Multi-objective low-carbon hybrid flow shop scheduling via an improved teaching-learning-based optimization algorithm

Wenjie Wang [1], Xuesheng Zhou [2], Guangdong Tian [1*], Amir M. Fathollahi-Fard [3], Peng Wu [4], Chaoyong Zhang [5], Chengwen Liu [6] and Zhiwu Li [7,8]

1. Key Laboratory of High Efficiency and Clean Mechanical Manufacture (Ministry of Education), National Demonstration Center for Experimental Mechanical Engineering Education, School of Mechanical Engineering, Shandong University, Jinan 250061, China
2. College of Engineering, Nanjing Agricultural University, Nanjing 210095, China
3. Department of Electrical Engineering, École de Technologie Supérieure, University of Québec, 1100, Notre-Dame St. W., Montréal, Canada.
4. Fuzhou University, School of Econ & Management, Fuzhou 350108, China
5. State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
6. Ruixing Group Co., Ltd, Taian 271509, China
7. Institute of Systems Engineering, Macau University of Science and Technology, Taipa 999078, Macau, China
8. School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China

*Corresponding Author: Guangdong Tian. E-mail: tiangd2013@163.com. Tel: +86 15104317455,

Postal Address: No. 17923, Jingshi Road, Jinan, China.

Abstract: In this article, for achieving an effective and environmental-friendly production scheduling, we investigate a multi-objective low-carbon hybrid flow shop scheduling problem (MLHFSP) with the consideration of machines with varied energy usage ratios. The problem is formulated by a multi-objective mathematical model with two optimization objectives, i.e., minimizing total carbon emission (TCE) and makespan ($C_{\max}$). We primarily analyse on the formation of TCE and derive its mathematical expression. MLHFSP is NP-hard, therefore, to tackle the model, an improved multi-objective teaching-learning-based optimization (ITLBO) algorithm is proposed. The ITLBO algorithm mainly contains global search based teaching phase and local search based learning phase. In ITLBO, a solution is represented by two vectors, i.e., job sequence vector and machine assignment vector. Sigma method is utilized to quantify each individual, and to avoid local optimum, sequential neighbourhood search (SNS) method is also adopted. Experimental results validate the feasibility and effectiveness of proposed ITLBO in addressing MLHFSP. The research findings help manufacturing engineers to seek a sophisticated balance between carbon emission reduction and makespan reduction.

Keywords: Production scheduling; Carbon emission; Makespan; Hybrid flow shop scheduling; Teaching-learning-based optimization

1. Introduction

Production scheduling is taken as the key technique and core issue to the management of numerous manufacturing enterprises [1-3]. Nowadays, manufacturing industry has become the main source of resource consumption and pollutant emission all over the

world, which leads to continuous deterioration of ecological environment and hinders the healthy development of global economy [4-5]. According to Efficiency [6], the manufacturing industry is responsible for about 30% of energy consumption and about 40% of CO_2 emission. Strict laws and regulations are being promulgated by governments and relevant organizations to shape the manufacturing industry into an environment-friendly one [7-10].

It has been proven that scheduling is an effective way in reducing energy consumption of manufacturing industry [11]. Hybrid flow shop is close to manufacturing industry and is commonly adopted by manufacturing enterprises. It can be regarded as the combination of the general flow shop and parallel locomotive shop, thus has a more flexible production process and is more in line with practical situations [12-14]. Its derivative problem, hybrid flow shop scheduling problem (HFSP) can be found in many real production systems, such as chemical industry, paper and logistics industry [15]. In addition, the HFSP owns a theoretical significance due to its NP-hard property.

Fu et al. considered a stochastic HFSP with deteriorating jobs with two time-related optimization objectives, i.e., makespan and total tardiness and introduced a novel evolutionary algorithm to seek for optimal solutions [16]. Li et al. proposed an improved artificial bee colony algorithm for addressing distributed HFSP with sequence-dependent setup times where makespan was optimized [17]. A modified single-player Monte-Carlo tree search-based method was used to minimize the makespan of HFSP with multi-constraint [18].

With the global warming and energy resources shortage becoming more and more serious, increasing attention has been paid to energy conservation [19] and researches on energy-aware HFSP have become a hot topic recently. Chen et al. studied the energy-aware HFSP with lot streaming in which production makespan and electric power consumption were minimized [20]. A multi-objective genetic algorithm was designed to obtain Pareto solutions. An energy-aware HFSP with forging operations was proposed by Liu et al. [21]. They established a multi-objective model with diverse time elements considered and employed the Turn On/Off strategy to obtain a smaller energy consumption. Hasani et al. considered a HFSP with machine-dependent processing stages and aimed to seek a balance between energy consumption and production costs by non-dominated sorting genetic algorithm (NSGA-II) [22]. Li et al. discussed an energy-aware HFSP with different objective importance [23]. Compared with objectives of makespan and total tardiness, total energy consumption was most concentrated. Imperialist competitive algorithm was improved to find promising solutions and computational results verified solution efficiency and effectiveness. Zheng et al. investigated a two-stage HFSP with consideration of minimizing total energy costs and makespan for job batches, and developed a mixed ant colony optimisation algorithm to address this problem [24]. First input first output and Johnsons rules were utilized to schedule batches. Compared with NSGA-II and the strong Pareto evolutionary algorithm, effectiveness of the mixed algorithm was validated. Considering the varying energy prices, Schulz et al. established an energy-aware hybrid flow shop scheduling model where makespan, total energy costs and peak power were taken as the optimization objectives [25].

The methods used for solving production scheduling in HFSP related literature can mainly be divided into three groups, i.e., exact methods, heuristic algorithms and intelligent optimization algorithms. However, exact methods are only suitable for HFSP with a small scale and heuristic algorithms are sensitive to the characteristics of

actual problems [26]. Intelligent optimization algorithms which get inspired from natural phenomena and natural principles are favoured by the majority of scholars due to their characteristics of simple implementation, fast convergence and strong search ability for optimal solutions in addressing NP-hard problems [27-29]. Recently, Ding et al. adopted a discrete particle swarm optimization algorithm for addressing the energy-aware HFSP, where tabu search method was integrated to enhance the solution quality in search space [30]. Dai et al. proposed an energy- aware model to illustrate HFSP with two optimization objectives of total energy consumption and makespan [31]. To solve this model, genetic algorithm and annealing algorithm were combined to refine potential solutions.

Table **1** lists the relevant papers for solving HFSP. On the whole, most papers concentrated on either time-related objectives, cost-related objectives or energy-related objectives, such as makespan, total tardiness, production costs, total energy cost, total energy consumption, electric power consumption and so on. On the other hand, discharged carbon emission in various shops has great impacts on the environment and studying a low-carbon HFSP is an important and practical direction. However, there is little research on the carbon emission formulation and optimization. Moreover, in reality, machines often have different energy consumption characteristics and little research considers this yet. To address these gaps, this study considers a multi-objective low-carbon HFSP (MLHFSP) with the consideration of machines with different energy usage ratios and try to make a comprehensive trade-off between total carbon emission and makespan.

Please insert Table 1 here

The teaching-learning-based optimization (TLBO) algorithm, as a newly developing intelligent algorithm, has arisen extensive attention from scholars recently. TLBO algorithm firstly proposed by Rao et al. is an emerging nature-inspired population algorithm whose implementation imitates the actual teaching process of teachers and the learning process of students [32]. Because of its simplicity, faster convergence speed, strong convergence ability and less control parameters, the TLBO algorithm has been widely concerned in many fields, such as, machine learning [33], power distribution [34], parameter optimization [35], especially in the field of shop scheduling [36-39]. However, TLBO algorithm is primarily designed to deal with those continuous optimization problems, which is infeasible to address the studied discrete MLHFSP. Besides, TLBO algorithm can easily fall into local optimum and its convergence efficiency needs to be further enhanced. So, in this paper, we make several modifications on basic TLBO algorithm and design an improved TLBO (ITLBO) to solve the MLHFSP.

Compared with previous work, this work makes the following three contributions.

1) Investigating the multi-objective low-carbon hybrid flow shop scheduling problem (MLHFSP) with processing machines that have different energy usage ratios, where total carbon emission and makespan objectives are optimized simultaneously.

2) Formulating a multi-objective mathematical model with several constraints to express this NP-hard scheduling problem and designing an improved TLBO (ITLBO) algorithm to solve it.

3) Through numerical experiments, indicating the feasibility and effectiveness of the proposed ITLBO algorithm, compared with NSGA-II and MOEA/D under several evaluation metrics, in addressing such scheduling problems.

The rest of the paper is arranged as follows. Section 2 gives detailed problem descriptions on MLHFSP and shows the process of mathematical model

establishment. The ITLBO algorithm is introduced in Section 3. Comparison experiments with other algorithms are conducted on several designed instances, and their results and discussions are drawn in Section 4. Section 5 concludes the paper and gives future researches.

2. Problem descriptions and model establishment

2.1. Problem description

As shown in **Fig. 1**, the studied MLHFSP can be organized as follows. There are a set of jobs to be processed at a series of processing stages sequentially from the first stage to the final stage and each stage is composed of two or more identical parallel machines with distinct energy usage ratios. The problem is to decide the job sequence for each processing stage and machine assignment for each job on the purpose of minimizing total carbon emission (TCE) and makespan ($C_{\max}$).

In the studied problem, the processing time of a job is fixed no matter which machine is used. But machines at a same stage are with different energy usage ratios. If a lower total carbon emission is favoured, jobs will prefer to be processed on machines with large energy usage ratio, which will leave machines with small energy usage ratio idle for a long time. Obviously, it will lead to a higher makespan and vice versa. In other words, there exists a trade-off between TCE and $C_{\max}$.

Please insert Fig. 1 here

Also, assumptions declared to describe the studied scheduling problem are organized as follows: (1) No job can be processed on two or more machines at a time, and a machine can only process one job at a time; (2) Once a job is processed on a specific machine, it will stay on the machine until finished; (3) Jobs travels sequentially through all stages and stage skipping of all jobs is not permitted; (4) A job will be transported to next processing stage instantly once it is finished at a specific stage, and transportation times for jobs between any two consecutive stages are ignored; (5) Machines are in good condition and random failures are not considered; (6) Setup times are sequence-independent, thus could be included in processing time; (7) Machines will be turned on once the first job assigned to them and will be shut down once all jobs assigned to them are finished.

2.2. Notation illustration

Symbol	Description
J	Set of jobs to be scheduled and $J=\{1,2,\ldots,i,\ldots,n\}$.
S	Set of processing stages and $S=\{1,2,\ldots,j,\ldots,s\}$.
M_j	Set of identical parallel machines at stage j and $M_j=\{1,2,\ldots,k,\ldots,\mathrm{m}_j\}$
M_{jk}	k-th machine at stage j
L	Set of machine positions and $L=\{1,2,\ldots,l,\ldots,n\}$.

$t_{i,j}$	Processing time of job *i* at stage *j*.
$s_{i,j}$	Start time of job *i* at the stage *j*.
$e_{i,j}$	Completion time of job *i* at the stage *j*.
$B_{j,k,l}$	Beginning time of the job at position *l* of machine *k* in stage *j*.
$E_{j,k,l}$	Ending time of the job at position *l* of machine *k* in stage *j*.
P_{work}^{j}	Power of machines working jobs at stage *j*.
P_{idle}^{j}	Idle power of machines in stage *j*.
Q	A very large positive number.
$r_{j,k}$	Energy usage ratio of machine *k* in stage *j*.

2.3. Two optimization objectives

$$\mathrm{TCE} = K_c \times \left[\sum_{i\in J} \sum_{j\in S} \sum_{k\in M_j} \sum_{l\in L} \frac{P_{work}^{j} \cdot t_{i,j} \cdot x_{i,j,k,l}}{r_{j,k}} + \sum_{j\in S} \sum_{k\in M_j} \sum_{l\in\{1,2,\ldots,n-1\}} \left(B_{j,k,l+1} - F_{j,k,l} \right) \cdot P_{idle}^{j} \right] \quad (1)$$

$$C_{\max} = \max_{i\in J} \left\{ e_{i,s} \right\} \quad (2)$$

Eq. (1) shows the total carbon emission in the shop, elements in brackets refers to total energy consumption in the period. According to [40], there exists a linear relationship between carbon emission and energy consumption in manufacturing. Parameter K_c represents the electricity carbon emission factor and is set to 0.1524 g CO_2/kJ according to [40]. The left symbol denotes the energy consumption when machines are in processing state; and the right symbol denotes the energy consumption when machines are in idle state.

Eq. (2) formulates the makespan, i.e., the maximum of completion times of all jobs at the last processing stage.

2.4. Mathematical model

According to the characteristics of the studied MLHFSP, its corresponding mathematical model is established as bellows:

$$\mathrm{Min}\ f_1 = \mathrm{TCE} \quad (3)$$

$$\mathrm{Min}\ f_2 = C_{\max} \quad (4)$$

s.t.

$$\sum_{k\in M_j}\sum_{l\in L} x_{i,j,k,l} = 1 \quad \forall i \in J, j \in S \tag{5}$$

$$\sum_{i\in J} x_{i,j,k,l} \le 1 \quad \forall j \in S, k \in M_j, l \in L \tag{6}$$

$$\sum_{i\in J} x_{i,j,k,l} \ge \sum_{i\in J} x_{i,j,k,l+1} \quad \forall j \in S, k \in M_j, l \in \{1,2,\ldots,n-1\} \tag{7}$$

$$e_{i,j} = s_{i,j} + t_{i,j} \times \sum_{k\in M_j}\sum_{l\in L} x_{i,j,k,l} \quad \forall\ i \in J, j \in S \tag{8}$$

$$F_{j,k,l} = B_{j,k,l} + t_{i,j} \times \sum_{i\in J} x_{i,j,k,l} \quad \forall j \in S, k \in M_j, l \in L \tag{9}$$

$$e_{i,j} \le s_{i,j+1} \quad \forall i \in J, j \in \{1,2,\ldots,s-1\} \tag{10}$$

$$F_{j,k,l} \le B_{j,k,l+1} \quad \forall j \in S, k \in M_j, l \in \{1,2,\ldots,n-1\} \tag{11}$$

$$B_{j,k,l} \le s_{i,j} + \mathrm{M} \times \left(1 - x_{i,j,k,l}\right) \quad \forall i \in J, j \in S, k \in M_j, l \in L \tag{12}$$

$$B_{j,k,l} \ge s_{i,j} - \mathrm{M} \times \left(1 - x_{i,j,k,l}\right) \quad \forall i \in J, j \in S, k \in M_j, l \in L \tag{13}$$

$$s_{i,j} \ge 0 \quad \forall i \in J, j \in S \tag{14}$$

$$B_{j,k,l} \ge 0 \quad \forall j \in S, k \in M_j, l \in L \tag{15}$$

$$x_{i,j,k,l} = \begin{cases} 1, & \text{if job } i \text{ is assigned to the } l\text{-th position of machine } k \text{ at stage } j \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

The objective (3) is to minimize the TCE, and the objective (4) is to minimize the makespan $C_{\max}$. Constraint (5) guarantees that each job can only be assigned to only one position of one machine at per processing stage. Constraint (6) ensures that no two or more jobs can be assigned to a same position of one machine at per processing stage. Constraint (7) makes sure that jobs must be assigned to preceding positions of one machine and position skip is not allowed. Constraints (8) and (9) ensure that job operations cannot be interrupted. Constraints (10) and (11) specify the order of time. (10) makes sure that a job cannot be transferred to next stage until its current processing operations have been done. (11) makes sure that a machine can only process the next job after finishing the current one. Constraints (12) and (13) try to establish the relationship between $B_{j,k,l}$ and $s_{i,j}$. Constraints (14)-(16) describe some variables and (16) is a binary variable.

3. Improved teaching-learning-based optimization (ITLBO) algorithm

In this section, we introduce an improved TLBO algorithm (ITLBO) to address the MLHFSP and its procedures are summarized in **Fig. 2**. Compared with basic TLBO, we make the following five improvements: (1) a two-vector encoding mechanism and an integrated initialization strategy are designed to represent solutions efficiently and

improve the quality of initial population; (2) an external archive with fixed capacity is set to store the elite teachers iteratively updated via Sigma method to guide the whole population; (3) three crossover operators are specially designed and conducted between teachers and students in teaching phase to enhance the global search ability; (4) three neighbourhood structures based sequential neighbourhood search (SNS) method is introduced and implemented on all individuals in learning phase to enhance the local search capability; (5) self-learning is removed from basic TLBO and individuals that haven't been improved in learning phase in several rounds will be abandoned and replaced by a new individual so as to skip local optimum.

Please insert Fig. 2 here

3.1. Encoding and decoding representation

The basic TLBO algorithm is primarily designed to deal with those continuous optimization problems, and due to that the studied MLHFSP is essentially a discrete problem, encoding and decoding methods ought to be highly addressed to suit the actual condition. The studied problem contains two sub-problems, i.e., job sequence and machine assignment. To achieve an effective schedule, a solution is encoded by two vectors: job sequence vector (JV) and machine assignment vector (MV).

For the MLHFSP with *n* jobs, *m* stages, a solution is represented in **Fig. 3**. The JV represents a feasible job sequence and has the same dimension as job count. π_i indicates job index and *n* denotes the job count. The MV is a $n \times s$ matrix that records the machine selection along sequential processing stages. z_{ij} represents the machine which job *i* assigned at stage *j*.

Please insert Fig. 3 here

To get corresponding schedules, the decoding process is also designed. Its detailed procedure can be divided into following steps:

Step 1: Initialize available times of all machines and completion times of all jobs to zero.

Step 2: At processing stage 1(i.e., current stage index *j*=1), pick out job π_i from SV one-by-one and do followings: (1) select its assigned machine z_{i1} from MV and available time of this machine; (2) compute completion time of job π_i by adding its processing time; (3) update available time of machine z_{i1}. $j = j + 1$, go to **Step 3**.

Step 3: If current processing stage index satisfies $j \le s$, implement **Step 4**; if not, implement **Step 5**.

Step 4: Based on First Come First Serve (FCFS) principle, arrange completion times of all jobs at previous stage in ascending order and obtain a new job sequence $\left[\pi_1', \pi_2', \ldots, \pi_n'\right]$. And then pick out job π_i' from it one-by-one and do followings: (1) select its assigned machine $z_{\pi_i' j}$ from MV and available time of this machine; (2) compute completion time of job π_i' by adding its processing time; (3) update available time of machine $z_{\pi_i' j}$. $j = j + 1$, go to **Step 3**.

Step 5: A feasible schedule is produced, and then output TCE and $C_{\max}$ via Eqs. (1) and (2).

3.2. Population initialization

In the ITLBO algorithm, each student/teacher represents a potential solution. Obviously, the quality of initial populations has great influences on the convergence speed and solution quality of an algorithm. Random generation, as a common initialization method, can be easily coded and implemented. However, the initial population generated by this is not ideal in terms of not only the optimal solution quality also the convergence speed. If the corresponding initialization method is designed based on the characteristics of specified optimization objectives, more desired solutions can be produced at the beginning, thus improving the algorithm efficiency. As a result, random generation and strategy determination are combined to generate a set of initial solutions with high quality.

As discussed in **Section 2.1**, there exists a trade-off between TCE and $C_{\max}$. In this paper, random generation is used for 20% of the initial population, that is, randomly sorting the numbers from 1 to n in SV and randomly assigning machines to MV with constraint of limited machine count per processing stage. 40% is built based on a low TCE strategy, that is, choosing machines with high energy usage ratios as much as possible in MV. For the rest 40%, we adopt a low $C_{\max}$ strategy, that is, distributing jobs as evenly as possible to parallel machines so as to enable jobs to enter next stage earlier and shorten $C_{\max}$. Clearly, the integrated initialization method is easy to conduct and can improve the solution quality and diversity of initial population. The population is simply denoted as set P.

3.3.Multi-objective optimization, sigma method and archive maintenance

As discussed in **Section 2**, two objectives are formulated in this paper, i.e., TCE and $C_{\max}$. When objectives are in a state of conflict, there will be no optimal solution that can make all the objectives reach the maximum or minimum simultaneously, we can only seek non-dominated solutions or Pareto solutions where the improvement of one objective is at cost of spoiling other objectives.

After implementation of initialization process, population P of size N is produced. In order to quantitatively evaluate all the students/individuals, sigma method proposed by [41] is used. Its calculation procedures contain the following two steps:

Step 1: All of the solutions will be sorted by employing the fast non-dominated sorting technique [42] and finally, each solution will get its corresponding rank.

Step 2: Calculate the sigma values for all individuals via:

$$Sigma_i = \sum_{k=1}^{2} \left[f_k(X_i) \Big/ \sum_{j=1}^{N_{\text{rank}}} f_k(j) \right] + (\text{rank}_i - 1) \times 2 \tag{17}$$

where $Sigma_i$ is the sigma value of individual X_i; $f_k(\text{X}_i)$ is the k-th objective value of individual X_i; rank_i corresponds to the rank layer index in which individual X_i lies; and N_{rank} represents the count of individuals that with the same rank rank_i. Note that individuals with small sigma values are preferred. Sigma method is also presented in **Fig. 4**. For more details on sigma method and its applications in scheduling field, please also refer to [43-45].

Please insert Fig. 4 here

In proposed ITLBO algorithm, an external archive Ω that preserves the best individuals (Pareto solutions) so far is set and Ω will be updated each generation. Meanwhile, there also exists a finite set Ψ called teacher set in ITLBO. Ψ stores relatively good individuals with size of $N_T (1 < N_T < N)$ and those individuals are treated as teachers. In other words, there exists N_T teachers in the whole class to improve the students' scores. And according to [46], it is feasible and effective to employ more than one teacher to solve NP-hard optimization problems.

As **Fig. 2** shows, Ω works as follows: At the beginning of ITLBO algorithm, Ω is formed by non-dominated solutions in initial population; at the end of a specific generation, Ω will be combined with P' and fast non-dominated sorting technique is utilized to this mixed population. The individuals with rank 1 will be thrown into new Ω, that is how archive Ω is updated. Note that Ω is an external set with an unfixed capacity. On the other hand, Ψ works as follows: At the beginning of ITLBO algorithm, sigma method is employed into initial population and individuals with N_T best sigma values make up for Ψ; at the end of a specific generation, previous population P evolves to population P' after teaching phase and learning phase, and then sigma method will be used again on P' to determine N_T best solutions, that is how set Ψ is updated. Note that a bigger N_T can occupy huge computing resources and is not conducive to the population evolution. Therefore, N_T should be carefully determined.

3.4. Teaching phase

Teaching phase aims to explore promising regions in the search space. In this phase, individuals/students get their performance improved through learning knowledge from the teacher. The crossover operator is commonly used to imitate the information sharing process because of its strong global search capability. Considering the characteristics of the studied problem, three crossover operators are designed for JV and MV to improve the diversity of population.

(1) two crossover operators for JV: **Two-point crossover (TPX)** and **Order crossover (OX)** are designed and their corresponding processes are presented in **Fig. 5**.

(2) one crossover operator for MV: Due to the difference in quantity of parallel machines in those processing stages, a simple crossover operator is designed for MV. It works as follows: 1) determine a teacher $X_{teacher}$ and a student X_i; 2) randomly select two positions that satisfy $1 \le \text{Position}1 < \text{Position}2 \le ns$; 3) replace the elements between the two positions in X_i with the elements in $X_{teacher}$ on same positions. The crossover operator for MV is called **Direct crossover (DX)** and its procedure is also visualized in **Fig. 6**.

Please insert Fig. 5 here

Please insert Fig. 6 here

In this teaching phase, each generation individual gets its scores enhanced by learning from one teacher that randomly selected from set Ω, as presented in **Fig. 2**. At this time, two conditions exist: I) If the current individual happens to be a constituent member of Ω, then it learns knowledge from one of other teachers in Ω; II) If the current individual doesn't belong to Ω, then directly pick up one teacher from Ω

and learn from him.

The interaction among three designed crossover operators guarantees the diversity of population in search space, and only new individuals X_i^{new} are produced while $X_{teacher}$ remains unchanged. It should be noted that executing all the three crossover operators is time-consuming, thus we developed a novel teaching strategy where operators are triggered by two probabilities p_1 and p_2, which is presented in **Fig. 2**. Usually, solving job sequence problem is more complicated and difficult than addressing machine assignment problem, thus we give a bigger opportunity for JV to execute its crossover operations. We set p_2 as 0.6 and TPX shares a same probability with OX, which is $p_1 = 0.3$ based on a number of experiments.

3.5. Learning phase

Learning phase aims to enhance the local search capability of algorithm, it simulates the phenomenon that students enrich their knowledge with the assistance of their mutual interaction. In ITLBO algorithm, an effective SNS method is adopted to refine a candidate solution around promising areas. The SNS method is composed of three neighbour structures, including neighbour structure job insertion, neighbour structure job swap and neighbour machine assignment. The former two is designed for JV and the last one is for MV, which is presented as follows:

Neighbour Structure Job Insertion (NS1): Randomly select a job in JV and insert it to a different position. Meanwhile, MV keeps unchanged. An example is shown in **Fig. 7**.

Neighbour Structure Job Swap (NS2): Randomly pick two jobs in JV and exchange them. Similar to NS1, MV keeps unchanged. An example is shown in **Fig. 7**.

Neighbour Structure Machine Assignment (NS3): Randomly select one element in MV and find its corresponding processing stage. Assigned a different machine index constrained by machine quantity in this stage to it. In NS3, JV remains unchanged. An example is shown in **Fig. 7**.

Please insert Fig. 7 here

To avoid falling into local optimum, an individual will be abandoned and replaced by a new randomly generated one if a certain number of neighbourhood execution cannot find a better individual. In ITLBO, the certain number is controlled by a positive integer $Limit > 1$ and neighbourhood execution index is set as *count* . Overall, the SNS method in the learning phase (also visualized in **Fig. 2**) is as follows:

Step 1: Set execution index *count* to 0.

Step 2: Apply NS1 on current individual X_i to obtain a new individual X_i^{new}. If X_i^{new} dominates X_i, go to **Step 6**; else continue to the next step, i.e., **Step 3**.

Step 3: Apply NS2 on current individual X_i to obtain a new individual X_i^{new}. If X_i^{new} dominates X_i, go to **Step 6**; else continue to the next step, i.e., **Step 4.**

Step 4: Apply NS3 on current individual X_i to obtain a new individual X_i^{new}. If X_i^{new} dominates X_i, go to Step 6; else do *count* = *count* +1 and go to **Step 5**.

Step 5: If *count* ≤ Limit, go to **Step 2**; else go to **Step 7**.

Step 6: Replace X_i with the better X_i^{new}.

Step 7: Replace X_i with a new randomly produced individual.

The SNS continues to run until all of individuals in P are updated.

3.6. Termination rules

As shown in **Fig. 2**, maximum CPU time is determined as the stop condition of ITLBO algorithm. By this, we can apply more computational resources for instances with large scales. The maximum CPU time is characterized as $n \times s \times v$ (unit: ms), and recall that n refers to the count of jobs and s denotes the count of processing stages. v is a fixed value and is set as 600 based on our pretesting results.

4. Experiments and results

In this section, we carry out extensive experiments to test the performance of proposed ITLBO algorithm in dealing with MLHFSP. Note that all the experiments are coded in MATLAB and runs on an Intel i7-7700 2.80GHz PC with 8 GB memory.

4.1 Test instances generation

As described above, test instances used for addressing the MLHFSP are composed of count of jobs (n), count of processing stages (s), count of unrelated parallel machines in each processing stage (m_j), working time of jobs on stages ($t_{i,j}$), processing/idle powers of machines (P_{work}^j / P_{idle}^j) and energy usage ratios of machines ($r_{j,k}$).We initialize the above data by follows. Specifically, there is a set of jobs $n \in [20,30,40,60,80,100]$ available. The count of processing stages has four levels $s \in [3,5,8,10]$ and m_j are randomly produced from a discrete uniform distribution with arrange of $U[2,5]$, where U represents the normal distribution. $t_{i,j}$ are randomly created from $U[1, 99]$. P_{work}^j and P_{idle}^j are randomly produced from discrete uniform distribution on $[4,8]$ and $[1,3]$, respectively. At last, $r_{j,k}$ are sampled from $U[0.7,1.0]$.

Considering the levels of n and s, we have 24 problem configurations in total. For each configuration, five instances will be created and therefore, we can get a group of 120 instances for experimental tests.

4.2 Comparative algorithms and evaluation metrics

To test the performance of ITLBO in addressing MLHFSP with 120 instances, two popular multi-objective optimization algorithms, NSGA-II [42] and MOEA/D [47] are chosen as its peers. NSGA-II is based on the evolution of gene and MOEA/D is on the basis of the idea that a multi-objective problem can be decomposed into several single objective optimization problems. The two algorithms are commonly used in solving optimization problems and show good performance.

Several performance metrics that can be employed to assess the results of multi-objective optimization algorithms. In the experiment, three following evaluation metrics are selected to evaluate algorithm performance quantitatively finally. Note that $\mathrm{PF}_{\mathrm{obtain}}$ and $\mathrm{PF}_{\mathrm{true}}$ refer to the PF obtained by a specific algorithm and the PF

gotten from PFs among all runs of different algorithms after Pareto dominance, respectively.

1) **Generational Distance Metric** (GD): this metric is used to measure how close $\mathrm{PF}_{\mathrm{obtain}}$ is to $\mathrm{PF}_{\mathrm{true}}$, formulated as [48]:

$$\mathrm{GD}=\frac{1}{N}\sqrt{\sum_{i=1}^{N}d_i^2} \tag{18}$$

where N refers to the size of $\mathrm{PF}_{\mathrm{obtain}}$, d_i refers to the Euclidean distances between the i-th solution in $\mathrm{PF}_{\mathrm{obtain}}$ and its nearest point in $\mathrm{PF}_{\mathrm{true}}$. Usually, a smaller GD value indicates an algorithm is with better convergence.

2) **Spread Metric** (Δ): this metric is to illustrate the diversity of solutions in $\mathrm{PF}_{\mathrm{obtain}}$, formulated as [48]:

$$\Delta=\frac{d_f+d_l+\sum_{i=1}^{N-1}\left|d_i-\bar{d}\right|}{d_f+d_l+\left(N-1\right)\bar{d}} \tag{19}$$

where $\bar{d}$ refers to the average of all distances d_i, d_f and d_l represent the Euclidean distances between extreme solutions in $\mathrm{PF}_{\mathrm{obtain}}$ and boundary solutions in $\mathrm{PF}_{\mathrm{true}}$. The smaller Δ is, the better the $\mathrm{PF}_{\mathrm{obtain}}$ is in terms of distribution and diversity. To eliminate the influence of dimensions, a normalization process is utilized.

3) **Inverted Generation Distance Metric** (IGD): this a comprehensive metric to reflect both convergence and diversity, formulated as [48]:

$$\mathrm{IGD}=\frac{1}{N^*}\sqrt{\sum_{i=1}^{N^*}d_i^{*2}} \tag{20}$$

where N^* represents the count of solutions in $\mathrm{PF}_{\mathrm{true}}$. As a variation of the GD indicator, d_i^* in IGD refers to Euclidean distances between the i-th solution in $\mathrm{PF}_{\mathrm{true}}$ and its nearest point in $\mathrm{PF}_{\mathrm{obtain}}$. Similar to the GD indicator, a smaller IGD is preferred. In summary, lower GD, Δ, and IGD values are more desired.

4) **Hyper-volume Metric** (HV): this metric is to illustrate the diversity and advancement of solutions in $\mathrm{PF}_{\mathrm{obtain}}$, formulated as [49]:

$$\mathrm{HV}=\bigcup_{i=1}^{\left|\mathrm{PF}_{\mathrm{obtain}}\right|}V_i \tag{21}$$

where $\left|\mathrm{PF}_{\mathrm{obtain}}\right|$ refers to the size of $\mathrm{PF}_{\mathrm{obtain}}$. V_i represent the hyper-volume produced from ith non-dominated solution in $\mathrm{PF}_{\mathrm{obtain}}$ to a reference point which is dominated by all non-dominated solutions. The MLHSFP is a minimization problem and the reference point is set as (1.2, 1.2). Meanwhile, we normalize the objective values of solutions in $\mathrm{PF}_{\mathrm{obtain}}$ into $\left[0,1\right]$.Usually, a bigger HV indicates a better approximation and distribution.

4.3 Parameter setting

As mentioned above, three main parameters, i.e., population size N, teacher number N_{T}, and maximum neighbourhood execution count *Limit* in proposed ITLBO

algorithm should be determined. In this paper, we use design of experiment (DOE) method to determine an optimal parameter combination of the three main parameters. As **Table 2** represents, each parameter has four different levels and therefore, a L_{16} orthogonal array is established.

Please insert Table 2 here

The proposed ITLBO algorithm will run 20 times independently on problem configuration 30×5, and the average IGD value for 20 times is selected as average responding value (ARV). The computational results are reported in **Table 3**. Based on the data in **Table 3**, **Table 4** presents significant rank of three main parameters by DOE. Also, **Fig. 8** shows the parameter trends. It can be seen from **Table 4** that $Limit$ has the biggest influence on the performance of ITLBO, followed by parameters N and N_{T}. Given a fixed running time, if $Limit$ is too small, individuals may not occupy enough opportunities to refine themselves. Population size N plays a second important role as it has a direct relationship with population diversity and global search capability. The existence of multiple teachers can accelerate the convergence speed of algorithm, however, if N_{T} is too large, the premature problem will occur. From **Fig. 8**, ITLBO can gain a better performance when $N=80$, $N_{\mathrm{T}}=0.2*N$ and $Limit=20$. This parameter configuration will be used in the following numerical experiments.

Please insert Fig. 8 here
Please insert Table 3 here
Please insert Table 4 here

4.4 Comparison results

In this section, we are going to compare the performance of the proposed ITLBO algorithm with NSGA-II and MOEA/D based on GD, Δ and IGD indicators. For an instance, each algorithm will run five times independently and its corresponding three indicator values will be obtained, and then obtained values are averaged and grouped under the same problem configuration $n\times s$. Computational results are reported in **Table 5** where best results among the algorithms are highlighted. In addition, to provide the results with confidence, additional statistical tests are conducted by using one-factor analysis of variance (ANOVA) technology in which the algorithm type is taken as a single factor. Mean plots with Tukey honestly significant difference intervals (95% confidence level) of GD, Δ and IGD indicators are presented in **Figs. 9-11**, respectively. Besides, experimental results on the mean hyper-volume values are recorded and they are also reported in **Table 6**, where the best results among the three algorithms are highlighted.

Please insert Fig. 9 here
Please insert Fig. 10 here
Please insert Fig. 11 here

From **Table 5**, regarding the comparison results on GD, the GD values of ITLBO vary from 0.0098 to 0.0862 while those are between 0.0086 and 0.3240 for comparative algorithms NSGA-II and MOEA/D, which indicates the good solving capability of ITLBO. From the Δ comparison results we can find that, ITLBO outperforms the comparative algorithms in most instances, and in one case it is the opposite. The average Δ values indicate that ITLBO is superior than the rest in terms of solution distribution and NSGA-II is inferior than it. Based on IGD computational

results, ITLBO performs better than NSGA-II and MOEA/D in convergence ability to true Pareto fronts and solution diversity. As presented in **Table 6**, although the proposed ITLBO algorithm performs slightly poorly when test instance is 80×5, for most instances (in 23 out of 24), the ITLBO algorithm receives higher HV values than NSGA-II and MOEA/D, which indicates that ITLBO performs better than NSGA-II and MOEA/D.

Regarding **Figs. 9-11**, we can find that the proposed ITLBO algorithm gets the lowest mean GD, Spread and IGD values among the three algorithms. ITLBO receives the smallest operating deviation in terms of GD and IGD metrics. Although NSGA-II is slightly more stable than ITLBO for Spread metric, but its mean value is not ideal. From the above results, we can draw a conclusion that proposed ITLBO is competitive and performs significantly better in generating promising candidates than its peers. The success of ITLBO relies on its algorithm design. First, two-vector encoding mechanism and hybrid initialization strategy provide a solid foundation. Second, crossover operators and neighbourhood structures are specially designed to enhance the global search ability and local search capability. Third, we introduce some techniques to skip the local optimum.

Please insert Table 5 here

Please insert Table 6 here

To intuitively represent the performance of ITLBO and its peers, **Fig. 12** plots Pareto fronts gotten by those algorithms in four problem configurations. The horizontal axis represents the maximum makespan and the vertical axis refers to the total carbon emission. It can be seen that the ITLBO can find the non-dominated solutions with better approximation and good distribution, and the Pareto front of ITLBO is lower than that of the rest compared algorithms. Therefore, the effectiveness of proposed ITLBO is confirmed and verified. What is more, as shown in **Fig. 12**, ITLBO generates a wide range of solutions, which can provide manufacturing enterprises/manufacturers with more choices. Manufacturers with a strong sense of time can choose a schedule with small makespan, while manufacturers who cares about the processing impacts on the environment may prefer a schedule with low carbon emission.

Please insert Fig. 12 here

4.5 Managerial insights

This work owns important practical significance for management. One of the most essential aspects of the hybrid flow shop management is to design an appropriate schedule to manufacture different kinds of products or jobs. Besides, the MLHFSP takes total carbon emission and makespan as its optimization goals, which has great significance in realizing a green and effective schedule and can gain a comprehensive trade-off between environmental impacts and production efficiency. In addition, the MLHFSP considers machines with different energy usage ratios, which is a general configuration and can be applied to various real-world manufacturing enterprises. Moreover, this work uses ITLBO algorithm to address the proposed scheduling problem effectively.

5. Conclusion

In this paper, we investigated a MLHFSP with total carbon emission and makespan. A

multi-objective mathematical model based on those two optimization objectives is established. To tackle it, ITLBO algorithm is proposed, where a two-vector encoding mechanism and an integrated initialization strategy are designed to represent solutions. In ITLBO, an external archive with fixed capacity is set to store the elite teachers iteratively updated via Sigma method to guide the whole population. Three crossover operators and three neighbourhood structures are specially designed to enhance the global and local search capability. In addition, to skip local optimum, individuals that haven't been improved in several rounds will be abandoned and replaced by a new individual. Regarding the experimental results, ITLBO can deal with the studied problem more effectively, leading to non-dominated solutions with better approximation and good distribution. Our future works will concentrate on two directions: (1) encode the ITLBO algorithm and implant codes into a computer-assisted support system for solving MLHFSP intelligently; (2) introduce some fuzzy theories into this filed for dealing with uncertain processing details [50-51].

Acknowledgements

This work is supported in part by National Natural Science Foundation of China (Grant Nos 52075303 and 51775238), in part by State Key Laboratory of Robotics and Systems (Grant No. SKLRS-2021-KF-09), and in part by State Key Laboratory of Fluid Power & Mechatronic Systems (Grant No. GZKF-202012), and in part by Fundamental Research Funds for the Central Universities (Grant No. 2019GN048).

Disclosure Statement

The authors declare that they have no competing interests.

Authors' Contribution Statement

Wenjie Wang: Data curation, Writing - original draft, preparation.
Xuesheng Zhou: Investigation, Data curation.
Guangdong Tian: Conceptualization, Methodology, Software, Supervision.
Amir M. Fathollahi-Fard: Software, Validation.
Peng Wu: Supervision.
Chaoyong Zhang: Supervision.
Chengwen Liu: Software
Zhiwu Li: Review & editing.

References

1. Gong, G.L., Chiong, R., Deng, Q.W., et al. "A memetic algorithm for multi-objective distributed production scheduling: minimizing the makespan and total energy consumption", *Journal of Intelligent Manufacturing*, **31**(6), pp. 1443-1466 (2020).
2. Tan, M., Chen, Y., Su, Y.X., et al. "Integrated optimization model for industrial self-generation and load scheduling with tradable carbon emission permits", *Journal of Cleaner Production*, **210**, pp. 1289-1300 (2019).
3. Kannan, G., Haq, A.N. and Devika, M. "Analysis of closed loop supply chain using genetic algorithm and particle swarm optimisation", *International Journal of Production Research*, **47**(5), pp. 1175-1200 (2009).

4. Cao, Y.R., Liu, J., Yu, Y., et al. "Impact of environmental regulation on green growth in China's manufacturing industry-based on the Malmquist-Luenberger index and the system GMM model", *Environmental Science and Pollution Research*, **27**(33), pp. 41928-41945 (2020).
5. Meng, K., Qian, X.M., Lou, P.H., et al. "Smart recovery decision-making of used industrial equipment for sustainable manufacturing: belt lifter case study", *Journal of Intelligent Manufacturing*, **31**(1), pp. 183-197 (2020).
6. Efficiency, E. "Tracking industrial energy efficiency and CO_2 emissions", *International Energy Agency*, **34**(2), pp. 1-12 (2007).
7. Far, M.H., Haleh, H. and Saghaei, A. "A flexible cell scheduling problem with automated guided vehicles and robots under energy-conscious policy", *Scientia Iranica*, **25**(1), pp. 339-358 (2018).
8. Xiong, G., Wang, F.Y., Nyberg, T.R., et al. "From mind to products: Towards social manufacturing and service", *IEEE-CAA Journal of Automatica Sinica*, **5**(1), pp. 47-57 (2018).
9. Tian, G.D., Zhang, H.H., Feng, Y.X., et al. "Green decoration materials selection under interior environment characteristics: A grey-correlation based hybrid MCDM method", *Renewable & Sustainable Energy Review*, **81**(1), pp. 682-692 (2018).
10. Cao, Y.R., Liu, J., Yu, Y., et al. "Impact of environmental regulation on green growth in China's manufacturing industry-based on the Malmquist-Luenberger index and the system GMM model", *Environmental Science and Pollution Research*, **27**(33), pp. 41928-41945 (2020).
11. Li, Y.B., Huang, W.X., Wu, R., et al. "An improved artificial bee colony algorithm for solving multi-objective low-carbon flexible job shop scheduling problem", *Applied Soft Computing*, **95**, pp. 106544 (2020).
12. Gholami, H.R., Mehdizadeh, E. and Naderi, B. "Mathematical models and an elephant herding optimization for multiprocessor-task flexible flow shop scheduling problems in the Manufacturing Resource Planning (MRPII) system", *Scientia Iranica*, **27**(3), pp. 1562-1571 (2020).
13. Fu, Y.P., Tian, G.D., Fathollahi-Fard, A.M., et al. "Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint", *Journal of Cleaner Production*, **226**, pp. 515-525 (2019).
14. Chen, J., Wang, M.L., Kong, X.T.T., et al. "Manufacturing synchronization in a hybrid flowshop with dynamic order arrivals", *Journal of Intelligent Manufacturing*, **30**(7), pp. 2659-2668 (2019).
15. Fonseca, G.B., Nogueira, T.H. and Ravetti, M.G. "A hybrid Lagrangian metaheuristic for the cross-docking flow shop scheduling problem", *European Journal of Operational Research*, **275**(1), pp. 139-154 (2019).
16. Fu, Y.P., Zhou, M.C., Guo, X.W., et al. "Scheduling dual-objective stochastic hybrid flow shop with deteriorating jobs via bi-population evolutionary algorithm", *IEEE Transactions on Systems Man Cybernetics-Systems*, **50**(12), pp. 5037-5048 (2020).
17. Li, Y.L., Li, X.Y., Gao, L., et al. "An improved artificial bee colony algorithm for distributed heterogeneous hybrid flowshop scheduling problem with sequence-dependent setup times", *Computers & Industrial Engineering*, **147**, pp. 106638 (2020).
18. Guo, J., Shi, Y.Y., Chen, Z., et al. "Improved SP-MCTS-based scheduling for multi-constraint hybrid flow Shop", *Applied Sciences*, **10**(18), pp. 6220 (2020).

19. Tian, G.D., Zhang, H.H., Feng, Y.X., et al. "Operation patterns analysis of automotive components remanufacturing industry development in China", *Journal of Cleaner Production*, **164**, pp. 1363-1375 (2017).
20. Chen, T.L., Cheng, C.Y. and Chou, Y.H. "Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming", *Annals of Operations Research*, **290**(1–2), pp. 813–836 (2020).
21. Liu, Z.F., Yan, J., Cheng, Q., et al. "The mixed production mode considering continuous and intermittent processing for an energy-efficient hybrid flow shop scheduling", *Journal of Cleaner Production*, **246**, pp. 119071 (2020).
22. Hasani, A. and Hosseini, S.M.H. "A bi-objective flexible flow shop scheduling problem with machine-dependent processing stages: Trade-off between production costs and energy consumption", *Applied Mathematics and Computation*, **386**, pp. 125533 (2020).
23. Li, M., Lei, D.M. and Cai, J.C. "Two-level imperialist competitive algorithm for energy-efficient hybrid flow shop scheduling problem with relative importance of objectives", *Swarm and Evolutionary Computation*, **49**, pp. 34-43 (2019).
24. Zheng, X., Zhou, S.C., Xu, R., et al. "Energy-efficient scheduling for multi-objective two-stage flow shop using a hybrid ant colony optimisation algorithm", *International Journal of Production Research*, **58**(12), pp. 4103-4120 (2019).
25. Schulz, S., Neufeld, J.S. and Buscher, U. "A multi-objective iterated local search algorithm for comprehensive energy-aware hybrid flow shop scheduling", *Journal of Cleaner Production*, **224**, pp. 421-434 (2019).
26. Meng, L.L., Zhang, C.Y., Shao, X.Y., et al. "More MILP models for hybrid flow shop scheduling problem and its extended problems", *International Journal of Production Research*, **58**(13), pp. 3905–3930 (2020).
27. Tian, G.D., Zhou, M.C. and Li, P.G. "Disassembly sequence planning considering fuzzy component quality and varying operational cost", *IEEE Transactions on Automation Science and Engineering*, **15**(2), pp. 748-760 (2018).
28. Tian, G.D., Ren, Y.P., Feng, Y.X., et al. "Modeling and planning for dual-objective selective disassembly using AND/OR graph and discrete artificial bee colony", *IEEE Transactions on Industrial Informatics*, **15**(4), pp. 2456-2468 (2019).
29. Wang, W.J., Tian, G.D., Chen, M.N., et al. "Dual-objective program and improved artificial bee colony for the optimization of energy-conscious milling parameters subject to multiple constraints", *Journal of Cleaner Production*, **245**, pp. 118714 (2020).
30. Ding, J.W., Schulz, S., Shen, L.J., et al. "Energy aware scheduling in flexible flow shops with hybrid particle swarm optimization", *Computers & Operations Research*, **125**, pp. 105088 (2021).
31. Dai, M., Tang, D.B., Zheng, K, et al. "An improved genetic-simulated annealing algorithm based on a hormone modulation mechanism for a flexible flow-shop scheduling problem", *Advances in Mechanical Engineering*, pp. 124903 (2013).
32. Rao, R.V., Savsani, V.J. and Vakharia, D.P. "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems", *Computer-Aided Design*, **43**(3), pp. 303-315 (2010).
33. Zhou, Y.K. and Zheng, S.Q. "Climate adaptive optimal design of an aerogel glazing system with the integration of a heuristicteaching-learning-based algorithm in machine learning-based optimization", *Renewable Energy*, **153**, pp. 375-391 (2020).

34. Garcia, J.A.M. and Mena, A.J.G. "Optimal distributed generation location and size using a modified teaching-learning based optimization algorithm", *International Journal of Electrical Power & Energy Systems*, **50**, pp. 65-75 (2013).
35. Sahu, B.K., Pati, S., Mohanty, P.K., et al. "Teaching-learning based optimization based fuzzy-PID controller for automatic generation control of multi-area power system", *Applied Soft Computing*, **27**, pp. 240-249 (2015).
36. Lei, D.M., Su, B. and Li, M. "Cooperated teaching-learning-based optimisation for distributed two-stage assembly flow shopscheduling", International Journal of Production Research, doi: 10.1080/00207543.2020.1836422 (2020).
37. Shao, W.S., Pi, D.C. and Shao, Z.S. "An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem", *Applied Soft Computing*, **61**, pp. 193-210 (2017).
38. Zhang, M., Wu, K.H. and Hu, Y.M. "Improved teaching-learning-based optimization algorithm for solving job shop scheduling problem", *Control and Decision*, **32**(2), pp. 349-57 (2017).
39. Boudjelida, A. "On the robustness of joint production and maintenance scheduling in presence of uncertainties", *Journal of Intelligent Manufacturing*, **30**(4), pp. 1515-1530 (2019).
40. Lin, W.W., Tian, G.D., Li, Z.W., et al. "Flow shop scheduling with low carbon emission and variable machining parameters", *Proceedings of The Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture*, **233**(5), pp. 1561-1572 (2019).
41. Enayatifar, R., Yousefi, M., Abdullah, A.H., et al. "MOICA: A novel multi-objective approach based on imperialist competitive algorithm", *Applied Mathematics and Computation*, **219**(17), pp. 8829-8841 (2013).
42. Deb, K., Pratap, A., Agarwal, S., et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, **6**(2), pp. 182–197 (2002).
43. Li, D.S., Zhang, C.Y., Tian, G.D., et al. "Multiobjective program and hybrid imperialist competitive algorithm for the mixed-model two-sided assembly lines subject to multiple constraints", *IEEE Transactions on Systems Man Cybernetics-Systems*, **48**(1), pp. 119-129 (2018).
44. Li, D.S., Zhang, C.Y., Shao, X.Y., et al. "A multi-objective TLBO algorithm for balancing two-sided assembly line with multiple constraints", *Journal of Intelligent Manufacturing*, **27**(4), pp. 725-739 (2016).
45. Wang, W.J., Tian, G.D., Yuan, G., et al. "Energy-time tradeoffs for remanufacturing system scheduling using an invasive weed optimization algorithm", *Journal of Intelligent Manufacturing*, doi: 10.1007/s10845-021-01837-5 (2021).
46. Lei, D.M., Gao, L. and Zheng, Y. L. "A novel teaching-learning-based optimization algorithm for energy-efficient scheduling in hybrid flow shop", *IEEE Transactions on Engineering Management*, **65**(2), pp. 330-340 (2018).
47. Zhang, Q. F. and Li, H. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition", *IEEE Transactions on Evolutionary Computation*, **11**(6), pp. 712-731 (2007).
48. Li, X.Y., Lu, C., Gao, L., et al. "An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop", *IEEE Transactions on Industrial Informatics*, **14**(12), pp. 5400-5409 (2018).

49. Auger, A., Bader, J., Brockhoff, D., et al. "Articulating user preferences in many-objective problems by sampling the weighted hypervolume", in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, ACM, pp. 555-562 (2009).
50. Wang, W.J., Tian, G.D., Zhang, T.Z., et al. "Scheme selection of design for disassembly (DFD) based on sustainability: A novel hybrid of interval 2-tuple linguistic intuitionistic fuzzy numbers and regret theory", *Journal of Cleaner Production*, **281**, pp. 124724 (2021).
51. Tian, G.D., Hao, N.N., Zhou, M.C., et al. "Fuzzy grey choquet integral for evaluation of multicriteria decision making problems with interactive and qualitative indices", *IEEE Transactions on System Man Cybernetics-Systems*, **51**(3), pp. 1855-1868 (2021).

List of Figures

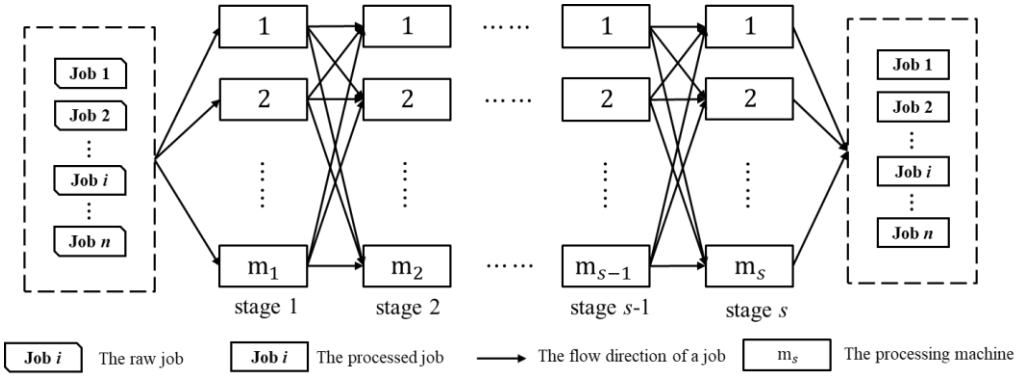

Fig 1. The diagrammatic sketch of the studied MLHFSP

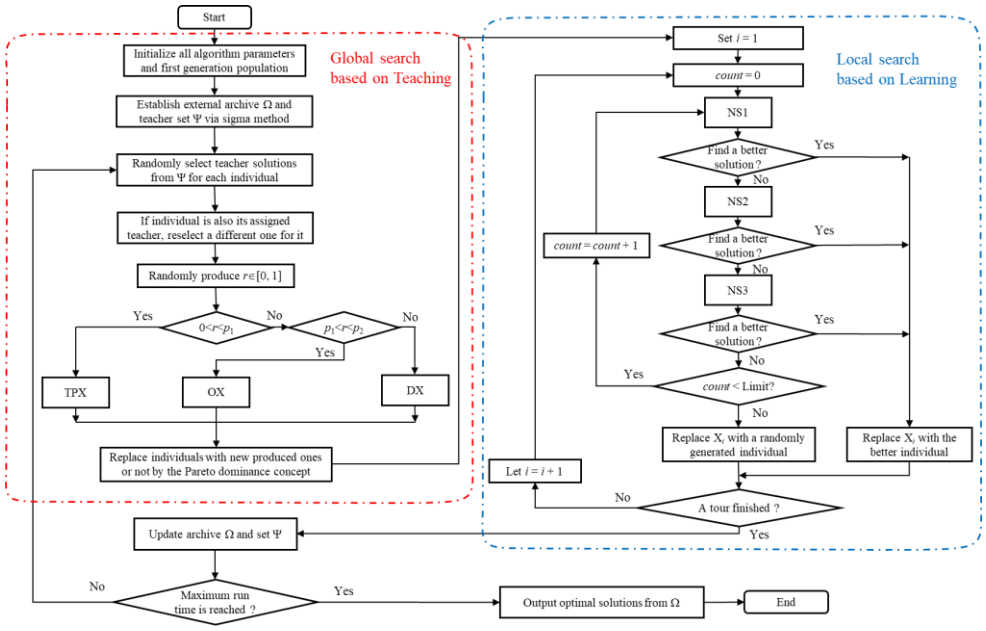

Fig 2. Flowchart of ITLBO

JV $\left[\pi_1, \pi_2, \ldots, \pi_n\right]$

MV $\left[z_{11}, z_{21}, \ldots, z_{n1}, \ldots, z_{1s}, \ldots, z_{ns}\right]$

Fig 3. Two-vector encoding

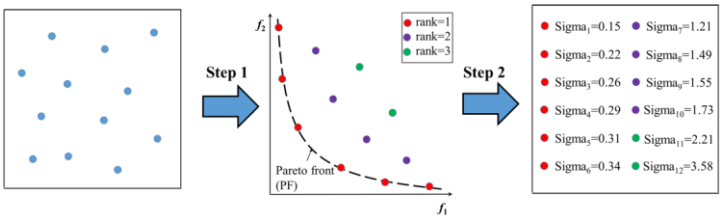

Fig 4. The procedure of sigma method

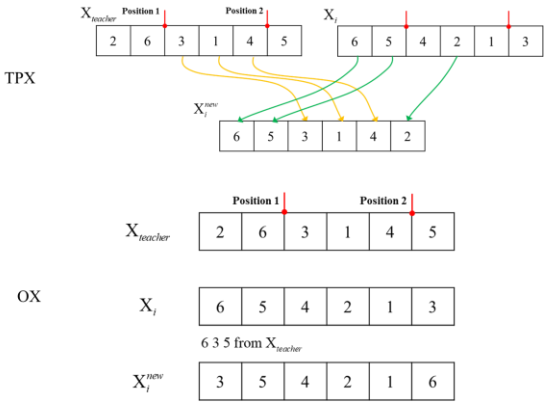

Fig 5. Two crossover operators for JV

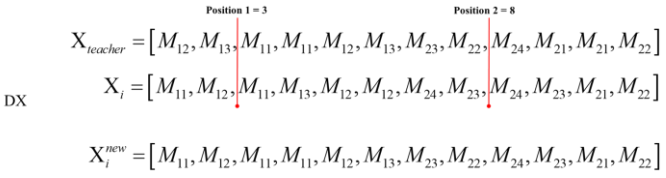

Fig 6. The crossover operator for MV

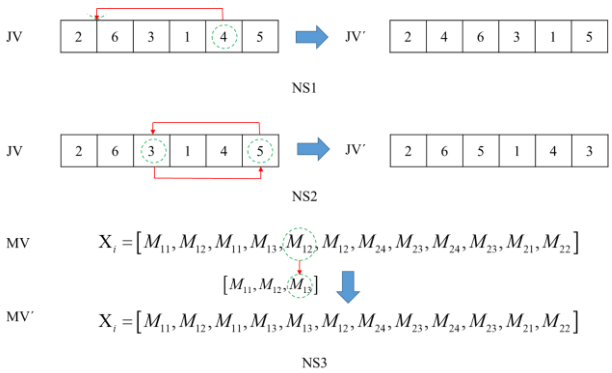

Fig 7. An example of three neighbourhood structures NS1, NS2, and NS3

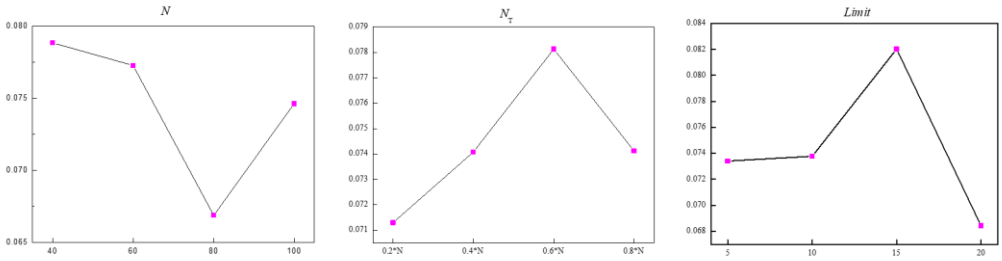

Fig 8. The trends of parameter with different levels in ITLBO

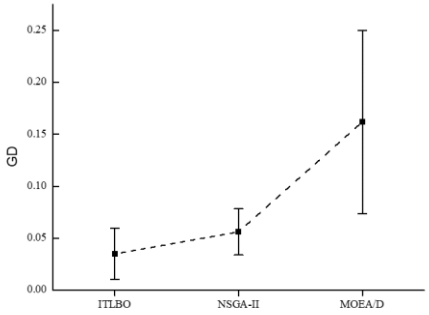

Fig 9. Mean plot and error bar of GD results

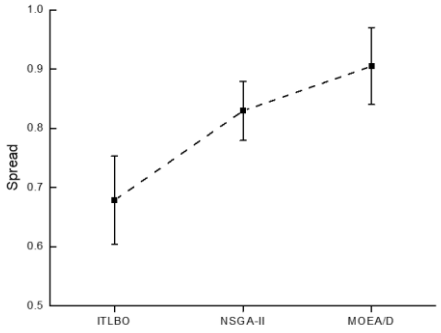

Fig 10. Mean plot and error bar of Spread results

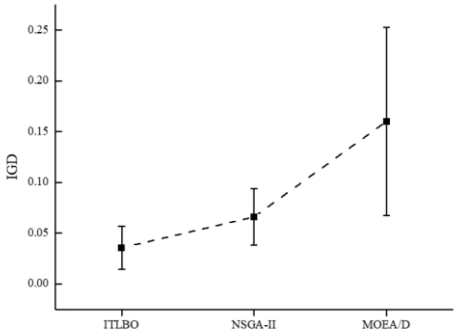

Fig 11. Mean plot and error bar of IGD results

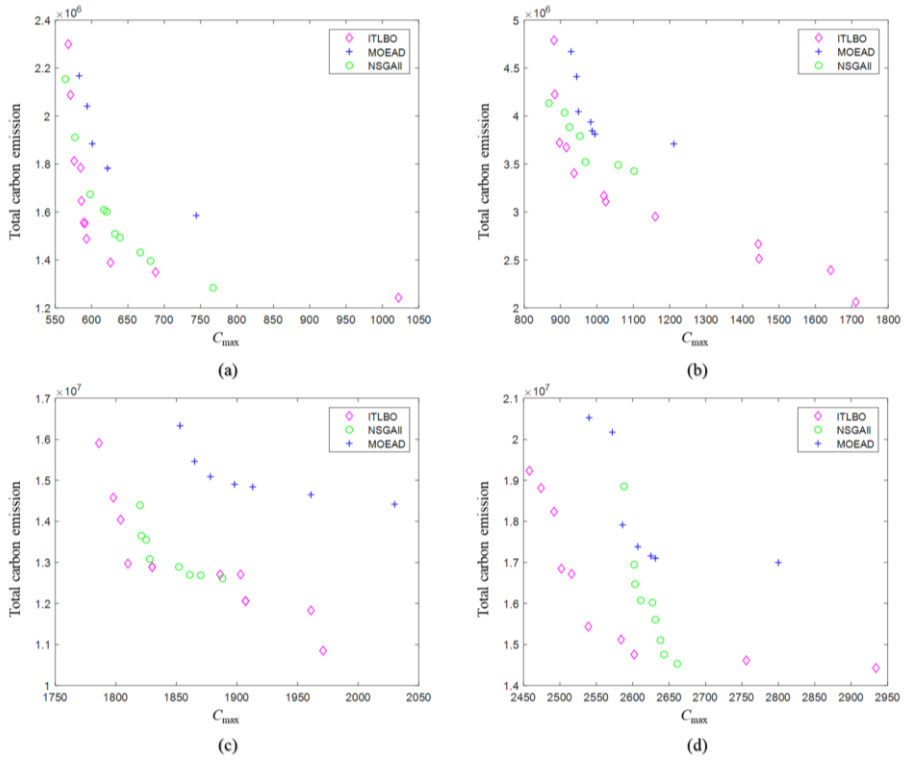

Fig 12. Pareto fronts obtained by ITLBO and its peers in different problem configuration: (a) 20×3, (b) 30×5, (c) 60×8, and (d) 100×5.

List of Tables

Table 1. The relevant published papers on HFSP

Refs	Objective	Methodology	Characteristics
[15]	Makespan and total tardiness	Evolutionary algorithm	Deterioration jobs
[16]	Makespan	Artificial bee colony algorithm	Distributed hybrid flow shops
[17]	Makespan	Monte-Carlo tree search	Large-scale problems
[18]	Makespan and electric power consumption	Genetic algorithm	Lot streaming
[19]	Makespan and energy consumption	Genetic algorithm	Forging operation
[20]	Production costs and energy consumption	NSGA-II	Machine-dependent processing stages
[21]	Total tardiness, makespan and total energy consumption	Imperialist competitive algorithm	Different objective importance
[22]	Total energy costs and makespan	Ant colony optimisation algorithm	Limited processing stages
[23]	Makespan, total energy costs and peak load	Iterated local search	Varying energy prices
[26]	Total tardiness and electric power costs	Particle swarm optimization algorithm	Varied processing speeds and electricity prices.
[27]	Total energy consumption and makespan	Genetic algorithm and annealing algorithm	Varied processing speeds

Table 2. Input control parameters and their levels

Parameters	Level			
	1	2	3	4
N	40	60	80	100
N_T	0.2* N	0.4* N	0.6* N	0.8* N
Limit	5	10	15	20

Table 3. L_{16} orthogonal array and computational results

Combination number	Parameters			ARV
	N	N_T	$Limit$	
1	1(40)	1(0.2* N)	1(5)	0.0745
2	1(40)	2(0.4* N)	2(10)	0.0826
3	1(40)	3(0.6* N)	3(15)	0.0829
4	1(40)	4(0.8* N)	4(20)	0.0754
5	2(60)	1(0.2* N)	2(10)	0.0685
6	2(60)	2(0.4* N)	1(5)	0.0765
7	2(60)	3(0.6* N)	4(20)	0.0842
8	2(60)	4(0.8* N)	3(15)	0.0799
9	3(80)	1(0.2* N)	3(15)	0.0760
10	3(80)	2(0.4* N)	4(20)	0.0479
11	3(80)	3(0.6* N)	1(5)	0.0725
12	3(80)	4(0.8* N)	2(10)	0.0711
13	4(100)	1(0.2* N)	4(20)	0.0662
14	4(100)	2(0.4* N)	3(15)	0.0893
15	4(100)	3(0.6* N)	2(10)	0.0729
16	4(100)	4(0.8* N)	1(5)	0.0701

Table 4. Response of ARV for ITLBO

Level	N	N_T	$Limit$
1	0.07885	0.07130	0.07340
2	0.07728	0.07408	0.07378
3	0.06687	0.07813	0.08203
4	0.07462	0.07412	0.06843
Delta	0.0120	0.0068	0.0136
Rank	2	3	1

Table 5 GD, Δ and IGD comparison values of ITLBO and its peers

Problems	ITLBO			NSGA-II			MOEA/D		
	GD	Δ	IGD	GD	Δ	IGD	GD	Δ	IGD
20×3	0.0862	**0.7341**	**0.0660**	**0.0805**	0.7886	0.0789	0.1589	0.8247	0.1098
20×5	**0.0745**	**0.7217**	**0.0615**	0.0823	0.8185	0.0831	0.2014	0.9265	0.1237

20×8	**0.0721**	**0.7020**	**0.0554**	0.0834	0.8547	0.0821	0.2139	0.9785	0.1823
20×10	**0.0649**	**0.7823**	**0.0722**	0.0811	0.7963	0.0725	0.1655	0.8962	0.1560
30×3	**0.0585**	**0.7288**	**0.0622**	0.0765	0.9218	0.1131	0.2478	0.9155	0.2300
30×5	**0.0691**	**0.8055**	**0.0702**	0.0782	0.9370	0.1237	0.1850	0.8740	0.1545
30×8	**0.0622**	**0.7851**	**0.0658**	0.0722	0.8416	0.0945	0.2032	0.9023	0.1721
30×10	**0.0427**	**0.7024**	**0.0531**	0.0752	0.8722	0.1043	0.1922	0.9341	0.1623
40×3	**0.0119**	**0.5688**	**0.0129**	0.0538	0.7589	0.0712	0.2548	0.9229	0.2856
40×5	**0.0284**	**0.6623**	**0.0290**	0.0594	0.7852	0.0735	0.2766	0.9647	0.3440
40×8	**0.0223**	**0.6245**	**0.0241**	0.0645	0.8364	0.0748	0.3152	0.9886	0.3641
40×10	**0.0098**	**0.5649**	**0.0116**	0.0687	0.8629	0.0825	0.3240	0.9702	0.3014
60×3	**0.0216**	**0.6756**	**0.0301**	0.0289	0.7388	0.0322	0.1847	0.8612	0.1913
60×5	**0.0122**	**0.6232**	**0.0146**	0.0408	0.8741	0.0437	0.2068	0.8755	0.2198
60×8	**0.0105**	**0.5796**	**0.0127**	0.0391	0.7604	0.0500	0.0954	0.7397	0.0980
60×10	**0.0302**	**0.7088**	**0.0318**	0.0411	0.8729	0.0458	0.1023	0.7546	0.1344
80×3	**0.0116**	**0.5911**	**0.0119**	0.0236	0.8341	0.0331	0.0346	0.9457	0.0402
80×5	0.0124	0.6582	0.0136	**0.0086**	**0.7864**	**0.0132**	0.0285	0.9612	0.0424
80×8	**0.0230**	**0.8057**	**0.0314**	0.0248	0.8707	0.0302	0.0412	0.9828	0.0533
80×10	**0.0145**	**0.6843**	**0.0257**	0.0244	0.8554	0.0317	0.0321	0.9214	0.0388
100×3	**0.0342**	**0.7117**	**0.0355**	0.0628	0.8043	0.0649	0.0633	0.8523	0.0754
100×5	**0.0181**	**0.5778**	**0.0153**	0.0554	0.7872	0.0598	0.0895	0.8928	0.0894
100×8	**0.0234**	**0.6488**	**0.0243**	0.0625	0.8310	0.0670	0.1255	0.9012	0.1324
100×10	**0.0228**	**0.6376**	**0.0252**	0.0596	0.8144	0.0622	0.1426	0.9342	0.1501

Table 6 HV comparison values of ITLBO and its peers

Problems	ITLBO	NSGA-II	MOEA/D	Problems	ITLBO	NSGA-II	MOEA/D
20×3	**1.2620**	1.2466	0.4563	60×3	**0.8456**	0.5165	0.2150

20×5	**1.2723**	1.2210	0.6582	60×5	**1.3120**	1.1866	0.6379
20×8	**1.3066**	1.2632	0.4030	60×8	**1.0412**	0.8374	0.3129
20×10	**1.1009**	0.8354	0.5313	60×10	**0.9561**	0.8587	0.6325
30×3	**1.3566**	1.2211	0.9547	80×3	**1.1810**	0.9361	0.7630
30×5	**1.0059**	0.6083	0.4224	80×5	1.0389	**1.1238**	0.8966
30×8	**1.2136**	1.1252	0.8263	80×8	**0.7856**	0.5214	0.3659
30×10	**1.3189**	1.0397	0.7632	80×10	**1.2578**	1.1583	0.7239
40×3	**0.9845**	0.9365	0.8521	100×3	**0.8416**	0.6215	0.4322
40×5	**0.9523**	0.9033	0.8316	100×5	**1.2554**	1.0386	0.5951
40×8	**0.8963**	0.7825	0.7365	100×8	**1.1368**	0.9626	0.6542
40×10	**1.2645**	0.9984	0.8976	100×10	**0.8863**	0.7263	0.6215

Biographies

Wenjie Wang is currently working toward the Ph.D. degree in mechanical engineering at Shandong University, Jinan, China. His current research interests include intelligent optimization and green manufacturing.

Xuesheng Zhou is currently working toward the Ph.D. degree in mechanical engineering at Nanjing Agricultural University, Nanjing, China. His research interests include intelligent transportation, Petri nets and optimization schedule with applications.

Guangdong Tian is currently a Professor with the School of Mechanical Engineering at Shandong University, Jinan, China. His research focuses on remanufacturing and green manufacturing, green logistics and transportation, intelligent inspection and repair of automotive, decision-making, and intelligent optimization.

Amir Mohammad Fathollahi-Fard is currently a Research Associate at the École de Technologie Supérieure, University of Québec, Montreal, QC, Canada. His research is about supply chain management, sustainable operations management, transportation and logistics optimization, and health care management.

Peng Wu is currently a Professor with the School of Economics and Management at Fuzhou University, Fuzhou, China. His research interest includes optimization of complex systems, such as intelligent transportation systems and production systems.

Chaoyong Zhang is currently a Professor with the School of Mechanical Science and Engineering at Huazhong University of Science and Technology, Wuhan, China. His research mainly focuses on optimization and scheduling for production

manufacturing systems, sustainable manufacturing including clean and high efficient manufacturing processes.

Chengwen Liu is currently an Engineer at Ruixing Group Co., Ltd, Taian, China. His research mainly focuses on raw and auxiliary materials.

Zhiwu Li is currently a Professor with the Institute of Systems Engineering at Macau University of Science and Technology, Macau, China. His research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, system reconfiguration, game theory, and data and process mining.