

Research Article

Hybrid Estimation of Distribution Algorithm for Solving Three-Stage Multiobjective Integrated Scheduling Problem

Chao Deng ¹, Rong Hu ^{1,2}, Bin Qian,^{1,2} and Huai P. Jin²

¹Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, China

²Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

Correspondence should be addressed to Rong Hu; ronghu@vip.163.com

Received 8 February 2021; Accepted 26 April 2021; Published 1 June 2021

Academic Editor: Xue-bo Jin

Copyright © 2021 Chao Deng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at reducing the total energy consumption of three stages processing-transportation-assembly in the assembly manufacturing industry, a three-stage multiobjective integrated scheduling problem with job batch transportation considering the energy consumption (3sMISP_JBTEC) is proposed, and a comprehensive energy consumption model of multistage of 3sMISP_JBTEC with an improved turn off/on strategy in the processing stage and considering speed in the transportation stage is formulated. Then, a hybrid estimation of distribution algorithm with variable neighborhood search (HEDA_VNS) is developed to solve the scheduling problem. In the HEDA_VNS, the reasonable coding/decoding rules and speed scheduling scheme (SSS) are designed. Moreover, two local search strategies are designed to further enhance the performance of HEDA_VNS. Among them, three types of neighborhood search strategies are devised in Local Search I to improve the search efficiency while retaining the structure of the original high-quality solution. A variable neighborhood hybrid operation based on the speed scheduling set is designed in Local Search II to further improve the quality of the solution while balancing the optimization goals. Finally, simulations and comparisons show the efficiency of the proposed HEDA_VNS.

1. Introduction

As environmental pollution becomes a serious challenge for the world, the development of the manufacturing industry is gradually changing towards the green manufacturing. According to relevant surveys, the energy efficiency of the manufacturing industry is low and pollution emissions are high at moment. For instance, in China, the proportion of industrial GDP, 34.3%, is obtained by consuming 68% of the national energy in 2015 [1]. The production process plays an important role in the manufacturing industry. In recent years, in order to reduce energy consumption in the production process, many studies have been conducted on improving the energy efficiency [2–6]. However, among the methods, production scheduling is the most effective and prioritized way to improve the energy efficiency of the manufacturing industry [7].

The assembly manufacturing industry is an important sector of manufacturing enterprises. The production process

includes multiple stages of processing, transportation, and assembly. Energy consumption should not be ignored in the whole production process. The existing researches of assembly manufacturing industry have focused on the two-stage scheduling problem (2sSP) containing the processing-assembly and the three-stage scheduling problem (3sSP) containing the processing-transport-assembly. Compared with 2sSP, the transportation between the processing and assembly stages has a nonnegligible process in 3sSP. However, a large number of studies assumed that the transportation time of each job is constant in the transportation stage [8–11]; i.e., the number or load of transportation vehicles is unlimited, which is far from the actual problem. Therefore, Deng et al. [12] have integrated production and transportation, and the three-stage integration scheduling problem with job batch transportation (3sISP_JBT) of processing-transportation-assembly has been studied, which makes the problem model of the assembly manufacturing industry more realistic. However, the

study only considers minimizing the makespan as the optimization goal. Although a close connection among the three stages has been established through the optimization, it leads to production process not compact enough in the single stage. Especially in the processing stage, a large idleness may be generated between two adjacent jobs on the same machine in the process of processing, which causes energy waste seriously. Furthermore, in the transportation stage, the energy consumption is affected by the load and speed of the vehicle closely. In terms of computational complexity, 2sSP has been proven to be NP-hard [13]. It is reduced to 3sSP. Also, 3sSP can be reduced to 3sISP_JBT. Therefore, the considered problem is also NP-hard. To sum up, the comprehensive energy consumption optimization for 3sISP_JBT needs to be further studied.

For the energy-optimized multiobjective green scheduling problem of the production process, the existing research has largely focused on optimizing the machining process with methods of machine speed control, turn off/on machine strategy, etc. [14–18]. However, the transport of workpieces is ignored in the production process. Then, Lu et al. [19] investigated a permutation flow shop scheduling problem with sequence-dependent setup and controllable transportation time in the processing stage from a real-world manufacturing enterprise, where the objects include the makespan and energy consumption, and a hybrid multi-objective backtracking search algorithm (HMOBSA) has proposed to solve the problem. Dai et al. [20] studied a flexible job shop scheduling problem considering the transportation time in the processing stage, where the objects include makespan and the total energy consumption, and an improved genetic algorithm (GA) has been proposed to solve it. However, in the above literature, many scholars considered transportation in the energy optimization problem of workshop scheduling, but they mainly consider the impact of the transportation time on the entire makespan and do not study the energy consumption of transportation in the energy consumption index. After that, Tanimizu and Amano [21] studied the integrated scheduling problem of the processing-transportation stage, a processing-transportation comprehensive energy consumption model has been constructed to minimize the total delay time and the energy consumption, and a genetic algorithm (GA) based on heuristic rules has been adopted to solve this problem. Guo et al. [22] also studied the integrated scheduling problem of the processing-transportation stage, and a hybrid memetic algorithm has been proposed to minimize the total cost and energy consumption. Liu et al. [23] studied a novel integrated green scheduling problem of flexible job shop and crane transportation in the processing stage, a comprehensive energy consumption model has been built to optimize the comprehensive energy consumption and makespan, and an integrated algorithm of a genetic algorithm (GA) and glowworm swarm optimization (GSO) with green transport heuristic strategy has been proposed. It is noted that the above results have considered the transportation time and transportation process in the workshop scheduling optimization and have conducted energy consumption optimization research on this issue, while mainly

considering the impact of load factors (constraints) of vehicles on the transportation stage and ignoring the impact of transportation speed on energy consumption. However, the impact of speed on energy consumption cannot be ignored in practice in the actual transportation process. A large number of existing studies have verified the impact of transportation speed on the energy consumption and scheduling schemes on the green transportation scheduling problems considering speed [24–26]. Hence, for 3sISP_JBT of the assembly manufacturing industry, in addition to designing effective energy-saving strategies for the two production stages of processing and assembly, it is necessary to further consider the impact of transportation speed on the energy consumption in the transportation stage to improve the entire production process energy optimization.

At present, a large number of intelligent algorithms have emerged to solve scheduling problems [27–29]. The estimation of distribution algorithm (EDA) is an emerging evolutionary algorithm based on statistics. It learns high-quality individuals in the population in a statistical manner and constructs the probability model, and then the probability model is sampled to generate a new population. It guides the search direction of the algorithm and has a good global search capability. In recent years, EDA has been successfully applied to solve a variety of scheduling problems, including parallel machine scheduling problem [30], flow shop scheduling problem [31], job shop scheduling problem [32, 33], energy-saving job shop scheduling problem [34], vehicle transportation scheduling problem [35], and integrated scheduling problem [12]. Judging from the research status of EDA in scheduling optimization problems, EDA itself has the advantage of probabilistic model global search. In addition, it can be reasonably mixed with other effective algorithms and operations, which can greatly enrich the search ability and improve the algorithm performance.

This paper aims to optimize the three-stage multi-objective integration scheduling problem with job batch transportation considering the energy consumption (3sMISP_JBTEC) in the assembly manufacturing industry. In the processing stage, the machining environment is a heterogeneous parallel machine of multiprocessing with release time. In the transportation stage, the transportation facilities are the same type vehicles. In the assembly stage, only one machine is used for assembly. The focus of this paper is to explore the interrelationship between the three stages processing-transportation-assembly and the key factors affecting energy consumption at each stage to further optimize the comprehensive energy consumption of the assembly manufacturing industry. The contributions of this paper are as follows:

- (1) A mathematic model is formulated for the 3sMISP_JBTEC, in which the objective is to minimize the comprehensive energy consumption and makespan of the production task.
- (2) In the proposed model, a comprehensive energy consumption model with an improved turn off/on strategy in the processing stage and speed factor in

the transportation stage is built to optimize the comprehensive energy efficiency of the three stages, processing-transportation-assembly.

- (3) A hybrid estimation of distribution algorithm with variable neighborhood search (HEDA_VNS) is presented to solve the proposed model.
- (4) In HEDA_VNS, the reasonable coding/decoding rules and speed scheduling scheme are designed based on the specific characteristics. Then, two local search strategies are designed to further enhance the HEDA_VNS's performance. Among them, three types of neighborhood search strategies are devised in Local Search I to improve the search efficiency while retaining the structure of the original high-quality solution. A variable neighborhood hybrid operation based on the speed set is designed in Local Search II to further improve the quality of the solution while balancing the optimization goals.

The remainder of this paper is organized as follows. A mathematic model of the proposed 3sMISP_JBTEC is developed in Section 2. HEDA_VNS algorithm is designed for solving 3sMISP_JBTEC in Section 3. Section 4 details the HEDA_VNS. Simulation experiment results and analysis is conducted in Section 5. Section 6 presents the conclusions.

2. Mathematical Formulation

In this section, the introduced 3sMISP_JBTEC is described in detail based on the assembly manufacturing environment. Then, a mathematical model is proposed to formulate the 3sMISP_JBTEC with the consideration of comprehensive energy consumption and makespan.

2.1. Problem Description and Assumptions. The 3sMISP_JBTEC problem is described as follows. According to the customer order (demand), we set the number of processed products as H_p , the number of processing machines in the processing stage as m_1 , the number of transport vehicles in the transportation stage as H_{TR} , and the number of assembled machines in the assembly stage as one. Each product is composed of multiple jobs, and each job needs to be processed in different procedures. The procedures of the jobs need to be processed in sequence on the machine that meets the processing constraints at the processing stage. After the jobs are processed, they are loaded by the same type vehicle in the transportation stage. Transport at a certain speed v to the assembly stage to wait for assembly. The jobs of each product are processed in three stages in the order of processing. Then, the model diagram is shown in Figure 1.

Without loss of generality, the 3sMISP_JBTEC is investigated based on the following reasonable assumptions:

- (1) Any processing machine in the processing stage can only process one job at the same time, and different jobs have release time; a process of job is immediately transferred to the next process after being processed, and the transfer time of job between the required machine is ignored; the preparation time of all

processes of job is not related to the processing sequence and is included in the processing time of the process. Different processes of the same job cannot be processed at the same time; each process of each job cannot be interrupted once it starts processing.

- (2) The assembly machine in the assembly stage can only assemble the same product at the same time, and the setup time between different products is set to zero; the initial state of the assembly machine is idle, and at zero time, any task is feasible.
- (3) If the same raw material is processed through the same process in the processing stage, it is the same job, and one or more jobs that make up different products are not the same.

2.2. Notation. The main notations used in formulating the mathematical model of the 3sMISP_JBTEC problem are listed as follows:

P : product set. $z \in P, P = \{1, 2, \dots, H_p\}$.

Q : jobs set. $j \in Q, Q = \{1, 2, \dots, q\}$.

O : process set. $i \in O, O = \{1, 2, \dots, H_O\}$.

T : vehicle set. $t \in T, T = \{1, 2, \dots, H_{TR}\}$.

M : machine set in processing stage.
 $k \in M, M = \{1, 2, \dots, m_1\}$.

H_k : the number of processes on the processing machine k .

w_t : the number of round trips of vehicle t . $w_t \geq 1$.

H_T : the total number of transport vehicles. $H_T \neq \infty$.

V_T : the vehicle rated load (fixed value).

Dis_{p-A} : the distance from the processing stage to the transportation stage.

S : the vehicle speed set.

W_{idle}^T : the unload weight of the vehicle.

SV_{use} : the speed use set. $SV_{use} = \{v_1, \dots, v_t, \dots, v_{H_T}\}$.

$\pi^K = [\pi_{[1]}^K, \pi_{[2]}^K, \dots, \pi_{[H_p]}^K]$: the permutation of the processes of all jobs in the processing stage. The jobs in the permutation are allocated to the corresponding machines for processing from left to right according to rules and parallel processing constraints.

$\pi^k = [\pi_{[1]}^k, \pi_{[2]}^k, \dots, \pi_{[i]}^k, \dots, \pi_{[H_k]}^k]$: the permutation of the process of the job on the processing machine k in the processing stage, where $\pi_{[i]}^k$ is the process of the job at the position i of π^k .

$\pi^{TR} = [\pi_{[1]}^{TR}, \pi_{[2]}^{TR}, \dots, \pi_{[j]}^{TR}, \dots, \pi_{[q]}^{TR}]$: the permutation of all the jobs in the transportation stage, where $\pi_{[j]}^{TR}$ is the job at the position j of π^{TR} .

$\pi^{TR-V} = [\pi_{[1]}^{TR-V}, \pi_{[2]}^{TR-V}, \dots, \pi_{[t]}^{TR-V}, \dots, \pi_{[H_T]}^{TR-V}]$: the speed permutation of all vehicles in the transportation phase, where $\pi_{[t]}^{TR-V}$ is the vehicle at the position t of π^{TR-V} .

$\pi^P = [\pi_{[1]}^P, \pi_{[2]}^P, \dots, \pi_{[z]}^P, \dots, \pi_{[H_p]}^P]$: the permutation of the products to be processed on the assembly machine

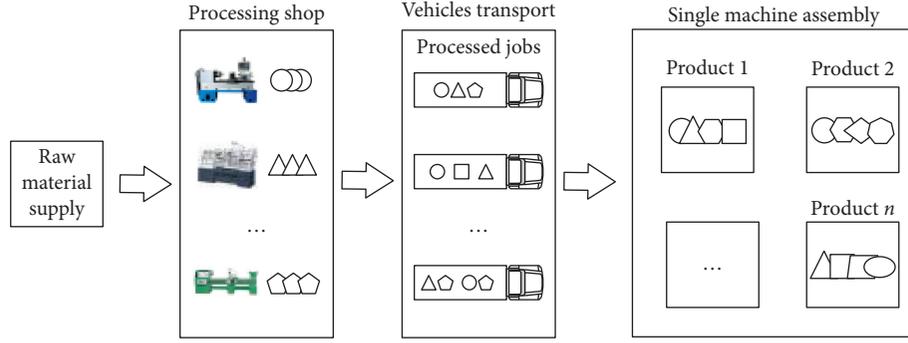


FIGURE 1: The problem model of 3sMISP_JBTEC.

in the assembly stage, where $\pi_{[z]}^P$ is the product at the position z of π^P .

$P(\pi_{[i]}^k)$: the processing time of $\pi_{[i]}^k$.

$R(\pi_{[i]}^k)$: the release time when the job at the position i of π^k first arrived at the processing machine k .

$P(\pi_{[z]}^P)$: the assembly time of $\pi_{[z]}^P$.

$V(\pi_{[j]}^{\text{TR}})$: the weight of the job at position j of π^{TR} .

$v(\pi_{[t]}^{\text{TR}-V})$: the speed at of the vehicle at position t of $\pi^{\text{TR}-V}$.

T_{on} : the time required to turn on a machine.

T_{off} : the time required to turn off a machine.

$T_k^{\text{Num-off}}$: the number of turning off the machine k .

e_k^{idle} : the energy consumption of idle time of machine k .

e_k^R : the energy consumption of release time of machine k .

e_k^P : the average energy consumption of processing time of machine k .

e_{tv}^T : the energy consumption of vehicle t at a speed v .

$E_{\text{off-on}}$: the energy consumption requirement for turn off, then turn on the machine.

E_T^{idle} : the energy consumption of vehicle t return one time without load.

$\text{pre}_m(\pi_{[i]}^k)$: the machine number of the previous processing operation of $\pi_{[i]}^k$; the permutation of the previous process of $\pi_{[i]}^k$ with the machine number $\text{pre}_m(\pi_{[i]}^k)$ expressed as $\pi_{[i]}^{\text{pre}_m(\pi_{[i]}^k)}$. If the job $\pi_{[i]}^k$ is processed for the first time, then $\text{pre}_m(\pi_{[i]}^k) = 0$.

$\text{pre}_k(\pi_{[i]}^k)$: the process number of the previous processed job when $\pi_{[i]}^k$ processing on the machine k . The position of the process in the permutation is recorded as $\pi_{\text{pre}_k(\pi_{[i]}^k)}^k$. If the machine k is processing the job for the first time, then $\text{pre}_k(\pi_{[i]}^k) = 0$.

$C(\pi_{[i]}^k)$: the processing complete time of $\pi_{[i]}^k$.

$C_{\text{st1}}(\pi_{[j]}^{\text{TR}})$: the processing completion time of the job j of π^{TR} .

$S(\pi_{[j]}^{\text{TR}})$: the transportation starting time of the job j of π^{TR} in vehicle t .

$C(\pi_{[j]}^{\text{TR}})_{w_t}$: the transportation completion time of the job j of π^{TR} in w_t trips of vehicle t .

$C_{\text{st2}}(\pi_{[j]}^{\text{TR}})$: the transportation completion time of the job j of π^{TR} .

$S(\pi_{[z]}^P)$: the assembly start time of $\pi_{[z]}^P$.

$C_{\text{st3}}(\pi_{[z]}^P)$: the assembly completion time of $\pi_{[z]}^P$.

2.3. Comprehensive Energy Consumption Model. In the section, a comprehensive energy consumption model of 3sMISP_JBTEC is given in the three stages, respectively. In the processing stage, the energy consumption includes the release process, the processing process, and the idle process. Besides, an improved turn off/on strategy is proposed to better save energy and extend the life of the machine. In the transportation stage, the impact factor of speed and load is both considered to calculate energy consumption during transportation.

2.3.1. Energy Consumption of the Processing Stage (EC_{st1}). The processing stage is a multiprocess heterogeneous parallel machine scheduling problem with release time. The processing workshop is a flexible workshop, and multiple processes of the jobs can be processed on different machines. It is flexible and versatile, but it is also one of the industrial workshops with the highest energy consumption [36]. Figure 2 is the energy consumption of the flexible machine k in the processing stage of the problem. It can be seen from Figure 2 that the energy consumption of this processing stage (EC_{st1}) is composed of the energy consumption of the release process (EC_R^{st1}) when the jobs arrives at the machine for the first time, the energy consumption of the processing process (EC_P^{st1}) of the different jobs in machines, and the energy consumption of the idle process ($EC_{\text{idle}}^{\text{st1}}$). If the energy consumption caused by job transformation between different machines is ignored, then the total energy consumption of the processing stage (EC_{st1}) is calculated by using equation (1). In order to simplify the calculation, this paper simplified the unit energy consumption of each part to the average value:

$$EC_{\text{st1}} = EC_R^{\text{st1}} + EC_P^{\text{st1}} + EC_{\text{idle}}^{\text{st1}}. \quad (1)$$

(1) **Energy Consumption of Release Process (EC_R^{st1}).** The energy consumption of the release process depends on the release time of jobs. Because there are differences between the jobs (i.e., size, shape, material, and weight) and

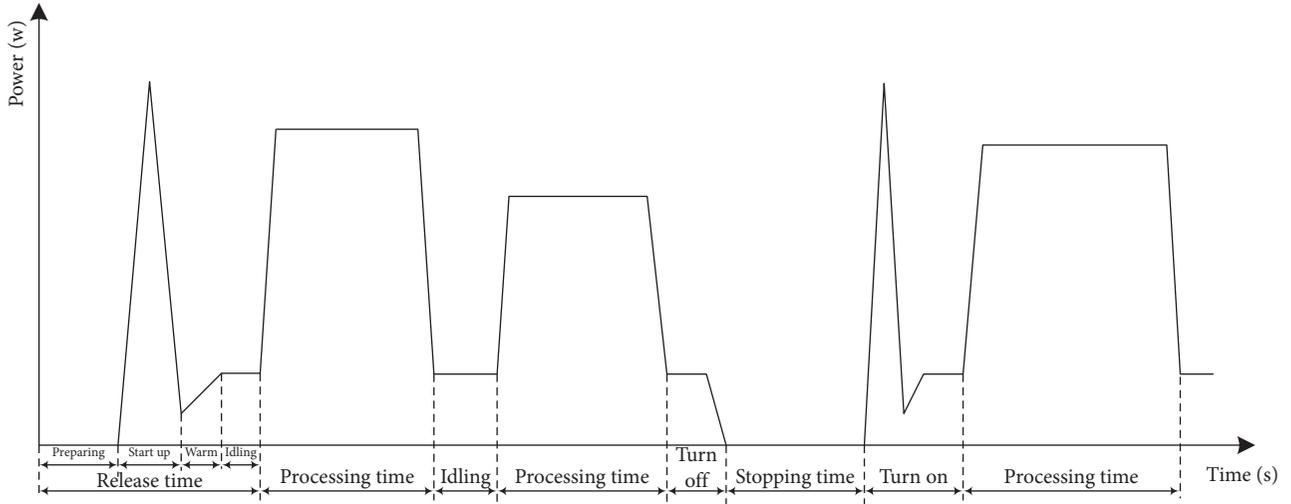


FIGURE 2: Example of energy consumption of a processing machine.

machines, it directly affects EC_R^{st1} . Thus, the energy consumption of the machine's release process is calculated by the following equation:

$$EC_R^{st1} = \sum_k e_k^R \cdot R(\pi_{[i]}^k), \quad i = 1, \dots, H_k; k = 1, \dots, m_1. \quad (2)$$

(2) Energy Consumption of the Processing Process (EC_P^{st1}).

The energy consumption of the process depends on the processing time of the corresponding process of the job on the machine. The energy consumption of all machines in the processing process is calculated as

$$EC_P^{st1} = \sum_i \sum_k e_k^P \cdot P(\pi_{[i]}^k), \quad i = 1, \dots, H_k; k = 1, \dots, m_1. \quad (3)$$

(3) *Energy Consumption of the Idle Process (EC_{idle}^{st1}) and an Improved Turn Off/On Strategy.* The energy consumption of the process depends on the idle time of the machine. The idle time i on the machine k can be expressed as $P_{k,i}^{idle} = S(\pi_{[i+1]}^k) - C(\pi_{[i]}^k)$, where $P_{k,i}^{idle} > 0$. If the machine's idle time is too long, it will cause a waste of energy. Therefore, the machine needs to be turned off to save energy at the appropriate time. The work in [37] gives the decision time period T_B for whether or not to shut down the machine, where $T_B = \max\{E_{off-on}/e_k^{idle}, T_{on} + T_{off}\}$. If $P_k^{idle} \geq T_B$, the machine should be turned off. Although the power-off method can reduce energy consumption, it frequently may

shorten the service life of the machine. Therefore, the work in [38] gives the maximum allowable number of machine turn off, where $T_{Num-off}$. Then, in order to further save energy and prolong the life of the machine, an energy-saving strategy for turn on/off of machine was proposed in [19]. However, the closeness of adjacent idles in the same machine is not considered. If the adjacent idles are too close and $P_k^{idle} \geq T_B$, turning the machine on and off frequently would still affect the life of the machine in the actual production process. Therefore, this paper designs an improved energy-saving strategy based on the above studies. The steps are as follows:

Step 1: calculate P_k^{idle} of the machine k , then record the time $T_{k,i}^{dis}$ between two adjacent idles, $T_{k,n}^{dis} = SP_{k,i+1}^{idle} - CP_{k,i}^{idle}$, where $T_{k,i}^{dis} > 0$.

Step 2: compare with $P_{k,i}^{idle}$ and T_B in turn. If $P_{k,i}^{idle} \geq T_B$ and $T_{k,i}^{dis} \leq \varepsilon$ (where ε is a certain value), then put $\max\{P_{k,i}^{idle}, P_{k,i+1}^{idle}\}$ into set Ω and put $\min\{P_{k,i}^{idle}, P_{k,i+1}^{idle}\}$ in the taboo table to avoid repeated comparisons, else put $P_{k,i}^{idle}$ (where $P_{k,i}^{idle} \geq T_B$) into the set Ω directly.

Step 3: sort P_k^{idle} which in the set Ω in nonascending order. Then, select the first number $T_{Num-off}$ of P_k^{idle} in the sequence and put them in the set Ω' which is recorded as $P_k^{Stop-idle}$.

From the above, the energy consumption of the idle process is calculated as

$$\begin{aligned} EC_{idle}^{st1} &= \sum_i \sum_k e_k^{idle} \cdot P_{k,i}^{idle} - \sum_i \sum_k e_k^{idle} \cdot P_{k,i}^{Stop-idle} + \sum_i \sum_k E_{off-on} \cdot T_{Num-off} \\ &= \sum_i \sum_k e_k^{idle} \cdot (S(\pi_{[i+1]}^k) - C(\pi_{[i]}^k)) - \sum_i \sum_k e_k^{idle} \cdot P_{k,i}^{Stop-idle} \\ &\quad + \sum_i \sum_k E_{off-on} \cdot T_{Num-off}, \quad i = 1, \dots, H_k - 1; k = 1, \dots, m_1; P_k^{Stop-idle} \in \Omega'. \end{aligned} \quad (4)$$

2.3.2. Energy Consumption of Transportation Stage (EC_{st2}).

In the transportation stage, the jobs completed in the processing stage will be transported to the assembly place for assembly by multiple vehicles of the same type under certain constraints of load and speed. The constant Dis_{p-A} is the distance from the processing stage to the assembly stage. The vehicle can go back and forth multiple times, and thus, the vehicle is empty during the return process. The speed of each vehicle without load is a constant value v_c , and the energy consumption is a constant value E_T^{idle} . It should be emphasized that once all vehicles begin to transport, the constant speed of vehicles should be maintained throughout the transportation process.

In this line, some studies have been conducted on the energy consumption of vehicle transportation. Palmer [39] incorporated the energy consumption indicators into the vehicle transportation problem and proposed an energy consumption model that considers the total length or

distance of the path. Kara et al. [40] proposed an energy consumption model considering the load of vehicle and distance. Kuo [24] further considered the vehicle speed and proposed a time-dependent speed and vehicle load energy consumption model. Demir et al. [25] developed a mathematical model of energy consumption of multiple parameters with speed, distance, and load. In the equation, P_{tract} (kilowatts) is the unit traction power of the vehicle, τ is the acceleration, $M = w + f$ (w is the mass of the vehicle and f is the mass of the cargo carried by the vehicle), and the rest is the vehicle-related parameter. In this paper, the uniform speed of the vehicles is considered, where $\tau = 0$, but the influence of the gradient of the vehicle is not considered. Then, the per unit time of the vehicle t is shown in equation (6). Therefore, the energy consumption of the transportation stage can be given as shown in equation (7), where the unit of speed is m/s:

$$P_{tract} = \frac{(M\tau + Mg \sin \theta + 0.5C_d\rho Av^2 + MgC_r \cos \theta)v}{1000}, \quad (5)$$

$$P_t^{tract} = \frac{(0.5C_d\rho Av^2 + MgC_r)v}{1000}, \quad (6)$$

$$\begin{aligned} EC_{st2} &= \sum_t \frac{P_t^{tract} \cdot Dis_{p-A}}{v(\pi_{[t]}^{TR-V})} + \sum_t \sum_w E_T^{idle} \cdot w_t \\ &= \sum_t \frac{(0.5C_d\rho A \cdot v(\pi_{[t]}^{TR-V})^2) + (W_{idle}^T + \sum_j V(\pi_{[j]}^{TR})gC_r) \cdot Dis_{p-A}}{1000} \\ &\quad + \sum_t \sum_w \frac{(0.5C_d\rho Av_c^2 + W_{idle}^T gC_r) \cdot Dis_{p-A} \cdot w_t}{1000}, \quad j = 1, \dots, q; t = 1, \dots, H_T; v \in S; w = 1, 2, \dots \end{aligned} \quad (7)$$

2.3.3. Energy Consumption of Assembly Stage (EC_{st3}).

The assembly stage is a single machine scheduling problem. Since the assembly time of each product on the machine is unchanged, it will not affect the processing energy consumption so that the energy consumption is only related to the machine's idle time. Therefore, energy consumption of the assembly process is calculated as

$$EC_{st3} = \sum_z e_{m_2}^{idle} \cdot (S(\pi_{[z+1]}^P) - C(\pi_{[z]}^P)), \quad z = 1, 2, \dots, H_p. \quad (8)$$

In summary, the total energy consumption of 3sMISP_JBTEC is calculated as

$$EC_{total} = EC_{st1} + EC_{st2} + EC_{st3}. \quad (9)$$

2.4. Formulation of Multiobjective Model of 3sMISP_JBTEC.

The 3sMISP_JBTEC is formulated as the following mathematical model, in which two objectives of makespan and total energy consumption are considered:

- (1) The processing stage: it is a multiprocess heterogeneous parallel machine scheduling problem with release time in the processing stage. If the job $\pi_{[i]}^k$ is processed for the first time, then $pre_m(\pi_{[i]}^k) = 0$; If the machine that processed job $\pi_{[i]}^k$ is the first time to process the job, then $pre_k(\pi_{[i]}^k) = 0$. The completion time $C(\pi_{[i]}^k)$ of $\pi_{[i]}^k$ is calculated as follows:

$$\begin{aligned}
C(\pi_{[i]}^k) &= \min\{P(\pi_{[i]}^k) + R(\pi_{[i]}^k)\}, \text{pre-}m(\pi_{[i]}^k) = 0; \text{pre-}k(\pi_{[i]}^k) = 0, \\
C(\pi_{[i]}^k) &= \min\left\{\max\left\{C\left(\pi_{[\text{pre-}k(\pi_{[i]}^k)]}^k\right), R(\pi_{[i]}^k)\right\} + P(\pi_{[i]}^k)\right\}, \text{pre-}m(\pi_{[i]}^k) = 0; \text{pre-}k(\pi_{[i]}^k) \neq 0, \\
C(\pi_{[i]}^k) &= \min\left\{C\left(\pi_{[i]}^{\text{pre-}m(\pi_{[i]}^k)}\right) + P(\pi_{[i]}^k)\right\}, \text{pre-}m(\pi_{[i]}^k) \neq 0; \text{pre-}k(\pi_{[i]}^k) = 0, \\
C(\pi_{[i]}^k) &= \min\left\{\max\left\{C\left(\pi_{[i]}^{\text{pre-}m(\pi_{[i]}^k)}\right), C\left(\pi_{[\text{pre-}k(\pi_{[i]}^k)]}^k\right)\right\} + P(\pi_{[i]}^k)\right\}, \text{pre-}m(\pi_{[i]}^k) \neq 0; \text{pre-}k(\pi_{[i]}^k) \neq 0, \\
C_{\text{st1}}(\pi_{[j]}^t) &= C(\pi_{[i]}^k), \quad k = 1, 2, \dots, m_1; j = 1, 2, \dots, q; i = 1, 2, \dots, H_k.
\end{aligned} \tag{10}$$

(2) The transportation stage: it is a multivehicle single-point delivery problem with the constraints of load and speed in the transportation stage. The total load of all jobs loaded in the vehicle t cannot exceed the

vehicle rated load (V_T). The transportation time of the vehicle t is determined by the speed $v(\pi_{[i]}^{\text{TR-V}})$. The transportation completion time of the job j of π^{TR} is calculated as follows

$$\begin{aligned}
S(\pi_{[j]}^{\text{TR}})_{w_t} &= \max\{C_{\text{st1}}(\pi_{[j]}^{\text{TR}})_{w_t}\}, \quad w_t = 1, \\
S(\pi_{[j]}^{\text{TR}})_{w_t} &= \max\left\{C_{\text{st1}}(\pi_{[j]}^{\text{TR}})_{w_t}^t, S(\pi_{[j]}^{\text{TR}})_{w_{t-1}} + \left(\frac{\text{Dis}_{p-t}}{v(\pi_{[i]}^{\text{TR-V}})} + \frac{\text{Dis}_{p-t}}{v_c}\right)\right\}, \quad w_t \neq 1, \\
C_{\text{st2}}(\pi_{[j]}^{\text{TR}})_{w_t} &= S(\pi_{[j]}^{\text{TR}})_{w_t} + \frac{\text{Dis}_{p-t}}{v(\pi_{[i]}^{\text{TR-V}})}, \quad j = 1, 2, \dots, q; w_t = 1, 2, \dots, l; t = 1, 2, \dots, H_T.
\end{aligned} \tag{11}$$

(3) The assembly stage: it is a single-machine scheduling problem with arrival time or release time. The

calculation formula for the assembly completion time $C_{\text{st3}}(\pi_{[z]}^P)$ of the product π_z^P is shown as follows:

$$\begin{aligned}
S(\pi_{[z]}^P) &= \max\{C_{\text{st2}}(\pi_{[j]}^{\text{TR}})\}, \\
C_{\text{st3}}(\pi_{[z]}^P) &= \max\{C_{\text{st3}}(\pi_{[z-1]}^P), S(\pi_{[z]}^P)\} + P(\pi_{[z]}^P), \quad z = 1, 2, \dots, H_P, j = 1, 2, \dots, q.
\end{aligned} \tag{12}$$

(4) The optimization goal: the optimization goal is to find an optimal sequence $\pi^* = (\pi^K, \pi^{\text{TR}}, \pi^{\text{TR-V}}, \pi^P)$ in the set Π of the three-stage sequence from processing and transportation to assembly of the job

which composed products so that the makespan C_{max} and the total energy consumption E_{total} of the three stages are minimized:

$$\begin{aligned}
C_{\text{max}}(\pi^*) &= \min_{(\pi^K, \pi^{\text{TR}}, \pi^{\text{TR-V}}, \pi^P) \in \Pi} \{C_{\text{max}}(\pi^K, \pi^{\text{TR}}, \pi^{\text{TR-V}}, \pi^P)\}, \\
E_{\text{total}}(\pi^*) &= \min_{(\pi^K, \pi^{\text{TR}}, \pi^{\text{TR-V}}, \pi^P) \in \Pi} \{E_{\text{total}}(\pi^K, \pi^{\text{TR}}, \pi^{\text{TR-V}}, \pi^P)\}, \\
\pi^* &= \arg\left\{ \begin{array}{l} C_{\text{max}}(\pi^K, \pi^{\text{TR}}, \pi^{\text{TR-V}}, \pi^P), \\ E_{\text{total}}(\pi^K, \pi^{\text{TR}}, \pi^{\text{TR-V}}, \pi^P) \end{array} \right\} \longrightarrow \min, \quad \forall (\pi^K, \pi^{\text{TR}}, \pi^{\text{TR-V}}, \pi^P) \in \Pi.
\end{aligned} \tag{13}$$

3. HEDA_VNS for 3sMISP_JBTEC

3.1. *Characteristics of 3sMISP_JBTEC.* The characteristic of 3sMISP_JBTEC have the following three aspects:

- (1) The problem is holistic and systematic. 3sMISP_JBTEC is a multistage integration problem. The scheduling result of the previous stage directly affects the latter stage. Therefore, the overall optimization should be considered when designing the solution algorithm.
- (2) The connection between stages is close. Each stage of 3sMISP_JBTEC is not independent, and there are constraints between the process set, the job set, and the product set. Moreover, all the processes belonging to the same job must be processed before the job is transported, and the jobs belonging to the same product must be transported to the assembly place before assembly. Therefore, there is strong coupling between stages.
- (3) The solution space of the problem is complicated. 3sMISP_JBTEC is a multistage and multiobjective scheduling problem, and its solution space is very complicated. In addition to the processing sequence π^K , the transportation job sequence π^{TR} , and the product sequence π^P , the problem also involves the speed-based vehicles sequence π^{TR-V} . Therefore, it is necessary to analyze and use the characteristics of the problem to reasonably narrow the search space. Meanwhile, π^{TR-V} in the transportation stage needs to be designed with reasonable operation in order to solve the problem better.

Based on the above analysis, the following rules are proposed:

Rule 1. Jobs belonging to the same product are assigned the same priority in the processing stage.

Rule 2. Jobs belonging to the same product should be put into the same vehicle as far as possible under the vehicle loading constraints.

Rule 3. Transport vehicles are filled first and transported first.

Rule 4. All jobs belonging to the same product are assembled first when they arrive at the assembly place first in the assembly stage.

3.2. *Design of HEDA_VNS Algorithm.* In this section, the HEDA_VNS is designed to improve the efficiency for solving 3sMISP_JBTEC. Because the problem solution space is very complicated, HEDA_VNS only searches the solution space of the subproblems in the processing stage (π^K). Then, π^{TR} and π^P are generated according to the characteristics of the problem and rules proposed in Section 3.1. Meanwhile, a reasonable speed scheduling scheme (SSS) is proposed for π^{TR-V} in the transportation stage to further improve the quality of the nondominated solution of the problem.

3.2.1. Encoding and Decoding

(i) *Encoding and decoding of the processing stage.* The encoding of the processing stage is the process-based permutation. Its length is determined by the product-job-process layer. Obviously, the job set P_z is determined by the product z , and the process set Q_{zj} is determined by the job j , which is in the set P_z , $z \in P$, $j \in Q$. The decoding adopts the method of earliest completion time (ECT) combined with insertion rule to determine the starting time of each process, while the machine allocation is completed. Then, the completion time of the job j in the processing stage and the energy consumption of the machine can be determined.

(ii) *Encoding and decoding of the transportation stage.* In the transportation stage, the encoding and decoding of the job and the vehicle are involved. The job encoding of the transportation phase is based on the sorting of the jobs in the nondescending order of the completion time in the processing stage. The job decoding of the transport phase is based on Rule 2 in Section 3.1. $V_{[t]}$ is the current load of the vehicle t , and set $V_{[t]} = 0$ as the initial situation. The specific steps of decoding are given as follows:

Step 1: select the jobs $\pi_{[j]}^{TR}$ belonging to the product $\pi_{[z]}^P$ from left to right in π^{TR} .

Step 2: put the selected $\pi_{[j]}^{TR}$ into the vehicle t in turn under the conditions $V(\pi_{[j]}^{TR}) + V_{[t]} \leq V_T$. If $V(\pi_{[j]}^{TR}) + V_{[t]} > V_T$, put the job j into the vehicle $t + 1$ and then update $V_{[t]}$ and the set of jobs to be transported ($\pi^{TR'}$).

Step 3: check whether there is a job $\pi_{[k]}^{TR}$ in the set $\pi^{TR'}$ that satisfies $V(\pi_{[k]}^{TR}) + V_{[t]} \leq V_T$. If $V(\pi_{[k]}^{TR}) + V_{[t]} \leq V_T$, put the job $\pi_{[j]}^{TR}$ into the vehicle t until $V(\pi_{[k]}^{TR}) + V_{[t]} > V_T$, update $V_{[t]}$ and $\pi^{TR'}$, $t = t + 1$, $z = z + 1$.

Step 4: turn to step 1, until $\pi^{TR'}$ is empty.

Following the above steps, the start transportation time of the job is determined by the job decoding in the transportation stage.

The vehicle encoding is based on the sorting of vehicle speed π^{TR-V} . The decoding of vehicle is based on Rule 3, and the vehicle speed can be obtained by the speed scheduling set. Then, the transportation completion time of the job and transportation energy consumption are determined.

(iii) *Encoding and decoding of the assembly stage.* The assembly encoding is the product-based sorting in the assembly stage, and the product encoding π^P is arrangement of nondescending order which is determined by the time when all the jobs belonging to the product arrive at the assembly workshop. The assembly decoding is based on Rule 4, and then, the product completion time and energy consumption are determined. Thus, the three-stage encoding of 3sMISP_JBTEC is shown in Figure 3.

3.2.2. *The Speed Scheduling Scheme.* The speed interval set in the transportation stage of 3sMISP_JBTEC is $v \in [v_{\min}, v_{\max}]$. If the speed of vehicle is generated

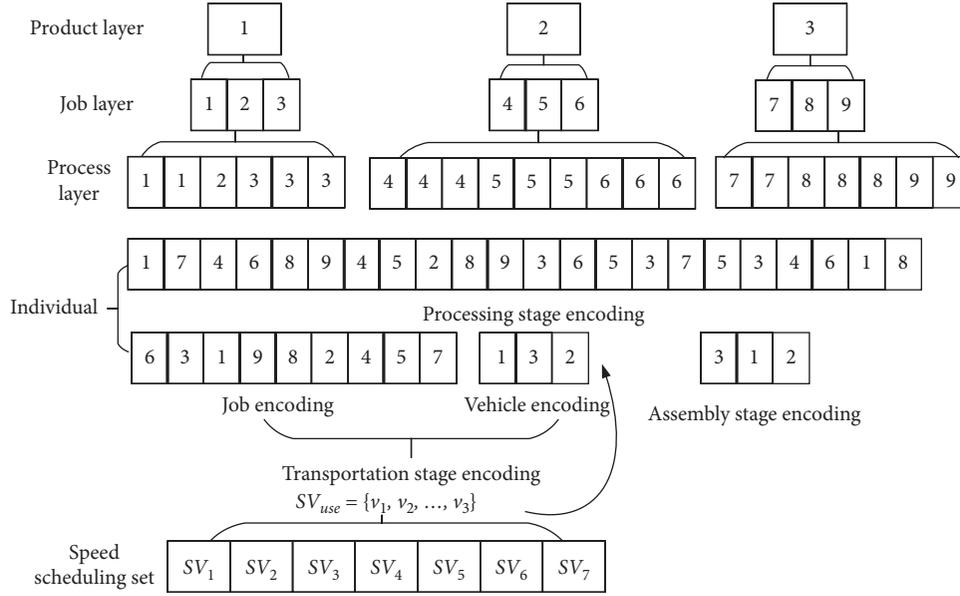


FIGURE 3: The coding method of scheduling solution.

randomly, there is a large span in this interval. Then, it is not easy to find a more accurate nondominated solution to the problem in a short time. Therefore, in the section, a speed scheduling scheme (SSS) is designed to further refine the speed range, which can better solve the problem. The specific operations are as follows:

- (1) The speed interval is divided into three speed sub-intervals: low, medium, and high speed. The low-speed interval is $SV_{low} = [v_{min}, v_a]$, the medium is $SV_{mid} = [v_a, v_b]$, and the high is $SV_{high} = [v_b, v_{max}]$, where $v_{min} < v_a < v_b < v_{max}$.
- (2) The speed scheduling set SV_{use} can be randomly generated from the following seven intervals. SV_{use} obtained by different modes can be traversed to the interval of different speeds while continuously updating the speed set to improve the quality of the solution:
 - ① $SV_{use} \subset SV_{low}$, marked as SV_1 .
 - ② $SV_{use} \subset SV_{mid}$, marked as SV_2 .
 - ③ $SV_{use} \subset SV_{high}$, marked as SV_3 .
 - ④ $SV_{use} \subset (SV_{low} \cup SV_{mid})$, marked as SV_4 .
 - ⑤ $SV_{use} \subset (SV_{low} \cup SV_{high})$, marked as SV_5 .
 - ⑥ $SV_{use} \subset (SV_{mid} \cup SV_{high})$, marked as SV_6 .
 - ⑦ $SV_{use} \subset (SV_{low} \cup SV_{mid} \cup SV_{high})$, marked as SV_7 .

4. Methodology of HEDA_VNS

4.1. HEDA_VNS Global Search

4.1.1. Initialization of Population, Probability Model, and Speed Scheduling Set. According to Rule 1, the product aggregation (PA) method is designed to generate the number of $\eta \times \text{popsize}$ individuals in the processing stage, and the remaining individuals are generated randomly to initialize the population. The PA is a rule to generate an individual. The specific steps about PA are as follows (shown in Figure 4): first, a

product sequence is randomly generated. Then, a job sequence that belongs to the same product is randomly generated. At last, a process sequence that is based on the same job is randomly generated. According to PA, the jobs that make up the same product can be more concentrated. After the jobs have been processed in the processing stage, they can be transported to the assembly stage faster and more concentrated.

According to the encoding method in Section 3.2.1, the process arrangement matrix ρ is used as the probability model. In order to ensure that the algorithm searches the solution space uniformly as much as possible, the probability model $\rho_{i,j}$ adopts a uniform distribution, and $\rho_{i,j}(0) = (1/A)$. The initial speed scheduling set is $SV_{use} \subset SV_7$ so that the transportation speed is generated randomly in $[v_{min}, v_{max}]$.

4.1.2. Update Probability Matrix. For multiobjective optimization, the nondominated solution is selected in iteration to update the probability model. $I_{i,j}$ is the number of times that the job j appeared at or before the position i of the process sequence vector. $E_{i,j}(g)$ is the probability obtained through the nondominated solution statistics in the iteration g . This can be updated by $\rho_{i,j}(g)$, where $\gamma \in (0, 1)$ is the learning rate:

$$I_{i,j} = \begin{cases} 1, & \text{job } j \text{ appears at or before position } i, \\ 0, & \text{else,} \end{cases}$$

$$E_{i,j}(g) = \frac{I_{i,j}}{\sum_{i=1}^n I_{i,j}}, \quad \forall i, j,$$

$$\rho_{i,j}(g+1) = (1-\gamma)\rho_{i,j}(g) + \gamma E_{i,j}. \quad (14)$$

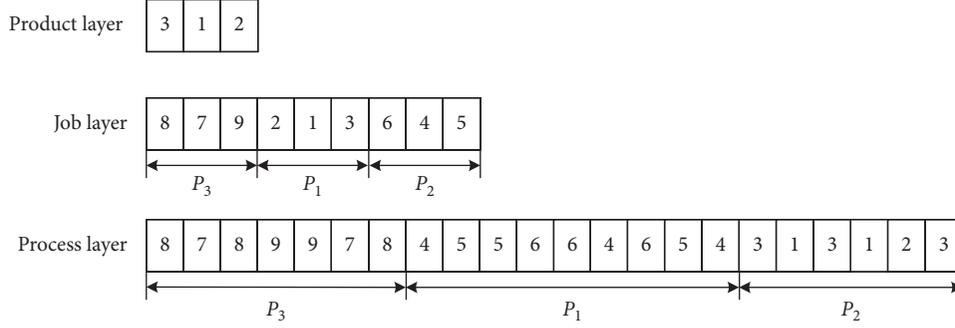


FIGURE 4: An example of the PA rule.

4.1.3. Sampling Method. In subsequent iteration of the algorithm, a new population is generated by sampling the probability matrix. The sampling method in this paper is designed for the problem model. The purpose is to make the jobs belong to the same product more likely to be aggregated together to produce higher-quality individuals, but each sampling uses roulette operation which guarantees the diversity of the population to a certain extent. Once a new individual is generated, it constitutes a new population for the next iteration. Specific steps are as follows:

Step 1: when $i = 1$, the roulette method is used to sample the matrix to select the workpiece directly and judge whether $j \in P_z$. The probability that the job j appears in position i is $\rho_{i,j}(g)$. If the number of job j occurrences in the matrix ρ is equal to the total number of processes of job j , set the column j of the matrix ρ to zero and normalize each row.

Step 2: when $i \neq 1$, add the probabilities of the job j belonging to P_z of the row i to get $\sum_{j=m}^k \rho_{i,j}$, where $j \in P_z$:

Step 2.1: if $\sum_{j=m}^k \rho_{i,j} = 0$, the roulette method is used to sample directly.

Step 2.2: if $\sum_{j=m}^k \rho_{i,j} \neq 0$, a random number $\varepsilon, \varepsilon \in (0.5, 0.8)$ is generated and compared with $\sum_{j=m}^k \rho_{i,j}$. If $\sum_{j=m}^k \rho_{i,j} \geq \varepsilon$, the roulette method is used to sample directly; if $\sum_{j=m}^k \rho_{i,j} < \varepsilon$, then readjust the probability of job j on the current row i to obtain a new probability. Among them, the probability of job j belonging to the product P_z is calculated using formula (16) or otherwise using formula (17). Use roulette to sample after adjustment. If the number of job j in the matrix ρ is equal to the total number of processes of job j , set the column of the matrix to zero and normalize each row:

$$\rho_{i,j}^* = \frac{\rho_{i,j}^z}{\sum_{j_z=m}^k \rho_{i,j}^z} \times \varepsilon, \quad j \in P_z, \quad (15)$$

$$\rho_{i,j}^* = \frac{\rho_{i,j}^z}{1 - \sum_{j_z=m}^k \rho_{i,j}^z} \times (1 - \varepsilon), \quad j \notin P_z. \quad (16)$$

Step 3: judge whether the job j in the row i of sampling by Step 2 belongs to P_z ; if job $j \in P_z$, then $j = j + 1$ directly. If $j \in P_z^*$, then order $P_z^* = P_z$ and $j = j + 1$, until all the processes of the job j belonging to P_z are selected completely.

4.2. Local Search Strategy

4.2.1. Local Search I. After new individuals of the population are sampled through the probability matrix, the probability matrix is updated with the individuals who have obtained nondominated solutions. A local search strategy is designed with Rule 1 in three situations to better guide the global search direction and beneficial to avoid some invalid search. Meanwhile, the original solution structure can be better retained.

For the nondominant solution π_r obtained in the population (where $r = 1, 2, \dots, m$), two positions u, v ($u \neq v$) in π_r are selected randomly, and $\pi_r(u)$ is the job encoding of the position u of π_r .

Situation 1. If $\pi_r(u) = \pi_r(v)$, it means that the two positions are the same job; the Swap operation will be invalid, then the Insert operation is performed at this time. If $u < v$, insert forward. Otherwise, insert back and to generate a new individual π_r' . If $f_1(\pi_r) > f_1(\pi_r')$ and $f_2(\pi_r) > f_2(\pi_r')$, let $\pi_r = \pi_r'$, then update the nondominated solution set.

Situation 2. If $\pi_r(u) \neq \pi_r(v)$ and it does not belong to the same product, let $x = \max(u, v) - \min(u, v)$ and $k = (H_O/H_P - 1)$ (where H_O is the total number of processes of all products, i.e., the length of the individual π_r in the processing stage and H_P is the total number of products); if $x \leq k$, the Swap operation is performed to generate a new individual. If $f_1(\pi_r) > f_1(\pi_r')$ and $f_2(\pi_r) > f_2(\pi_r')$, let $\pi_r = \pi_r'$, update the nondominated solution set.

Situation 3. If $\pi_r(u) \neq \pi_r(v)$, it means that the two positions belong to different jobs. If they belong to the same product, then a hybrid operation based on the variable neighborhood local search of Insert and Swap is performed. Specific steps are given as follows (Algorithm 1).

4.2.2. Local Search II. In the transportation stage, the faster the vehicle transportation speed and the shorter the transportation time, and the jobs will be faster, reaching the assembly stage for assembly. Meanwhile, the makespan decreases, but the energy consumption increases. Hence, the two target values conflict. In order to balance the two goals and improve the quality of nondominated solutions, it is also necessary to search for the transportation speed and vehicle order. Combining the speed scheduling scheme given in Section 3.2.2, a variable neighborhood search hybrid operation based on the speed set is designed in this section.

For the nondominant solution set $\kappa = \{\pi_1, \pi_2, \pi_3, \dots, \pi_m\}$, $\pi_a^{\text{TR-V}}$ is the vehicle sequence corresponding to individual π_a in the transportation stage. Two positions u, v ($u \neq v$) are selected randomly in $\pi_a^{\text{TR-V}}$, and $\pi_o^{\text{TR-V}}(u)$ is the vehicle encoding of the position u of the individual $\pi_o^{\text{TR-V}}$. $\text{SV}_{\text{use}} = \{v_1, \dots, v_{H_T}\}$ is the current speed scheduling set, and SV_x is the speed scheduling set x , where $x = 1, \dots, 7$. The specific operations are as follows (Algorithm 2).

4.3. Species Diversity Determination and Control Mechanism. With the continuous iteration of the algorithm, the diversity of the population will decrease, and the individuals in the population will become very similar, which will cause the algorithm to fall into a local optimum. In order to solve the problem, a population control mechanism of literature [41] is used in this paper to calculate the diversity value θ_{div} of contemporary populations. A diversity threshold δ is given; when $\theta_{\text{div}} \leq \delta$, the current population is adjusted to retain noninferior individuals in the contemporary population, and $\varphi \times \text{popsize}/3$ individuals are randomly generated by the PA rule, and the rest are randomly generated.

4.4. The Framework of HEDA_VNS. In HEDA_VNS, firstly, a probability model is adopted to learn high-quality solution information and guides the algorithm to search globally. Secondly, a Local Search I is designed by using the rules obtained from the analysis of the problem in Section 3.1. Then, in order to improve the quality of the solution, the Local Search II is combined with the speed scheduling set to realize an effective search in the velocity space. Finally, a population control mechanism is designed to maintain the diversity of the population. The framework of HEDA_VNS is shown in Figure 5.

5. Simulation Results and Discussion

5.1. Experimental Settings and Evaluation Index

5.1.1. Experimental Settings. All test questions of 3sMISP_JBTEC are based on the structural of the three-stage integrated system. For test data, the number of jobs composed of each product and the number of procedures corresponding to the jobs are randomly generated in the interval of [2, 5] and [1, 3], respectively. The processing time of each job and the assembly time of each product are randomly generated in the interval of [20, 80] and [100, 200],

respectively. The load of each job is randomly generated in the interval of [20, 50], and the unit energy consumption of the processing machine is randomly generated in the interval of [3, 5]. According to the literature [24], the parameters of vehicle-related are as follows: $C_d = 0.7$, $C_r = 0.01$, $\rho = 1.2041$, $A = 3.912$, and $g = 9.81$. The distance of vehicle transportation is $\text{Dis}_{P-A} = 8000$.

In addition to the number of products and machines, the results of the test problems are also affected by the number of transport vehicles. Therefore, the test problems listed in this paper are expressed as P_M_T . All algorithms and test procedures are implemented by Delphi 2010 programming, and the operating system is WinXP.

5.1.2. Evaluation Index. In the comparison, two indexes are used to evaluate the algorithm [42]. There are $R_N(S_j)$ and $N_N(S_j)$. $R_N(S_j)$ is used to evaluate the quality of the nondominated solutions (shown in equation (17)). When $R_N(S_j) = 0$, the solutions in S_j are all dominated by other solutions. Also, when $R_N(S_j) = 1$, the solutions in S_j are all nondominated solutions in S . $N_N(S_j)$ is the number of solutions in S_j which are not dominated by other solutions (shown in equation (18)). The larger the value of $N_N(S_j)$, the greater the number of nondominated solutions. Obviously, $R_N(S_j)$ is the dominant ratio, which evaluates the overall quality of the algorithm to obtain nondominated solutions, and $N_N(S_j)$ is the number of dominants, which evaluates the ability of the algorithm to obtain nondominated solutions. There is a positive correlation between these two indicators, and the correlation coefficient is $1/S_j$:

$$R_N(S_j) = \left| \frac{S_j - x \in S_j | \exists y \in S: y < x}{|S_j|} \right|, \quad (17)$$

$$N_N(S_j) = |S_j - x \in S_j | \exists y \in S: y < x|. \quad (18)$$

5.2. Key Parameter Setting and Influence of HEDA_VNS. In the section, the three key parameters of HEDA_VNS which are population size (popsize), iterative probability sampling parameters (η), and learning rate (γ) are analyzed by the Design of Experiment (DOE) method of medium-scale problems. On the basis of combining existing research, four levels of each parameter are determined (shown in Table 1); thereby, an orthogonal experimental table is established with a scale of $L_{16}(4^3)$. The algorithm runs independently for each set of parameters for 10 medium-scale questions of 20 times, with $2 \times P \times M \times T$ seconds as the termination condition for each scale and the average value AVG of each scale under different parameter combinations of the algorithm as the evaluation index to determine the appropriate parameters combination. The results are shown in Table 2.

From the experimental results (shown in Table 3 and Figure 6), it can be found that different parameter combinations have a greater impact on the performance of the algorithm. Based on the experimental analysis results, the best parameters popsize = 50, $\eta = 0.4$, and $\gamma = 0.3$ are

```

Set loop = 1;
Repeat
  Set count = 0; max_count = 2;
  Repeat
    If count = 0, then the Insert operation is performed on  $\pi_r$  to get  $\pi_r'$ ;
    If count = 1, then the Swap operation is performed on  $\pi_r$  to get  $\pi_r'$ ;
    If  $f_1(\pi_r) > f_1(\pi_r')$  and  $f_2(\pi_r) > f_2(\pi_r')$ , then count = 0,  $\pi_r = \pi_r'$ , update nondominated solution set;
    Else, let count = count + 1;
  Until count = max_count;
Until loop = max_loop;
End.

```

ALGORITHM 1: Local search insertSwap method.

```

For  $a = 1$  to  $m$ 
  For  $x = 1$  to 7
     $SV_{use} = SV_x$ ;
    Set loop = 1;
    Repeat
      Set count = 0; max_count = 2;
      Repeat
        If count = 0, then the Insert operation is performed on  $\pi_a^{TR-V}$  to get  $(\pi_a^{TR-V})'$ ;
        If count = 1, then the Swap operation is performed on  $\pi_a^{TR-V}$  to get  $(\pi_a^{TR-V})'$ ;
        If  $f_1(\pi_a, \pi_a^{TR-V}, SV_{use}) > f_1(\pi_a, (\pi_a^{TR-V})', SV_{use})$  and  $f_2(\pi_a, \pi_a^{TR-V}, SV_{use}) > f_2(\pi_a, (\pi_a^{TR-V})', SV_{use})$  then set
count = 0,  $\pi_a^{TR-V} = (\pi_a^{TR-V})'$ , update nondominated solution set;
        Else, set count = count + 1;
      Until count = max_count;
    Until loop = max_loop;
  End;
End.

```

ALGORITHM 2: Local search speed mechanism.

set in this paper, and further performance testing and comparison will be carried out based on this parameter setting.

5.3. Analysis of the Results of Energy Consumption and Makespan under a Certain Problem Scale. In this section, HEDA_VNS is used to solve the 3sMISP_JBTEC with a problem scale of (3_3_2). The scale of the problem can be described as follows: 3 heterogeneous parallel machines in the processing stage, 2 vehicles with the same load in the transportation stage, and one machine in the assembly stage. 3 products are required to be processed and assembled during the three stages. The product composition is shown in Figure 7. The 3 products are assembled from multiple jobs. Each job has multiple processes, and each process can be processed on multiple machines. The vehicle's return to no-load speed is $v_c = 40$. The data of each stage of the product are shown in Table 4.

Figure 8 shows the noninferior solution set of the problem, and other scales are similar. It can be seen from Figure 8 that all the solutions are independent of each other due to conflicts and constraints between the makespan and total energy consumption (TEC). The

conflicts and containments are embodied in both machine processing stage and vehicle transportation stage. The solution at point A in Figure 8 shows that it has a large span in the machine processing stage, but it has close connection between the stages, and all vehicles are transported at high speed. Meanwhile, the makespan is relatively smallest, but energy consumption is relatively highest. On the one hand, high-speed transportation can reduce the transportation time and the accumulation of intermediate inventory and effectively prevent delays in the manufacturing process. On the other hand, high-speed transportation may cause that jobs arrive early for assembly, which is prone to bottlenecks and a backlog of finished products. Therefore, it is not a blind pursuit of makespan, and it is necessary to limit the maximum speed level of transportation vehicles. The solution at point B in Figure 8 represents that the machining span is small, but the connection between the stages is not tight enough, and all vehicles are transported at low speed. Meanwhile, the makespan is relatively largest, but energy consumption is relatively lowest. In this case, although the processing stage is compact, the delivery delay time and the intermediate inventory level will be at the highest level, and the delivery of the finished product may be delayed. It can be

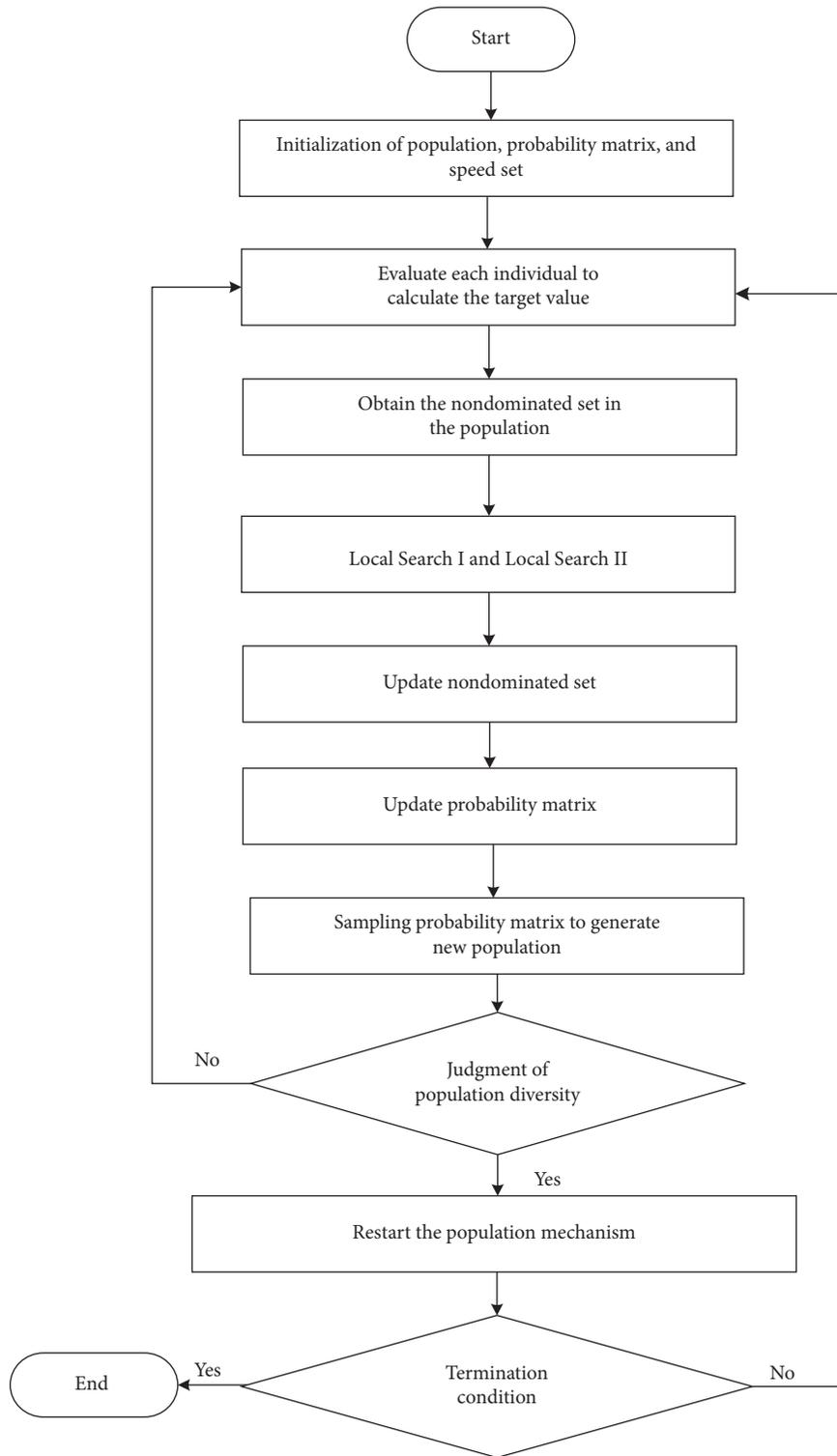


FIGURE 5: The framework of HEDA_VNS.

seen that although it is the same problem, the noninferior solution will be affected due to different vehicle transportation speeds. Hence, enterprise managers need to make decisions based on the company itself and related policies, taking into account factors such as order time and demand; for example, the selected solution will be

different if the same order is completed in a shorter planning period and the selected solution is completed in a longer planning period, but under normal circumstances, decision makers generally choose point C because the makespan and energy consumption are at a normal and acceptable level.

TABLE 1: Parameter level.

Parameter	Level			
	1	2	3	4
Popsize	30	40	50	60
η	0.3	0.4	0.5	0.6
γ	0.1	0.2	0.3	0.4

TABLE 2: Orthogonal table and average statistics.

Parameter combination	Level			AVG \times 10
	Popsize	η	γ	
1	1	1	1	0.501
2	1	2	2	0.447
3	1	3	3	0.393
4	1	4	4	0.299
5	2	1	2	0.375
6	2	2	3	0.491
7	2	3	4	0.386
8	2	4	1	0.293
9	3	1	3	0.607
10	3	2	4	0.583
11	3	3	1	0.412
12	3	4	2	0.286
13	4	1	4	0.456
14	4	2	1	0.486
15	4	3	2	0.363
16	4	4	3	0.368

TABLE 3: The response value of parameters.

Level	Popsize	η	γ
1	0.410	0.485	0.423
2	0.386	0.502	0.368
3	0.472	0.389	0.465
4	0.419	0.312	0.431

5.4. *Comparison of Experimental Results of HEDA_VNS with Other Intelligent Algorithms.* In order to further verify the effectiveness of HEDA_VNS compared with HMOBSA [19] and GA-GSO-GTHS [23], a large number of experiments can be carried out by means of the generated different types of problems. HMOBSA is an effective multiobjective backtracking algorithm. The performance of HMOBSA is superior to the well-known multiobjective algorithm (nondominated sorting genetic algorithm, NSGAI) and MOEA/D in solving energy consumption problems. The turn off/on strategy is also used when calculating energy consumption in the processing stage. GA-GSO-GTHS is a multiobjective hybrid optimization algorithm with strategy. The performance of GA-GSO-GTHS is better than most of the latest multiobjective optimization algorithms that are widely used in scheduling problems. It is effective in solving production energy consumption problems that consider transportation. In addition, the parameters of HMOBSA are set as follows: popsize = 50, crossover probability $p_c = 0.9$, and mutation probability $p_m = 0.1$. The parameters of GA-GSO-GTHS are set as follows: popsize = 100, $p_c = 0.6$, $p_m = 0.05$, step size $s = 5$,

maximum perceptual radius $r_s = 20$, volatility coefficient $\rho = 0.4$, enhancement factor $\gamma = 0.6$, perceptual radius scaling factor $\beta = 0.08$, $n_t = 5$, and $l(0) = 5$. The parameters selected by the three algorithms are all good parameters. Besides, HEDA_VNS runs for 200 generations, and other algorithms run the same time as HEDA_VNS. The comparative results of the three algorithms to solve 3sMISP_JBTEC under the randomly generated speed in $[v_{\min}, v_{\max}]$ are listed in Table 5. Among them, R_N and N_N , respectively, represent the index values of the evaluation solutions corresponding to the three algorithms. The data in Table 5 are the average values of the algorithm after 20 runs.

It can be seen from Table 5, for the 25 examples to be tested in this paper, the proposed HEDA_VNS finds more and better nondominated solutions for 23 instances of the 25 problem instances in terms of R_N and N_N . Moreover, as the problem size increases, HEDA_VNS is more effective. This is because HEDA_VNS is designed based on the characteristic of the problem, and the advantages of HEDA_VNS are mainly reflected in the following three aspects:

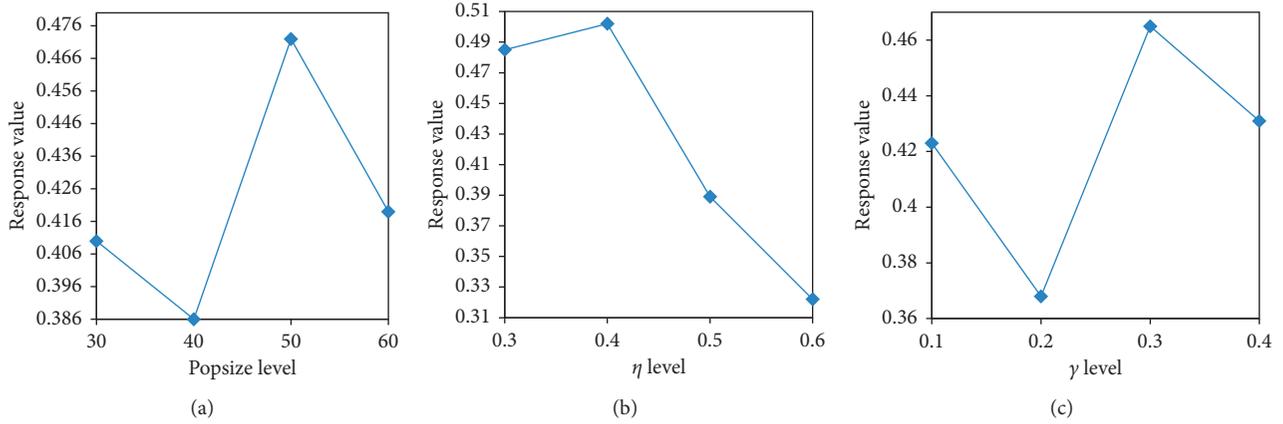


FIGURE 6: The level trend of the influence of each parameter on the algorithm performance.

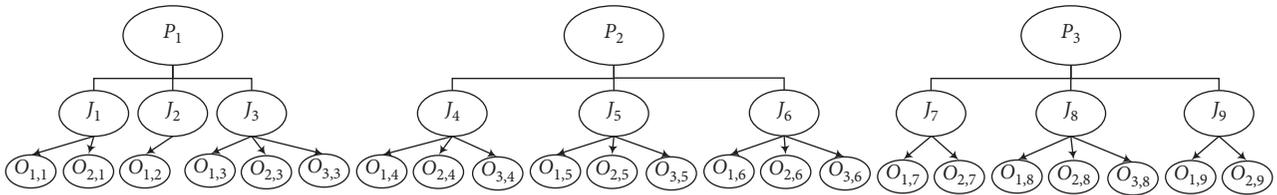


FIGURE 7: The product composition of the problem scale of (3_3_2).

TABLE 4: Data at each stage of the problem scale of (3_3_2).

		P_1			P_2						P_3										
		J_1	J_2	J_3	J_4		J_5		J_6		J_7	J_8		J_9							
P_{ijk}	m_1	27	—	99	—	—	116	91	—	113	75	54	—	87	—	—	—	80	98	—	
	m_2	25	—	—	39	—	75	101	110	62	—	—	31	112	—	59	—	—	—	—	80
	m_3	77	107	—	59	28	57	56	—	33	26	101	—	50	60	—	—	97	97	83	—
R_{jk}	m_1	55	93	—	62	—	—	41	—	—	101	—	—	66	—	39	—	—	84	—	59
	m_2	88	75	—	87	—	—	70	—	—	101	—	—	47	—	80	—	—	37	—	49
	m_3	25	55	—	69	—	—	60	—	—	64	—	—	88	—	114	—	—	72	—	30
V_j		33	48	—	25	—	—	52	—	—	6	—	—	36	—	31	—	—	25	—	33
A_z				161							128								99		

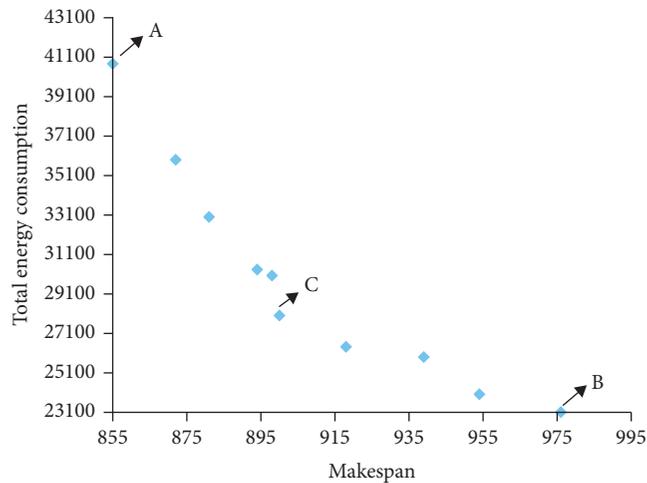


FIGURE 8: The noninferior solution set of the problem scale of (3_3_2).

TABLE 5: Comparison of HEDA_VNS with HMOBSA and GA-GSO-GTHS.

No.	P_M_T	HMOBSA		GA-GSO-GTHS		HEDA_VNS	
		R_N	N_N	R_N	N_N	R_N	N_N
01	2_3_2	0.20	0.50	0.55	1.30	0.47	1.25
02	3_2_2	0.05	0.10	0.50	0.97	0.52	1.30
03	3_3_2	0.02	0.05	0.35	1.35	0.80	2.15
04	4_4_2	0.05	0.30	0.75	1.55	0.60	1.50
05	4_5_3	0.07	0.02	0.35	1.15	0.68	1.80
06	5_3_3	0.03	0.01	0.27	0.45	0.81	1.60
07	5_4_3	0.05	0.15	0.52	0.85	0.75	1.55
08	5_5_3	0.15	0.35	0.13	0.65	0.80	1.95
09	5_6_3	0.02	0.05	0.15	0.35	0.92	2.05
10	6_4_3	0.00	0.00	0.13	0.35	0.95	2.20
11	6_5_3	0.05	0.10	0.27	0.50	0.86	2.05
12	6_6_3	0.02	0.05	0.22	0.65	0.90	2.20
13	6_4_4	0.03	0.10	0.28	0.50	0.83	1.95
14	6_5_4	0.02	0.05	0.32	0.85	0.78	1.90
15	6_6_4	0.00	0.00	0.11	0.35	0.92	2.65
16	7_5_4	0.00	0.00	0.16	0.30	0.95	2.80
17	7_6_4	0.02	0.05	0.24	0.45	0.93	2.25
18	7_7_4	0.03	0.05	0.12	0.35	0.93	2.05
19	8_8_4	0.00	0.00	0.25	0.95	0.95	2.60
20	9_9_4	0.00	0.00	0.10	0.25	0.95	1.85
21	9_9_5	0.05	0.15	0.15	0.20	0.93	1.70
22	10_10_5	0.02	0.05	0.08	0.15	0.92	2.05
23	10_10_8	0.00	0.00	0.03	0.10	0.96	1.60
24	15_15_5	0.00	0.00	0.02	0.05	0.95	1.95
25	15_15_10	0.00	0.00	0.01	0.05	0.96	1.55

- (1) The solution generated by the PA rule in the initialization phase can better improve the quality of the initial solution.
- (2) A sampling mechanism based on the nature of the problem is designed so that the jobs belonging to the same product can be selected with a higher probability, which can more clearly guide the algorithm to search in the high-quality solution area.
- (3) Two local search strategies are designed to improve the performance of the algorithm. Among them, three different search methods are designed in the Local Search I to perform a detailed search on the noninferior solution high-quality regions to improve the efficiency of the solution, which can effectively avoid some invalid searches and better retain the original high-quality solution structure. Then, a variable neighborhood hybrid operation based on the speed set is designed in Local Search II to search for transportation speed and vehicle order, which can balance the two optimization goals while further improving the quality of the solution.

6. Conclusion

This paper aims at solving the three-stage multiobjective integrated scheduling problem considering the energy consumption (3sMISP_JBTEC) of processing-transport-assembly and studying the multiobjective problem modeling and solution algorithm with makespan and energy consumption. In terms of modeling, a mathematical model of

multiobjective optimization problems is established. Then, the improved energy-saving strategy is adopted to save energy and extend the life of the machine in the processing and assembly stages. Meanwhile, the impact of speed on energy consumption is considered in the transportation stage. For solving the problem, the HEDA_VNS is designed to solve 3sMISP_JBTEC based on the characteristics of the problem. There is a conflict between the makespan and the energy consumption through the results of testing same-scale problems, and decision-makers need to make decisions based on different actual conditions. Moreover, HEDA_VNS is compared with other algorithms on large-scale problems, and comparative simulations show the efficiency of the proposed HEDA_VNS. This obtained result can provide a reference for the implementation of intelligent production in the assembly manufacturing industry.

Data Availability

Data were curated by the authors, the link for the test data is (<https://dx.doi.org/10.13140/RG.2.2.29751.44969>).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61963022 and 51665025)

and Scientific Research Foundation of Yunnan Provincial Department of Education (Grant no. 2021J0052).

References

- [1] N. Salahi and M. A. Jafari, "Energy-performance as a driver for optimal production planning," *Applied Energy*, vol. 174, no. 15, pp. 88–100, 2016.
- [2] J. M. Bloemhof, P. Beek, L. Hordijka et al., "Interactions between operational research and environmental management," *European Journal of Operation Research*, vol. 85, no. 2, pp. 229–243, 1995.
- [3] G. May, M. Taisch, B. Stahl et al., "Toward energy efficient manufacturing: a study on practices and viewpoint of the industry," *Competitive Manufacturing for Innovative Products and Services*, vol. 8, 2012.
- [4] Y. Liu and A. Tiwari, "An investigation into minimising total energy consumption and total completion time in a flexible job shop for recycling carbon fiber reinforced Polymer," *Procedia CIRP*, vol. 29, no. 3, pp. 722–727, 2015.
- [5] Q. Zhu, J. F. Lu, A. Mayyas et al., "Production energy optimization using low dynamic programming, a decision support tool for sustainable manufacturing," *Journal of Cleaner Production*, vol. 15, no. 2, pp. 135–147, 2014.
- [6] K. Geng, C. Ye, Z. h. Dai, and L. Liu, "Bi-objective re-Entrant hybrid flow shop scheduling considering energy consumption cost under time-of-use electricity tariffs," *Complexity*, vol. 2020, no. 1, pp. 1–17, 2020.
- [7] B.-H. Zhou and C.-Y. Shen, "Multi-objective optimization of material delivery for mixed model assembly lines with energy consideration," *Journal of Cleaner Production*, vol. 192, no. 1, pp. 293–305, 2018.
- [8] C. Koulamas and G. J. Kyparisis, "The three-stage assembly flowshop scheduling problem," *Computers & Operations Research*, vol. 28, no. 7, pp. 689–704, 2001.
- [9] S. Hatami, S. Ebrahimnejad, M. R. Tavakkoli et al., "Two meta-heuristics for three-stage assembly flowshop scheduling with sequence-dependent setup times," *International Journal of Advanced Manufacturing Technology*, vol. 50, no. 12, pp. 1153–1164, 2010.
- [10] A. Maleki and I. Seyedi, "A three-stage assembly flow shop scheduling problem with blocking and sequence-dependent setup times," *Journal of Industrial Engineering International*, vol. 8, no. 1, pp. 1–7, 2012.
- [11] S. C. Campos and J. E. C. Arroyo, "NSGA-II with iterated greedy for a bi-objective three-stage assembly flowshop scheduling problem, Conference on Genetic & Evolutionary Computation," *ACM*, vol. 12, no. 1, pp. 429–436, 2014.
- [12] C. Deng, B. Qian, R. Hu et al., "Rule-based hybrid EDA for three-stage assembly integrated scheduling problem with job batches transportation problem," *Control and Decision*, vol. 35, no. 10, pp. 2507–2513, 2020.
- [13] J. N. D. Gupta, "Two-stage, hybrid flowshop scheduling problem," *Journal of the Operational Research Society*, vol. 39, no. 4, pp. 359–364, 1988.
- [14] J. Y. Ding, S. Song, and C. Wu, "Carbon-efficient scheduling of flow shops by multi-objective optimization," *European Journal of Operational Research*, vol. 248, no. 3, pp. 758–771, 2015.
- [15] J. Yan, L. Li, F. Zhao, F. Zhang, and Q. Zhao, "A multi-level optimization approach for energy-efficient flexible flow shop scheduling," *Journal of Cleaner Production*, vol. 137, no. 1, pp. 1543–1552, 2016.
- [16] D. Tang, M. Dai, M. A. Salido, and A. Giret, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization," *Computers in Industry*, vol. 81, no. 1, pp. 82–95, 2016.
- [17] R. Zhang and R. Chiong, "Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption," *Journal of Cleaner Production*, vol. 112, no. 7, pp. 3361–3375, 2015.
- [18] X. Zheng, S. Zhou, R. Xu et al., "Energy-efficient scheduling for multi-objective two-stage flow shop using a hybrid ant colony optimisation algorithm," *International Journal of Production Research*, vol. 58, no. 13, pp. 1–18, 2020.
- [19] C. Lu, L. Gao, X. Li, Q. Pan, and Q. Wang, "Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm," *Journal of Cleaner Production*, vol. 144, no. 6, pp. 228–238, 2017.
- [20] M. Dai, D. Tang, A. Giret, and M. A. Salido, "Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 59, no. 1, pp. 143–157, 2019.
- [21] Y. Tanimizu and K. Amano, "Integrated production and transportation scheduling for multi-objective green supply chain network design," *Procedia CIRP*, vol. 57, no. 12, pp. 152–157, 2016.
- [22] Z. Guo, L. Shi, L. Chen, and Y. Liang, "A harmony search-based memetic optimization model for integrated production and transportation scheduling in MTO manufacturing," *Omega*, vol. 66, no. 2, pp. 327–343, 2017.
- [23] Z. Liu, S. Guo, and L. Wang, "Integrated green scheduling optimization of flexible job shop and crane transportation considering comprehensive energy consumption," *Journal of Cleaner Production*, vol. 211, no. 20, pp. 765–786, 2019.
- [24] Y. Kuo, "Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem," *Computers & Industrial Engineering*, vol. 59, no. 1, pp. 157–165, 2010.
- [25] E. Demir, T. Bektaş, and G. Laporte, "An adaptive large neighborhood search heuristic for the pollution-routing problem," *European Journal of Operational Research*, vol. 223, no. 2, pp. 346–359, 2012.
- [26] M. Salehi, M. Jalalian, and M. M. Vali Siar, "Green transportation scheduling with speed control: trade-off between total transportation cost and carbon emission," *Computers & Industrial Engineering*, vol. 113, no. 3, pp. 392–404, 2017.
- [27] J.-Q. Li, Y. Du, K.-Z. Gao et al., "A hybrid iterated greedy algorithm for a crane transportation flexible job shop problem," *IEEE Transactions on Automation Science and Engineering*, vol. 99, no. 1, pp. 1–18, 2021.
- [28] J. Q. Li, Y. Q. Han, P. Y. Duan et al., "Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems," *Journal of Cleaner Production*, vol. 250, no. 2, pp. 223–356, 2019.
- [29] J. Li, Z.-M. Liu, C. Li, and Z. Zheng, "Improved artificial immune system algorithm for type-2 fuzzy flexible job shop scheduling problem," *IEEE Transactions on Fuzzy Systems*, vol. 99, no. 1, pp. 1–191, 2020.
- [30] Z.-w. Sun and X.-s. Gu, "A novel hybrid estimation of distribution algorithm for solving hybrid flowshop scheduling problem with unrelated parallel machine," *Journal of Central South University*, vol. 24, no. 8, pp. 1779–1788, 2017.

- [31] B. Qian, Z.-c. Li, and R. Hu, "A copula-based hybrid estimation of distribution algorithm for m-machine reentrant permutation flow-shop scheduling problem," *Applied Soft Computing*, vol. 61, no. 3, pp. 921–934, 2017.
- [32] X. Wang, K. Xing, X. Li, and J. Luo, "An estimation of distribution algorithm for scheduling problem of flexible manufacturing systems using Petri nets," *Applied Mathematical Modelling*, vol. 55, no. 6, pp. 776–788, 2018.
- [33] Z. C. Li, B. Qian, R. Hu, L. L. Chang, and J. B. Yang, "An elitist nondominated sorting hybrid algorithm for multi-objective flexible job-shop scheduling problem with sequence-dependent setups," *Knowledge-Based Systems*, vol. 173, no. 6, pp. 83–112, 2019.
- [34] M. Dai, Z. Zhang, A. Giret, and M. A. Salido, "An enhanced estimation of distribution algorithm for energy-efficient job-shop scheduling problems with transportation constraints," *Sustainability*, vol. 11, no. 11, pp. 3085–3098, 2019.
- [35] Z. J. Yang, Z. Wang, Y. Wang et al., "Two-stage estimation of distribution algorithm to solve multi-vehicle carpooling problem," *Journal of Transportation Systems Engineering and Information Technology*, vol. 16, no. 2, pp. 164–168, 2016.
- [36] H. Mokhtari and A. Hasani, "An energy-efficient multi-objective optimization for flexible job-shop scheduling problem," *Computers & Chemical Engineering*, vol. 104, no. 4, pp. 339–352, 2017.
- [37] M. B. Yildirim and G. Mouzon, "Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm," *IEEE Transactions on Engineering Management*, vol. 59, no. 4, pp. 585–597, 2012.
- [38] M. Dal, D. Tang, A. Giret et al., "Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm," *Robotics and Computer Integrated Manufacturing*, vol. 29, no. 9, pp. 418–429, 2016.
- [39] A. Palmer, *The Development to Fan Integrated Routing and Carbon Dioxide Emissions Model for Goods Vehicles*, Cranfield University, School of Management, Cranfield, UK, 2007.
- [40] L. Kara, B. Y. Kara, and M. K. Yetis, "Energy minimizing vehicle routing problem," *International Conference on Combinatorial Optimization and Applications*, vol. 105, no. 5, pp. 916–1024, 2007.
- [41] Q.-K. Pan and R. Ruiz, "An estimation of distribution algorithm for lot-streaming flow shop problems with setup times," *Omega*, vol. 40, no. 2, pp. 166–180, 2012.
- [42] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, 2003.