International Journal of Industrial Engineering Computations 15 (2024) 387–414

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

A case study of whale optimization algorithm for scheduling in C2M model

Hongying Shan[a], Xinze Shan[a], Libin Zhang[b*], Mengyao Qin[a], Peiyang Peng[a] and Zunyan Meng[a]

[a]*School of Mechanical and Aerospace Engineering, Jilin University, Changchun 130000, China*
[b]*School of Transportation, Jilin University, Changchun 130000, China*

C H R O N I C L E

Article history:
Received October 26 2023
Received in Revised Format January 28 2024
Accepted February 17 2024
Available online February 17 2024

Keywords:
Worker scheduling
Learning curve
Whale optimization algorithm
Elite Non-dominant Sorting
Multi-objective optimization

A B S T R A C T

With the continuous upgrading of industrial technology and information technology, consumers can deeply participate in the whole life cycle of products and realize customized production. These unprecedented changes have brought consumers and manufacturers closer together, resulting in the intelligent business model of "Internet + Customized Production" and "Customer to Manufacturer (C2M)". C2M has been adopted by more and more companies. However, the transition from traditional business models to C2M is a problem that every company must face and solve. Personalized orders of many varieties and small lots put enormous pressure on the production of mainly labor-intensive electronic assembly companies. The theoretical findings of Industry 4.0 and Lean Manufacturing show that people play a central role in assembly operations. As an important element of the production system, worker scheduling has a direct impact on delivery time and cost. Worker scheduling requires not only matching people to jobs, but also considering flexible employment. According to the "Learning Curve" theory, workers with learning potential can continuously enrich their skills and work efficiency will show dynamic changes. Therefore, under the condition of shortest order delivery time and lowest cost, worker scheduling considering the learning effect becomes a challenge for enterprise decision makers. Firstly, the production method of manufacturing industry in C2M environment is studied. Then, based on single-skill task and multi-skill task, respectively, a learning curve-based model of dynamic change in worker skill level is constructed. And this model is used as the input of the assembly line worker scheduling model. Secondly, an Elite Non-dominant Sorting Whale Optimization Algorithm (ENS-WOA) is designed for this multi-objective optimization problem. The correctness and feasibility of the proposed algorithm are verified by selecting classical arithmetic cases for experimental comparison with other algorithms. Finally, the established worker efficiency change model, worker scheduling model and the proposed algorithm are applied to optimize the assembly line of water pump products of Company B, which is being transformed to C2M, and solved by MATLAB software. The results show that the model proposed in this paper is effective, stable and practical compared with the worker costs and delivery period required to complete the order in the original assembly line. Worker costs were reduced by 29.02% and orders were completed approximately 10 days earlier.

1. Introduction

Customer to Manufacturer (C2M), is a model developed from the mass customization of Alvin Toffler's *Future Shock* in 1970. It is characterized by the consumer facing the manufacturer and the manufacturer providing personalization to the consumer, removing the intermediate link. In the C2M model, factories need the deep integration of flexible manufacturing technology, digital technology, and industrial information technology to realize customized production. Previously, research on flexible production systems has mostly focused on control theories and methods such as machine and equipment, production planning and decision optimization, ignoring workers as an important factor. As the core element in enterprise production organization, workers' learning ability, memory and flexibility enable production systems to effectively cope with complex and changing environments, but at the same time bring new challenges to optimally control production. In recent years, China's labor market has become increasingly complex, with rising labor costs and increasing mobility of personnel. This is especially true in the

* Corresponding author
E-mail 1977916962@qq.com (L. Zhang)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2024 Growing Science Ltd.
doi: 10.5267/j.ijiec.2024.2.002

C2M business model, where a large number of workers need to be employed to enable rapid reconfiguration of production lines, customized production and increased production flexibility, making worker scheduling an even more important issue.

The rest of this paper is organized as follows: Section2 shortly analyzes the literature on the manufacturing industry under the C2M model, the worker scheduling problem under the flexible employment strategy considering the learning effect and the improved whale optimization algorithm; Section 3 proposes a dynamic change model of the operational efficiency of single-skilled workers and multi-skilled workers based on the learning curve, and uses it as the input of the assembly line worker scheduling model; Section 4 improves the whale optimization algorithm and designs the Elite Non-dominant Sorting Whale Optimization Algorithm (ENS-WOA) for solving the mathematical model, introducing non-dominant sorting, congestion distance, elite selection strategy ,crossover and variation in the original whale algorithm to improve the overall performance of the algorithm; Section 5 demonstrates the practicality of the proposed model and algorithm through cases; Section 6 summarizes the contributions made in this paper and the sustainable research directions for the future.

2. Literature review

2.1 The C2M model in manufacturing

Temu has adopted a C2M strategy that allows consumers to communicate directly with manufacturers through digital media to customize personalized products. In 2020, the first Suning C2M industry belt was established, followed by cooperation agreements with more than 20 companies in a row. Manufacturers can design and manufacture products with guaranteed demand as well as to attract consumers and meet their individual needs at low prices. Companies such as SAIC Volkswagen and SAIC Datsun have invested heavily in market research for this business to explore the individual needs of customers in the C2M model. Nissan Motor has proposed the “Any volume, Anytime, Anybody, Anywhere, Anything” plan to help customers realize customized production. Home furnishing enterprises such as Sofia have successfully completed the transformation and upgrading of C2M by relying on the trinity development of “Big data + Internet e-commerce platform + Intelligent factory”. The above research shows that the C2M model is in the development stage in China. Domestic and international research on practical solutions for production systems in the C2M model is scarce, especially for assembly lines in the manufacturing sector.

2.2 Worker scheduling problems considering learning effect

In 1936, Wright (1936) proposed the "log-linear model", also known as Wright's model. However, Wright's model did not provide the best fit in all cases, and various types of learning curves, such as Stanford-B model, DeJong model, Plateau model, and S-curve model, were developed for different applications. In recent years, there have been many studies on theoretical research and applications of learning effect, focusing on task repetition-based scheduling in manufacturing workshops (Azzouz et al., 2018; Y. Li et al., 2019), construction (Lee et al., 2015; Tai et al., 2021) and healthcare (Prasad et al., 2022; Tang et al., 2022; Valsamis et al., 2018), but there are fewer studies on worker scheduling based on learning effect theory. However, the learning effect has a non-negligible impact on the production system of enterprises and is an effective way to reduce production costs. Especially in the production system with mainly manual work, it is important to consider the learning effect of workers. Cohen and Ezey Dar-El (1998) studied the equilibrium problem of determining the optimal number of workstations for an assembly line considering the learning effect, pursuing cost minimization and profit maximization. Anzanello and Fogliatto (2011) did a systematic review of learning curve models and applications, and concluded that by modeling the learning curve can better assign tasks to workers, plan production more efficiently, and reduce production costs. Karaoz and Albeni (2005) and others integrated learning curves and indices describing technical aspects to assess worker performance under long production runs. Lohmann et al. (2019a) integrated learning curve modeling and cluster analysis methods to propose a homogeneous group formation based on workers' learning situations framework that helps managers to better organize production in mass customization scenarios. To fully utilize the potential of the workforce, Cavagnini et al. (2020) established exponential learning curves as a basis for quantifying worker allocation decisions. Neidigh and Harrison (2010) optimized the learning rate for order scheduling. Liu et al. (2016) performs dynamic staff allocation for a fiber optic connector manufacturing firm based on worker learning and forgetting effect to minimize the sum of inventory holding cost and the out-of-stock cost. Lohmann et al. (2019b) grouped workers with similar learning profiles by integrating learning curve modeling and cluster analysis. Fichera et al. (2017) investigated the problem of job scheduling, machine scheduling, and man-machine assignment in Flow shop considering the learning ability of workers.

Most of the above studies homogenize employees' learning and work to simplify the learning curve model, but there are few visual demonstrations of workers' learning effect. Meanwhile, papers devoted to learning curve models seldom consider worker scheduling assignments.

2.3 Worker scheduling under flexible employment strategy

Flexible employment was born in the 1920s during the Great Depression in the U.S. The common strategy was to recruit temporary workers, including fixed-term contract workers, temporary helpers, on-call workers and seasonal workers. IBM replaced its all-hire model with a flexible hiring model. General Motors Company hired about 200 temporary workers to replace absent workers on the assembly line. The Honda plant in Marysville, Ohio, called on office employees to complete assembly line tasks in response to worker shortages. Foote and Folta (2002) used temporary workers as an option for workforce expansion decisions. Kim et al. (2018) increased productivity by introducing unskilled temporary workers to minimize total workstation costs, total costs of skilled and unskilled temporary workers, the minimizing the cycle time and work load as the objective to develop a mathematical model and verify the validity of this idea with examples. Corominas et al. (2008) also proposed the use of temporary workers to enhance the flexibility of workers to respond to seasonal demand orders. Similarly, due to the diversity of seasonal demand, Erdem (2011)opened new production lines, shift changes, and hired seasonal workers to increase production capacity. The key scheduling issue in a temporary labor environment is the hiring and allocation of permanent and temporary workers. Pinker and Larson (2003) developed a model to size permanent and temporary workers to minimize expected labor and backlog costs in an uncertain demand environment. Liu et al. (2022) found that a limited temporary and mobile worker configuration can solve the risk aversion problem in assembly line balancing. Emmons and Fuh (1997) studied the worker scheduling problem considering leave in the presence of both temporary and permanent workers. Stratman et al. (2004) analyzed the effect of worker skill dynamics using an analytic experimental design to compare the effect of using temporary and permanent workers on manufacturing cost performance. Mathur and Süer (2013) studied the practical problems of textile companies and proposed overtime as a useful strategy to adjust production capacity to reduce the amount of delayed work. To curb the slowdown of large assembly lines due to absenteeism and absenteeism, (Pilati et al., 2021)split the assembly line into several sections.

2.4 The improved whale optimization algorithm

In 2016, Mirjalili and Lewis (2016) proposed a new metaheuristic optimization algorithm, Whale Optimization Algorithm (WOA) for short, by studying the unique hunting foraging behavior of humpback whales. The algorithm is conceptually simple, with few parameters and easy to program, and has been applied in many fields, such as feature selection, data clustering, medical image diagnosis, neural network training, optimal resource allocation, economic scheduling, and unit combination. In order to solve the respective problems better with the help of algorithms, numerous scholars have proposed improved whale optimization algorithms. Paul et al. (2023) proposed a quasi-oppositional-based whale optimization algorithm (QOWOA) for solving the economic scheduling problem of cogeneration and compared it with some state-of-the-art algorithms to judge the effectiveness and stability of QOWOA. Chakraborty et al. (2022) proposed ImWOA for the image segmentation problem, and modified the selection process of random solutions in the search prey phase to prevent falling into local optimum. Ruihan (2021) proposed an improved whale optimization algorithm (AWOA) by introducing adaptive step factor function and adaptive difference variation factor to solve the problems of slow convergence and weak search ability in the late iteration. Singh and Singh (2023) combined WOA with NSGA-II algorithm to propose a bio-inspired re-initialization and decomposition whale optimization algorithm (R&D WOA). To address the drawbacks of WOA in large-scale optimization problems such as slow convergence and jumping out from the extremum, (Sun et al., 2022)proposed an improved whale optimization algorithm and cross-optimization algorithm (MWOA-CS) by using a new nonlinear convergence factor and nonlinear inertia weights to adjust the exploitation and exploration capabilities. El-Dabah et al. (2022) proposed a non-dominated ranked whale optimization algorithm (NSWOA) to solve single and multi-objective optimal power flow (OPF) problems.

Table 1
Literature Analysis

Literature	Research Background	Type	Solving method	Learning effect	Objective 1	Objective 2
(Zhi & Xu, 2019)	Apparel manufacturing	Flexible Flow Shop Scheduling (FFS)	NSGA-II, εbound	√	Factory profit	Worker satisfaction
(Q. Li et al., 2019)	Construction Project Management	Multi-skilled project scheduling issues	εbound	√	Project Validity	Skill level
(Azizi et al., 2010)	Manufacturing cell	Worker Job Rotation Issues	SAMED-JR, SA, GA	√	Total Delay	/
(F. Liu et al., 2021)	Seru production	Cross-trained worker assignment issues	NSGA-II based modal algorithm, NSGA-II algorithm based on K-means	√	Minimal completion time	Minimal workload imbalance
(R. Liu et al., 2021)	Manufacturing	Allocation of multi-skilled workers considering energy consumption	PT-ECSFR, NSGA-II, MOSA		Total worker cost	Energy
(Shi et al., 2023)	Manufacturing	Consider delay-free shop scheduling for overtime	GASA		Minimal total inventory	Minimal overtime costs
(Tian et al., 2023)	Aerospace industry	Dynamic energy-saving scheduling for multi-variety, low-volume flexible job shop	BDABC		Energy	Processing cost
This paper	Automotive Industry	Scheduling of assembly line workers considering learning effect and flexible labor	ENS-WOA	√	Order delivery time	Total Cost

This paper analyzes the relevant literature as shown in Table 1. Given the research perspectives of the above literature, it aims to provide solutions to the worker scheduling problem in the implementation of C2M transformation and upgrading in the assembly plant of enterprise B. The goal is to improve the overall performance of the production system by considering the learning effect of workers and rationally allocating permanent and temporary workers according to the skill levels of heterogeneous workers. The main contributions of this paper are as follows (1) a learning curve-based model for the dynamic change of operation efficiency of single-skilled workers and multi-skilled workers is proposed; (2) a mathematical model based on the multi-objective optimization problem of worker scheduling is established with the optimization objectives of minimizing worker cost and minimizing order completion time to solve the order-assembly unit-operation-worker series problem; (3) an elite non-dominant sorting whale optimization algorithm (ENS-WOA) is designed to solve the mathematical model, and introduce non-dominant sorting, congestion distance, elite selection strategy and crossover variation in the original whale algorithm to improve the overall performance of the algorithm.

3. Modeling of assembly line worker scheduling considering learning effect and flexible labor

3.1 Modeling of operational efficiency changes

In the De Jong learning curve model, the time required to complete the task decreases gradually as the worker repeats the task and the skills acquired by the worker improve gradually. At the same time, the De Jong model limits the extent to which the task completion time decreases, and the completion time decreases at a slower rate as experience is accumulated (De Jong, 1957). However, regardless of the accumulated experience, the completion time is≥0. This is consistent with the situation of workers on assembly lines in manufacturing. In this paper, the variation of workers' operational efficiency based on the De Jong learning curve model of Eq. (1) is modeled.

$$T_s = T_1(M + \frac{1-M}{s^{\beta}}) \tag{1}$$

T_1 represents the time required for the worker to complete the task in the first round; T_s represents the time required for the s^{th} round; M represents the incompressibility factor, which is related to the task type, $0 \le M \le 1$. In the infinite successive task cycles $T_{\infty} = T_1 M$;β represents the learning ability of the worker ($0 \le \beta \le 1$). In order to describe the relationship between skill level change and task time in detail, two aspects of single-skill tasks and multi-skill tasks were studied separately.

Assume that worker i masters skill s, skill level is l_k^i .The execution time required for single-skill task j at that skill level is $T_{l_k}^i$, then the completion time of single-skill task j is:

$$T_{l_k}^i = T_{l_0}(M + (1 - M) \cdot \varphi^{\beta_i} \cdot t^{-\beta_i}) \tag{2}$$

Here, $\beta_i (0 \le \beta_i \le 1)$ is the learning ability of worker i. β_i is heterogeneous and is related to the worker's work experience and education level. The larger β_i is, the faster the worker learns, and the faster the skill level increases. φ is the coefficient in the worker skill level change model. M is the incompressible factor, which is related to the task type, and the maximum value of task completion time at an infinite cumulative task time is $T_{l_0} M$. T_{l_0} is the operating time at the initial skill level l_0 . The time required for worker i to master skill s at a level from l_0 to level l_k is shown in Eq. (3).

$$t = \varphi \log \beta_i (\frac{1-M}{\frac{T_{l_k}^i}{T_{l_0}} - M}) \tag{3}$$

The time required to upgrade worker i's skill s from skill level l_k to the next skill l_{k+1} is calculated by Eq. (4) as:

$$\Delta_{l_k^i+1} = t_{l_k^i+1} - t_{l_k^i} = \varphi \log \beta_i (\frac{1-M}{\frac{T_{l_k+1}^i}{T_{l_0}} - M}) - \varphi \log \beta_i (\frac{1-M}{\frac{T_{l_k}^i}{T_{l_0}} - M}) = \varphi \log \beta_i (\frac{\frac{T_{l_k}^i}{T_{l_0}} - M}{\frac{T_{l_k+1}^i}{T_{l_0}} - M}) \tag{4}$$

Assume that a single-skill task j requires a skill type and the minimum skill level required for task j is NS_{js} , and the actual completion time T_{ij}^{real} allocated to worker i at that level is calculated as shown in Eq. (5):

$$T_{ij}^{real} = \frac{T_{l_k}^i}{T_{NS_{js}}} T_j \tag{5}$$

where, $T_{NS_{js}}$ is the completion time of task j at skill level NS_{js}.

After a worker completes task j, the worker's skill is updated by introducing a decision variable W_{is}. It indicates whether the cumulative completion time of worker i at skill s of level condition meets the time requirement for promotion to the next level. If it is met, $W_{is} = 1$, otherwise, $W_{is} = 0$. Update the skill level of the worker by Eq. (6).

$$A_s^i = (T_{ij}^{real} + A_s^i) \cdot (1 - W_{is}) \tag{6}$$

$$l_k^i = l_{k+W_{is}}^i \tag{7}$$

If a task j requires more than two types of skills, it is called a multi-skilled task. Assume that the types of skills required for task j are $\{j_{s1}, j_{s2}, ...j_{sS}\}$, and that the requirements for different skills for task j will vary. The minimum level of skill required to complete task j is $\{NS_{j_{s1}}, NS_{j_{s2}}, ...NS_{j_{sS}}\}$. If worker i has all the skills needed to complete task $j\{j_{s1}, j_{s2}, ...j_{sS}\}$, the skill level under each skill is $\{sl_k^i, s = 1, 2, ...S\}$, and the time taken to complete task j at each skill level is $\{T_{1l_k}, T_{2l_k}, ...T_{Sl_k}\}$. $\{T_{1l_0}, T_{2l_0}, ...T_{Sl_0}\}$ is the operating time under both initial skill of l_0. Since the completion of multi-skilled task j requires the participation of 2 and more skills, considering the effect of multiple skills on the worker's completion time, the operation time of task j at the initial skill level l_0 is $NS_j = Max\{NS_{j_{s1}}, NS_{j_{s2}}, ...NS_{j_{sS}}\}$. The minimum level of skill required to complete a multi-skilled task j is $NS_j = Max\{NS_{j_{s1}}, NS_{j_{s2}}, ...NS_{j_{sS}}\}$. And the time to complete the task $T_{l_k}^i$ under a multi-skilled task varies as a function of time t as:

$$T_{l_k}^i = T_{l_0}(M + (1 - M) \cdot \varphi^{\beta_i} \cdot t^{-\beta_i}) \tag{8}$$

The minimum time required for worker i to master skill s from "l_0" level to "l_k" level is:

$$t = \varphi \log \beta_i \left(\frac{1 - M}{\frac{T_{l_k}^i}{T_{1l_0}} - M} \right) \tag{9}$$

The completion time at the minimum skill level required to perform the task is T_j, and the actual completion time T_{ij}^{real} assigned to worker i is calculated as shown in Eq. (10):

$$T_{ij}^{real} = \frac{T_{l_k}^i}{T_{l_j}} T_j \tag{10}$$

where, T_{l_j} is the execution time of task j at the skill level of NS_j.

The model quantifies the effect of learning effect of heterogeneous workers on operational efficiency by improving the De Jong learning curve, and updates workers' skill levels in real time through the learning curve based on the amount of work they complete at each stage.

3.2 Problem Description

The production tasks of a C2M model manufacturing company are based on personalized orders. Before building a mathematical model, the problem needs to be described, as shown schematically in Fig. 1.

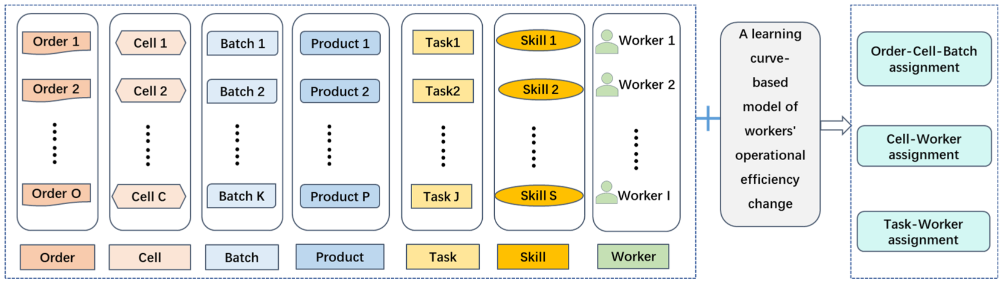


Fig. 1. Personnel assignment chart

The flow of an assembly line consists of j tasks. These tasks form a serial assembly line, where the previous tasks are completed before the later ones can be started and the process cannot be interrupted. Each task requires some specific skills to complete successfully. The workforce consists of N permanent workers and M temporary workers. There are p types of products, o orders, s types of skills, c assembly cells and k batches per assembly cell. Prior to worker allocation, there will be differences in the level of proficiency of workers, considering their prior work experience and level of education. As a result, their learning rates are different, but workers are trained accordingly before they start work. For all the skills required for the job processes in product manufacturing, the worker's initial skill matrix P_{is} and the task-skill correlation matrix Q_{js} are constructed. The dynamic skill level of workers can be obtained from Eq. (7).

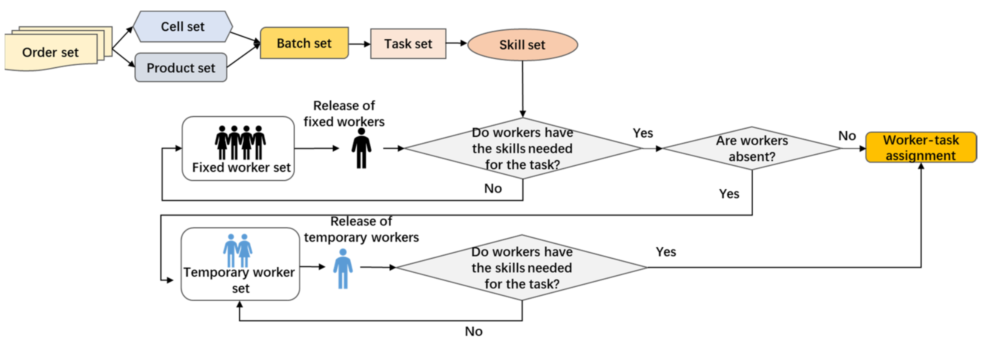


Fig. 2. Worker-task assignment chart under flexible employment strategy

As shown in Fig. 2, permanent workers can maximize their individual capabilities and can be given assignments as long as their skill level meets the requirements of the task. For casual workers, they can only perform job-specific tasks based on the job requirements of the production line and their limited skill level and can experience higher product defect rates. This paper will additionally consider the product quality and cost differences between permanent and temporary workers. Workers are paid a basic base salary and performance pay, with monthly performance pay related to the number of hours a worker operates the line each month; basic pay is related to the type and level of worker and not to the specific task to which they are assigned.

3.3 Establishment of hypothetical conditions

In order to bring the research process closer to the way companies produce in the C2M e-commerce model and. At the same time, to facilitate the solution of the mathematical model, the following assumptions were made to simplify the model:

（1）Assume that the entire manufacturing system is predominantly manual, that the number of workers and their learning capacity are known, and that permanent workers have the potential to operate multiple operations;

（2）Assume that workers maintain a stable operational capability after completing skill acquisition, disregarding the effect of the forgetting curve;

（3）Assume that an order can only be assigned to one order batch sequence in an assembly cell;

（4）Assume that workers are heterogeneous and differ in terms of learning rates;

（5）Assume that the maximum number of tasks that can be performed simultaneously by each multi-skilled worker is 3;

（6）Assume that the ability of a worker to complete a task and the time spent on it depends on the level of skill and the number of skills acquired by the worker;

（7）Assume all casual workers have the same cost of quality loss per unit of time;

（8）Assume that the number of skills required for each task does not exceed3 and that the maximum number of skills acquired by a permanent worker at the same time is 3, with skill levels l_1-l_4;

（9）Assume no blocking stoppages or other time between tasks;

（10）Assume that an order has only one product type.

3.4 Parameter and variable setting

（1）Parameter Setting

i: {$1,2,3\ldots\ldots I$} Worker set

N: Number of permanent workers

M: Number of temporary workers

O: {$1,2,3\ldots\ldots O$} Order set

p: {$1,2,3\ldots\ldots P$} Product set

j: {$1,2,3\ldots\ldots J$} Task set

s: {$1,2,3\ldots\ldots S$} Skill set

l_k: $\left\{l_1, l_2, \ldots\ldots l_k\right\}$ Skill level set

c: {$1,2,3\ldots\ldots C$} Assembly cell set

k: {$1,2,3\ldots\ldots K$} Batch set

C_d: Quality loss cost per hour

s_p: Lead time for product p

T: Worker hours per shift

Q_o: Demand for order O

D_o: Delivery period for order O

H_i: Number of skills possessed by worker i

sl_0^i : Initial skill level of worker i mastery skill s

sl_k^i : Skill level of worker i mastery skill s

$$z_{js} = \begin{cases} 1,\text{if task } j \text{ requires the skill } s \\ 0,\text{otherwise} \end{cases}$$

$$x_{is} = \begin{cases} 1,\text{if worker } i \text{ acquires the skill } s \\ 0,\text{otherwise} \end{cases}$$

$$H_{ock} = \begin{cases} 1,\text{if the order } o \text{ is assigned to batch } k \text{ of assembly cell } c \\ 0,\text{otherwise} \end{cases}$$

$$x_i = \begin{cases} 1, \text{ if worker } i \text{ is a single-skilled worker} \\ 0, \text{ otherwise} \end{cases}$$

$$y_i = \begin{cases} 1, \text{ if worker } i \text{ is a multi-skilled worker} \\ 0, \text{ otherwise} \end{cases}$$

$$z_i = \begin{cases} 1, \text{ if worker } i \text{ is a temporary worker} \\ 0, \text{ otherwise} \end{cases}$$

$I_c^{\min}$: Minimum number of workers that can be assigned to assembly cell c

$I_c^{\max}$: Maximum number of workers that can be assigned to assembly cell c

NS_{js} : Minimum skill level for completion of task j worker mastery skill s

sl_k : Level of skill s, $sl_1 = 1, sl_2 = 2, sl_3 = 3, sl_4 = 4$

$$z_{op} = \begin{cases} 1, \text{ if the product type of the order } o \text{ is } p \\ 0, \text{ otherwise} \end{cases}$$

$$y_{ip} = \begin{cases} 1, \text{ if worker } i \text{ can assemble product } p \\ 0, \text{ otherwise} \end{cases}$$

β_i : The worker i's learning rate

φ : Coefficient

T_{ij}^{real} : The actual time for worker i to complete task j

T_o : Lead time for order O

$$T_o = \begin{cases} 0, \text{if } H_{ock} = 1, k = 1 \text{ or } H_{ock} = H_{o'c(k-1)} = H_{o'c(k+1)}, z_{op} = z_{o'p} \\ s_P, \text{otherwise} \end{cases}$$

C_{sk} : Cost of man-hours with skill s and level l_k

C_a : Daily base salary for single -skilled permanent workers

C_b : Daily base salary for multi-skilled permanent workers

C_t : Daily base salary for temporary workers

$$u_{ijs} = \begin{cases} 1, \text{ if worker } i \text{ has the minimum skill required for task } j \\ 0, \text{ otherwise} \end{cases}$$

Number of workers in assembly cell c who can execute order O: $\sum_{i=1}^{I} f_{oc} z_{op} y_{ip}$

$$P_{is} = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1s} \\ \dots & \dots & \dots & \dots \\ P_{i1} & P_{i2} & \dots & P_{is} \end{bmatrix} \qquad Q_{js} = \begin{bmatrix} Q_{11} & Q_{12} & \dots & Q_{1s} \\ \dots & \dots & \dots & \dots \\ Q_{j1} & Q_{j2} & \dots & Q_{js} \end{bmatrix}$$

（2）Decision Variables

$$y_{ij}:\begin{cases}1, & \text{if worker } i \text{ is assigned to complete task } j\\ 0, & \text{otherwise}\end{cases} \qquad f_{oc}=\begin{cases}1, & \text{if order } o \text{ is assigned to assembly cell } c\\ 0, & \text{otherwise}\end{cases}$$

$$x_{ic}=\begin{cases}1, & \text{if worker } i \text{ is assigned to assembly cell } c\\ 0, & \text{otherwise}\end{cases} \qquad x_{o}=\begin{cases}1, & \text{if order } o \text{ can be picked up}\\ 0, & \text{otherwise}\end{cases}$$

3.5 Determination of the objective function

$$\min\left(\sum_{s=1}^{S}\frac{C_{sk}}{3600}+\sum_{i=1}^{I}\frac{x_iC_a+y_iC_b+z_iC_t}{3600T}\right)\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{o=1}^{O}y_{ij}T_{ij}^{real}Q_o+\sum_{o=1}^{O}T_o\right)+\sum_{o=1}^{O}\sum_{j=1}^{J}\sum_{i=1}^{I}\frac{T_{ij}^{real}z_iC_dQ_o}{3600}) \tag{11}$$

$$\min\ \left(\sum_{i=1}^{I}\sum_{o=1}^{O}\sum_{j=1}^{J}y_{ij}T_{ij}^{real}Q_o+\sum_{o=1}^{O}T_o\right) \tag{12}$$

subject to

$$\sum_{i=1}^{N+M}y_{ij}=1,\forall i\in I \tag{13}$$

$$\sum_{j=1}^{J}y_{ij}=1,\forall i\in M \tag{14}$$

$$1\le\sum_{j=1}^{J}y_{ij}\le 3,\forall i\in N \tag{15}$$

$$1\le\sum_{s=1}^{S}x_{is}\le 3,\forall i\in I \tag{16}$$

$$x_{is}\in\{0,1\},\forall i\in I,s\in S \tag{17}$$

$$z_{js}\in\{0,1\},\forall j\in J,s\in S \tag{18}$$

$$y_{ij}\in\{0,1\},\forall i\in I,j\in J \tag{19}$$

$$sl_k,k\in\{1,2,3,4\} \tag{20}$$

$$\min\ (sl_k^i y_{jsl_k^i})\ge NS_j \tag{21}$$

$$\sum_{c=1}^{C}\sum_{k=1}^{K}H_{ock}=x_o,\forall o \tag{22}$$

$$I_c^{\min}\le\sum_{i=1}^{I}x_{ic}\le I_c^{\max},\forall c \tag{23}$$

$$\sum_{c=1}^{C}x_{ic}=1,\forall i \tag{24}$$

$$\sum\nolimits_{o=1}^{O}z_{op}=1,\forall p \tag{25}$$

The objective function (11) minimizes the sum of the total costs of permanent and temporary workers in the case of fulfilling orders. Three components are included: hourly pay, basic base pay, and quality loss pay. The objective function (2) ensures that all orders have the shortest completion time. Constraint (13) ensures that there are workers to complete each task and each task can only be assigned to one worker. Constraint (14) ensures that each temporary worker can only be assigned one task. Constraint (15) indicates that the number of permanent workers can be assigned tasks is 1-3. Constraint (16) requires workers to master at least one skill and at most 3 skills; Constraints (17), (18), (19) ensure the corresponding 0-1 variables. Constraint (20) ensures that the skill level of workers is l_1-l_4. Constraint (21) ensures that only workers whose skill level meets the minimum requirements needed for the operation can be assigned to that operation process. Constraint (22) indicates that accepted orders can only be assigned to one order batch sequence in the assembly cell, and rejected or changed orders will not be scheduled to any order batch sequence. In order to balance the workload among workers, constraint (23) ensures that the number of workers in each assembly unit is within a certain interval. Constraint (24) indicates that each worker can be assigned to only one assembly cell. Constraint (25) indicates that an order contains only one product type. Constraint (26) indicates a logical constraint.

4. Design of ENS-WOA

4.1 Flow of the original whale optimization algorithm

The original whale optimization algorithm consists of 3 operational steps: encircling the prey, bubble net attack and random prey search, and the pseudocode is shown in Table2. The whale population represents multiple potential solutions to the optimization problem, each of which is also referred to as a “search agent”. The ultimate goal is to find the optimal search agent location for the objective function. Since the original whale optimization algorithm is proven feasible on the benchmark function, there is much room for research in solving the practical problem of this paper. In this chapter, a unique whale optimization algorithm will be designed based on the assembly line worker scheduling problem.

Table 2
The WOA pseudocode

WOA	
Input	Population size N, Parameter a, A, l, C, p, Crossover probability p_c ,Mutation probability p_m , Maximum number of iterations t_{Max}
Output	Optimal solution x^*, Fitness value $Fit(f(x))$
	Initialize the population $X_i = (X_i^1, X_i^2, ...X_i^n)$, $i = 1,2,\ldots\ldots N$. Set the maximum number of iterations t_{Max}
	Coding of whale populations X_i ,Calculate the fitness value $Fit(f(x))$
	Find the best search agent location x^*
	While t< t_{Max}
	for i=1,2,……n
	Update a, A, l, C, p
	if p<0.5
	if $\lvert A \rvert < 1$
	Update the current search agent location by shrink-wrapping mechanism
	else
	Select a random search agent X_{rand}
	Update the location of the current search agent via the search method
	end if
	else
	Update search agent location via spiral update method
	end if
	Calculate the fitness value for each search agent
	t=t+1
	end while
	Output the best search agent location x^*

4.1.1 Surrounding the prey

The humpback whale leader determines the prey position by continuously iterating updates, and the rest of the whales follow to update to the best position. Since the best position in the search space cannot be predicted in advance, the algorithm sets the current best whale position as the position closest to the target prey. After specifying the best search agent position, other search agents try to approach the best search agent and keep moving to update its position to gradually surround the prey.

$$\vec{x}(t+1)=\vec{x}^{*}(t)-\vec{A}\cdot\vec{D} \tag{27}$$

$$\vec{D}=\left|\vec{C}\cdot\vec{x}^{*}(t)-\vec{x}(t)\right| \tag{28}$$

where, t denotes the current number of iterations; $\vec{A}$ and $\vec{D}$ are the coefficient vector; $\vec{x}^{*}(t)$ is the best position of the humpback whale so far; $\vec{x}(t)$ is the current position of the individual whale; $\vec{x}(t+1)$ is the position of the individual whale in the $t+1^{th}$ iteration; D denotes the distance; and the dot operator denotes element-by-element multiplication.

$$\vec{A}=2\vec{a}\cdot\vec{r}_1-\vec{a} \tag{29}$$

$$\vec{C}=2\vec{r}_2 \tag{30}$$

where, $\vec{a}$ is the convergence factor. As t increases, $\vec{a}$ decreases linearly from 2 to 0. The expression is $\vec{a}=2-(2t/t_{Max})$,where t_{Max} is the maximum number of iterations; $\vec{r}_1$, $\vec{r}_2$ are the random vector uniformly distributed in the range of $[0,1]$, and $\vec{C}$ is the swing factor.

4.1.2 Bubble net attack

During predation, humpback whales swim around their prey in a gradually shrinking circle. According to the value of the probability factor p ($0\le p\le 1$), the whale has two ways of movement. Assuming that the probability of an individual whale randomly choosing one way to complete position updating is 0.5. When $p<0.5$, it enters the contracting encirclement phase; when $p\ge 0.5$, it enters the spiral updating position phase.

（1）Shrink to surround the prey

The shrink-wrapped prey is similar to the global search prey. However, the difference is that A takes values in the range $[-1,1]$. Shrinking the enclosed prey is mainly achieved by reducing the $\vec{a}$ in Eq. (29). When choosing the best whale position, the optimal solution in the previous iteration is considered as the reference solution to promote the remaining whales to be close to the best candidate whale position. As shown in Fig. 3, (X,Y) is the original position of the whale, and (X^{*},Y^{*}) represents the current best position of the whale.

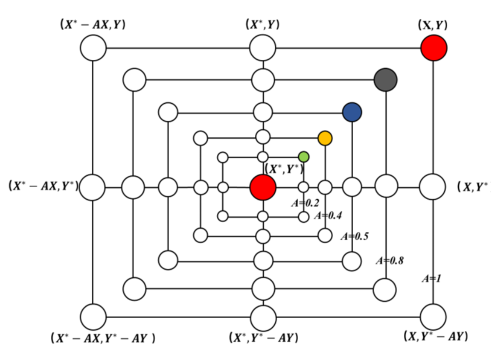


Fig. 3. Two-dimensional diagram of the shrink-wrapping mechanism

（2）Spiral Position Update

It refers to the process of constructing 1 logarithmic spiral curve based on the current whale position and the best whale candidate position, and the individual whale slowly approaches the best whale position and exhales bubbles to catch prey. Eq. (31) describes the position transformation of the whale spiral movement.

$$\vec{x}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l)\vec{x}^*(t) \tag{31}$$

$$\vec{D}' = \left| \vec{x}^*(t) - \vec{x}(t) \right| \tag{32}$$

where, $\vec{D}'$ is the distance between the whale individual and the current best whale individual at the t^{th} iteration. $\vec{x}(t+1)$ is the position of the individual whale at the $t+1^{th}$ iteration. l is a random number with controlled values and b is the logarithmic spiral shape constant.

In summary, the mathematical model of a humpback whale attacking its prey with a bubble net is as follows:

$$\vec{x}(t+1) = \begin{cases} \vec{x}^*(t) - \vec{A} \cdot \vec{D}, p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l)\vec{x}^*(t), p \geq 0.5 \end{cases} \tag{33}$$

P is the random number generated by uniform distribution between [0,1].

4.1.3 Random prey search

In addition to using the bubble net attack method to find prey, humpback whales also use a random search feeding method, as shown in Fig. 4. Random search is achieved by controlling the change of A value. When $|A| \geq 1$, the whale swims outside the constricted envelope and keeps moving in the direction of the random whale individual thus expanding the search range away from the current target prey and moving to other better prey. The mathematical model is as follows:

$$\vec{x}(t+1) = \vec{x}_{rand}(t) - \vec{A} \cdot \vec{D} \tag{34}$$

$$\vec{D} = \left| \vec{x}_{rand}(t) \cdot C - \vec{x}(t) \right| \tag{35}$$

where, $\vec{x}_{rand}(t)$ is a random selection of whales from the current population.

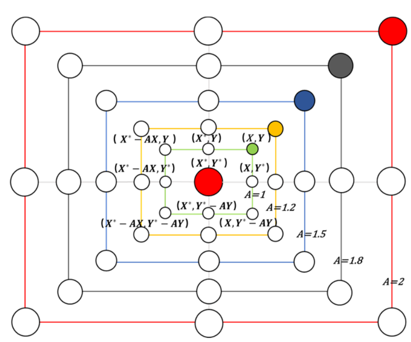


Fig. 4. Two-dimensional diagram of random prey search mechanism

4.2 Design of ENS-WOA

Similar to other population intelligence optimization algorithms such as particle swarm optimization algorithms and difference optimization algorithms, whale optimization algorithms must also maintain a balance between global and local search. However, the whale optimization algorithm currently has problems, such as: when the dimensionality of the optimization problem increases, the WOA will lose diversity and mature prematurely at a later stage of evolution. To address the above problems, an improved whale optimization algorithm is designed from the following aspects:

1. Combine the advantages of the original WOA with the non-dominated sorting technique to make full use of the advantages of both. The fast convergence property of WOA and the effectiveness of non-dominated sorting for solving multi-objective

optimization problems are very useful for solving complex and large iterative computations and can significantly increase the performance of WOA.

2. The crowding distance mechanism is incorporated into the original WOA. The more crowded individuals are able to enter the next generation population in preference, and the rest are eliminated to maintain diversity. Meanwhile, the elite selection strategy is incorporated to improve the quality of the solution set.

3. The introduction of crossover operator and variational operator can both global random search capability and prevent premature convergence.

4.2.1 Coding of whale locations

A three-level chromosome real number coding method was developed in order to make chromosomes contain the association information between orders, cells, workers and tasks at the same time, as shown in Fig. 5.

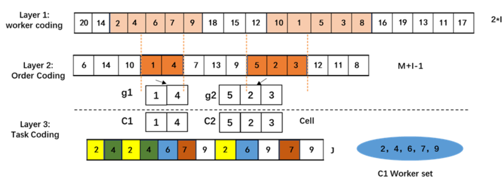


Fig. 5. Three-level chromosome real number coding

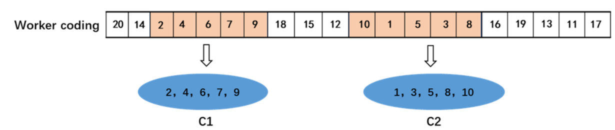


Fig. 6. Worker coding mechanism

First, determine how many assembly cells to build and to which assembly cell each worker is assigned. The first level of the chromosome gene value contains 2 × I numbers, where I is the number of workers. The first segment indicates the worker-assembly cell assignment: 2 × I numbers are arranged in random combinations in order to determine the number of assembly cells. Where, "1,2,......I" denotes the worker number and the number greater than I denotes the assembly cell interval to obtain the worker-assembly cell assignment. Take O=5, C=2 and I=10 as an example, as shown in Fig. 6. The first layer of the chromosome contains 20 genes, and the 10 workers are finally assigned to 2 assembly cells according to the coding method. $C_1 = \{I_2, I_4, I_6, I_7, I_9\}$, $C_2 = \{I_1, I_3, I_5, I_8, I_{10}\}$

Next, determine which assembly cell each order is assigned to and the order's batch order within the assembly cell. The second layer is coded in a sequential manner, with M orders and I workers as an example. The second layer of chromosomal gene values has a total of M+I-1 genes, dividing the orders into G groups. The number 1,2,...,M represents the order number, and the number M+I-1 represents the redundancy code, which is used to split the cell. According to the coding result, if G≤C, group g and assembly cell c can achieve one-to-one correspondence; if G>C, the excess group g is then assigned to each assembly cell c in turn. In the following, two examples are explained, as shown in Fig. 7 and Fig. 8.

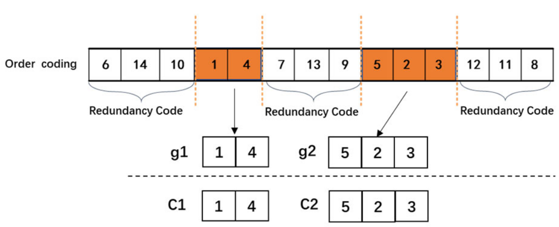


Fig. 7. Example of coding for 5 orders, 10 workers, $G \leq C$

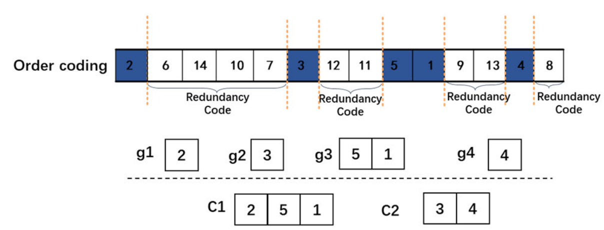


Fig. 8. Example of coding for 5 orders, 10 workers, $G>C$

M=5, I=10, the number of genes on the chromosome was 14, and the 5 orders were divided into 2 groups. $g_1 = \{O_1, O_4\}$, $g_2 = \{O_5, O_2, O_3\}$ .Since $C = 2$ and $G = C$, it follows from the encoding that $C_1 = \{O_1, O_4\}$, $C_2 = \{O_5, O_2, O_3\}$.

M=5, I=10, the number of genes on the chromosome was 14, and the 5 orders were divided into 4 groups. $g_1 = \{O_2\}$, $g_2 = \{O_3\}$, $g_3 = \{O_5, O_1\}$, $g_4 = \{O_4\}$ .Since $C = 2$ and $G > C$, it follows from the encoding that $C_1 = \{O_2, O_5, O_1\}$, $C_2 = \{O_3, O_4\}$.

Finally, the allocation relationship between workers and tasks in the cell is determined. Based on the results of the order-assembly cell allocation, the products to be completed in each cell and their tasks can be derived. The number of genes in the third tier is equal to the number of tasks in each cell. As shown in Fig. 9, the assembly cell $C_1 = \{I_2, I_4, I_6, I_7, I_9\}$, then the numbers 2, 4, 6, 7, 9 are randomly coded.

Fig. 9. Task code of C_1

4.2.2 Whale population initialization

After finishing encoding and decoding, the initial solutions of whale populations are generated based on the encoding rules. In this paper, we use a random generation method to generate two initial populations P_A and P_B of size N. The current evolutionary generation Gen=1 is set, and the position of each individual whale represents one solution of the objective function, i.e., an allocation scheme.

In the whale population, each individual consists of an n-dimensional random vector with the following equation:

$$X_i = x_{\min} + rand(1, D)(x_{\max} - x_{\min}) \tag{36}$$

where, $X_i = (X_i^1, X_i^2, ...X_i^n), i = 1, 2, ...N$ .N is the population size and D is the dimensionality of the objective function. $x_{\max}$ and $x_{\min}$ are the upper and lower bounds. $rand(1, D)$ is a random number within [0,1]

4.2.3 Fitness function

Both objective functions in this paper are minimized, and the following conversion method can be used, as shown in Eq. (37).

$$Fit(f(x)) = \begin{cases} f(x), & \text{when the objective function is to find the maximum value} \\ \dfrac{1}{f(x)}, & \text{when the objective function is to find the minimum value} \end{cases} \tag{37}$$

4.2.4 Non-dominant sorting

The non-dominated ranking is based on the domination level to rank the Pareto optimal solutions. The multi-objective WOA based on Pareto ranking has one main point: the algorithm seeks the set of Pareto solutions instead of one Pareto solution. Therefore, the crowding degree operator needs to be added to ensure the diversity of the population. The flow is shown in Fig. 10.

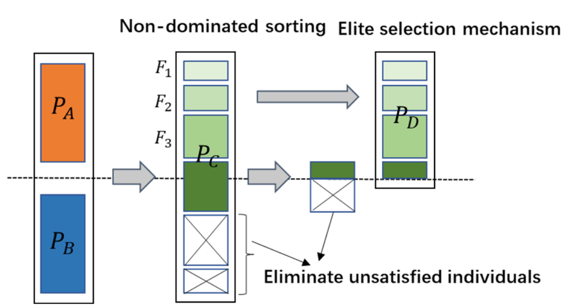


Fig. 10. Workflow of Non-dominant sorting

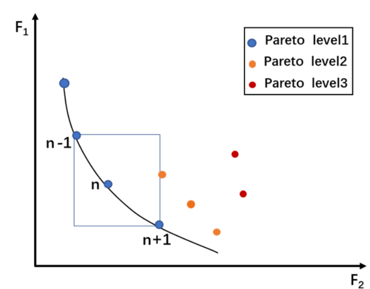


Fig. 11. Congestion diagram

Select individuals to be retained by ranking F_{rank} and crowding distance n_d . When the ranks of two individuals are different, the higher ranked individual is selected; when the ranks of two individuals are the same, the individual with the higher n_d is selected. It is shown in Fig. 11.

$$n_d = \sum_{n=1}^{m} (\left| f_j^{n+1} - f_j^{n-1} \right|) \tag{38}$$

where, n_d denotes the congestion at point n, f_j^{n+1} denotes the j^{th} objective function value at point $n+1$, and f_j^{n-1} denotes the j^{th} objective function value at point n-1. The elite selection strategy can make the algorithm run faster. It can also expand the search space and prevent the satisfactory solutions already found from being discarded, thus improving the quality of the population.

4.2.5 Crossover and variation

The traditional crossover operator is a fixed probability with a small search space. In this paper, we introduce the arithmetic crossover operator, which takes two matched waiting crossover individuals in the population and obtains two new individuals respectively by corresponding mathematical operations, as shown in Eq. (39):

$$\begin{cases} X'_a = \alpha X_b + (1-\alpha) X_a \\ X'_b = \alpha X_a + (1-\alpha) X_b \end{cases} \tag{39}$$

where, X_a and X_b are the two individuals in the population waiting to cross over. X'_a and X'_b are two new individuals. α is a uniform random number of $[0,1]$.

Variation is used as an auxiliary way of crossover operations to prevent the generation of local optima. The common variation methods for real number coding include uniform variation, non-uniform variation, Gaussian variation, normal variation, adaptive variation and boundary variation. The uniform variation used in this paper is to replace the original gene with a random number f uniformly distributed in some range with a small probability.

If x_i is the value of the original gene location, $i = 1, 2......n$ .The maximum value of x_i that can be taken is U_i and the minimum value is L_i. The value of x'_i after uniform variation is as follows:

$$x'_i = \begin{cases} x_i + \beta \times (U_i - x_i) & f > 0.5 \\ x_i - \beta \times (x_i - L_i) & f \le 0.5 \end{cases} \tag{40}$$

where, β is a uniform random number between $[0,1]$.

4.2.6 Process of ENS-WOA

As shown in Fig. 12, ENS-WOA can be achieved by the following steps:

Step 1: Initialize the system parameters *a*, *A*, *l*, *C*. Randomly generate 2 initial population of whales with Worker *No.*, Order *No.* and Task *No.* attached to two chromosomes, both with population size *N*. The crossover probability is p_c , the variation probability is p_m, and the maximum number of iterations is t_{Max} . Define the number of iterations t=0.

Step 2: Input the parameters of the worker scheduling model, specify the fitness function, and calculate the fitness value for each search agent (whale).

Step 3: The operations of non-dominated sorting, arithmetic crossover, uniform variation, calculation of crowding distances and elite selection are performed on population P_A .

Step 4: In the initial population P_B , the fitness value is calculated and the optimal agent position x^* is determined. Use Eq. (27) and Eq. (34) to update the location of the search agents (whales)

Step 5: If the conditions of Step 4 are not satisfied, then use Eq. (33) to spiral update the location of the search agent (whale) to find the global best agent x^*. Record the fitness value of the search agent (whale) in population P_B at this time.

Step 6: The optimized population P_A in Step 3 is merged with the WOA-optimized population P_B in Step 5 to form a new population P_C with a population size of 2*N*.

Step 7: Take a fast non-dominant sort on the new population P_C and record the best for each generation. Increase the number of population generations by 1. Repeat Steps 4 - 7.

Step 8: The crowding distance of the new population P_C is calculated and an elite selection strategy is used to filter individuals into the new population P_D until the population P_D is full (the number of individuals reaches *N*) .The remaining solutions are eliminated.

Step 9: Determine whether ENS-WOA runs to the t_{Max} . If “yes”, output the Pareto solution sets. Otherwise, skip to Step 3.

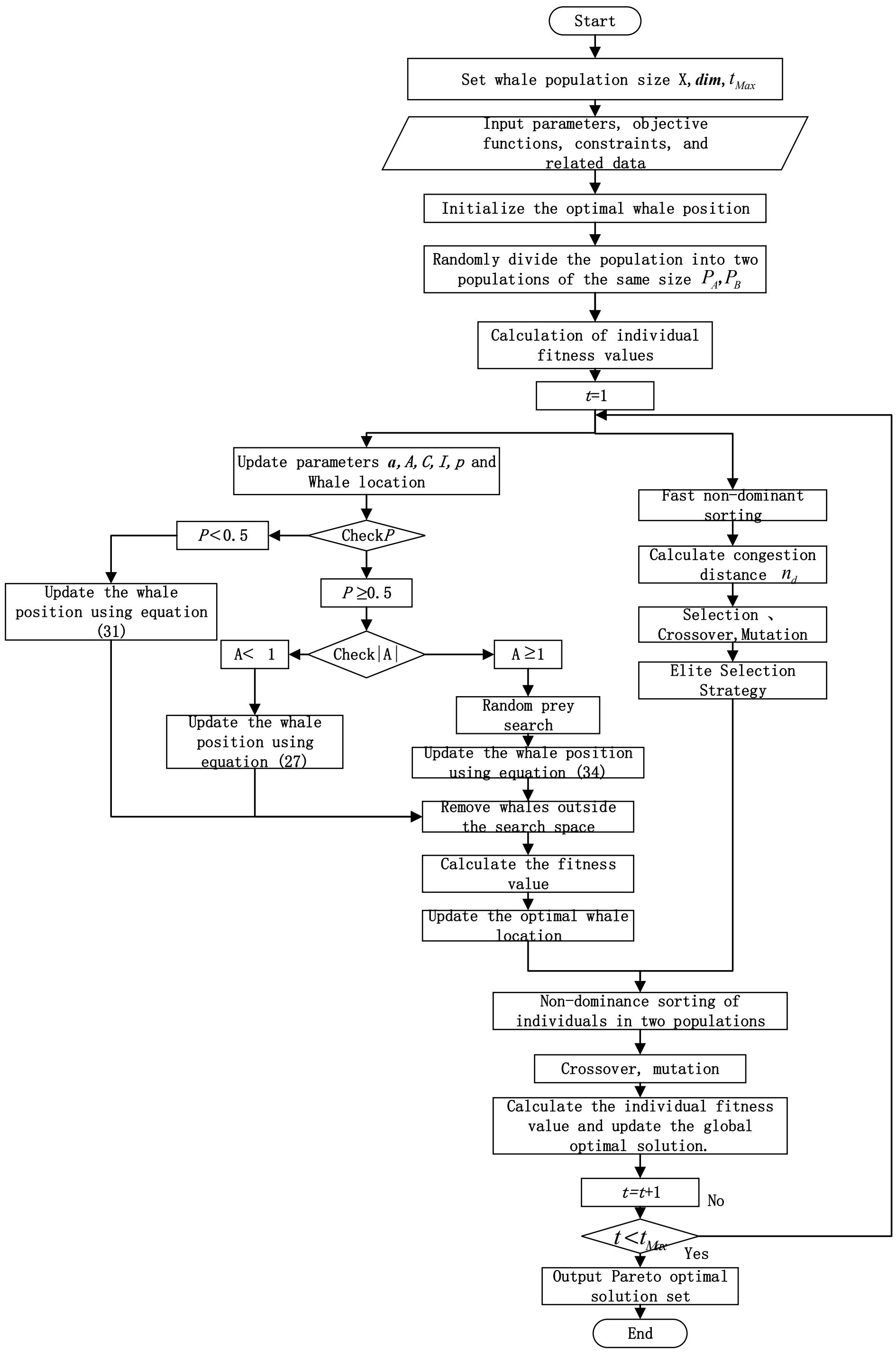


Fig. 12. Flowchart of ENS-WOA algorithm

4.3 Algorithm verification

4.3.1 Evaluation Metrics for Algorithms

The performance of a multi-objective optimization algorithm is usually evaluated in terms of three aspects: running time, occupied memory and quality of the solution. The quality of the solution includes the convergence, diversity and distribution of the algorithm.

（1）Convergence and diversity - Inverse Generation Distance (*IGD*)

The *IGD* metric uses the real Pareto solution set as a reference to calculate the average of the distances between each reference point and the nearest non-dominant solution in a known Pareto front(Van Veldhuizen & Lamont, 2000).

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^{\,2}}}{n} \tag{41}$$

$$d_i = \min(\sum_{k=1}^{m} \left| f_m^i - f_m^j \right|) \tag{42}$$

where, n is the number of solutions in the true pareto front, d_i is the Euclidean distance between the i^{th} solution in the pareto front and the closest solution in the solution set, and m is the number of objective functions. From Eq. (41), it can be seen that the smaller the value of d_i, the smaller the value of *IGD*, and the better the convergence and diversity of the algorithm.

（2））Distribution—Spacing（*Sp*）

Spacing (*Sp*) estimates whether the set of Pareto front solutions is uniformly distributed (Schott, 1995), and the formula is as follows:

$$Sp = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\bar{d} - d_i)^2} \tag{43}$$

where, $i, j = 1, ...n$. d_i is the Euclidean distance between the i^{th} solution in the pareto front and the closest solution in the solution set, and m is the number of objective functions. $\bar{d}$ is the average of all d_i. The smaller the value of *Sp*, the better the uniformity of the algorithm.

4.3.2 Classical arithmetic testing and comparative analysis

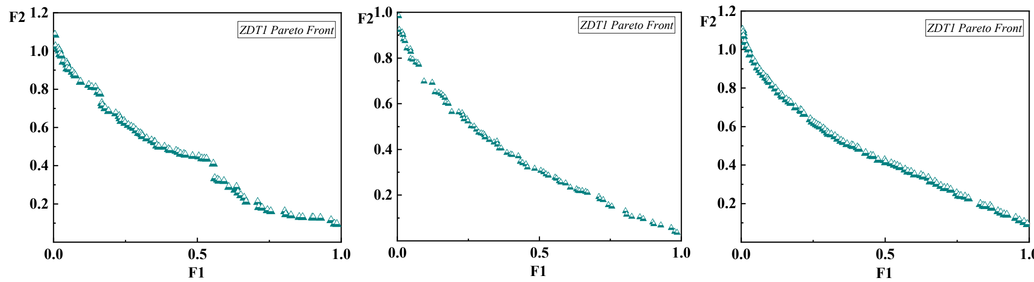
To evaluate the performance of the proposed ENS-WOA to solve the personnel assignment problem, it was coded in MATLAB R2016b and run on a computer configured with an Intel(R) Core(TM) i5-12500H (CPU)、frequency of 2.50 GHz, 16 GB RAM and Windows 11 operating system. Tests were performed using ZDT1 and ZDT2 from the ZDT family of test functions as shown in Eq. (44) and Eq. (45) (Zitzler et al., 2000). Where, the leading edge of ZDT1 is concave and the leading edge of ZDT2 is convex. Also, the performance of the comparative ENS-WOA is evaluated using the WOA, NSGA-II algorithm as a control.

$$ZDT1 = \begin{cases} \min f_1(x_1) = x_1 \\ \min f_2(x) = g(1 - \sqrt{f_1/g}) \\ g(x) = 1 + 9\sum_{i=2}^{n} x_i / (n-1) \\ S.T. 0 \le x_i \le 1, i = 1, 2, ...n \end{cases} \tag{44}$$

$$ZDT2 = \begin{cases} \min f_1(x_1) = x_1 \\ \min f_2(x) = g(1 - (f_1/g)^2) \\ g(x) = 1 + 9\sum_{i=2}^{n} x_i / (n-1) \\ S.T. 0 \le x_i \le 1, i = 1, 2, ...n \end{cases} \tag{45}$$

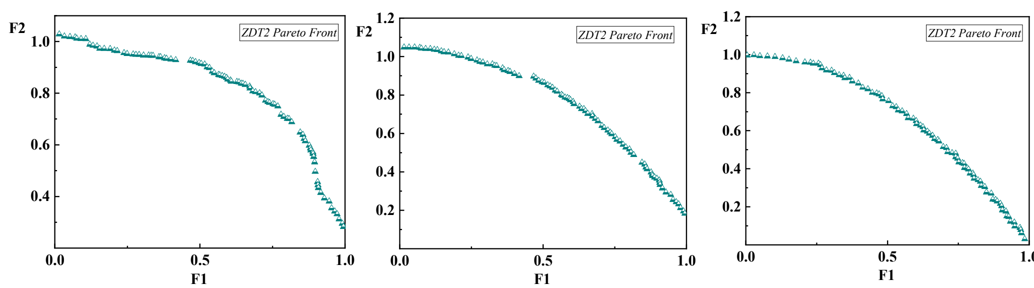
The comparison of algorithms requires setting uniform parameters. The population size is 200, the dimension of the test function dim is 50, the maximum number of iterations is 500, the crossover probability of the ENS-WOA and NSGA-II algorithms is set to 0.8, and the variance probability is set to 0.05. $l=[-1,1]$, $b=1$.15 sets of experiments are conducted for each function, for a total of 30 sets of experiments.

Fig. 13 and Fig. 14 show the Pareto solution sets obtained by the three algorithms for the test functions ZDT1 and ZDT2 after 30 sets of experiments, respectively. The running time and other evaluation metrics of ZDT1 and ZDT2 under different algorithms are shown in Table 3.



a） WOA　　（b） NSGA-II　　（c）ENS-WOA

Fig. 13. ZDT1 Pareto frontier of three algorithms



（a） WOA　　（b） NSGA-II　　（c） ENS-WOA

Fig. 14. ZDT2 Pareto frontier of three algorithms

Table 3
Evaluation metrics of test functions ZDT1 and ZDT2 under three algorithms

Evaluation Metrics	WOA		NSGA-II		ENS-WOA	
	ZDT1	ZDT2	ZDT1	ZDT2	ZDT1	ZDT2
Running time/s	10	11	7	8	4	8
Number of solutions	94	95	85	93	92	89
Sp	0.005	0.0031	0.00358	0.00195	0.00162	0.00093
IGD	0.0034	0.0022	0.00244	0.00202	0.00183	0.0008

The results of ZDT1 and ZDT2 show that the WOA and NSGA-II algorithms obtain larger values of *Sp*, *IGD* for the solution set, indicating that the combined performance of WOA and NSGA-II is worse than ENS-WOA. ENS-WOA has better convergence, distribution and uniformity. However, the benchmark case experiment does not clearly show the advantage of the operational efficiency of the ENS-WOA algorithm.

5. Case Study

Company B is always pursuing innovation and striving to achieve the transformation to C2M. The company's main products include wiper systems, airbags, water pumps, fans, and window crank motors that are suitable for different car models. In recent years, with the popularity of customized cars, the production of components has also faced a huge change. Multi-species and small-lot custom orders have become mainstream, and single-piece personalized orders are increasing year by year. In addition, customers are demanding higher and higher delivery periods for their products. Therefore, companies need to make transformation and adjustment to achieve flexible production. Take pump products as an example, the assembly process of different models of pumps is basically the same, mainly including rotor assembly, stator assembly and shell assembly. The product is finished on 1 conveyor belt and other workbenches in Company B, as shown in Fig. 15. Due to company secrecy, only the code is shown. The operation method of the assembly line is "manual operation + semi-automatic operation", except for ST4200, ST4300, ST4400, ST6500, ST6600 and ST6700 processes, which are semi-automatic operations, all other processes are manual operations. The operation flow is shown in Table 4.

Table 4
Information of the task of all models of water pumps

Worker No.	Task No.	Task code	Worker/equipment standard processing time (sec/piece)
I_1	1	ST2000	20.9
	2	ST2100	8.0
I_2	3	ST2200	11.4
	4	ST2300	18.1
	5	ST2400	9.8
I_3	6	ST2550	15.3
	7	ST2700	15.1
I_4	8	ST4100	17.0
	9	ST4200	22.1
I_5	10	ST4300	12.0
	11	ST4400	12.2
I_6	12	ST6000	15.2
I_7	13	ST6100	21.9
	14	ST6200	6.0
I_8	15	ST6251	16.9
	16	ST6280	10.8
	17	ST6300	20.6
I_9	18	ST6400	22.1
	19	ST6500	18.6
I_{10}	20	ST6600	17.6
	21	ST6700	12.0
I_{11}	22	ST6750	12.8
	23	ST6800	7.5
I_{12}	24	ST6900	20.9
I_{13}	25	ST7000	18.2
I_{14}	26	ST7100	23.0
	27	ST7200	15.7
I_{15}	28	ST7300	12.4
	29	ST7400	12.2

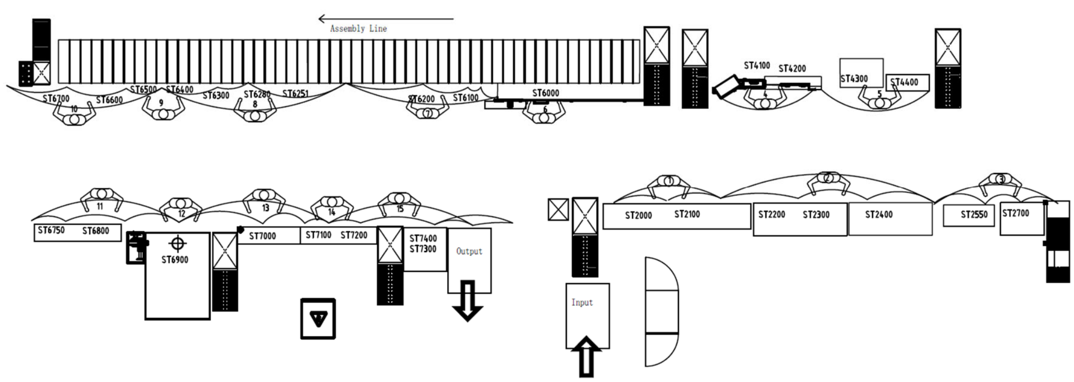


Fig. 15. Company B pump assembly line

At present, the pump assembly line has 15 Permanent workers per shift. The workers are trained to work in a unified way, and the production is organized in a single shift system. The workers work five days a week, and the working hours are 10h per shift. The workers' salary is composed of base salary and overtime pay. The cost per unit work hour is the same for all workers, which is ￥10/h. If overtime production is performed, the overtime pay per unit time is 1.5 times of the normal pay. However, the relationship between quality loss and workers' salary is not considered.

5.1 Determination of workers' initial operating efficiency level

First, the task-skill correlation matrix Q_{js} is delineated in detail based on standard work instructions. The technical experts and assembly line leaders are arranged to measure the skill mastery of workers in the production line. The learning rate of workers β_i is set, and the matrix of workers' mastery of initial skills Q_i is developed. The judgment criteria relied upon in filling in the data are comprehensive, including education level, worker skill competition test results, work experience, and skill certificates. The status is "1" and "0". "1" means mastered and "0" means not mastered, and each element is marked with Q_i, which indicates the mastery of skill s by employee *i*. Table 5 describes the worker's mastery of the initial skill matrix P_{is}. The task-skill association matrix Q_{js} is shown in Fig. 16.

Table 5
Worker Mastery Initial Skills Matrix P_{is}

	S_1	S_2	S_3	S_4	S_5	S_6	Total
I_1	1	0	1	0	1	0	3
I_2	0	0	1	1	0	1	3
I_3	1	0	1	0	0	1	3
I_4	0	1	1	0	0	0	2
I_5	0	0	1	0	1	0	2
I_6	1	1	0	0	0	1	3
I_7	0	0	0	0	1	1	2
I_8	0	0	1	0	1	1	3
I_9	1	1	0	0	0	1	3
I_{10}	1	1	0	1	0	0	3
I_{11}	0	0	1	0	0	1	2
I_{12}	0	1	1	0	0	0	2
I_{13}	0	0	0	1	1	1	3
I_{14}	1	0	1	0	1	0	3
I_{15}	1	1	1	0	0	0	3
Total	7	6	10	3	6	8	—

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}
S_1	1		1	1		1		1			1		1		
S_2		1	1	1		1		1		1		1			1
S_3													1		1
S_4					1		1		1	1					
S_5				1											
S_6					1		1		1		1			1	
	J_{16}	J_{17}	J_{18}	J_{19}	J_{20}	J_{21}	J_{22}	J_{23}	J_{24}	J_{25}	J_{26}	J_{27}	J_{28}	J_{29}	
S_1	1			1	1	1		1			1			1	
S_2		1					1		1		1	1	1		
S_3		1					1	1							
S_4			1	1					1				1		
S_5			1			1				1		1			
S_6			1							1					

Fig. 16. Task-skill correlation matrix Q_{js}

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}
S_1	l_1		l_1	l_1		l_1		l_2			l_1		l_1		
S_2		l_2	l_2	l_2		l_1		l_1		l_1		l_1			l_1
S_3													l_2		l_2
S_4					l_2		l_1		l_1	l_1					
S_5				l_1											
S_6					l_1		l_1		l_2		l_1			l_1	
	J_{16}	J_{17}	J_{18}	J_{19}	J_{20}	J_{21}	J_{22}	J_{23}	J_{24}	J_{25}	J_{26}	J_{27}	J_{28}	J_{29}	
S_1	l_2			l_1	l_1	l_2		l_1			l_1			l_1	
S_2		l_1					l_1		l_1		l_1	l_1	l_1		
S_3		l_2					l_1	l_1							
S_4			l_1	l_1					l_1				l_2		
S_5			l_2			l_2				l_1		l_1			
S_6			l_2							l_1					

Fig. 17. Minimum skill level required to complete task *j*, NS_j

The minimum skill level required for single-skill task *j* is NS_{js}, and the minimum skill level required for multi-skill task *j* is NS_j, as shown in Fig. 17. According to the types of skills acquired by workers, workers with a single skill are level I, workers with two skills are level II, and workers with three or more skills are level III. According to (Lev & Withers, 2002) 's literature study, the learning rates β_i for workers at different levels are 0.152, 0.312, and 0.515, respectively. The specific information of workers is counted in Table 6. Since the phase of skill level update is set to one month, the coefficient in the model of worker skill level change is 22.

Table 6
Information on worker levels, study rates, etc.

Worker	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
Level	III	III	III	II	II	III	II	III
β_i	0.515	0.515	0.515	0.312	0.312	0.515	0.312	0.515
Worker	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	
Level	III	III	II	II	III	III	III	
β_i	0.515	0.515	0.312	0.312	0.515	0.515	0.515	

The value of incompressibility factor M differs due to the different process contents of each job. The values of the incompressibility factor M for jobs {J_1, J_2, ……J_{29}} are shown in Table 7.

Table 7
The value of incompressibility factor M for each task

Task	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}
	0	0	0	0	0	0	0	0	0.6	0.6	0.6	0	0	0	0
Task	J_{16}	J_{17}	J_{18}	J_{19}	J_{20}	J_{21}	J_{22}	J_{23}	J_{24}	J_{25}	J_{26}	J_{27}	J_{28}	J_{29}	
	0	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0	0	

The workers' mastery levels of each skill were updated by Eq. (6) and Eq. (7) based on the measurement of the workers' hours of completing each task in the previous stage, as shown in Table 8.

Table 8
The level of mastery of each skill by permanent workers sl_k^i

	S_1	S_2	S_3	S_4	S_5	S_6
I_1	l_2		l_2		l_2	
I_2			l_2	l_2		l_2
I_3	l_3		l_2			l_2
I_4		l_2	l_1			
I_5			l_1		l_2	
I_6	l_2	l_2				l_2
I_7					l_2	l_2
I_8			l_2		l_2	l_2
I_9	l_3	l_1				l_3
I_{10}	l_1	l_2		l_1		
I_{11}			l_2			l_2
I_{12}		l_3	l_2			
I_{13}				l_3	l_2	l_3
I_{14}	l_2		l_2		l_2	
I_{15}	l_3	l_3	l_3			

5.2 Development of worker scheduling program

The skills acquired by workers are divided into 6 types. There are 29 assembly tasks for various types of pumps, and the number of permanent workers is 15. There is a batch of orders, the number of orders is 6, and the number of products is 6. This case considers the learning effect of workers and flexible employment strategy, and assumes that there are 10 alternative workers available in the temporary worker database, whose skill mastery is shown in Table 9. Table 10 describes the basic information of the case, and Table 11 shows the algorithm parameter settings.

Table 9
Skill mastery of temporary workers P_{is}

	S_1	S_2	S_3	S_4	S_5	S_6
M_1	l_1					
M_2						
M_3						l_2
M_4		l_2				
M_5	l_2					
M_6	l_2					
M_7						l_2
M_8	l_2					
M_9	l_1					
M_{10}		l_2				

Table 10
Basic information of the case

Parameters	Parameter Value
Number of orders	6
Number of tasks	29
Number of permanent workers	15
Number of temporary workers	10
Number of skills	6
Skill Level	l_1-l_4
Skill set size	3
Maximum number of multi-skilled workers working in the same cell	3
Product Category	6
Daily base salary for single-skilled workers C_a (yuan)	110
Daily base salary for multi-skilled workers C_b (yuan)	130
Daily base salary for temporary workers C_t (yuan)	80
Mass loss cost per unit hour C_d (yuan)	0.5
Maximum number of workers assigned to cell C I_c^{max}	25
Minimum number of workers assigned to cell C I_c^{min}	5

Table 11
Algorithm parameter setting

Parameters	Value
Population size N	200
Maximum number of iterations t_{Max}	100/200/500
Crossover probability p_c	0.8
Mutation probability p_m	0.05
a	[0,2]
r_1, r_2	[0,1]
l	[-1,1]
p	[0,1]
b	1

Table 12 shows the order information, including the order quantity demanded, the product type, and the delivery period. Table 13 shows the lead time for different types of products, and Table 14 shows the tasks for different types of products. Where "1" means that the task is included in the product and "-" means that the task is not included in the product. Table 15 describes the values for different skill levels.

Table 12
Order Information

Order	1	2	3	4	5	6
Product Type	A	A	B	C	C	D
Demand	1000	1500	500	450	510	440
Delivery Time	12	15	14	13	13	14

Table 13
Lead times for different types of products（minutes）

Product Type	A	B	C	D	E	F
Lead time	10	12	22	13	10	24

Table 14
Tasks for different types of products

Product Type	A	B	C	D	E	F
Task1	1	1	1	1	1	1
Task 2	-	1	1	1	1	-
Task 3	1	-	-	1	1	-
Task 4	1	-	-	1	-	-
Task 5	-	-	1	-	-	1
Task 6	-	1	-	1	1	1
Task 7	-	1	-	1	1	-
Task 8	-	1	-	-	1	1
Task 9	-	1	-	-	1	-
Task 10	-	1	-	-	1	1
Task 11	1	-	-	-	-	-
Task 12	-	-	1	-	-	-
Task 13	-	-	-	-	-	-
Task 14	1	1	-	-	-	1
Task 15	-	-	1	-	-	1
Task 16	1	1	1	1	-	1
Task 17	-	-	-	1	-	1
Task 18	-	-	1	-	1	1
Task 19	-	-	1	-	-	1
Task 20	-	-	1	-	1	1
Task 21	-	-	-	-	1	-
Task 22	-	-	-	1	1	-
Task 23	1	-	-	-	1	-
Task 24	1	1	-	-	1	-
Task 25	-	1	-	-	-	-
Task 26	-	-	1	-	-	-
Task 27	-	-	-	-	1	-
Task 28	-	-	-	-	1	-
Task 29	1	1	1	1	1	1

Table 15
The C_{sk} value at different skill levels（Yuan）

Level	C_{sk}	C_{sk}	C_{sk}	C_{sk}	C_{sk}	C_{sk}
$l_k = l_1$	C_{11}	C_{21}	C_{31}	C_{41}	C_{51}	C_{61}
	1	1	1	1	1	1
$l_k = l_2$	C_{12}	C_{22}	C_{32}	C_{42}	C_{52}	C_{62}
	2	2	2	2	2	2
Level	C_{sk}	C_{sk}	C_{sk}	C_{sk}	C_{sk}	C_{sk}
$l_k = l_3$	C_{13}	C_{23}	C_{33}	C_{43}	C_{53}	C_{63}
	3	3	3	3	3	3
$l_k = l_4$	C_{14}	C_{24}	C_{34}	C_{44}	C_{54}	C_{64}
	5	5	5	5	5	5

Before optimizing worker scheduling, all six orders were completed in the assembly line, and the products took longer to prepare for processing. If overtime is not considered, all orders are required to be completed within the specified working hours. The specific orders were completed as shown in Table 16, and the delivery rate of the orders was 66.67%. The base salary of all permanent workers is ￥100/day. According to 15 permanent workers with 10h working hours a day, it takes 650736s to complete 6 orders cumulatively, which is about 18 days. The cost of workers needed is ￥27,000.

Table 16
Delivery of orders under assembly line production

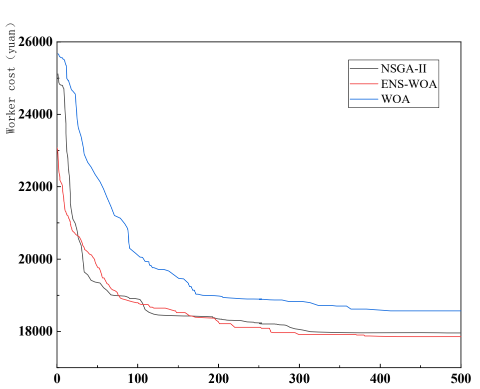
Order No.	Order sequence	Number of workers	Order completion time	Order delivery time	Order Delivery
1	5	15	33.3h	146.36h	Delay
2	6	15	34.4h	180.76h	Delay
3	4	15	33h	113.06	On time
4	1	15	21.89h	21.89h	On time
5	2	15	24.8h	46.69h	On time
6	3	15	33.37h	80.06h	On time

Since the performance of the algorithm is greatly affected by the evolution time, the number of iterations is taken as 100, 200 and 500 for testing respectively. The experimental results are taken as the average results of 10 runs of the algorithm, and the results are all retained to two decimal places, as shown in Table 17.

Table 17
Analysis of optimization results

Algorithm	Iteration	Running time (s)	F_1		F_2	
			Optimum value	Average value	Optimum value	Average value
WOA	100	351	19825.63	22976.23	351056.90	418766.11
	200	489	18979.40	19205.33	322000.20	330982.69
	500	590	18720.00	18844.27	285200.42	298203.05
NSGA-II	100	406	18910.34	21410.37	350098.02	403821.84
	200	672	18344.92	18460.66	341064.85	345038.47
	500	821	17962.70	18164.92	285944.71	296233.56
ENS-WOA	100	399	18797.70	20618.18	370057.89	418531.04
	200	482	18266.60	18582.32	346000.70	333662.99
	500	565	17857.40	17968.43	284873.00	288726.85

The evolution process of the optimal solution for each objective function is shown in Fig. 18. The convergence curves show that the average convergence ability of ENS-WOA is stronger than that of NSGA-II algorithm and WOA, and it finds the Pareto solution faster than these two algorithms.



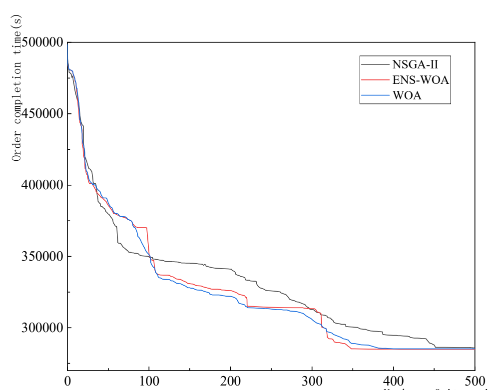


Fig. 18. Evolutionary process of optimal solutions of functions F_1 and F_2

Finally, to determine the quality of the algorithms, the diversity, uniformity index *IGD* and *Sp* values of the solutions of the objective function under the three algorithms are calculated, as shown in Table 18. The results show that the results of *Sp* for functions F1 and F2 under both NSGA-II and ENS-WOA algorithms are found to be close, and both *Sp* values are smaller than the results under WOA. This indicates that both algorithms have comparable performance in terms of solution uniformity and both are better than WOA. However, the *IGD* values of F_1 and F_2 solved by ENS-WOA are smaller than those of the other two algorithms, indicating that ENS-WOA has the strongest convergence.

Table 18
Comparison of the three algorithms in two functions

Function	Evaluation Metrics	WOA	NSGA-II	ENS-WOA
F_1	*IGD*	3.081	2.569	**0.132**
	Sp	2.809	**1.035**	1.3
F_2	*IGD*	1.73	1.470	**0.328**
	Sp	10.370	7.192	**7.01**

In order to observe more intuitively the differences in the metrics of the three algorithms, Fig. 19 shows the box line plot of the function solution for a number of iterations of 500. The results of ENS-WOA solution are more stable, and the values of F_1 and F_2 are better. Therefore, ENS-WOA shows the best performance in solving the multi-objective optimization problem presented in this paper.

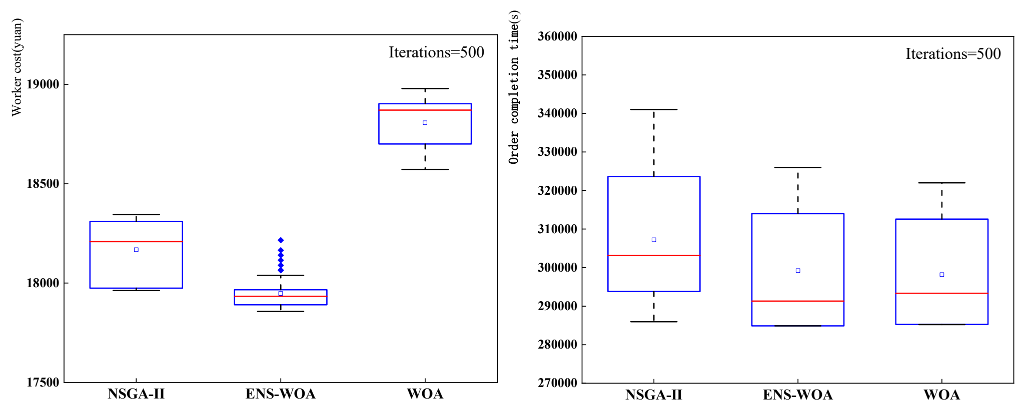


Fig. 19. Box line diagram of three algorithms

5.3 Comparative analysis of different solutions

When the number of iterations is 500 and the ENS-WOA is run 10 times, the algorithm generates 22 solutions, as shown in Fig. 20. The final Pareto frontier contains 10 solutions that do not directly provide a unique optimal solution for the decision maker, as shown in Table 19.

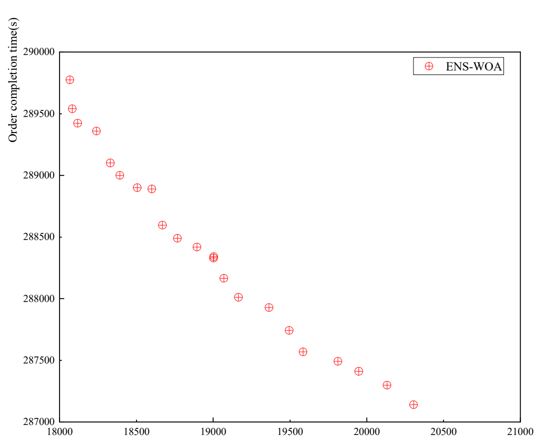


Fig. 20. Pareto solution set for 500 iterations of ENS-WOA

Table 19
Pareto Frontier Solutions

Solutions	Worker cost	Delivery time of the order (s)
1	18067.30	289774.96
2	18082.64	289540.75
3	18392.66	289000.67
4	18117.31	289423.22
5	19164.91	288011.09
6	18894.64	288419.04
7	18330.00	289100.66
8	19584.91	287569.30
9	20304.80	287140.20
10	18506.00	288900.00

The analysis of the Pareto frontier shows that option 1 is chosen when the company values the cost factor more, and option 9 is chosen when the company values the order completion time more. Different types of the same product exist. For the pump assembly line, after subdividing the skill level and type required for the operation and the skill level of workers, workers only need to continuously learn the corresponding process to reduce non-value-added time while improving operational efficiency.

At the same time, after optimizing the salary structure, workers are more motivated to work and learn, which is conducive to improving the overall operational efficiency of workers.

6. Conclusion

This paper has focused on three aspects of the optimization problem of personnel allocation on assembly lines in a C2M environment:

(1) Firstly, for the heterogeneity among workers and the ever-changing skill learning, a model of worker assignment efficiency change based on learning curve is constructed for single-skill and multi-skill tasks, respectively. And this model is used as the input of a multi-objective optimization number model for personnel assignment in assembly lines considering the learning effect.

(2) Secondly, by studying the solution methods of multi-objective optimization problems and the characteristics of each hybrid metaheuristic algorithm, it is proposed that the whale optimization algorithm can be combined with the non- dominant ranking, crowding operator and cross-variance operator in genetic algorithm to achieve complementary advantages. The improved multi-objective elite non-dominant sorting whale optimization algorithm (ENS-WOA) is designed. The classical arithmetic cases are selected for testing, and the experimental comparison and analysis with the WOA and the NSGA-II are conducted to verify the feasibility and superiority of the ENS-WOA.

(3) Finally, taking the water pump product assembly line of Company B, which is transforming to C2M, as an example, the operational efficiency of the existing workers is measured and the personnel scheduling of the assembly line is optimized. MATLAB software is applied to establish the ENS-WOA for solving the problem, and the results are compared with the worker cost and delivery time required to complete the order in the original assembly line. The results show that the model proposed in this paper is reliable, the algorithm is stable, and the optimal solution can be found quickly and accurately. It outperforms both the WOA and the NSGA-II algorithm in terms of convergence speed and quality of the solution. Worker costs were reduced by 29.02% and orders were completed approximately 10 days earlier.

The following conclusions are drawn from this research: when carrying out production with a wide variety of orders, short delivery periods, inconsistent worker skill levels, and assembly lines dominated by manual work, a mixed application of rapidly reconfigurable assembly cells, workers with multiple skill levels, and temporary workers can give full play to worker potential, significantly improve efficiency, and reduce worker cost.

References

Anzanello, M. J., & Fogliatto, F. S. (2011). Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics*, *41*(5), 573-583.

Azizi, N., Zolfaghari, S., & Liang, M. (2010). Modeling job rotation in manufacturing systems: The study of employee's boredom and skill variations. *International Journal of Production Economics*, *123*(1), 69-85.

Azzouz, A., Ennigrou, M., & Ben Said, L. (2018). Scheduling problems under learning effects: classification and cartography. *International Journal of Production Research*, *56*(4), 1642-1661.

Cavagnini, R., Hewitt, M., & Maggioni, F. (2020). Workforce production planning under uncertain learning rates. *International Journal of Production Economics*, *225*(7), 107590.

Chakraborty, S., Sharma, S., Saha, A. K., & Saha, A. (2022). A novel improved whale optimization algorithm to solve numerical optimization and real-world applications. *Artificial Intelligence Review*, *55*(6), 4605-4716.

Cohen, Y., & Ezey Dar-El, M. (1998). Optimizing the number of stations in assembly lines under learning for limited production. *Production Planning & Control*, *9*(3), 230-240.

Corominas, A., Pastor, R., & Plans, J. (2008). Balancing assembly line with skilled and unskilled workers. *Omega*, *36*(6), 1126-1132.

De Jong, J. R. (1957). The Effects of Increasing Skill on Cycle Time and Its Consequences for Time Standards. *Ergonomics*, *1*(1), 51-60.

El-Dabah, M., Ebrahim, M. A., El-Sehiemy, R. A., Alaas, Z., & Ramadan, M. M. (2022). A Modified Whale Optimizer for Single- and Multi-Objective OPF Frameworks [Article]. *Energies (19961073)*, *15*(7), 2378-N.PAG.

Emmons, H., & Fuh, D.-S. (1997). Sizing and scheduling a full-time and part-time workforce with off-day and off-weekend constraints. *Annals of Operations Research*, *70*(0), 473-492.

Erdem, Y. (2011). *Workforce Planning for Seasonal Demands* (Publication Number 28554014) Marmara Universitesi (Turkey)]. ProQuest Dissertations and Theses Full-text Search Platform.

Fichera, S., Costa, A., & Cappadonna, F. A. (2017). Heterogeneous workers with learning ability assignment in a cellular manufacturing system. *International Journal of Industrial Engineering Computations*, *8*(4), 427-440.

Foote, D. A., & Folta, T. B. (2002). Temporary workers as real options. *Human Resource Management Review*, *12*(4), 579-597.

Karaoz, M., & Albeni, M. (2005). Dynamic technological learning trends in Turkish manufacturing industries. *Technological Forecasting and Social Change*, *72*(7), 866-885.

Kim, D., Moon, D. H., & Moon, I. (2018). Balancing a mixed-model assembly line with unskilled temporary workers: algorithm and case study. *ASSEMBLY AUTOMATION*, *38*(4), 511-523.

Lee, B., Lee, H. S., Lee, S., & Park, M. (2015). Influence Factors of Learning-Curve Effect in High-Rise Building Constructions. *Journal of Construction Engineering and Management*, *141*(8), Article 04015019.

Lev, B., & Withers, D. (2002). Human Learning: From Learning Curves to Learning Organizations [Book Review]. *Interfaces*, *32*(1), 95-96.

Li, Q., Sun, Q., Tao, S., & Gao, X. (2019). Multi-skill project scheduling with skill evolution and cooperation effectiveness. *Engineering, Construction and Architectural Management*, *27*(8), 2023-2045.

Li, Y., Yang, X., & Yang, Z. (2019). Uncertain learning curve and its application in scheduling. *Computers & Industrial Engineering*, *131*, 534-541.

Liu, C., Wang, J., & Leung, J. (2016). Worker assignment and production planning with learning and forgetting in manufacturing cells by hybrid bacteria foraging algorithm. *Computers & Industrial Engineering*, *96*(C), 162-179.

Liu, F., Niu, B., Xing, M., Wu, L., & Feng, Y. (2021). Optimal cross-trained worker assignment for a hybrid seru production system to minimize makespan and workload imbalance. *Computers & Industrial Engineering*, *160*, 107552.

Liu, M., Liu, Z., Chu, F., Liu, R., Zheng, F., & Chu, C. (2022). Risk-averse assembly line worker assignment and balancing problem with limited temporary workers and moving workers. *International Journal of Production Research*, *60*(23), 7074-7092.

Liu, R., Liu, M., Chu, F., Zheng, F., & Chu, C. (2021). Eco-friendly multi-skilled worker assignment and assembly line balancing problem. *Computers & Industrial Engineering*, *151*, 106944.

Lohmann, M., Anzanello, M. J., Fogliatto, F. S., & da Silveira, G. C. (2019a). Grouping workers with similar learning profiles in mass customization production lines. *Computers & Industrial Engineering*, *131*(5), 542-551.

Lohmann, M., Anzanello, M. J., Fogliatto, F. S., & da Silveira, G. C. (2019b). Grouping workers with similar learning profiles in mass customization production lines. *Computers & Industrial Engineering*, *131*, 542-551.

Mathur, K., & Süer, G. A. (2013). Math modeling and GA approach to simultaneously make overtime decisions, load cells and sequence products. *Computers & Industrial Engineering*, *66*(3), 614-624.

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in engineering software*, *95*, 51-67.

Neidigh, R. O., & Harrison, T. P. (2010). Optimising lot sizing and order scheduling with non-linear production rates. *International Journal of Production Research*, *48*(7-8), 2279-2295.

Paul, C., Kumar Roy, P., & Mukherjee, V. (2023). Study of wind-solar based combined heat and power economic dispatch problem using quasi-oppositional-based whale optimization technique. *Optimal Control Applications and Methods*, *44*(2), 480-507.

Pilati, F., Faccio, M., & Cohen, Y. (2021). Absenteeism and Turnover as Motivation Factors for Segmenting Assembly Lines. *IFAC-PapersOnLine*, *54*(1), 613-616.

Pinker, E. J., & Larson, R. C. (2003). Optimizing the use of contingent labor when demand is uncertain. *European Journal of Operational Research*, *144*(1), 39-55.

Prasad, P., Wallace, L., Navidi, M., & Phillips, A. W. (2022). Learning curves in minimally invasive esophagectomy: A systematic review and evaluation of benchmarking parameters. *Surgery*, *171*(5), 1247-1256.

Ruihan, C. J. R. H. Z. Z. L. (2021). *An Adaptive Evolutionary Whale Optimization Algorithm*. College of Information Science & Technology, Beijing University of Chemical Technology；Systems Engineering Research Institute.

Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization* Massachusetts Institute of Technology.

Shi, S., Xiong, H., & Li, G. (2023). A no-tardiness job shop scheduling problem with overtime consideration and the solution approaches. *Computers & Industrial Engineering*, *178*, 109115.

Singh, S., & Singh, D. (2023). A Bio-inspired VM Migration using Re-initialization and Decomposition Based-Whale Optimization. *ICT Express*, *9*(1), 92-99.

Stratman, J. K., Roth, A. V., & Gilland, W. G. (2004). The deployment of temporary production workers in assembly operations: a case study of the hidden costs of learning and forgetting. *Journal of Operations Management*, *21*(6), 689-707.

Sun, G., Shang, Y., & Zhang, R. (2022). An Efficient and Robust Improved Whale Optimization Algorithm for Large Scale Global Optimization Problems. *Electronics*, *11*(1475), 1475.

Tai, H.-W., Chen, J.-H., Cheng, J.-Y., Hsu, S.-C., & Wei, H.-H. (2021). Learn Curve for Precast Component Productivity in Construction. *International Journal of Civil Engineering*, *19*(10), 1179-1194.

Tang, B., Liang, Y., Shi, J., & Li, T. (2022). Learning Curve of Robotic Right Hemicolectomy. *Journal of Gastrointestinal Surgery*, *26*(10), 2215-2217.

Tian, Z., Jiang, X., Liu, W., & Li, Z. (2023). Dynamic energy-efficient scheduling of multi-variety and small batch flexible job-shop: A case study for the aerospace industry. *Computers & Industrial Engineering*, *178*, 109111.

Valsamis, E. M., Chouari, T., O'Dowd-Booth, C., Rogers, B., & Ricketts, D. (2018). Learning curves in surgery: variables, analysis and applications. *POSTGRADUATE MEDICAL JOURNAL*, *94*(1115), 525-530.

Van Veldhuizen, D. A., & Lamont, G. B. (2000). Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation* *8*(2), 125-147.

Wright, T. P. (1936). Factors Affecting the Cost of Airplanes. *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, *3*(4), 122-128.

Zhi, R., & Xu, Y. (2019). *Optimal workforce and machine scheduling to maximize profit and worker satisfaction*. Glorious Sun School of Business and Management, Donghua University, Shanghai, China Glorious Sun School of Business and Management, Donghua University, Shanghai, China.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *EVOLUTIONARY COMPUTATION*, *8*(2), 173-195.