Energy-efficient model of hybrid flow-shop in manufacturing workshop and optimizing multi-objective multi-verse algorithm

Phung Tran Dinh [1*], Nguyen Manh Cuong [2]
[1-2] Faculty of Mechanical Engineering, Viet Hung Industrial University, Vietnam

* Corresponding Author: **Phung Tran Dinh**

Article Info

ISSN (online): 2582-7138
Volume: 05
Issue: 03
May-June 2024
Received: 10-04-2024
Accepted: 13-05-2024
Page No: 672-690

Abstract
Hybrid flow-shop scheduling problem (HFSP) is widely used in the field of intelligent manufacturing, which brings a variety of environmental benefits to the energy consumption problem of workshop through resource integration. In order to explore the energy-efficient potential of manufacturing workshop, an improved multi-objective multi-verse optimizer (IMOMVO) algorithm is proposed to solve the proposed multi-objective optimization HFSP model for energy-efficient based on mixed integer linear programming (MILP) model. In addition, a "shutdown-restart" energy-efficient strategy is proposed, which can reduce the energy consumption of the manufacturing workshop and prolong the life of the machine. Finally, through a test example, three manufacturing scenarios are designed. The improved genetic-simulated annealing algorithm (IGSA) in literature is compared with IMOMVO and the results validate the effectiveness and superiority of the proposed model and algorithm.

Keywords: IMOMVO; MILP; Energy-efficient; Hybrid flow-shop scheduling; Roulette wheel selection; shutdown-restart

1. Introduction

With the development of intelligent manufacturing technology, there are a lot of HFSP in manufacturing workshop production process (Gupta *et al*., 1991) [17]. Also known as flexible flow-shop scheduling problem (FFSP) (Lee *et al*., 1994) [26]. This problem was first raised by Salvador (1973) [40] in 1973 against the background of petroleum industry applications. The two-stage instance problem with at least one stage of parallel machines has been proved to be Non-deterministic Polynomial (NP)-hard problem (Gupta *et al*., 1988) [16]. And widely used in aerospace (Azami *et al*., 2018) [4], machinery (Zohali *et al*., 2019) [56], pharmacy (Zhi *et al*., 2018) [53], chemistry, steelmaking, textile, semiconductor and other industries. Therefore, the study of HFSP has important scientific research significance and engineering application value.

Intelligent algorithm usually uses fitness function to guide the algorithm to search solution space and solve it. It has the advantages of easy convergence, fast convergence speed and not easy to fall into local optimum. Therefore, more and more researchers tend to use intelligent algorithms to solve HFSP in view of the characteristics of NP-hard problems. For example, Qi *et al*. (2017) [39] proposed an improved hybrid variable neighborhood search genetic algorithm. Zhicong *et al*. (2016) [52] improved biogeography optimization algorithm to solve HFSP. Yunna *et al*. (2016) [48] designed an ant colony algorithm based on time window for multi-stage HFSP. The parameter setting of intelligent algorithm usually needs a lot of experiments to adjust, so some researchers put forward adaptive parameter adjustment algorithm. Chaoyong *et al*. (2017; 2019) [7] proposed the adaptive adjustment probability model, and designed the migratory bird optimization algorithm and efficient distribution estimation algorithm for solving HFSP. Hua *et al*. (2018) [20] proposed an adaptive genetic parameter adjustment strategy for improved genetic algorithm (IGA) to solve the reentry hybrid flow shop scheduling problem (RHFS-TWC) with the objective of minimizing the total weighted makespan. Zhonghua *et al*. (2016) [54] used two-population adaptive differential evolution algorithm to solve irrelevant parallel machine load balancing scheduling optimization problem. However, these algorithms often solve single-objective or non-energy-efficient objective problems.

Reasonable optimization of resource allocation can achieve the goal of improving efficiency, energy-efficient and emission reduction. Therefore, energy-efficient optimal scheduling is the key technology and research hot spot for enterprises to improve efficiency, increase efficiency and enhance competitiveness. Chaoyong *et al*. (2016; 2018) [9] proposed a MILP model aiming at minimizing energy consumption. In order to solve large-scale problems, a distribution algorithm estimation method based on a new decoding method is proposed proposed a new teaching-learning based optimization method (TTLBO) to solve the problem of energy-efficient and consumption reduction of HFSP proposed a non-dominant sorting genetic algorithm with variable local search (HNSGA-II) for low-carbon scheduling. For dynamic scheduling problem, Tang *et al*. (2016) [43] adopt improved particle swarm optimization algorithm to search Pareto optimal solution of dynamic flexible flow shop scheduling problem. Yan *et al*., (2016) [47] proposed an efficient multi-level optimization method for FFSP based on genetic algorithm and grey relational analysis. Liu *et al*, (2018) [29] proposed an optimization method based on non-dominant sequencing genetic algorithm II (NSGA-II) to solve the flexible flow shop model. Considering multi-objective problems, Zhang *et al*., (2019) [49] used energy-efficient flexible job shop scheduling (EFJSS) model based on non-dominated sequencing genetic algorithm II (NSGA-II) considering total production energy consumption and production cycle. It can be seen that most of the current research is single-objective problem or the mainstream of classical intelligent algorithms. In the process of solving, the algorithm is complex and needs a lot of iteration calculation. Multi-Verse Optimizer (MVO) is a new meta-heuristic algorithm. The algorithm is inspired by the theory of multiverse in physics. MVO algorithm was first proposed by Mirjalili and Seyedali (2016) [34] and applied to five practical engineering problems. Most of the experiment results are better than grey wolf optimization algorithm, particle swarm optimization algorithm, genetic algorithm and gravity search algorithm. Since then, MOMVO has been proposed (Mirjalili *et al*., 2017) [35] and compared with non-dominated sequential genetic algorithm (NSGA-II), differential evolution algorithm (MODE), multi-objective particle swarm optimization (MOPSO), multi-objective symbiotic biological search (MOSOS) and multi-objective collision body optimization (MOCBO) using test examples. The experiment results show that this method is superior to other algorithms. Since then, MVO and MOMVO algorithms have been applied to solve various problems, such as photonic crystal waveguide (PCW) related design (Mirjalili *et al*., 2016; 2017) [34, 35], Support Vector Machine (SVM) parameter optimization (Faris *et al*., 2016; Wang *et al*., 2018) [12, 44], parameter extraction of photovoltaic generator set (Ali *et al*., 2016), network-on-chip test scheduling (Hu *et al*., 2016), adjustment and optimization of artificial neural network (Peng *et al*., 2017) [38], parameter optimization of proton exchange membrane fuel cell (PEMFC) (Fathy *et al*., 2018) [13], parameter optimization of chaotic model, clustering optimization (Shukri *et al*., 2018) [42], parameter optimization of hybrid annual load forecasting model (Zhao *et al*., 2018) [51], Controller optimization (Kumar *et al*., 2018) [24], image segmentation (Abd Elaziz *et al*., 2019) [1] and so on. However, there is no relevant research on the application of MOMVO to HFSP.

Aiming at the characteristics of HFSP, the paper aims at energy-efficient multi-objective optimization HFSP model. The organization of the paper is as follows. Section 2 analyses the energy consumption of manufacturing workshop and establishes a multi-objective optimization HFSP model based on MILP model. Section 3 improves the MOMVO algorithm to solve multi-objective uncorrelated parallel machine problems. Section 4 chooses a test case to make an experiment comparison, which proves the validity of the model and the superiority of the method. Section 5 summarizes innovations and shortcomings of the paper and the future work.

2. Energy-efficient model of HFSP in manufacturing workshop

2.1. Mixed integer nonlinear programming model

2.1.1. Definition of parameters

In this section, an energy-efficient multi-objective optimization HFSP model is proposed, which can optimize the makespan and total energy consumption. To facilitate the establishment of the model, the relevant parameters of the model are shown in Table 1.

Table 1: Relevant parameters

Parameter	Definition
j	job number
n	number of jobs, n = 12
J	job set, {1,2,· · · n}, j ∈ J
m	machine serial number
k	number of machines, k = 9
k_s	the number of parallel machines in processing s stage,$k_1 = 3$, $k_2 = 2$, $k_3 = 4$
M	machine set, {1,2,⋯k}, m ∈ M
M_s	set of parallel machines, {1,2,⋯k_i}
s	stage serial number
i	Number of stages, i = 3
S	stage set, {1,2,⋯i}, s∈S
T_m	processing sequence number set of jobs on machine m
T_s	processing sequence number set of jobs on stage s
S_{jm}	starting time of job j on machine m
parameter	definition
T_{jm}	processing time of job j on machine m
C_{jm}	completion time of job j on machine m
C_{max}	makespan, indicating the completion time of the last processing

P_{jm}	processing power of job j on machine m
E_{ajm}	energy consumption in processing job j by machine m
E_{bjm}	idle energy consumption of job j after adopting energy-efficient strategy on machine m
E_{cjm}	idle energy consumption of job j on machine m
$E_{c'm}$	energy consumption of "shutdown-restart" for machine m
E_a	energy consumption in machine processing state
E_b	idle energy consumption of the machine after adopting the strategy of energy-efficient
E_c	energy consumption of machine in idle state
$E_{c'}$	energy consumption of "shutdown-restart" for machine
E_m	total energy consumption in manufacturing workshop
h	machine serial number of job j to be processed in the next stage after machine m processing
d	serial number of the last job processed before the machine m processes the job j
g	serial number of the next job processed after machine m has processed job j
$A_{n\times i}$	matrix of n rows and i columns represents scheduling schemes for n jobs in i stages
X(b,s)	matrix-coded real number of job b processed on stage s
X_{jm}	job j is processed on machine m, the value is 1, otherwise 0

2.1.2. Hybrid flow-shop scheduling problem

HFSP is a hybrid problem of classical flow shop scheduling problem (Zhao *et al*., 2019) [50] and parallel machine scheduling problem (Kramer *et al*., 2019) [23]. HFSP can be generally described as: on the basis of traditional flow shop scheduling problem, there are at least two stages, each stage has at least one machine and at least one stage has parallel machines. Each job is processed in the same sequence of stages and one M_s= {1,2,... k_s} of parallel machines in each stage. Each job $j \in J$, each machine $m \in M$, each stage $s \in S$. Each job has the corresponding processing time and power on each machine at each stage. HFSP layout diagram is shown in Fig. 1.

The constraints and assumptions of HFSP are as follows

- Each job is processed only once by one machine at each stage.
- Each job must be processed in the same sequence of stages.
- Each machine can only process one job at a time.
- There is no preemptive priority rule for each job in the same stage of processing. Once the processing starts, it must be completed without interruption.
- There are more than one parallel computers in at least one stage.
- The processing time and power of each machine for different jobs in each stage are known.
- The waiting interval between the two stages is infinite.

No consideration is given to the replacement time of the job machine and the transportation time of the job.

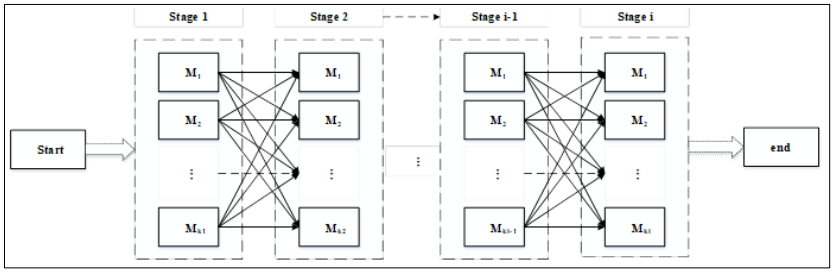

Fig 1: HFSP layout diagram

Process is to be completed in each stage. Each process of each job can be preemptively processed on any machine in the corresponding stage. To judge whether the processing performance of the machine is the same in the same stage, HFSP is divided into the same parallel machine HFSP (Shengyao *et al*., 2013), the uniform parallel machine HFSP (Zhou *et al*., 2016) [55] and the unrelated parallel machine HFSP (Meng *et al*., 2019) [31]. An example of HFSP is described as: there are n jobs, i.e. sets J = {1,2,... n}, and i stages, i.e. sets S = {1,2,... i}, and k machines, i.e. sets M = {1,2,... k}, and sets M_s= {1,2,... k_s} of parallel machines in each stage. Each job $j \in J$, each machine $m \in M$, each stage $s \in S$. Each job has the corresponding processing time and power on each machine at each stage. HFSP layout diagram is shown in Fig. 1.

The constraints and assumptions of HFSP are as follows

- Each job is processed only once by one machine at each stage.
- Each job must be processed in the same sequence of stages.
- Each machine can only process one job at a time.
- There is no preemptive priority rule for each job in the same stage of processing. Once the processing starts, it must be completed without interruption.
- There are more than one parallel computers in at least one stage.
- The processing time and power of each machine for different jobs in each stage are known.
- The waiting interval between the two stages is infinite.
- No consideration is given to the replacement time of the job machine and the transportation time of the job.

- Friction energy consumption and public energy consumption are not considered.

2.1.3. Energy consumption analysis

The paper assumes that there are *n* jobs, *i* stages and *k* machines in a manufacturing workshop. Workshop energy consumption mainly includes processing energy consumption and idle energy consumption. Workshop machines have four states: start-up, shut-down, idle and processing. Therefore, the energy consumption of workshop is analyzed and divided into four parts: a, b, c and d.

a. Processing energy consumption

Processing energy consumption refers to the energy consumed by the machine in the manufacturing workshop. Job *j* is processed by machine *m* in stage *s*, and its energy consumption E_{ajm} can be expressed as shown in Eq. (1):

$$E_{ajm} = P_{jm} \times T_{jm} \quad (1)$$

The total energy consumption of machine processing in the whole workshopE_a can be expressed as shown in Eq. (2):

$$E_a = \sum_{j \in J} \sum_{m \in M} P_{jm} \times T_{jm} \times X_{jm} \quad (2)$$

b. Idle energy consumption

The idle energy consumption refers to the energy consumed by the machines in the manufacturing workshop when they are idle. The idle state is generally due to loading and unloading work pieces, positioning and tightening work pieces, changing cutters or machine preheating. Therefore, the selection of idle power is consistent with the corresponding processing power of the job to be processed on the machine. If the job to be processed by machine *m* in *s* stage is job *j* and the last job is job *d*, its idle energy consumption E_{cjm} can be expressed as shown in Eq. (3):

$$E_{cjm} = P_{jm} \times (S_{jm} - C_{dm}) \quad (3)$$

The total idle energy consumption E_c of the machine in the whole workshop can be expressed as shown in Eq. (4):

$$E_c = \sum_{j \in J} \sum_{m \in M} \sum_{h \in J} P_{jm} \times (S_{jm} - C_{dm}) \times X_{jm} \quad (4)$$

c. Energy consumption of "shutdown-restart"

"Shutdown-restart" energy consumption refers to the energy consumed by the "shutdown-restart" strategy adopted by the manufacturing workshop machines. Considering only the factors of the machine itself, each machine has its own unique "shutdown-restart" energy consumption, which has nothing to do with other factors. "Shutdown-restart" energy-efficient schematic diagram is shown in Fig. 2. The abscissa of the schematic diagram is time, the ordinate is machine power, and the area enclosed by the broken line and the coordinate axis is energy consumption. The condition of using "shutdown-restart" strategy is that the idle energy consumption E_c of the machine in idle state is greater than or equal to "shutdown-restart" $E_{c'}$ when the job *j* to be processed will be processed on the machine *m*. In addition, the "shutdown-restart" time $T_{"shutdown-restart"}$ is negligible.

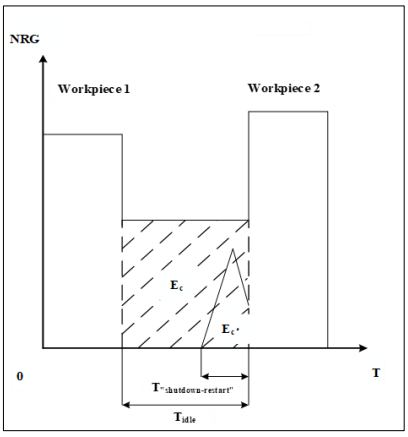

Fig 2: Energy-efficient schematic diagram of "shutdown-restart"

d. Total energy consumption in manufacturing workshop

The total energy consumption of manufacturing workshop E_m is the sum of total processing energy consumption E_a and total energy consumption E_b after "shutdown-restart" energy-efficient. Job *j* is processed by machine *m* in stage *s*. Energy consumption E_{bjm} after "shutdown-restart"energy-efficient can be expressed as shown in Eq. (5):

$$E_{bjm} = P_{jm} \times (S_{jm} - C_{dm}), \quad E_{cjm} \le E_{c'm} = E_{c'm}, \quad E_{cjm} \ge E_{c'm} \quad (5)$$

The total energy consumption E_b after "shutdown-restart" energy-efficient can be expressed as shown in Eq. (6):

$$E_b = \sum_{j \in J} \sum_{m \in M} \min(E_{cjm}, E_{c'm}) \quad (6)$$

Total energy consumption E_m of workshop can be expressed

as shown in Eq. (7):

$$E_m = E_a + E_b \quad (7)$$

In fact, there are many sources of energy consumption in production workshop. The paper does not consider other factors of energy consumption.

2.2. Multi-objective optimization HFSP model for energy-efficient

There are many studies on the mathematical model of HFSP. At present, MILP models are applied to HFSP. Although these Mixed Integer Nonlinear Programming (MINLP) model based on time target (Asadi-Gangraj *et al*., 2018; Lalitha *et al*., 2017; Chen *et al*., 2013; Mirsanei *et al*., 2011; He *et al*., 2010; Behnamian *et al*., 2010) [8, 25, 10, 36, 5], energy consumption target (Jiang *et al*., 2019; Meng *et al*., 2019) [21, 31], and cost target (Zohali *et al*., 2019; Gonzalez-Neira *et al*., 2016) [56, 15]. The paper studies a multi-objective HFSP optimization model aiming at minimizing the makespan f_1 and the total energy consumption f_2.

The objective function can be obtained by comprehensive analysis

$$\min f_1 = \max\left(\max_{j=1}^{n} C_{jm1}, \max_{j=1}^{n} C_{jm2}, \ldots\ldots \max_{j=1}^{n} C_{jmk}\right) \quad (8)$$

$$\min f_2 = \sum_{j=1}^{n}\sum_{m=1}^{k} E_{ajm} + \sum_{j=1}^{n}\sum_{m=1}^{k} \min\left(E_{cjm}, E_{c'm}\right) \quad (9)$$

Eq. (8) calculates the minimum the makespan. The first step is to find the makespan of each job on each machine, that is, the last job on the machine; the second step is to compare makespan of the last job processed on each machine, and the maximum is the makespan. Eq. (9) is to minimize the total energy consumption. The first part is the energy consumption in machine processing state. The second part is the idle energy consumption after adopting energy-efficient mechanism. Firstly, the idle energy consumption of each job on each machine is compared with that of the machine using "shutdown-restart" energy consumption. The smaller value is obtained, and then the sum is accumulated.

The constraints are as follows

$$\sum_{m=1}^{K_s} X_{jm} = 1 \quad (10)$$

$$C_{jm} = S_{jm} + X_{jm}T_{jm},\ X_{jm} = 1,\ j \in J, m \in M \quad (11)$$

$$C_{jm} \le S_{jh} \quad (12)$$

$$C_{jm} \le S_{gm} \quad (13)$$

$$X(j,s) \ne X(b,s),\ j \ne b \quad (14)$$

Eq. (10) indicates that each job can only be processed by one machine at each processing stage. Eq. (11) denotes that if job *j* is processed on machine *m*, the sum of the start processing time and the processing time is makespan. Eq. (12) indicates that makespan of job *i* in machine *m* is less than or equal to the start time of the next stage in machine *h*. Eq. (13) indicates that makespan of job *j* in machine m is less than or equal to the start time of the next job *g*. Eq. (14) is that each machine in each stage can only process one job at the same time. This constraint is realized by coding mechanism.

3. IMOMVO algorithm for energy-efficient multi-objective HFSP

3.1. MOMVO algorithm

The stochastic population-based MVO has three main concepts (white hole, black hole and wormhole) used to model. The Big Bang Theory (Khoury *et al*., 2001) [22] introduces that white holes are jet sources in the universe, which can provide material and energy to the outer regions, but can not absorb any matter and radiation from the outer regions. On the contrary, a black hole is a special universe object that only absorbs and does not emit. Wormholes exist objectively and are narrow tunnels connecting two different spaces in the universe. The MVO algorithm uses white holes and black holes to explore the search space, and wormholes help to develop the search space. Fig. 3 is a concept sketch of white holes, black holes and wormholes.

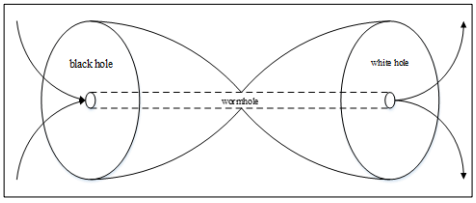

Fig 3: Concept sketch of white hole, black hole and wormhole

MOMVO algorithm is a generalization of MVO algorithm for solving multi-objective problems. In MOMVO algorithm, each problem is represented by a universe, and each universe has an inflation, which is expressed by the reciprocal of fitness function value. The following rules are followed in the optimization process:0

- The higher inflation, the greater the probability of white holes and the lower the probability of black holes.
- The universe with higher inflation sends matter through white holes, while the universe with lower inflation receives matter through black holes.
- All matter in the universe has a probability of moving to the best universe through wormholes.

The stages of MOMVO algorithm can be divided into standardization of objective function, parameter setting, initialization of universe, parameter updating, calculation of inflation and archiving, roulette selection, boundary checking and updating of universe.

Stage 1: weighted additive utility objective function standardization, which is used to standardize the objectives of different orders of magnitude and different units as comparable objectives, and then carry out weighted evaluation.

Stage 2: setting variable intervals, iteration times, weight coefficients and other relevant algorithm parameters.

Stage 3: initialize the universe randomly according to the given parameters.

Stage 4: updating the parameters of *WEP* and *TDR*.

Stage 5: calculate fitness values, sort and archive the best number of solutions in history according to fitness values.

Stage 6: deposit the archives by roulette mechanism and copy them by row, column or element.

Stage 7: boundary checking is used to check the elements of the coded matrix after universe updating. If the value is greater than the maximum value of the interval, the maximum value is taken; if the value is less than the minimum value of the interval, the minimum value is taken.
Stage 8: iterative updating of the universe.
The above process goes on until the condition is satisfied (the maximum number of iterations is reached), the algorithm is terminated and the optimal solution is output.
Fig. 4 shows the flow chart of the MOMVO algorithm:

3.2. Standardization of weighted additive utility objective function

Energy-efficient multi-objective HFSP is a multi-objective problem. There are two typical characteristics of this kind of problem. On the one hand, the incommensurability of multi-objective exists, and the physical meaning and magnitude of the unit of objective function are quite different. On the other hand, there is a non-linear or irrelevant relationship between multiple targets. Energy-efficient multi-objective HFSP model is an irrelevant parallel machine scheduling model. Specifically, the processing time and power of each machine for each job are different in each stage. It can be concluded that there is no linear relationship between minimizing the makespan and minimizing the total energy consumption. At the same time, the unit physical meaning of the objective function of the model is different and there are large orders of magnitude differences. Therefore, the weighted additive utility objective function standardization is used to solve the multi-objective decision-making problem.
Combined with MOMVO algorithm, the larger inflation, the better the optimization effect. Therefore, the total objective function U, i.e. fitness function, can be defined as shown in Eq. (15):

$$U = w_1 f_1 + w_2 f_2 \quad (15)$$

Specifically, w_1 and w_2 are the weight coefficients of the objective function, and the sum of weights is equal to 1. The weights are positive that is, $w_1 > 0$, $w_2 > 0$. By utilizing utility

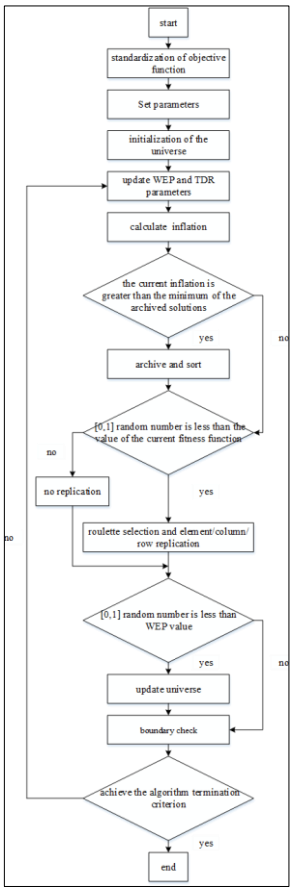

Fig. 4. Flow diagram of MOMVO algorithm

Function, the objective values of a multi-objective problem are combined into a single objective function for easy solution. Different objectives can be standardized to targets with small order of magnitude differences. The weighted additive utility function with normalized objective is defined as:

$$U' = w_1 f'_1 + w_2 f'_2 \quad (16)$$

Eq. (16) indicates that f'_1 and f'_2 are the standardized values of f_1 and f_2 respectively. The standardized objectives f' are defined as:

$$f' = \frac{f - f_{\min}}{f_{\max} - f_{\min}} \quad (17)$$

Eq. (17) indicates that f_{max} and f_{min} are the maximum and minimum values of the objective function f respectively, and they are known conditions. Generally, they can be estimated according to a large number of experiments. Specifically, in the paper, a statistical method is used. Firstly, the f_{max} and f_{min} values of f_1 and f_2 can be considered as two single-objective problems when they are searched through f_{max} and f_{min}. For f_1 and f_2, a matrix is randomly generated to iterate 10,000 (given) times and the image of inflation is observed. If inflation F is at the point of $(-\infty, 0)$ interval, the f_{min} set will be reduced, and if the point of (0, 1) interval appears, the f_{max} set will be increased. After several experiments, reasonable f_{max} and f_{min} estimates can be obtained. In the MOMVO algorithm, the larger inflation, the better the optimization effect. The HFSP model minimizes the multi-objective problem, so the inflation F is defined asshown in Eq. (18):

$$F = \frac{1}{U} = \frac{1}{w_1 f'_1 + w_2 f'_2} \quad (18)$$

3.3. Universe initialization

Universe initialization in MOMVO algorithm needs to consider the diversity of the universe, that is, the diversity of initial scheduling schemes. The effectiveness and convergence of MOMVO algorithm can be illustrated by maintaining the diversity of the initialization universe and obtaining optimal solutions eventually. Therefore, the initial universe is generated by a stochastic strategy according to the parameters of the limited interval range.

3.4. Coding and decoding

The encoding method of HFSP is matrix encoding based on elements and their positions. That is, a matrix represents a universe, that is, a scheduling scheme. The row of matrix represents the serial number of work piece, the list of matrix represents the processing stage, and the matrix element X (b, s) represents the real number of work piece b corresponding to the processing stage s in matrix coding. Assuming that n jobs need to be processed, each job needs to complete i stages of processing. The number of parallel machines in each stage is k_1, k_2... k_i, respectively. That is, the n×i encoding matrix is as shown in Eq. (19):

$$A_{n\times i} = \begin{bmatrix} X(1,1) & X(1,2) & \dots & X(1,i) \\ X(2,1) & X(2,2) & \dots & X(2,i) \\ . & . & & . \\ . & . & X(b,s) & . \\ . & . & & . \\ X(n,1) & X(n,2) & \dots & X(n,i) \end{bmatrix} \quad (19)$$

Elements X (b, s) $\in$ (1, k_s + 1), b = 1, 2... n, s = 1, 2... i. [X (b, s)] denotes the integral part of X (b, s), which is the machine serial number used in the processing of the b job in the s processing stage. If [X (j, s)]= [X (b, s)] and $j \neq b$, it means that the job j and b are processed by the same machine in the same s stage. It is stipulated that the smaller of X (j, s) and X (b, s) is preferred in processing order. For the first processing stage, the processing order of the same machine is sorted according to the size of matrix elements. For the second and higher processing stages, the processing sequence of a job is determined by makespan of the previous stage of the job and makespan of the previous job used in the current stage. Because the machine still consumes energy in the idle time, the less the idle time, the more energy-efficient. In other words, the processing start time is equal to the larger value of both. Assuming that there are three processing stages and three jobs in a HFSP, each job is required to complete three processing stages. Assume that the number of parallel machines is 2, 3 and 4, respectively. A random matrix $A_{3\times3}$ is generated by Python software. Examples are given as follows:

$$A_{3*3} = \begin{bmatrix} 1.861 & 1.786 & 2.933 \\ 1.253 & 2.209 & 2.127 \\ 2.862 & 3.192 & 1.853 \end{bmatrix}$$

Based on the encoding and decoding method, the processing sequence of the job in the first stage is parallel machine 1:[2,1]; parallel machine 2:[3]. The second stage of job processing sequence is parallel machine 1:[1]; parallel machine 2:[2]; parallel machine 3:[3]. The sequence of job processing in the third stage is parallel machine 1:[3]; parallel machine 2:[2,1]. Parallel computers 3 and 4 have no jobs to be processed.

The decoding method of HFSP follows the principle of first come first processed. According to the characteristics of multi-stage processing, the specific decoding rules are as follows:

Stage 1: s = 1;

Stage 2: determine the sequence number of components according to the row of the encoding matrix, and determine the processing stage according to the column.

Stage 3: the integer part of the matrix element value judges the machine serial number of each job in the corresponding processing stage of the column, and the decimal part judges the processing order of the job to be processed on each machine.

Stage 4: the bigger value of makespan of the previous stage

of the work piece and that of the previous work piece machined in the current stage is the start time of the current stage of the work piece. Other relevant time can be calculated according to Eq. (11). The idle power of machine m before processing job *j* after processing job *d* is equal to the processing power of job *j* in machine *m*. The idle energy can be calculated according to Eq. (3).
Stage 5: j = j + 1;
Stage 6: steps 2-5 until all processing stages are completed.

3.5. Improvement of roulette gambling mechanism

By improving the roulette mechanism, the equal difference probability roulette mechanism is proposed to model white holes, black holes and wormholes and exchange universe material. In each iteration, the current universe is ranked according to inflation of the universe and the universe with white holes is selected by roulette. In other words, the best solution set of the history stored in the archive is selected for replication. The random real number r_1 in the range of value [0,1] is compared with the reciprocal *U* of the current inflation *F*. When the solution set is better, that is, *U* is smaller. The smaller the probability of roulette betting, the better the solution is retained. Therefore, it is stipulated that when $r_1 > U$, replication should be made, otherwise no replication should be made. Because inflation of the optimal solution for archiving is different, the probability of choosing replication should increase with inflation. A roulette mechanism of equal difference probability is designed. The probability tolerance is *d*, the first one is *x*, the number of sequence is *n*, the sequence number of arbitrary solution set is *i*, and the probability p_i formula of replication can be expressed as shown in Eq. (20):

$$p_i = x + (i-1)d \tag{20}$$

The sum of replication probability of each solution set is one, which is in conformity with the norm. The calculation processing and that the job processing on the same machine has sequential constraints, column duplication is selected for equal difference probability roulette. The roulette mechanism of equal difference probability is conducive to exploring search space.

3.6. Improvement of universe renewal formula

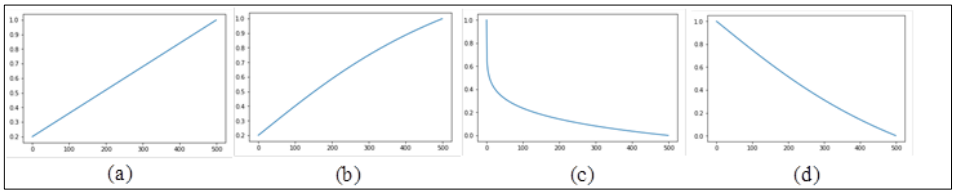

Fig 5: (a) WEP adaptive function figures. (b) Figure improvement by WEP adaptive function. (c) TDR adaptive function figure. (d) Figure improvement by TDR adaptive funtion

Formula can be expressed as shown in Eq. (21):

$$\sum_{i=1}^{n} p_i = x + x + d + x + 2d + \ldots\ldots + x + (n-1)d = 1 \tag{21}$$

Deformation to be expressed as shown in Eq. (22):

$$x = \frac{1 - \frac{n(n-1)d}{2}}{n} \tag{22}$$

To judge whether to replicate or not, it is assumed that the interval [0,1] is divided into n intervals, and the formula for calculating the value x_i of the interval endpoint can be expressed as shown in Eq. (23):

$$x_i = ix + \frac{i(i-1)d}{2} \tag{23}$$

The types of replication can be divided into three cases: row replication, column replication and element replication. Considering the characteristics of HFSP and the significance of encoding matrix, taking the selected optimal solution set in the archive as the replication object, row replication represents the replication of the job, that is, the machine allocation of replication rows corresponding to all stages of the job; column replication represents the progressive stage. Replication is the machine allocation of all the jobs in the corresponding stage in the replication column; element replication represents the machine allocation of the jobs in the corresponding stage in the replication column. Considering that the HFSP problem has the characteristics of the same job in different stages of machine processing and that the job processing on the same machine has sequential constraints, column duplication is selected for equal difference probability roulette. The roulette mechanism of equal difference probability is conducive to exploring search space.
In order to ensure the diversity of the universe and develop search space, wormholes can randomly change the matter in the universe without considering inflation of the universe. The update formula of MOMVO algorithm is improved can be expressed as shown in Eq. (24):

$$x_i = \begin{cases} X_i + TDR \times ((ub_i - lb_i) \times r_4 + lb_i \times r_5) & \left.\begin{matrix} r_3 < 0.5 \\ r_3 \geq 0.5 \end{matrix}\right\} r_2 < WEP \\ X_i - TDR \times ((ub_i - lb_i) \times r_4 + lbi \times r_5) & \\ x_i & r_2 \geq WEP \end{cases} \tag{24}$$

Among them, x_i represents the current universe, that is, the real number of the first element in the encoding matrix, and

X_i represents the real number of the ith element in the encoding matrix corresponding to the universe with the best history. *TDR* denotes the travel distance rate, *WEP* denotes the existence probability of wormholes, ub_i denotes the numerical upper bound of the real number of the first element, and lb_i denotes the numerical lower bound of the real number of the second element, r_2, r_3, r_4 and r_5 are all [0,1]. *TDR* and *WEP* are two adaptive formulas can be expressed as shown in Eq. (25)and Eq.(26):

$$WEP = \min + (\max - \min) \times \left(\frac{l}{L}\right) \tag{25}$$

$$TDR = 1 - \sqrt[p]{\frac{l}{L}} \tag{26}$$

In the paper, min = 0.2, max = 1. L is the maximum number of iterations set to 500, l is the current number of iterations, p is the accuracy of iterative development. The greater the P value, the faster and more accurate the iterative development. Considering the improvement of *WEP* and *TDR* adaptive formulas can be expressed as shown in Eq. (27) and Eq.(28):

$$WEP = 0.2 + 2 \times 0.8 \times \frac{2 \times \arctan(\frac{l}{L})}{\pi} \tag{27}$$

$$TDR = 1 - 2 \times \frac{2 \times \arctan(\frac{l}{L})}{\pi} \tag{28}$$

Drawing the *WEP* and *TDR* adaptive functions before and after improvement as shown in Fig. 5, abscissa are iteration times, longitudinal coordinates are *WEP* and *TDR* values respectively. Through comparison, it is found that the improved *WEP* has a faster growth rate in the early stage and a slower growth rate in the later stage, which is conducive to the rapid improvement of the probability of updating the universe in the early stage and the improvement of the optimization speed. *TDR* reduces the speed of decline, improves the accuracy of local search, and prevents early decline from falling too fast into local optimum.

4. Computational experiments and results

4.1. IMOMVO parameter setting

In the paper, the parameters of IMOMVO algorithm are set as shown in Table 2.

Table 2: Relevant parameters of IMOMVO algorithm.

parameter	value	Significance
max	1	maximum WEP
min	0.2	minimum WEP
L	500	total number of iterations
l	1-500	total number of iterations
w_1	0-1	weight coefficient of f_1 objective function
w_2	0-1	weight coefficient of f_2 objective function
f_{1_max}	96	maximum of f_1 objective function
f_{1_min}	26	minimum of f_1 objective function
f_{2_max}	320	maximum of f_2 objective function
f_{1_min}	215	minimum of f_2 objective function
r_1	0-1	random number
r_2	0-1	random number
r_3	0-1	random number
r_4	0-1	random number
r_5	0-1	random number

4.2. Effectiveness of mixed integer nonlinear programming model

In order to verify the feasibility and effectiveness of IMOMVO algorithm, which is used to solve energy-efficient multi-objective HFSP. With windows 7 operating system as the development environment and python as the programming language, the experiment is carried out on the notebook computer with Inter core (TM) i7-6700 CPU@3.40GHz and 8GB memory. The HFSP test example in the paper is selected from IGSA experiment (Dai *et al*., 2013). The details are as follows: 12 jobs need to be processed in three production stages, and the number of parallel machines in each stage is 3, 2 and 4 in turn. The processing time and machine power related data are shown in Table 3.

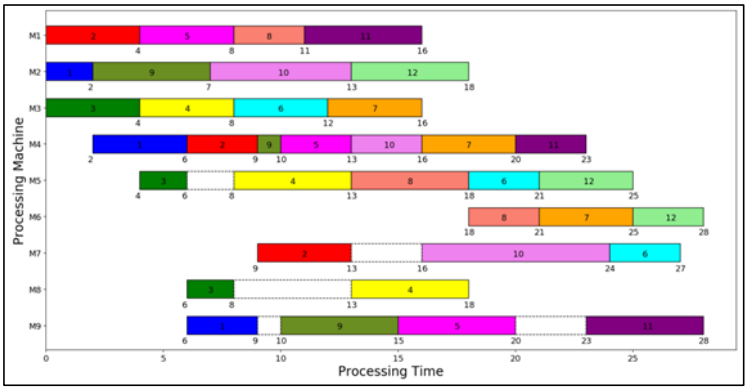

Fig 6: Sample gant chart of reference experiment

Table 3: Relevant data of test cases

Job		Process 1			Process 2		Process 3			
		M1	M2	M3	M4	M5	M6	M7	M8	M9
1	Processing time/min	2	2	3	4	5	2	3	2	3
	Unload power/kw	2.26	1.36	1.43	1.46	1.2	4.03	3.8	3.72	3.43
2	Processing time/min	4	5	4	3	4	3	4	5	4
	Unload power/kw	1.86	0.98	0.9	1.68	1.3	3.42	3.32	2.3	2.68
3	Processing time/min	6	5	4	4	2	3	4	2	5
	Unload power/kw	1	0.98	0.9	1.46	1.55	3.42	3.32	3.72	2.25
4	Processing time/min	4	3	4	6	5	3	6	5	7
	Unload power/kw	1.86	1.12	0.9	1.32	1.2	3.42	2.94	2.3	1.99
5	Processing time/min	4	5	3	3	1	3	4	6	5
	Unload power/kw	1.86	0.98	1.43	1.68	1.7	3.42	3.32	2.12	2.25
6	Processing time/min	6	5	4	5	3	4	3	9	5
	Unload power/kw	1	0.98	0.9	1.46	1.42	2.8	3.8	1.92	2.25
7	Processing time/min	5	2	4	4	6	4	4	3	5
	Unload power/kw	1.18	1.36	0.9	1.46	1.1	2.8	3.32	3.26	2.25
8	Processing time/min	3	5	4	7	5	3	3	6	4
	Unload power/kw	2.1	0.98	0.9	1.22	1.2	3.42	3.8	2.12	2.68
9	Processing time/min	2	5	4	1	2	7	8	6	5
	Unload power/kw	2.26	0.98	0.9	2.14	1.55	2.14	2.6	2.12	2.25
10	Processing time/min	3	6	4	3	4	4	8	6	7
	Unload power/kw	2.1	0.8	0.9	1.68	1.3	2.8	2.6	2.12	1.99
11	Processing time/min	5	2	4	3	5	6	7	6	5
	Unload power/kw	1.18	1.36	0.9	1.68	1.2	2.54	2.76	2.12	2.25
12	Processing time/min	6	5	4	5	4	3	4	7	5
	Unload power/kw	1	0.98	0.9	1.46	1.3	3.42	3.32	2.04	2.25

The validity of the MILP model for energy-efficient multi-objective HFSP is tested. The gant chart corresponding to a scheduling situation listed in reference experiment is shown in Fig. 6.

Design encoding matrix that conforms to the gant chart.

$$A_{12\times 3}=\begin{bmatrix} 2.1 & 1.1 & 3.1 & 3.2 & 1.2 & 3.3 & 3.4 & 1.3 & 2.2 & 2.3 & 1.2 & 1.4 \\ 1.1 & 1.2 & 2.1 & 2.2 & 1.4 & 2.4 & 1.6 & 2.3 & 1.3 & 1.5 & 1.7 & 2.5 \\ 4.1 & 2.1 & 3.1 & 3.2 & 4.3 & 2.3 & 1.2 & 1.1 & 4.2 & 2.2 & 4.4 & 1.3 \end{bmatrix}^{T}$$

Considering that only the "shutdown-restart" strategy can be adopted in the second and third stages of the test case, and the "shutdown-restart" energy consumption of M5-M9 machine is given only in IGSA experiment. It is observed that there are two "shutdown-restart" strategies for M9 machine, and the energy consumption value is the same. Therefore, it can be assumed that the "shutdown-restart" energy consumption of the machine is unique and constant. Table 4 is relevant data.

Table 4: Energy consumption data of M5-M9 machine "shutdown-restart"

Machine serial number	$E_{c'm}$/J
M5	8065.4
M6	76782.2
M7	32010.0
M8	41898.5
M9	35876.3

After testing the MILP model for multi-objective energy-efficient HFSP, it is calculated that f_1 is 28, f_2 is 298.36, and f_2 is 260.08 after "shutdown-restart" strategy is adopted. In

the last two paragraphs of the fifth part of the reference experiment, we can see that f_1 is 28, and f_2 is 260.36 after "shutdown-restart" strategy. The experiment results show that the model is accurate and effective.

4.3. Test examples, results and algorithm comparison

In the interval of [0,1], every interval of 0.5, a group of experiments were carried out, each group of experiments took different w_1 and w_2 to run 15 times. Three typical manufacturing scenarios were analyzed by adjusting the values of w_1 and w_2. In experiment, E_b refers to standby power consumption after "shutdown-restart" strategy. In IGSA experiment, E_u refers to standby power consumption without "shutdown-restart" strategy, so E_b and E_b/f_2 are not compared with the reference experiment E_u and E_u/f_2. In the scatter plot of the relationship between makespan and energy consumption, the black dot represents the IMOMVO results, and the white dot represents the results of the IGSA experiment.

4.3.1. Scenario one: minimum makespan scenario

Fig. 7 reflects the scatter plot of the relationship between makespan and energy consumption when the weight coefficient of the objective function is set to $w_1 = 1$ and $w_2 = 0$. Fig. 8 shows a box diagram with reference experiments under the makespan index. Fig. 9 is a 3D mesh drawing with makespan, energy consumption and inflation as three coordinate axes. Table 5 gives a quantitative analysis of the relationship between makespan and energy consumption.

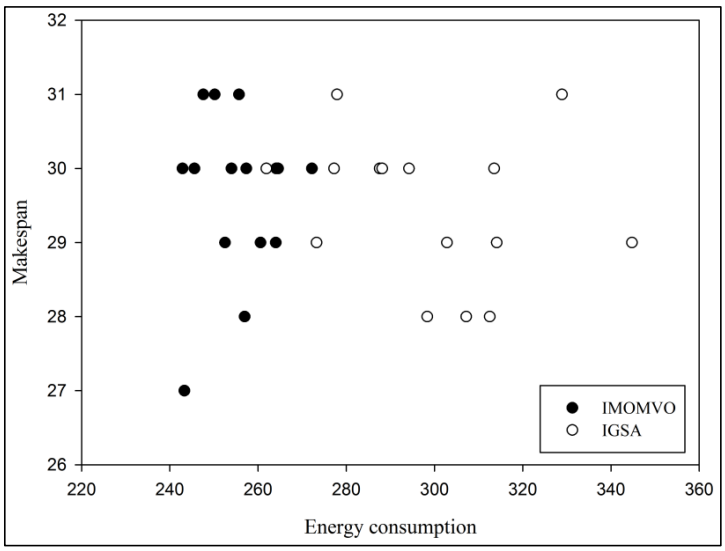

Fig 7: Plots of makespan versus energy consumption in scenario one

Table 5: Makespan and energy consumption related data in scenario one

Number	f_1/min	f_2/(kw·min)	E_b/(kw·min)	E_b/f_2
1	27 | 28	243.29 | 298.36	1.41	0.60%
2	28 | 28	256.95 | 307.20	6.90	2.70%
3	29 | 28	252.50 | 312.53	2.68	1.06%
4	29 | 29	260.55 | 273.23	4.21	1.60%
5	29 | 29	264.01 | 302.83	5.25	1.99%
6	30 | 29	242.85 | 314.13	0.97	0.40%
7	30 | 29	245.57 | 344.76	0.83	0.30%
8	30 | 30	253.95 | 261.86	3.43	1.35%
9	30 | 30	257.32 | 277.23	5.17	2.00%
10	30 | 30	264.09 | 287.55	3.23	1.22%
11	30 | 30	264.55 | 288.11	3.81	1.40%
12	30 | 30	272.19 | 294.24	5.82	2.10%
13	31 | 30	247.59 | 313.54	1.30	0.50%
14	31 | 31	250.17 | 277.89	4.45	1.78%
15	31 | 31	255.67 | 328.91	0.60	0.20%

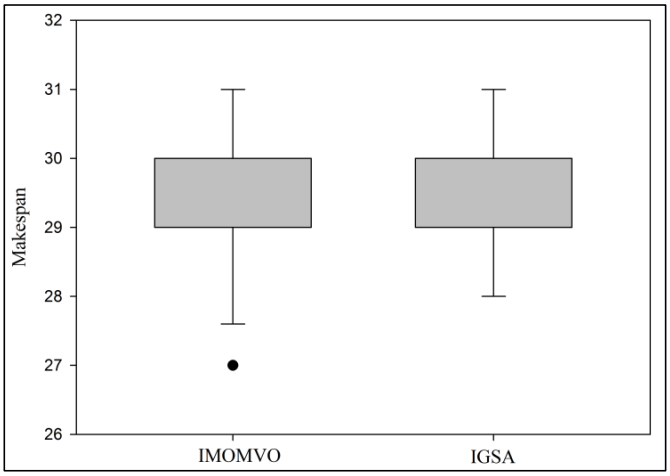

Fig 8: Comparison of makespan in scenario one

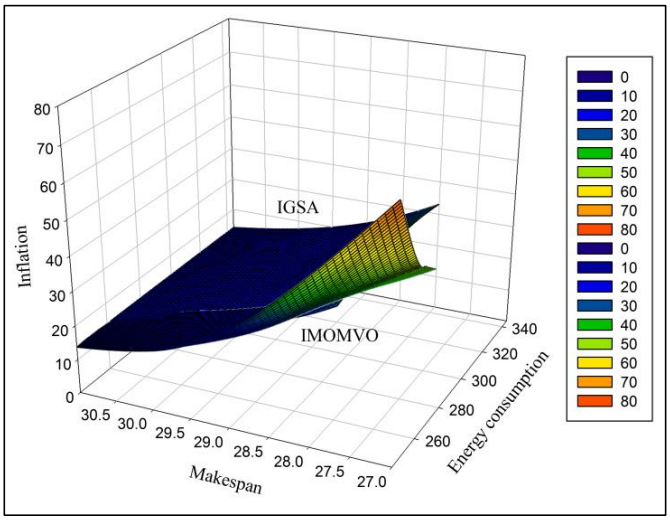

Fig 9: Comparison of pareto fronts and inflation in scenario one

As shown in Fig. 7, the distribution of black and white dots in the makespan f_1 of the longitudinal axis is approximately the same, which shows the effectiveness of IMVOVO algorithm and MILP model for energy-efficient multi-objective HFSP. Compared with the white dots, the distribution of black dots is left, which indicates that the "shutdown-restart" strategy can effectively reduce energy consumption. As shown in Fig. 8, in scenario one, the optimal result 27 of this experiment is better than that of the reference experiment, which shows the superiority of the IMOMVO without the influence of "shutdown-restart" strategy. Fig. 9 shows that IMOMVO results are superior to the IGSA experiments in terms of inflation, which shows the superiority of IMOMVO.

4.3.2. Scenario two: minimum energy consumption scenario

Fig. 10 reflects the scatter plot of the relationship between makespan and energy consumption when the weight coefficient of the objective function is set to $w_1 = 0$ and $w_2 = 1$. Fig. 11 shows a box diagram with reference experiments under the energy consumption index. Fig. 12 is a 3D mesh drawing with makespan, energy consumption and inflation as three coordinate axes. Table 6 shows the quantitative analysis of makespan and energy consumption.

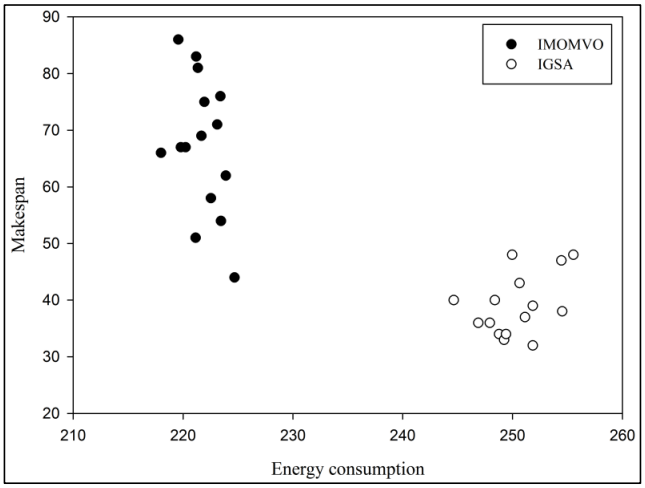

Fig 10: Plots of makespan versus energy consumption in scenario two

Table 6: Makespan and energy consumption related data in scenario two

Number	f_1/min	f_2/(kw·min)	E_b/(kw·min)	E_b/f_2
1	66 | 40	217.99 | 244.64	0	0.00%
2	86 | 36	219.57 | 246.88	0	0.00%
3	67 | 36	219.80 | 247.92	0	0.00%
4	67 | 40	220.24 | 248.38	0	0.00%
5	51 | 34	221.14 | 248.77	0.60	0.27%
6	83 | 33	221.19 | 249.22	1.30	0.59%
7	81 | 34	221.34 | 249.40	0	0.00%
8	69 | 48	221.67 | 249.97	0	0.00%
9	75 | 43	221.94 | 250.64	0	0.00%
10	58 | 37	222.53 | 251.12	0	0.00%
11	71 | 39	223.11 | 251.84	0	0.00%
12	76 | 32	223.40 | 251.84	0	0.00%
13	54 | 47	223.45 | 254.44	0	0.00%
14	62 | 38	223.89 | 254.52	0	0.00%
15	44 | 48	224.69 | 255.53	0	0.00%

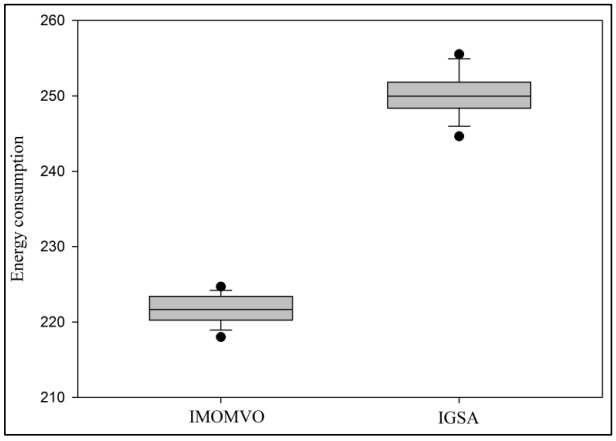

Fig 11: Comparison of energy consumption in scenario two

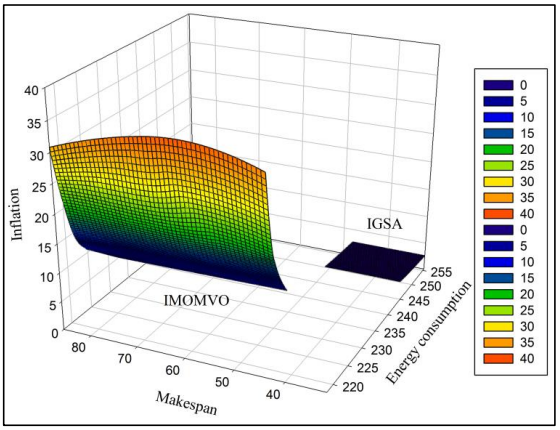

Fig 12: Comparison of pareto fronts and inflation in scenario two

Looking at Fig. 10 that the distribution of black dots is more left than that of white dots, the superiority of "shutdown-restart" strategy and IMOMVO is illustrated. However, the distribution of black dots is higher than that of white dots, which indicates that there is a contradiction between makespan and total energy consumption at the Pareto front. The reason is that there is a contradiction between energy consumption and makespan. There are two reasons for lower energy consumption of experiment. Firstly, the experimental results are obtained after the "shutdown-restart" strategy is adopted in this paper. Secondly, the IMOMVO has certain advantages in optimization problems. Fig. 11 shows that the IMOMVO data are better than the IGSA data. Fig. 12 also shows the advantages in terms of inflation. In addition, when the total energy consumption is low, the idle energy consumption E_b is almost zero, which indicates that when the machine is not idle, some high-energy-consuming machines are generally shut down in the whole process, and the total energy consumption is low, while makespan is higher. The main objective of scheduling problem is to reduce idle energy consumption and idle time. The conclusion is different from common sense perception because of the complex constraints of NP-hard problem in machine and job allocation. It is difficult to reduce the necessary energy consumption in processing by scheduling, but the idle time of machine can be reduced by optimizing the processing sequence, and the idle energy consumption can be reduced to achieve the goal of energy-efficient.

4.3.3. Scenario three: minimum unit time energy consumption scenario

Fig. 13 reflects the scatter plot of the relationship between makespan and energy consumption when the weight coefficient of the objective function is set to $w_1 = 0$ and $w_2 = 1$. Fig. 14 shows a box diagram with reference experiments under the inflation index. Fig. 15 is a 3D mesh drawing with makespan, energy consumption and inflation as three coordinate axes. Table 7 shows the data related to the maximum completion time and energy consumption.

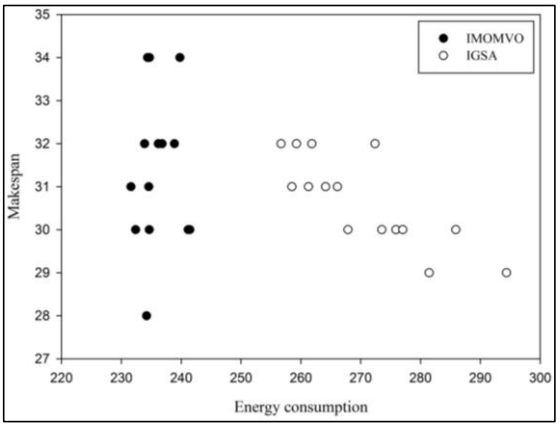

Fig 13: Plots of makespan versus energy consumption in scenario three

Table 7: Makespan and energy consumption related data in scenario three

Number	f_1/min	f_2/(kw·min)	E_b/(kw·min)	E_b/f_2
1	28 | 32	234.21 | 256.68	0	0.00%
2	30 | 31	232.38 | 258.50	0	0.00%
3	31 | 32	231.60 | 259.25	0	0.00%
4	30 | 31	234.66 | 261.28	0.53	0.23%
5	31 | 32	234.57 | 261.84	0	0.00%
6	32 | 31	233.88 | 264.13	1.28	0.55%
7	32 | 31	236.16 | 266.12	0.13	0.06%
8	32 | 30	236.81 | 267.89	0.13	0.05%
9	32 | 30	236.81 | 273.53	0.27	0.11%
10	34 | 32	234.38 | 272.41	1.28	0.55%
11	34 | 30	234.69 | 275.89	0.13	0.06%
12	30 | 30	241.16 | 277.02	0	0.00%
13	30 | 29	241.43 | 281.44	0.70	0.29%
14	32 | 30	238.87 | 285.90	1.88	0.79%
15	34 | 29	239.78 | 294.36	0.60	0.25%

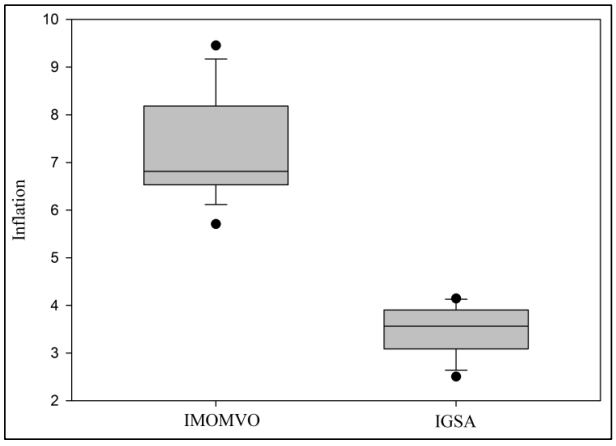

Fig 14: Comparison of inflation in scenario three

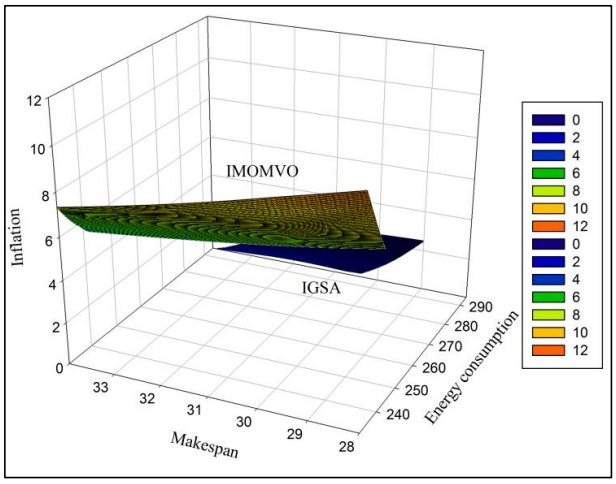

Fig 15: Comparison of pareto fronts and inflation in scenario three

As shown in Fig. 13, the distribution of black dots still is more left and more dispersed than that of white dots. Scenario three is a multi-objective problem, so inflation is needed as an evaluation index of experimental data. Fig. 14 and Fig. 15 show that inflation of IMOMVO data is higher than that of IGSA data from one-dimensional and three-dimensional perspectives, respectively. Many idle energy consumption is 0, that is, the machine has no idle time. The maximum and minimum energy consumption rates are 0.79% and 0, with an average value of 0.20%, which is larger than the average energy consumption rate of scene two and less than the average energy consumption rate of scene one 1.28%. It shows that the lower the idle energy consumption rate, the more energy-efficient, and the cost is to spend more time on completion.

By comprehensive analysis, the average data of this experiment and the reference experiment are listed as Table 8, and the results of this experiment are compared and analyzed. The following conclusions can be drawn: 1. Using "shutdown-restart" strategy can effectively reduce energy consumption; 2. Using "shutdown-restart" strategy can reduce idle energy consumption and idle energy consumption rate; 3. Maximum completion time; There is a contradiction between the total energy consumption and the Pareto frontier. 4. The limit of energy-efficient scheduling is the idleness of the machine. Usually it is shown that some high energy consumption machines do not work. By comparing the experimental results, the IMOMVO adopted in this experiment has certain advantages. Combined with Figs. 17-19, the production benefit of manufacturing scenario three is the best when the target of maximum completion time f_1 and total energy consumption f_2 need to be considered simultaneously, which has guiding significance for actual production and manufacturing.

4.3.4. Algorithmic comparison

In order to analyze the performance of this algorithm, IP is defined as the percentage of improvement of MOMVO algorithm compared with other algorithms. The Eq (29) is

$$IP = \frac{I_{IMOMVO} - I_X}{I_{IMOMVO}} \times 100\% \quad (29)$$

I_{IMOMVO} represents the best experimental result of IMOMVO algorithm, I_X is the best experimental result of algorithm X. According to the characteristics of the three scenarios, makespan, energy consumption and inflation are selected as the experimental results, and the IP results of IMOMVO and IGSA are calculated as shown in Table 9. It can be seen that the IMOMVO has the best optimization effect for multi-objective optimization problems.

Table 8: Average values of the IMOMVO experiment and IGSA experiment in each scenario

Scene	Type	f_1/min	f_2/kw·min	E_b/kw·min	E_b/f_2
1	IMOMVO	30	255.42	3.34	1.28%
1	IGSA	30	298.82	-	-
2	IMOMVO	67	221.73	0.13	0.06%
2	IGSA	39	250.34	-	-
3	IMOMVO	31	236.09	0.46	0.20%
3	IGSA	31	270.42	-	-

Table 9: IP of MOMVO and IGSA in three scenarios.

Scene	Index	IP
1	Makespan	4%
2	Energy consumption	12%
3	Inflation	56%

4.4. Effectiveness of "shutdown-restart"strategy

Fig. 16 shows the best experimental results of IMOMVO and the IGSA in three manufacturing scenarios, in which the "shutdown-restart" strategy is not used in the reference experiment. The energy consumption of IMMVO in this paper is lower than that of IGSA in three scenarios.

Taking the best experimental results in manufacturing scenario one as the research object, the effectiveness of "shutdown-restart" strategy is illustrated. The maximum completion time f_1 is 27, the corresponding total energy consumption f_2 is 243.29, and the idle energy consumption is 1.41. The idle energy consumption E_u without "shutdown-restart" strategy is 18.65, the idle energy consumption rate is 0.6%, and the idle energy consumption rate without "shutdown-restart" strategy is 3%. The production scheduling results are shown in Fig. 17. According to the data analysis, the total energy consumption is reduced by 17.24 kW min and the idle energy consumption rate is reduced by 2.4% by adopting the "shutdown-restart" strategy. Specifically, the third job is on standby for 1 minute on the fifth machine and the twelfth job is on standby for 5 minutes on the sixth machine. Specifically, as shown in Table 4, $E_{c'5}$ is 8065.4J, $E_{c'6}$ is 76782.2J, and the combined formula (5) of 0.1344 and 1.26 after conversion unit kw.min, respectively. The standby time consumption of these two periods is about 1.55 and 17.1 respectively when the "shutdown-restart" strategy is not adopted. That is to say, "shutdown-restart" strategy should be adopted in both periods of time.

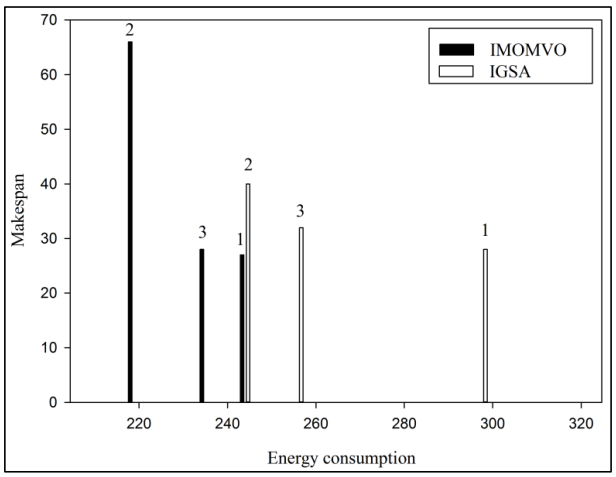

Fig 16: Comparison of the optimum results in three scenarios

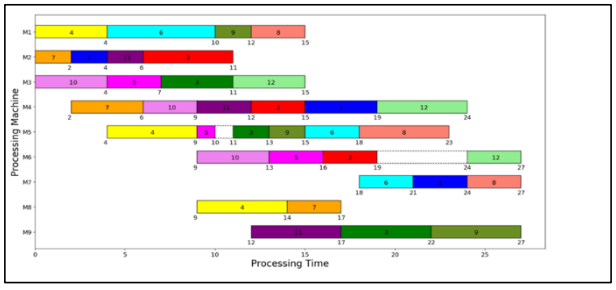

Fig 17: Gant chart of the optimum results production scheduling in scenario one

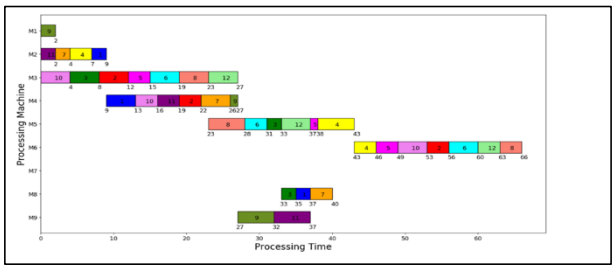

Fig 18: Gant chart of the optimum results production scheduling in scenario two

5. Conclusions and future work

Firstly, the hybrid pipeline scheduling problem (HFSP), intelligent algorithms and MOMVO algorithms for solving HFSP and their applications are reviewed. Secondly, a MILP model is established with the total energy consumption which is basic for Multi-objective optimization HFSP model. The weighted additive utility objective function standardization method is used to establish multi-objective evaluation for generating Pareto optimal solution, and the improved multivariate universe optimization algorithm (MOMVO) is used for HFSP solving. Specifically, element-based and element-based methods are adopted. The improved matrix coding of its position and the improved permutation decoding method based on the principle of first-come-first-processing are proposed. An equal-difference probability roulette mechanism and the update formula of MOMVO algorithm are improved.

Finally, three manufacturing scenarios are proposed by adjusting the weight coefficients of different combinations, using reference experiment conditions and data to test the algorithm. A "shutdown-restart"energy-efficient strategy of reference experiment is used to determine whether the

machine works standby or "shutdown-restart" in idle state. For the three manufacturing scenarios, the experimental results show that the IMOMVO can solve a set of Pareto optimal solutions, and the experimental results are better than the HGSA in reference. There are two main innovations in the paper. The first one is to apply MOMVO algorithm to energy-efficient multi-objective optimization HFSP problem for the first time. The second one is to improve MOMVO algorithm, specifically improve *WEP* and *TDR* parameter settings, and update formulas. The limitations of the paper lie in the limitation of data and the absence of uncertainty in the manufacturing environment. In the further research we will consider the dynamic multi-objective model for energy-efficient HFSP and data-driven optimization of HFSP. Specifically, agent model is used to optimize the objective function and reduce the optimization calculation time.

6. Acknowledgments
The authors would like to thank the editor and anonymous referees whose comments helped a lot in improving the paper.

7. Supporting Information
The Supporting Information is available free of charge on the https://mekhub.cn/live_now/IMOMVO.git.

9. References

1. Abd Elaziz M. Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer. Expert Systems with Applications. 2019;125:112-129.
2. Ali EE. Parameter extraction of photovoltaic generating units using multi-verse optimizer. Sustainable Energy Technologies and Assessments. 2016;17:68-76.
3. Asadi-Gangraj E. Lagrangian relaxation approach to minimizing makespan in hybrid flow shop scheduling problem with unrelated parallel machines. Scientia Iranica. 2018;25(6):3765-3775.
4. Azami A, Demirli K, Bhuiyan N. Scheduling in aerospace composite manufacturing systems: a two-stage hybrid flow shop problem. The International Journal of Advanced Manufacturing Technology. 2018;95(9-12):3259-3274.
5. Behnamian J, Ghomi S, Zandieh M. Development of a hybrid metaheuristic to minimise earliness and tardiness in a hybrid flowshop with sequence-dependent setup times. International Journal of Production Research. 2010;48(5):1415-1438.
6. Chaoyong Z. Hybrid flow-shop scheduling problems based on improved migrating birds optimization algorithm. Computer Integrated Manufacturing Systems. 2019;25(03):643-653.
7. Chaoyong Z, et al. Efficient estimation of distribution for flexible hybrid flow shop scheduling. Acta Automatica Sinica. 2017;43(02):280-293.
8. Chaoyong Z, et al. Modeling and optimization for energy-efficient hybrid flow-shop scheduling problem. Computer Integrated Manufacturing Systems; c2018. p. 1-26.
9. Chaoyong Z. Estimation of distribution algorithm for energy-efficient scheduling in turning processes. Sustainability. 2016;8(8):762-782.
10. Chen YY. A hybrid approach based on the variable neighborhood search and particle swarm optimization for parallel machine scheduling problems - A case study for solar cell industry. International Journal of Production Economics. 2013;141(1):66-78.
11. Dai M. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. Robotics and Computer-Integrated Manufacturing. 2013;29(5):418-429.
12. Faris H, Aljarah I, Mirjalili S. Training feedforward neural networks using multi-verse optimizer for binary classification problems. Applied Intelligence. 2016;45(2):322-332.
13. Fathy A, Rezk H. Multi-verse optimizer for identifying the optimal parameters of PEMFC model. Energy. 2018;143:634-644.
14. Friedrich S, Brussau K, Voss S. A scheduling approach for the production of synthetic granules. PPS Management. 2008;13(2):23-26.
15. Gonzalez-Neira EM, et al. Stochastic flexible flow shop scheduling problem under quantitative and qualitative decision criteria. Computers & Industrial Engineering. 2016;101:128-144.
16. Gupta JND. Two-Stage, Hybrid Flowshop Scheduling Problem. Journal of the Operational Research Society. 1988;39(4):359-364.
17. Gupta JND, Tunc EA. Schedules for a two-stage hybrid flowshop with parallel machines at the second stage. International Journal of Production Research. 1991;29(7):1489-1502.
18. He YH, Hui CW. A novel search framework for multi-stage process scheduling with tight due dates. AIChE Journal. 2010;56(8):2103-2121.
19. Hu C, Li Z, Zhou T, Zhu A, Xu C. A multi-verse optimizer with levy flights for numerical optimization and its application in test scheduling for network-on-chip. PloS one. 2016;11(12):e0167341.
20. Hua X. Reentrant hybrid flow shop scheduling problem based on total weighted completion time. Control and Decision. 2018;33(12):2218-2226.
21. Jiang SL, Zhang L. Energy-oriented scheduling for hybrid flow shop with limited buffers through efficient multi-objective optimization. IEEE Access. 2019;7:34477-34487.
22. Khoury J, et al. From big crunch to big bang. Physical Review D: Particles, Fields, Gravitation, and Cosmology. 2001;65(8):381-399.
23. Kramer A, Dell'Amico M, Iori M. Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. European Journal of Operational Research. 2019;275(1):67-79.
24. Kumar A, Suhag S. Effect of TCPS, SMES, and DFIG on load frequency control of a multi-area multi-source power system using multi-verse optimized fuzzy-PID controller with derivative filter. Journal of Vibration and Control. 2018;24(24):5922-5937.
25. Lalitha JL, Mohan N, Pillai VM. Lot streaming in +N(m) hybrid flow shop. Journal of Manufacturing Systems. 2017;44(1):12-21.
26. Lee CY, Vairaktarakis GL. Minimizing makespan in hybrid flowshops. Operations Research Letters. 1994 Oct 1;16(3):149-158.
27. Gao L. A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop. IEEE Transactions on Engineering Management. 2018;65(2):330-340.

Commented [1]: the HGSA in reference

28. Lin JT, Chen CM. Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. Simulation Modelling Practice and Theory. 2015;51:100-114.
29. Liu Y, Farnsworth M, Tiwari A. Energy-efficient scheduling of flexible flow shop of composite recycling. The International Journal of Advanced Manufacturing Technology. 2018;97(1-4):117-127.
30. Lopez JC, Giraldo JA, Arango JA. Reduction of the makespan in scheduling the production of a hybrid-flexible flow shop (HFS). Information Technology. 2015;26(3):157-172.
31. Meng LL. Mathematical modelling and optimisation of energy-conscious hybrid flow shop scheduling problem with unrelated parallel machines. International Journal of Production Research. 2019;57(4):1119-1145.
32. Mirjalili S, Mirjalili SM, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications. 2016;27(2SI2):495-513.
33. Mirjalili S. Optimization of problems with multiple objectives using the multi-verse optimization algorithm. Knowledge-Based Systems. 2017;134:50-71.
34. Mirjalili SM, Mirjalili SZ. Asymmetric oval-shaped-hole photonic crystal waveguide design by artificial intelligence optimizers. IEEE Journal of Selected Topics in Quantum Electronics. 2015;22(2):258-264.
35. Mirjalili SM, Mirjalili SZ. Single-objective optimization framework for designing photonic crystal filters. Neural Computing and Applications. 2017;28(6SI):1463-1469.
36. Mirsanei HS. A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times. Journal of Intelligent Manufacturing. 2011;22(6):965-978.
37. Pan QK. An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process. IEEE Transactions on Automation Science and Engineering. 2013;10(2):307-322.
38. Peng T. Streamflow forecasting using empirical wavelet transform and artificial neural networks. Water. 2017;9(6):406.
39. Qi C. Improved genetic algorithm variable neighborhood search for solving hybrid flow shop scheduling problem. Computer Integrated Manufacturing Systems. 2017;23(09):1917-1927.
40. Salvador MS. A solution of a special class of flow shop scheduling problems [C]//Proc of the Symposium on the Theory of Scheduling and Its Applications. Berlin; c1973. p. 83-91.
41. Wang SY. Estimation of distribution algorithm for solving hybrid flow-shop scheduling problem with identical parallel machine. Computer Integrated Manufacturing Systems. 2013;19(06):1304-1312.
42. Shukri S. Evolutionary static and dynamic clustering algorithms based on multi-verse optimizer. Engineering Applications of Artificial Intelligence. 2018;72:54-66.
43. Tang DB. Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. Computers & Industry. 2016;81(SI):82-95.
44. Wang XY. Estimates of energy consumption in China using a self-adaptive multi-verse optimizer-based support vector machine with rolling cross-validation. Energy. 2018;152:539-548.
45. Wu X. Multi-objective flexible flow shop scheduling problem considering variable processing time due to renewable energy. Sustainability. 2018;10(3):841-871.
46. Xu G, Shekofteh Y, Akgül A, Li C, Panahi S. A new chaotic system with a self-excited attractor: entropy measurement, signal encryption, and parameter estimation. Entropy. 2018;20(2):86.
47. Yan JH, et al. A multi-level optimization approach for energy-efficient flexible flow shop scheduling. Journal of Cleaner Production. 2016;137:1543-1552.
48. Yunna T. A time window-based approach for multi-stage hybrid flow shop. Journal of Mechanical Engineering. 2016;52(16):185-196.
49. Zhang ZW. An improved scheduling approach for minimizing total energy consumption and makespan in a flexible job shop environment. Sustainability. 2019;11(1):179.
50. Zhao FQ. A factorial based particle swarm optimization with a population adaptation mechanism for the no-wait flow shop scheduling problem with the makespan objective. Expert Systems with Applications. 2019;126:41-53.
51. Zhao HR, Han XY, Guo S. DGM (1,1) model optimized by MVO (multi-verse optimizer) for annual peak load forecasting. Neural Computing and Applications. 2018;30(6):1811-1825.
52. Zhicong L. Improved biogeography-based optimization algorithm used in solving hybrid flow shop scheduling problem. CIESC Journal. 2016;67(03):751-757.
53. Zhi P. Multi-stage no-wait pharmaceutical flexible job shop scheduling. Journal of Zhejiang University Engineering Science. 2018;52(12):2253-2261.
54. Zhonghua H, et al. Improved DE algorithm for hybrid flow shop load balancing scheduling problem. Computer Integrated Manufacturing Systems. 2016;22(02):547-557.
55. Zhou SC. An effective discrete differential evolution algorithm for scheduling uniform parallel batch processing machines with non-identical capacities and arbitrary job sizes. International Journal of Production Economics. 2016;179:1-11.
56. Zohali H. Reformulation, linearization, and a hybrid iterated local search algorithm for economic lot-sizing and sequencing if n hybrid flow shop problems. Computers & Operations Research. 2019;104:127-138.