


Fig. 8 Time-of-use plan rates for small-medium business (Source: Pacific Gas and Electric Company)

Table 2 Factor levels considered in the calibration of AMOGA

Parameter	Level		
	1	2	3
nPop	50	80	100
Pc	0.7	0.8	0.9
Pm	0.1	0.2	0.3
T_{size}	2	3	4

solutions. It is computed by Eq. (36) with the lower value being more desirable (Coello Coello and Reyes Sierra 2004).

$$IGD = \frac{\sqrt{\sum_{i=1}^{n^*}(d_i^*)^2}}{n^*} \tag{36}$$

In this equation, d_i^* is the Euclidean distance of each Pareto-front solution to the nearest schedule obtained by each algorithm, and n^* is the number of Pareto-front solutions. It is worth noting that in cases with unknown Pareto-front solutions, *IGD* can be estimated considering the integrated Pareto-front solutions found by all benchmark algorithms.

Parameter settings

Separate experiments are conducted to calibrate the parameters of each algorithm. For AMOGA, the population size, tournament selection configuration, crossover, and mutation probabilities are tuned. The parameters of SPEA2 include the population size, archive size, crossover probability, and mutation probability. For PESA2, the grid number needs to be considered in addition to the parameters of SPEA2. The Taguchi method is used for determining the optimal level of important known factors while the effects of uncontrollable factors are minimized. For a full description of how to perform the Taguchi method for tuning the algorithm's parameters, we refer the interested readers to the research conducted by Nabipoor Afruzi et al. (2013).

As a first step, trial-and-error experiments are conducted to determine the initial values for each parameter. Table 2 shows the parameter levels of AMOGA. Considering the number of known parameters, the respective test levels, and the Taguchi method, an orthogonal array L9 then is designed, which consists of different configurations of parameter levels. To assess which combination of parameter values results in the best algorithm performance, four randomly generated test problems of different scales (HS04, HS06, HS10, HS14) are considered. To ensure the reliability of the outcomes, each problem is solved five times, providing a total of 20 results for each trial.

Next, the performance metrics explained in the previous section are computed and normalized by the relative deviation index for every instance, calculated using Eq. (37). The average values associated with each trial are then determined. These performance indicators have different degrees of importance based on how they measure the quality and diversity of the Pareto-front solutions for solving a multi-objective optimization problem. SM = 1, DM = 1, MID = 2, QM = 3, and IGD = 4 are considered to measure the weighted average relative deviation index (WARDI; Eq. 38) for every item of the orthogonal array.

$$RDI_i = \frac{\left|\overline{Pc_i} - Pc_i^{best}\right|}{\left|Pc_i^{max} - Pc_i^{min}\right|} \times 100 \tag{37}$$

$$\mathrm{WA}RDI_i = \frac{\sum_{i=1}^{nM}\left(\mathrm{w}_i \times \overline{RDI_i}\right)}{\sum_{i=1}^{nM}\mathrm{w}_i} \times 100 \tag{38}$$

In these equations, $\overline{Pc_i}$ represents the average of performance criterion associated with the *ith* experiment over 20 runs; Pc_i^{best} is the best performance metric among all experimental results; Pc_i^{max} and Pc_i^{min} denote the maximum (and minimum) values of a certain performance metric among all experiments. It should be noted that WARDI_i is calculated for the *ith* experiment with $\overline{RDI_i}$ representing the average of RDIs of the four test problems, and *nM* referring to the number of metrics. The results are used as the response value in the Taguchi design approach, according to which the best combination of parameter values for AMOGA can be obtained.

The same procedure has been implemented for SPEA2 and PESA2. It should be noted that the levels of the common factor of SPEA2 and PESA2 are similar to AMOGA while the archive size levels for these two algorithms are L1(50)-L2(80)-L3(100). For PESA2, the potential levels of the grid's number are L1(20)-L2(40)-L3(60). The results of the Taguchi experiments for AMOGA, SPEA2, and PESA2 are summarized in Figs. 9, 10, and 11, respectively. Table 3 presents the optimal parameter values for each algorithm.

Experimental results and discussions

Given the calibration outcomes, this section compares the performance of the algorithms considering a total of twenty-and-four test instances of various scales. Table 4 summarizes the computational results of AMOGA, SPEA2, and PESA2 considering different performance measures, SM, MID, DM, QM, and IGD. The best outcomes in the benchmark are in bold for an easy track.

A statistical test of significance is then conducted to confirm if the difference is significant. Looking at the

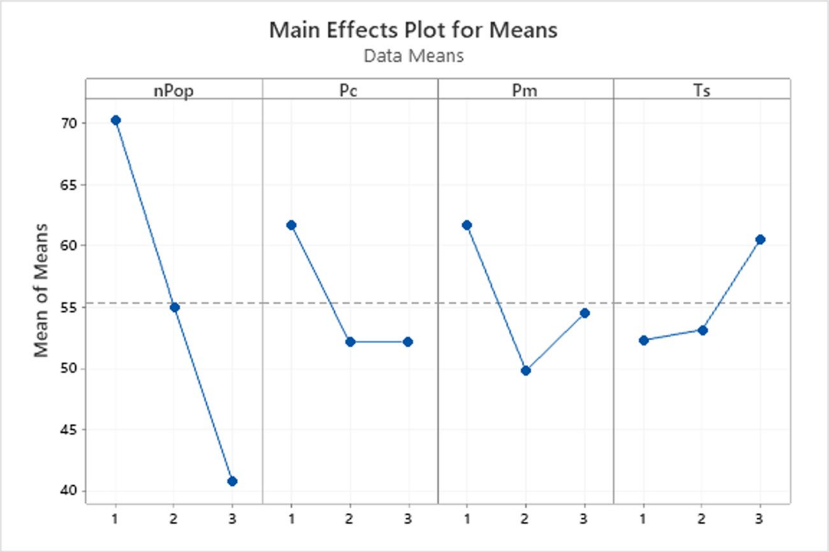


Fig. 9 The mean WARDI plot for each level of the parameters in Taguchi methodology-AMOGA

first criterion, SM, it is observed that the PESA2 outperforms both AMOGA and SPEA2 on 19 and 15 out of the 24 test instances, respectively. That is, better uniformity of the spread of non-dominated solutions can be obtained by PESA2. The reason behind the superiority of PESA2 lies in the fact that it benefits from a region-based approach, which leads to yielding results with a more uniform distribution compared with the other two algorithms. However, the difference is not statistically significant (null hypothesis is accepted with statistic = 2.5249 and p-value = 0.0874).

Concerning the second criterion, MID, AMOGA shows a relatively better performance with an average of 235.61 compared with 308.75 and 343.57 of SPEA2 and PESA2, respectively. Considering the number of instances, AMOGA and SPEA2 are comparable and outperform PESA2 in 19 and 16 instances, respectively. Although the statistical test does not confirm a significant difference between the overall performances, AMOGA results are more competitive for large-scale instances (HFSP/W20-24). That is, the proximity of the non-dominated solutions obtained by AMOGA from the ideal point is comparatively better for large-scale instances. The outperformance concerning MID is expected to be even more significant when industry-scale instances are solved.

Concerning the third metric (i.e., DM), it is observed that the diversity of the Pareto-front solution obtained by both AMOGA and SPEA2 is better than PESA2 with 20 and 18 out of the 24 tests being outperformed, respectively. Overall, AMOGA obtained better more diverse solutions with a DM value of 608.12, and it can be confirmed that

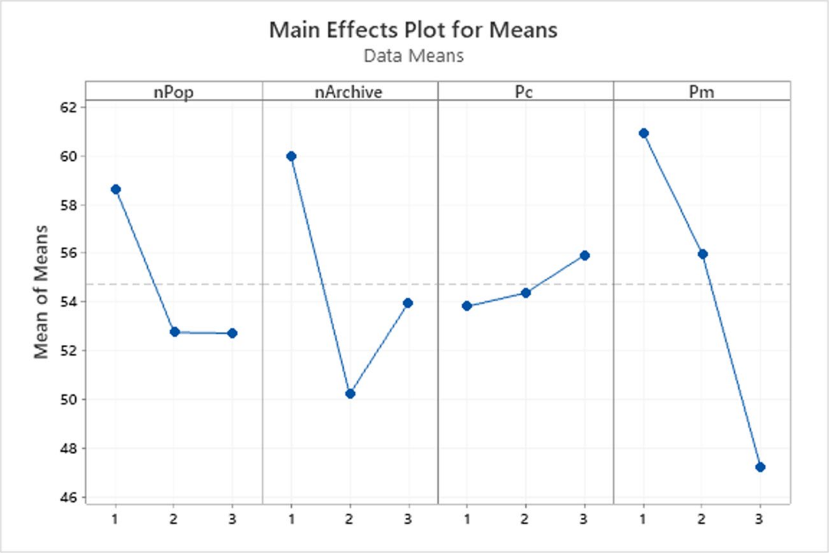


Fig. 10 The mean WARDI plot for each level of the parameters in Taguchi methodology-SPEA2

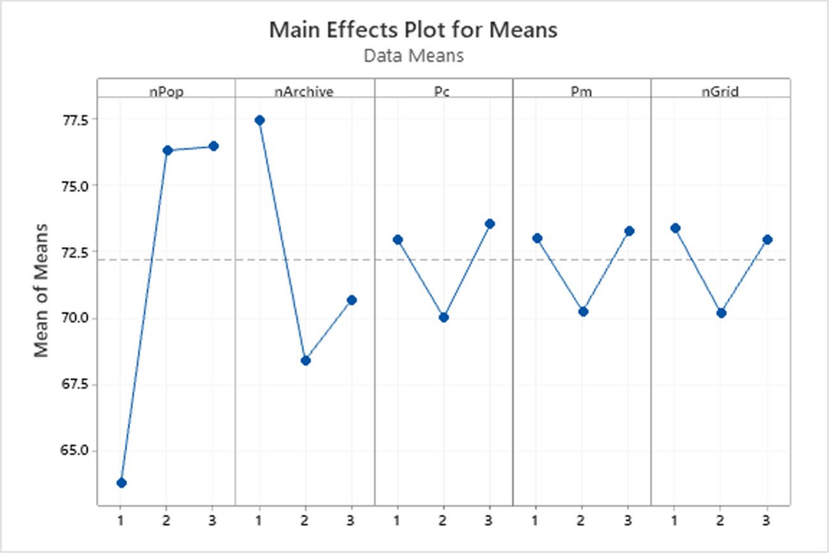


Fig. 11 The mean WARDI plot for each level of the parameters in Taguchi methodology-PESA2

AMOGA performs significantly better than SPEA2 with statistic = 3.4531, *p*-value = 0.0012.

Considering the number of unique non-dominated solutions obtained by each algorithm, i.e., the QM indicator, it is evident that AMOGA outperforms both SPEA2 and PESA2. Obtaining better QMs in all 24 test instances proves the superiority of AMOGA for exploring harder solution spaces, like non-convex areas. It is worth noting that an average of about 81% of non-dominated solutions were yielded by AMOGA, compared with only 9 and 11% for SPEA2 and PESA2, respectively. With 99% confidence, a statistic value of 243.5465 and a *p*-value of 0.0000 the null hypothesis are rejected, confirming that AMOGA performs significantly better in terms of finding unique non-dominated solutions compared to SPEA2 and PESA2.

In terms of the last performance metric, IGD, analysis of the results shows that AMOGA outperforms PESA2 in 23 test instances and SPEA2 yielded better convergence and diversity of solutions only for solving HFSP/W17. Considering the extent of the difference, which is 2.78 against 90.50 and 99.09, one can conclude that AMOGA is also superior considering IGD. Finally, the statistic value of 9.46145 and a *p*-value of 0.0002 confirm, with 99% confidence, that AMOGA showed better convergence and more diverse solutions compared to SPEA2 and PESA2.

Table 3 Parameter setting of AMOGA, SPEA2, and PESA2

Parameter	AMOGA	SPEA2	PESA2
nPop	100	100	50
Pc	0.9	0.7	0.8
Pm	0.2	0.3	0.2
T_size	2	-	-
nArchive	-	80	80
nGrid	-	-	40

To visually compare the non-dominated solutions obtained by the benchmark algorithms, six test instances of different scales, i.e., HFSP/W03, HFSP/W06, HFSP/W08, HFSP/W12, HFSP/W16, HFSP/W19, and HFSP/W24) are presented in Fig. 12a–f. One can observe that most non-dominated solutions obtained by AMOGA are better than those of SPEA2 and PESA2 in terms of production efficiency and saving total cost. That is, the proposed AMOGA appears to be more effective for solving different scales of the studied problem.

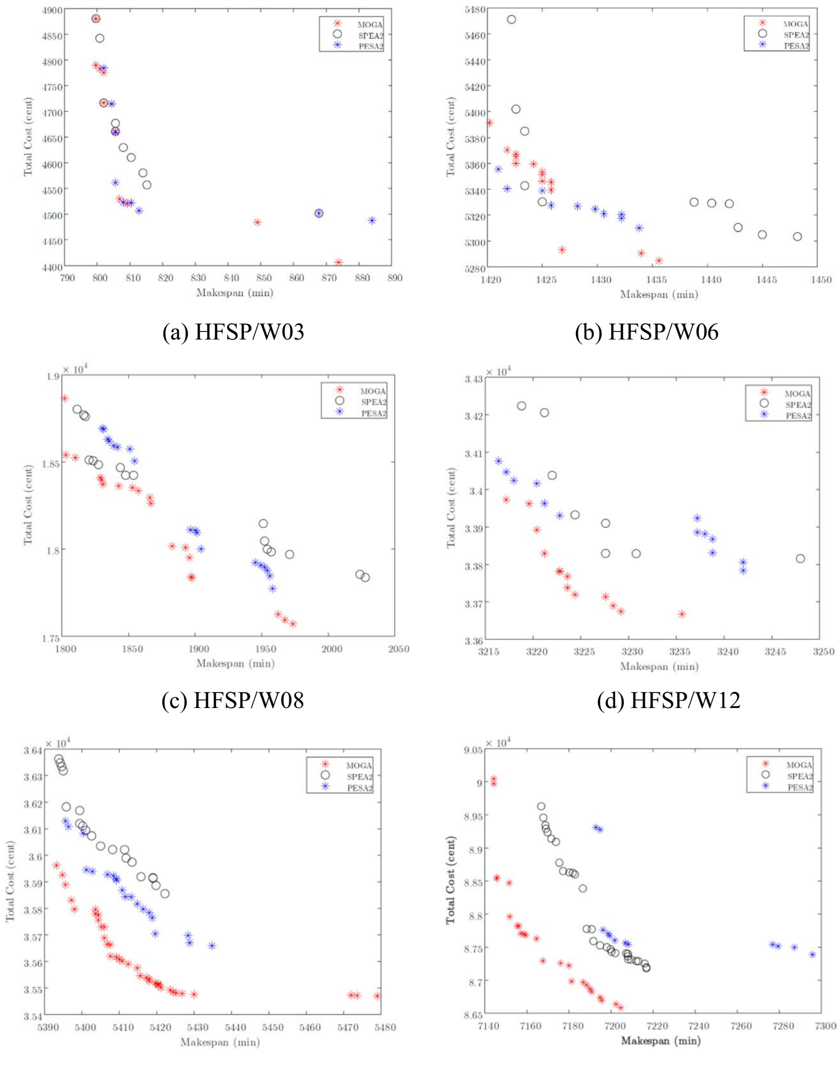
Furthermore, Fig. 13a–b displays the trend of the inverted generational distance of each algorithm for HFSP/W20 and HFSP/W24 instances, which showcases a better convergence give a fixed number of iterations. Finally, sample Gant Charts of the non-dominated solution for HFSP/W08 and HFSP/W11 instances are provided in Fig. 14 to showcase the production schedules obtained by our optimization approach.

Given a diverse set of near-optimum non-dominated schedules, production managers can flexibly weigh up between minimizing the total costs and the maximum completion time to select the solution that suits the current operational needs of the company. Taking mandates on the energy consumption and environmental performance as an example, the production manager may choose a schedule with lower energy consumption—but higher makespan—in normal conditions. However, when the delivery time of certain orders is a priority, the production schedule with a minimum makespan may be selected. Workforce productivity is another major consideration; having highly productive workers for the production orders that require substantial setups is one way to reconciliation between cost-effectiveness and responsiveness strategies.

Table 4 Computational results of AMOGA, SPEA2, and PESA2 based on performance metrics (best in bold)

Instance	Configuration	SM			MID			DM			QM			IGD		
		AMOGA	SPEA2	PESA2	AMOGA	SPEA2	PESA2	AMOGA	SPEA2	PESA2	AMOGA	SPEA2	PESA2	AMOGA	SPEA2	PESA2
W01	10–2-6–4	1.17	**0.00**	0.91	62.90	**61.32**	83.97	120.11	**122.64**	121.33	**0.80**	0.20	0.00	**0.00**	5.92	3.20
W02	10–3-9–5	0.96	**0.92**	1.11	74.04	**67.51**	121.06	**159.91**	121.02	129.03	**0.85**	0.08	0.08	**0.13**	16.59	15.85
W03	10–5-15–9	0.84	0.564	**0.561**	233.95	267.28	**184.92**	**480.02**	385.37	307.77	**0.65**	0.09	0.26	**0.00**	26.73	11.34
W04	10–10-25–16	0.87	**0.71**	0.85	565.28	**410.14**	712.40	**1014.82**	875.94	714.93	**0.78**	0.11	0.11	**1.67**	143.34	86.83
W05	20–2-4–3	0.49	**0.27**	0.53	**14.47**	18.11	14.53	**28.90**	11.67	15.05	**0.63**	0.13	0.25	**1.42**	7.28	4.40
W06	20–3-7–4	0.88	0.80	**0.67**	61.26	60.86	**44.59**	107.59	**169.70**	47.11	**0.50**	0.13	0.38	**3.71**	10.98	12.73
W07	20–5-16–10	0.80	1.16	**0.33**	206.13	**170.11**	264.37	**394.70**	162.82	196.19	**0.58**	0.25	0.17	**8.62**	52.85	70.24
W08	20–10-30–18	0.98	0.98	**0.94**	**621.76**	747.30	689.47	**1305.16**	989.26	928.10	**0.79**	0.17	0.04	**7.32**	87.61	99.10
W09	40–2-4–3	0.97	0.73	0.47	134.82	39.86	162.07	247.27	41.60	42.93	**0.90**	0.10	0.00	**0.00**	83.76	69.25
W10	40–3-10–6	0.70	0.74	0.50	69.28	57.85	73.57	110.58	177.27	72.31	**0.74**	0.00	0.25	**0.70**	13.22	6.78
W11	40–5-12–8	1.24	1.15	**0.87**	351.47	**271.37**	290.42	**1017.18**	824.37	686.73	**0.77**	0.05	0.18	**2.49**	40.37	58.09
W12	40–10-27–17	0.69	0.83	0.42	119.16	137.93	259.59	306.50	409.47	294.16	**0.81**	0.06	0.13	**11.82**	65.82	41.29
W13	60–2-3–2	0.96	**0.77**	0.82	55.92	**55.03**	143.53	135.97	**199.18**	73.65	**0.91**	0.09	0.00	**5.38**	24.64	68.81
W14	60–3-5–4	1.70	**1.43**	1.53	**195.87**	200.01	324.75	653.17	**713.59**	679.82	**0.94**	0.06	0.00	**0.00**	78.78	39.33
W15	60–5-12–7	1.02	**0.39**	0.60	**267.41**	524.82	376.35	535.05	138.81	**656.41**	**0.95**	0.05	0.00	**0.00**	240.38	74.38
W16	60–10-24–15	**0.73**	1.15	0.87	657.57	657.01	**511.63**	**1400.37**	726.35	528.93	**0.59**	0.00	0.41	**6.54**	149.00	121.58
W17	80–2-3–2	1.20	0.91	**0.65**	71.42	143.54	**51.47**	199.09	**263.37**	83.00	**0.45**	0.20	0.35	16.86	**14.37**	29.27
W18	80–3-6–4	0.72	0.65	**0.50**	102.83	**80.60**	189.83	**214.55**	20.09	48.02	**0.96**	0.04	0.00	**0.00**	44.21	77.15
W19	80–5-15–9	0.82	0.70	**0.68**	167.11	**154.56**	397.72	500.12	**507.65**	471.84	**0.97**	0.03	0.00	**0.00**	238.93	90.45
W20	80–10-30–19	0.87	1.25	**0.45**	**237.92**	775.36	809.30	572.38	**1102.44**	475.50	**0.94**	0.06	0.00	**0.00**	263.72	353.26
W21	100–2-3–2	0.89	0.91	**0.63**	**63.26**	132.61	216.96	**201.99**	194.90	112.86	**0.97**	0.03	0.00	**0.00**	39.00	120.04
W22	100–3-8–5	**0.71**	0.95	1.15	**125.78**	246.05	311.23	306.64	**914.74**	343.59	**0.96**	0.04	0.00	**0.00**	60.34	115.97
W23	100–5-13–8	1.39	1.19	**0.73**	**152.72**	638.26	726.02	**1122.00**	595.43	478.98	**0.94**	0.06	0.00	**0.00**	251.73	393.56
W24	100–10-31–19	1.10	**0.95**	1.34	**1042.29**	1492.47	1285.82	**3460.87**	2452.13	1929.54	**0.96**	0.04	0.00	**0.00**	212.43	415.28
Average	-	0.95	0.84	**0.75**	**235.61**	308.75	343.57	**608.12**	504.99	393.24	**0.81**	0.09	0.11	**2.78**	90.50	99.09

Fig. 12 Non-dominated solutions by AMOGA, SPEA2, and PESA2



(a) HFSP/W03

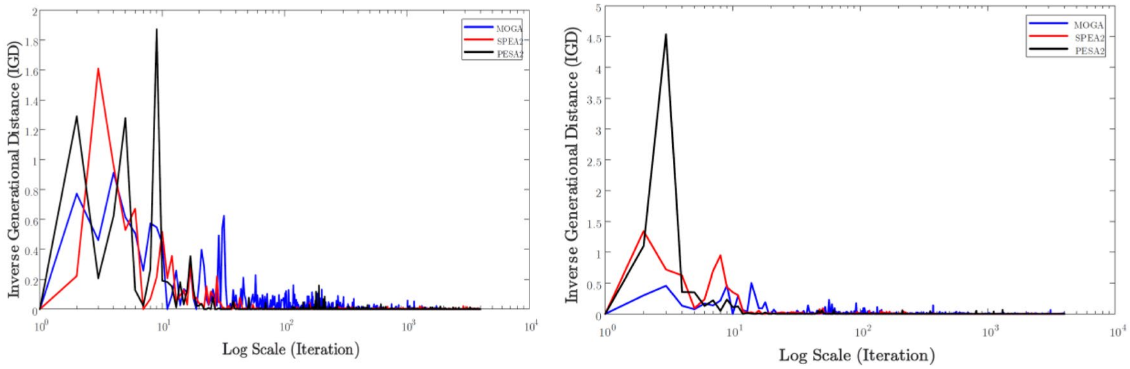
(b) HFSP/W06

(c) HFSP/W08

(d) HFSP/W12

(e) HFSP/W19

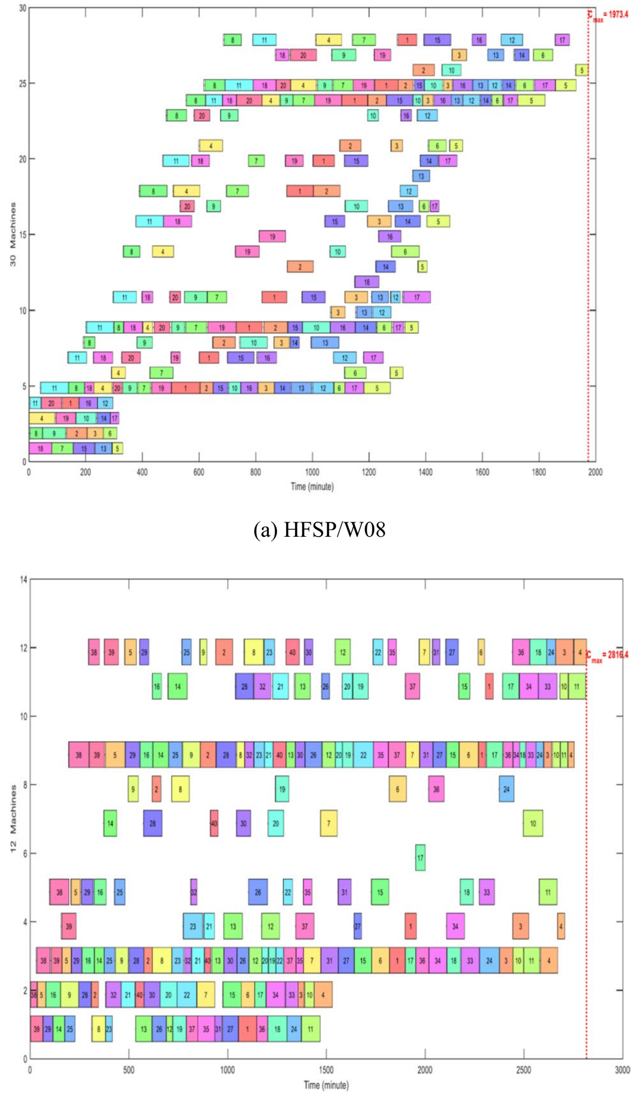
(f) HFSP/W24



(a) HFSP/W20

(b) HFSP/W24

Fig. 13 Inverse generational distance for AMOGA, SPEA2, and PESA2



(a) HFSP/W08

(b) HFSP/W11

Fig. 14 Visual illustration of a non-dominated solution of HFSP/W08 by AMOGA

Conclusions

The environmental concerns associated with the growing industrial activities have led to an array of initiatives seeking energy efficiency. As a prime example of net-zero initiatives, the time-of-use electricity policy is pushing manufacturing shop owners to reduce energy consumption. Both value-adding (i.e., production) and non-value-adding operations (i.e., setup and preparation) contribute to energy consumption on the shop floor. Production planning and control methods that address such concerns are of utmost relevance to industrial sustainable development. Besides, the workers' capabilities impact the energy efficiency of production operations; simultaneously incorporating these features improve the optimization outcomes. This study contributed to the literature by introducing an effective multi-objective optimization approach for scheduling hybrid flow-shops, considering energy consumption under time-of-use pricing models, workers' flexibility, and SDST features simultaneously. The objective was to minimize total costs comprising electric consumption and worker costs pertinent to the setup operations while minimizing the response time for fulfilling new demands. To the best of the authors' knowledge, this is the first study that integrates these factors.

To evaluate the performance of the proposed algorithm, different operational parameters and performance measures are considered within a comprehensive computational experiment. Comparing the performance of AMOGA with two state-of-the-art multi-objective evolutionary algorithms, SPEA2 and PESA2, we showed that most of the non-dominated solutions were yielded by AMOGA considering various tradeoffs between the optimization objectives. AMOGA proved to be more effective for solving various scales of HFSP instances considering MID, DM, QM, and IGD measures, which are advantageous for providing wider trade-offs for production planning, while PESA2 showed to perform better in terms of SM.

Future research may extend our study in three directions. First, our study is limited in that it considers a static time-of-use electricity pricing model. Machine learning can be used to predict future power consumption in the region and adjust the production schedules accordingly. Machine learning can also be used for weather forecasting in cases of using renewable energies, like solar panels and wind turbine. Second, the energy pricing model should be incorporated into other classes of scheduling problems, like in unrelated parallel machines. In so doing, situation-specific operational constraints, like a capacitated buffer in each stage, machine failure, and workers' availability constraints may be considered to explore their impact on the energy consumption factor. The proposed incorporation can also be studied in rescheduling models to reduce energy consumption in times of energy shortage and/or power disruption. In terms of the objective function, a balance of workers' workload and minimizing earliness/tardiness can be considered to highlight the possible changes in the energy cost in pick hours. Our study focused on the multi-objective feature of managing the Pareto-front members for which AMOGA was compared with SPEA2 and PESA2; the future research may consider comparing other search features to compare AMOGA with the multi-objective evolutionary algorithms based on decomposition (MOEA/D) and multi-objective particle swarm optimization (MOPSO). As a final suggestion for future direction, other methods of dealing with multiple conflicting objectives should be developed and compared with the Pareto-based method developed in our study. For this purpose, some of the properties of multi-objective decision-making models may be inspiring for further development in the multi-objective optimization field.

Author contribution All authors contributed to the study's conception. Material preparation, data collection, and analysis were performed by Ali Mokhtari-Moghaddam. Investigations was done by Pourya Pourhejazy. The model extension was performed by Ali Mokhtari-Moghaddam and Deepak Gupta. The submitted draft is written and revised by Pourya Pourhejazy. All authors read and approved the final manuscript.

Funding Open access funding provided by UiT The Arctic University of Norway.

Data availability The raw/processed data required to reproduce these findings can be provided upon reasonable request.

Declarations

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Competing interests The authors declare no competing interests.

References

Adler L, Fraiman N, Kobacker E et al (1993) BPSS: a scheduling support system for the packaging industry. Oper Res 41:641–648. https://doi.org/10.1287/opre.41.4.641

Chen L, Bostel N, Dejax P et al (2007) A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. Eur J Oper Res 181:40–58. https://doi.org/10.1016/j.ejor.2006.06.033

Cheng C-Y, Lin S-W, Pourhejazy P et al (2020) Greedy-based non-dominated sorting genetic algorithm III for optimizing single-machine scheduling problem with interfering jobs. IEEE Access 8:142543–142556. https://doi.org/10.1109/ACCESS.2020.3014134

CoelloCoello CA, Reyes Sierra M (2004) A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In: Monroy R, Arroyo-Figueroa G, Sucar LE, Sossa H (eds) MICAI 2004: Advances in Artificial Intelligence. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 688–697

Corne DW, Jerram NR, Knowles JD, Oates MJ (2001) PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO'01). Morgan Kaufmann Publishers Inc., San Francisco, CA, pp 283–290

Dai M, Tang D, Giret A et al (2013) Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. Robot Comput Integr Manuf 29:418–429. https://doi.org/10.1016/j.rcim.2013.04.001

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197. https://doi.org/10.1109/4235.996017

Gong G, Chiong R, Deng Q et al (2020) Energy-efficient flexible flow shop scheduling with worker flexibility. Expert Syst Appl 141:112902. https://doi.org/10.1016/j.eswa.2019.112902

Gong G, Deng Q, Gong X et al (2018a) A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators. J Clean Prod 174:560–576. https://doi.org/10.1016/j.jclepro.2017.10.188

Gong X, Deng Q, Gong G et al (2018b) A memetic algorithm for multi-objective flexible job-shop problem with worker flexibility. Int J Prod Res 56:2506–2522. https://doi.org/10.1080/00207543.2017.1388933

Gong G, Deng Q, Gong X, Huang D (2021) A non-dominated ensemble fitness ranking algorithm for multi-objective flexible job-shop scheduling problem considering worker flexibility and green factors. Knowl-Based Syst 231:107430. https://doi.org/10.1016/j.knosys.2021.107430

Jin ZH, Ohno K, Ito T, Se E (2002) scheduling hybrid flowshops in printed circuit board assembly lines*. Prod Oper Manag 11:216–230. https://doi.org/10.1111/j.1937-5956.2002.tb00492.x

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

Meng L, Zhang C, Zhang B, Ren Y (2023) Mathematical modeling and optimization of energy-conscious flexible job shop scheduling problem with worker flexibility. IEEE Access 7:68043–68059. https://doi.org/10.1109/ACCESS.2019.2916468

Li X, Lu C, Gao L et al (2018) An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop. IEEE Trans Ind Informatics 14:5400–5409. https://doi.org/10.1109/TII.2018.2843441

Lin S-W, Cheng C-Y, Pourhejazy P et al (2021) New benchmark algorithm for hybrid flowshop scheduling with identical machines. Expert Syst Appl 183:115422. https://doi.org/10.1016/j.eswa.2021.115422

Xue L, Wang X (2023) A multi-objective discrete differential evolution algorithm for energy-efficient two-stage flow shop scheduling under time-of-use electricity tariffs. Appl Soft Comput 133:109946. https://doi.org/10.1016/j.asoc.2022.109946

Linn R, Zhang W (1999) Hybrid flow shop scheduling: a survey. Comput Ind Eng 37:57–61. https://doi.org/10.1016/S0360-8352(99)00023-6

Luo H, Du B, Huang GQ et al (2013) Hybrid flow shop scheduling considering machine electricity consumption cost. Int J Prod Econ 146:423–439. https://doi.org/10.1016/j.ijpe.2013.01.028

Meng L, Zhang C, Zhang B, Ren Y (2019) Mathematical modeling and optimization of energy-conscious flexible job shop scheduling problem with worker flexibility. IEEE Access 7:68043–68059. https://doi.org/10.1109/ACCESS.2019.2916468

Moghadam MA, Wong KY, Piroozfard H (2018) Solving a hybrid job-shop scheduling problem with space constraints and reentrant process by a genetic algorithm: a case study. Int J Ind Eng Theory Appl Pract 24(5). https://doi.org/10.23055/ijietap.2017.24.5.2371

Moghadam AM, Wong KY, Piroozfard H, et al (2014) Solving an industrial shop scheduling problem using genetic algorithm. In: Materials, Industrial, and Manufacturing Engineering Research Advances 1.1. Trans Tech Publications Ltd, pp 564–568. https://doi.org/10.4028/www.scientific.net/amr.845.564

Moradi H, Zandieh M, Mahdavi I (2011) Non-dominated ranked genetic algorithm for a multi-objective mixed-model assembly line sequencing problem. Int J Prod Res 49:3479–3499. https://doi.org/10.1080/00207540903433882

Mouzon G, Yildirim MB, Twomey J (2007) Operational methods for minimization of energy consumption of manufacturing equipment. Int J Prod Res 45:4247–4271. https://doi.org/10.1080/00207540701450013

NabipoorAfruzi E, Roghanian E, Najafi AA, Mazinani M (2013) A multi-mode resource-constrained discrete time–cost tradeoff problem solving using an adjusted fuzzy dominance genetic algorithm. Sci Iran 20:931–944. https://doi.org/10.1016/j.scient.2012.12.024

Pan Q, Wang L, Mao K et al (2013) An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process. IEEE Trans Autom Sci Eng 10:307–322. https://doi.org/10.1109/TASE.2012.2204874

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

Luo Q, Deng Q, Xie G, Gong G (2023) A Pareto-based two-stage evolutionary algorithm for flexible job shop scheduling problem with worker cooperation flexibility. Robot Comput Integr Manuf 82:102534. https://doi.org/10.1016/j.rcim.2023.102534

Ribas I, Leisten R, Framiñan JM (2010) Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. Comput Oper Res 37:1439–1454. https://doi.org/10.1016/j.cor.2009.11.001

Ruiz R, Maroto C (2006) A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. Eur J Oper Res 169:781–800. https://doi.org/10.1016/j.ejor.2004.06.038

Ruiz R, Vázquez-Rodríguez JA (2010) The hybrid flow shop scheduling problem. Eur J Oper Res 1–18. https://doi.org/10.1016/j.ejor.2009.09.024

Salvador MS (1973) A solution to a special class of flow shop scheduling problems. In: Elmaghraby SE (ed) Symposium on the Theory of Scheduling and Its Applications. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 83–91

Tang D, Dai M, Salido MA, Giret A (2016) Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. Comput Ind 81:82–95. https://doi.org/10.1016/j.compind.2015.10.001

Tang L, Wang X (2011) A scatter search algorithm for a multistage production scheduling problem with blocking and semi-continuous batching machine. IEEE Trans Control Syst Technol 19:976–989. https://doi.org/10.1109/TCST.2010.2060201

Tavakkoli-Moghaddam R, Azarkish M, Sadeghnejad-Barkousaraie A (2011) A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem. Expert Syst Appl 38:10812–10821. https://doi.org/10.1016/j.eswa.2011.02.050

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 58:2283–2314. https://doi.org/10.1080/00207543.2019.1624857

Wardono B, Fathi Y (2004) A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. Eur J Oper Res 155:380–401. https://doi.org/10.1016/S0377-2217(02)00873-1

Wittrock RJ (1988) An adaptable scheduling algorithm for flexible flow lines. Oper Res 36:445–453. https://doi.org/10.1287/opre.36.3.445

Yan J, Li L, Zhao F et al (2016) A multi-level optimization approach for energy-efficient flexible flow shop scheduling. J Clean Prod 137:1543–1552. https://doi.org/10.1016/j.jclepro.2016.06.161

Yazdani M, Zandieh M, Tavakkoli-Moghaddam R (2019) Evolutionary algorithms for multi-objective dual-resource constrained flexible job-shop scheduling problem. Opsearch 56:983–1006. https://doi.org/10.1007/s12597-019-00395-

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 180:111–127. https://doi.org/10.1016/j.amc.2005.11.136

Zhang M, Yan J, Zhang Y, Yan S (2019) Optimization for energy-efficient flexible flow shop scheduling under time of use electricity tariffs. Procedia CIRP 80:251–256. https://doi.org/10.1016/j.procir.2019.01.062

Zheng X, Wang L (2016) A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem. Int J Prod Res 54:5554–5566. https://doi.org/10.1080/00207543.2016.1170226

Zhang TW, Dornfeld DA (2007) Energy use per worker-hour: evaluating the contribution of labor to manufacturing energy use. In: Takata S, Umeda Y (eds) Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses. Springer, London. https://doi.org/10.1007/978-1-84628-935-4_33

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. TIK-report 103. https://doi.org/10.3929/ethz-a-004284029

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