Application research of improved genetic algorithm based on machine learning in production scheduling

Kai Guo[1,2] · Mei Yang[1] · Hai Zhu[1]

Abstract

Job shop scheduling problem is a well-known NP problem. It is limited by various conditions. As the scale of the problem increases, the difficulty of finding the optimal solution will increase. It is a difficult combination problem. Limited by the constraints of the actual production environment, how to effectively arrange the processing order of each part will directly affect the production efficiency, the appropriate production scheduling algorithm can correctly and effectively plan the enterprise resources and rationally arrange the processing order and processing time of the workpiece. Proper use of existing resources, by optimizing production scheduling instructions, to meet the basic requirements of production scheduling, in order to obtain the optimization of total production time, has important theoretical significance for the actual production of enterprises. In this paper, the mathematical model is abstracted on the basis of the production scheduling problem. According to the different parts of the same machine and the different processes of the same part, the corresponding processing time and waiting time are obtained. At the same time, the genetic algorithm is improved by genetic algorithm. A dynamic genetic operator based on the number of iterations is proposed, which further enhances the convergence performance and search ability of the genetic algorithm. Through the simulation of MATLAB simulation program, combined with the scheduling standard example, the performance analysis of different algorithms is carried out, the search efficiency of genetic algorithm is improved, the convergence performance of the algorithm is improved, and different optimization choices are obtained for different time weights. The operation results of the system meet the requirements of production scheduling, which proves the feasibility and practicability of the improved genetic algorithm.

Keywords Machine learning · Legacy algorithm · Production scheduling · Mathematical model

1 Introduction

Production scheduling is the core of modern enterprise manufacturing execution systems and the most complex research issue in the production environment. Production scheduling refers to the sorting process of production operations in different production resources in the production process. Based on the capacity and efficiency of production equipment, according to the quantitative production work, the production sequence of each production task is comprehensively arranged, and the production plan is optimized. Select production equipment to reduce production time, increase production efficiency, and balance the production load of various machines and workers [1–4]. Under the premise of meeting production constraints as

✉ Kai Guo
  guokai@haust.edu.cn

  Mei Yang
  kdgzy@haust.edu.cn

  Hai Zhu
  zhuhai@haust.edu.cn

1 School of Management, Henan University of Science and Technology, Luoyang 471003, China

2 Henan Collaborative Innovation Center of Nonferrous Metals, Luoyang 471003, China

much as possible, determine the specific production sequence, production time and production machine for each production object to obtain the optimization of production time or production cost [5]. As a result, production capacity is optimized and production cycles are shortened, creating greater economic benefits for the company. For the theoretical research of production scheduling, there are currently many solutions, including linear programming algorithms, Petri nets, queuing network models, Markov chains, simulation methods and intelligent algorithms. These methods involve the production scheduling system. Logical hierarchy, random hierarchy and time hierarchy establish the theoretical framework of combined optimization dynamic system [6, 7]. Particularly in applied research, genetic algorithms have a stable place. Based on the characteristics of genetic algorithms, the application field is expanding, and the research on optimization and improvement of genetic algorithms is becoming more and more mature. It can be mixed and optimized with various algorithms. Improve its own performance. In recent years, there are also many production scheduling strategies based on genetic algorithms in the field of manufacturing [8–10].

When considering the actual production schedule, most companies cannot really estimate the actual production capacity and the number of production operations at the manufacturing site, and miscalculate the inventory of raw materials and the purchase rate so that enterprises can take orders to improve their own efficiency. It is not possible to carry out an effective production scheduling plan [11–13]. In order to improve the service level of customers and meet the requirements of completing production tasks on time, the production workshops have to meet the delivery time of production through overtime or outsourcing [14, 15]. At the same time, defective production plans cannot meet the constraints of production materials and production capacity. For example, the procurement progress of raw materials cannot meet the production schedule of the factory floor, resulting in great waste of production resources of the factory, which will affect the entire production schedule, and bringing huge production costs to the company [16–18]. In order to solve the above problems, an efficient and appropriate planning of production resources, meet customer needs, control production costs, assist production personnel to develop practical production scheduling, maximize production efficiency, maximize production resource utilization, and production. The shortest production scheduling algorithm has been extremely urgent [19–21].

Yan et al. proposed an integrated optimization production scheduling method based on alternating iterative genetic algorithm. The operational constraints that guarantee mass production are first determined. Then, based on the nonlinear mixed integer programming, an integrated model of production planning and scheduling is established. The hybrid heuristic algorithm (AIHGA) alternate iterative method is used to solve the problem. The operation steps are as follows: a hybrid genetic algorithm is used to solve the scheduling scheme; another hybrid genetic algorithm is used to give a solution scheme. Two hybrid genetic algorithms run alternately to optimize planning and scheduling at the same time. Finally, AIHGA is compared with the overall optimization method based on hybrid genetic algorithm [22]. Hai-Jun et al. proposed a particle swarm optimization algorithm based on dimensional information sharing for shop scheduling problem. The cognitive process and update process of particle swarm were studied, and the concepts of dimensional information sharing and dynamic cognition were introduced. The dimensional information of the optimization problem can be transmitted and exchanged. By increasing the disturbance factor to overcome the premature convergence of the algorithm, the ability of PSO to adapt to the optimization problem is improved. Finally, the best mean value and standard deviation are obtained through the test of three continuous function options [23]. Zhang and other scholars proposed a parallel production scheduling method based on hybrid genetic simulation algorithm. For the parallel production model, the principles and methods of operator identification and design in genetic algorithm are studied. The strategy of retaining elites has also been studied in genetic algorithms. In order to ensure the quality of the initial population chromosomes in the example, the initial population chromosomes of the genetic algorithm are generated using weighted benefits, and the final result is obtained in this way [24]. Soares et al. proposed a new genetic algorithm structure that describes the multi-objective fitness function used, the set of possible individual selection techniques, and the adjustment values of the crossover and mutation operators. Applying the developed genetic algorithm to two manufacturing scenarios, the most important parameters of the genetic algorithm configuration are determined. This study shows that the use of genetic algorithms is a viable technique for solving the MPS problem; however, its applicability still depends heavily on the scale of the manufacturing scheme [25].

In this paper, the mathematical model is abstracted on the basis of the production scheduling problem. According to the different parts of the same machine and the different parts of the same part, the corresponding processing time and waiting time are obtained. At the same time, the genetic algorithm is improved by genetic algorithm. A dynamic genetic operator based on the number of iterations is proposed, which further enhances the convergence performance and search ability of the genetic algorithm. Through the simulation of MATLAB simulation program, combined with the scheduling standard example, the

performance analysis of different algorithms is carried out, the search efficiency of genetic algorithm is improved, the convergence performance of the algorithm is improved, and different optimization choices are obtained for different time weights. The operation results of the system meet the requirements of production scheduling, which proves the feasibility and practicability of the improved genetic algorithm.

2 Proposed method

2.1 Production scheduling problem solving problem

The production scheduling problem generally refers to the scheduling of production operations, the rational scheduling of production resources, for a detachable project, under the premise of fully considering various constraints, coordinating the use of various resources, planning processing time, arranging the processing sequence, and finally obtaining the optimization of total production time and utilization of production resources. At the time of the production plan, although the detailed order was generated, the production activities were arranged according to the production order, but there were many unforeseen accidents in the actual production environment. In the actual production execution system, there are unexpected production deviations and production failures from time to time. Therefore, the production management personnel need to pay attention to the production status and maintain the production order in the actual production process to prevent the occurrence of production accidents. In special cases, it can solve production problems in a timely and accurate manner and ensure the steady operation of the production system.

2.1.1 Classification of performance indicators

In the production scheduling process, the general performance indicators are divided into the following three categories.

The first category: the maximum productivity indicator. It refers to the maximum output rate of production operations, the shortest total production time, etc. Through these constraints, the purpose of strengthening the production capacity of enterprises and improving the production efficiency of enterprises is achieved. Under the premise of determining the inventory according to the customer's order, the enterprise formulates the production strategy, improves the utilization rate of production resources, achieves the goal of shortening the total production time and maximizes the production capacity.

The second category: the minimum investment indicator. When the enterprise is producing, it will produce various losses and form production costs, including warehouse inventory cost, resource production cost, transportation cost and insufficient production cost. The enterprise needs to reduce the cost of each part, maximize profit and profit, and minimize the capital investment.

The third category: customer satisfaction level indicators. The production operations must meet the requirements of minimum delivery delay, close delivery time, etc.

The traditional production scheduling is aimed at the maximum production capacity, with the minimum production time, the minimum production investment capital, and the customer's demand as the standard. In the actual production enterprise, it is also necessary to consider the relationship between the produced product and the delivery date, in advance. The products produced need to be stored in the warehouse to the delivery date, and the products that are lagging to be produced must pay liquidated damages, which results in production costs. Therefore, the actual production situation is much more complicated, and various factors need to be considered comprehensively.

2.1.2 Characteristics of production scheduling problems

The main features of the production scheduling problem are as follows:

1. Complexity: In the actual production system, each production operation has arrival time, startup time, loading and unloading time, processing time, waiting time, production quantity, production batch, etc. Each processing machine also has loss, between the machine and the machine. There are transportation distances, operators are also skilled, and various factors are mutually influential, not only considering the processing sequence of production operations, but also considering various external factors, so the shop scheduling problem is quite complicated. As the scale of the problem increases, the amount of computation required for solving the problem is also rising sharply. The time taken to find the optimal solution is immeasurable, so finding the suboptimal solution or the better solution becomes the best choice.

2. Randomness: In the actual production system, there will be many random bursts, such as the inherent randomness of the production system, the randomness of the production process, and the randomness of the production environment. For example, the arrival time of the production operation is uncertain, and the processing is performed. Uncertain time, processing equipment failures, changes in production requirements, emergency workpiece insertion, customer order

cancelation, etc., are unforeseen uncertainties, so production scheduling has a large dynamic randomness.

3. Multiple constraints: The production sequence of production and processing operations has strict regulations. The processing order of each process cannot be changed at will. The number of production and processing machines and the production capacity of the machine are limited. Personnel, raw materials and storage are not. Infinite and other constraints constrain the production scheduling system, making its modeling and solving more complicated.

4. Multi-purpose: The performance indicators of production scheduling are divided into three categories, the maximum productivity index, the minimum investment fund index and the customer satisfaction degree index. The corresponding production schedules need to refer to these indicators to achieve production goals. Minimize production by maximizing productivity, minimizing total production time to meet production time and job delivery deadlines, reducing production losses, lowering inventory costs, maximizing resource utilization, and increasing business profitability to reduce production costs. These goals are mutually influential and mutually constrained. It is quite difficult to meet all the objectives. This requires the production scheduling strategy to comprehensively consider various factors, meet the production targets as much as possible, and ensure the efficiency and stability of the production system.

2.2 Genetic algorithm (GA)

2.2.1 Principles of genetic algorithms

When dealing with problems, genetic algorithm first needs to plan the solution space of the solved problem. All feasible solutions in space must be coded and mapped. A set of feasible solutions can be produced by the conversion of binary code or real code. Population: Individuals in a population are called chromosomes and are composed of gene codes that represent individual characteristics. The characteristics of individuals are mapped to chromosomes with specific gene codes by coding. By the three operators of the genetic algorithm, under the operation of the crossover and the mutation operator, new individuals are generated between the individuals and the individuals themselves, forming a larger population, by selecting the operator selection, according to the fitness function. Individuals with high adaptability are selected according to the criteria to enter the next generation population, and the above process is repeated until the evolutionary algebra is

reached or the optimal solution is converged. According to the criteria of survival of the fittest and the survival of the fittest, the next-generation population has higher fitness and better characteristics than the previous-generation population. After multiple iterations, the optimal individual produced by the last generation of population is decoded by decoding. According to the criteria of survival of the fittest and survival of the fittest, the next generation population has higher adaptability and better characteristics than the previous generation population. After multiple iterations of evolution, the optimal individuals generated by the last generation population become the near optimal solution or optimal solution of the problem after decoding.

2.2.2 The execution process of the genetic algorithm

The program structure of the standard genetic algorithm is shown in Fig. 1: where $P(t)$ is the current population and $P(t + 1)$ is the next-generation population.

As can be seen from the block diagram, the process of finally solving the genetic algorithm is not unique. It can be used in various combinations. The middle operation can get different solutions as long as there is one difference.

The basic steps for a simple GA to solve a problem are as follows:

1. The solution to the coding problem, each code string after writing represents a feasible solution;
2. By randomly defining n code strings to form an initial group pop_0, then the population is a set of solutions;
3. Set the problem to be solved into the initial code string group, and obtain the fitness value that the individual in the group can adapt to the problem;
4. In the initial population pop_0, select according to the individual's fitness, select excellent individuals to form a new population, that is, the father's population F_k, and eliminate the inferior individuals;
5. Propagating the parental population F_k, that is, crossing the crossover probability p_c to generate a new population C_k;
6. Perform the mutation of the new population C_k, that is, perform the operation with the mutation probability p_m to obtain another population $pop_{(k+1)}$.

By repeating steps 3–6, the individual is constantly evolving, and finally the individual best adapts to the problem environment, that is, the optimal solution to the problem. From the basic steps above, it can be seen that the genetic algorithm has three basic operations: selection, crossover and mutation.

Fig. 1 Genetic algorithm block diagram
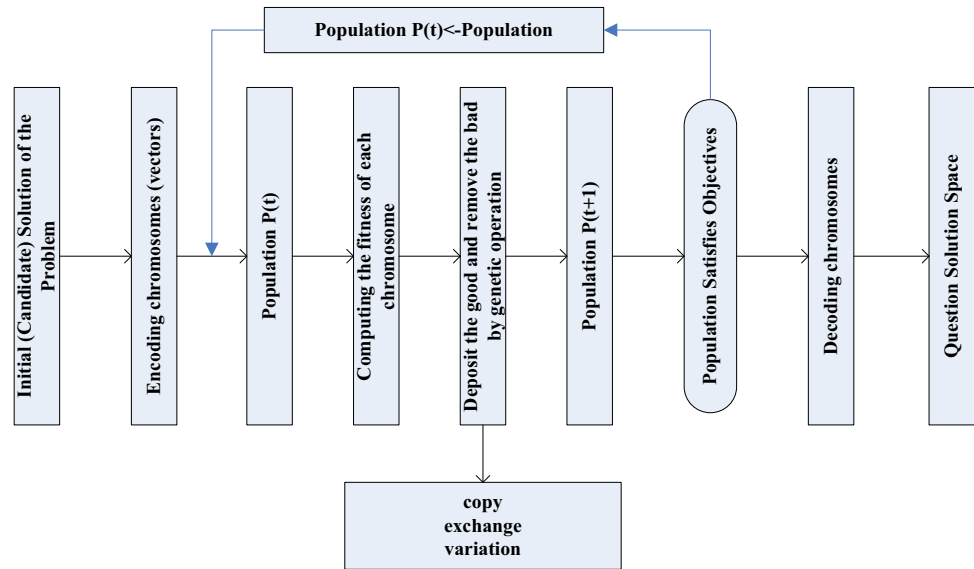
2.2.3 Mathematical description of three basic genetic operations

Let the individual coding alphabet $A = \{0, 1, \ldots, a - 1\}$, the length of the individual string is l, then the individual space is represented as $A = \{0, 1, \ldots, a - 1\}$, and $|S| = a^l$. In the genetic process, the population size is fixed to M, then the population space and the parent space are expressed as:

$$S^M = \{(s_1, s_2, \ldots, s_M) | s_i \in S, \quad i = 1, 2, \ldots, M\} \quad (1)$$

$$S^2 = \{(s_1, s_2) | s_1, s_2 \in S\} \quad (2)$$

This gives a definition of the probability of the above three operations.

Select T_s: For any given population $X \in S^M$, the probability of selecting a T_s parent under the influence of $(s_1, s_2) \in S^2$ is:

$$P\{T_s(X) = (s_1, s_2)\} = \frac{f(s_1)}{\sum_{s_i \in X} f(s_i)} \times \frac{f(s_2)}{\sum_{s_j \in X} f(s_j)} \quad (3)$$

Among them $1 \leq i \leq M$, $1 \leq j \leq M$.

Cross T_c: For a given parent s_i, $i = 1, 2$, the probability of generating an individual s with probability P_c is:

$$P\{T_c[(s_1, s_2)] = s\} = \begin{cases} \dfrac{k \cdot P_c}{l}, & y \neq s_1 \\ (1 - P_c) + \dfrac{k \cdot P_c}{l}, & y = s_1 \end{cases} \quad (4)$$

where $k = k(s_1, s_2)$ is the number of intersection locations and y indicates the individuals generated by the intersections, which are intermediate variables.

Variation T_m: The mutation operator is defined as:

$$P\{T_m(s_i) = s\} = p_m^{|s_i - s|} (1 - P_m)^{l - |s_i - s|} \quad (5)$$

where $|s_i - s|$ is the Hamming distance between individual s_i and s.

2.3 Improvement of genetic algorithm

In the genetic algorithm, the selection operator and the crossover operator ensure the global search ability of the genetic algorithm. The mutation operator enhances the ability of the genetic algorithm to perform local optimization. The three operators cooperate with each other so that the genetic algorithm can solve the solution space. Global search and local search ensure efficient and accurate search ability of genetic algorithm to solve combinatorial optimization problems. When the genetic algorithm uses the crossover operator to solve the optimal solution and the solution to the optimal solution converges, the local search ability of the mutation operator can be used to speed up the approach to the optimal solution. The value of the variation probability should be compared. It should be compared that the value of change probability is small, so as to prevent the large mutation probability from destroying the building block of genetic algorithm approximate optimal solution. In the current situation, move closer to the global optimal direction. The crossover operator guarantees the global search performance of the genetic algorithm, while the mutation operator guarantees the local search performance of the genetic algorithm. The two operators work together to make the individuals in the population more diverse and improve the crossover operation to some extent. The resulting damage to good individuals avoids the occurrence of precocity.

Therefore, the crossover probability P_c and the mutation probability P_m dominate the convergence efficiency and optimization performance of the genetic algorithm. The

genetic algorithm guarantees the global search ability through the crossover operator, searches all the solution spaces, guarantees the local search ability through the mutation operator and converges to the optimal solution. The larger the P_c, the faster the new individual will be generated, and the larger the search space will be, but it will easily destroy the excellent individuals that have been produced; the smaller the P_c, the slower the new individual will be, and the algorithm will easily stop. The larger the P_m, the easier the genetic algorithm is to perform undirected search, and then degenerate into random search; the smaller the P_m, the smaller the population is not likely to mutate, making the algorithm fall into local optimum. The mutation probability and crossover probability of the general genetic algorithm take fixed values. Although the operation is simple, it limits the search performance of the genetic algorithm, and it is easy to cause the genetic algorithm to fall into a precocious situation.

2.3.1 Adaptive genetic algorithm (AGA)

The formula corresponding to the adaptive genetic algorithm is shown in Eqs. (6) and (7).

$$P_c = \begin{cases} \dfrac{k_1(f_{max} - f')}{f_{max} - f_{avg}}, & f' \geq f_{avg} \\ k_2, & f' < f_{avg} \end{cases} \quad (6)$$

$$p_m = \begin{cases} \dfrac{k_3(f_{max} - f)}{f_{max} - f_{avg}}, & f \geq f_{avg} \\ k_4, & f < f_{avg} \end{cases} \quad (7)$$

Among them, f' indicates the larger value of the two individuals waiting for the parent to cross, f indicates the appropriate value of the individual waiting for the mutation of the parent, f_{max} indicates the maximum value of all the individuals in the population, and f_{avg} indicates the average value of all individuals in the population, k_1, k_2 indicate the set crossover probability, and k_3 and k_4 indicate the set mutation probability. It can be found that when the individual fitness value is higher than the average fitness, that is, the current crossover and mutated individuals belong to the better individuals, the cross mutation probability is lower than the set initial crossover probability k_1, ensuring that the better individuals are not destruction; when the individual fitness value is lower than the average fitness, that is, when the current cross and mutated individuals belong to the inferior individual, the cross mutation probability is greater than the set initial crossover probability k_3 so that the inferior individual performs genetic recombination, so that the genetic algorithm performs global Looking for excellence. When the individual fitness value is the largest, the crossover and mutation probability is 0, and the overall optimal individual preservation is saved.

This adaptive genetic algorithm is more suitable in the late stage of genetics. All individuals are basically in the vicinity of the optimal solution, and the probability of crossover and mutation is small, which guarantees that the excellent structure will not be destroyed. However, in the early stage of genetic evolution, all individuals are in a state of waiting. In the inferior stage of evolution, this method makes the evolutionary optimization process extremely slow, and the resulting optimal individual may be a local optimal solution.

2.3.2 Improved adaptive genetic algorithm

An improved adaptive genetic algorithm, the formula is shown in Eqs. (8) and (9).

$$p_c = \begin{cases} p_{c1} - \dfrac{(p_{c1} - p_{c2})(f' - f_{avg})}{f_{max} - f_{avg}}, & f' \geq f_{avg} \\ p_{c1}, & f' < f_{avg} \end{cases} \quad (8)$$

$$p_m = \begin{cases} p_{m1} - \dfrac{(p_{m1} - p_{m2})(f - f_{avg})}{f_{max} - f_{avg}}, & f \geq f_{avg} \\ p_{m1}, & f < f_{avg} \end{cases} \quad (9)$$

This method guarantees that all individuals have large variations and crossover probabilities in the early stage of evolution. Even the population optimal solution has the smallest crossover probability p_{c2} and mutation probability p_{m2}, which ensures the rapid replacement search and global optimization of individuals in the early stage of evolution. However, considerable variation and crossover probability were preserved in the later stages of evolution, leading to the destruction of good individuals.

2.3.3 Improved algorithm in this paper

This paper proposes an optimal individual retention method and an adaptive genetic algorithm based on the number of iterations, as shown in Eqs. (10) and (11).

$$P_c = \begin{cases} 0, & f' = f_{max} \\ \left(\dfrac{t_{max} - t}{t_{max}}\right)\left(p_{c1} - \dfrac{(p_{c1} - p_{c2})(f' - f_{avg})}{f_{max} - f_{avg}}\right), & f_{max} > f' \geq f_{avg} \\ p_{c1}, & f' < f_{avg} \end{cases}$$
$$(10)$$

$$p_m = \begin{cases} 0, & f = f_{max} \\ \left(\dfrac{t_{max} - t}{t_{max}}\right)\left(p_{m1} - \dfrac{(p_{m1} - p_{m2})(f - f_{avg})}{f_{max} - f_{avg}}\right), & f_{max} > f \geq f_{avg} \\ p_{m1}, & f < f_{avg} \end{cases}$$
$$(11)$$

Among them, t indicates the current number of evolutions, t_{max} indicates the maximum number of evolutions, p_{c1} indicates the upper limit of the crossover probability, here is 0.9, p_{c2} indicates the lower limit of the crossover

probability, here is taken as 0.6, p_{m1} indicates the upper limit of the probability of variation, here is taken as 0.1, p_{m2} indicates the lower limit of the probability of mutation, which is taken as 0.001 here. Each generation adopts the optimal individual preservation method, and the individuals with the highest fitness value are directly transferred to the next-generation population. The individuals above the average use the dynamic change intersection and mutation operator, and the individuals below the average value adopt the fixed maximum. The probability of mutation ensures the global optimization ability and prevents the situation from falling into local optimum. At the same time, the low crossover and mutation probability of the good individual are destroyed by the anti-solution structure, and the optimal solution is accelerated. When the number of iterations is small, the probability of variation crossover is large, preventing the genetic algorithm from stagnating. When the number of iterations is large, the probability of crossover and mutation is small, and it is guaranteed that the excellent structure will not be destroyed.

3 Experiments

3.1 Description of the experimental problem

According to the actual environment of production scheduling, analyze the processing situation of the workshop. Many batch scheduling problems can be described as: there are N kinds of workpieces, each workpiece contains multiple identical parts, there are many different processes, there are M machine tools, each certain process of the workpiece can be processed by a single or a plurality of different machines, and the processing time of each process on different processing machines can be different. For this scheduling model, how to distribute the machining sequence of the workpiece and the machining machine is solved, so that the total machining time of the workpiece is minimized. At the same time, the problem also includes the following constraints:

1. The processing time of a certain machine for processing a certain process is determined. The processing time of a certain process on different machinable machines can be different, which ensures that the machine can process a specific workpiece at a specific process time at any time. Is certain;
2. The sequence of the processing of the same workpiece is determined, and a certain process of a workpiece must be processed after the completion of the previous process to prevent the disorder of the processing;

3. The same process of each type of workpiece can only be carried out once, to prevent repeated processing in a certain process when multiple machines can be processed;
4. A certain process of the same workpiece can be processed on different machines, as long as it can machine the workpiece, it can be processed, which ensures the diversity of the processing path, reduces the waiting time of the workpiece and improves the utilization of the machine rate;
5. The processing steps of different workpieces are not constrained in sequence, which ensures that when the machine is idle at a certain time, the qualified workpieces can be processed;
6. The production cycle is divided into processing time, queuing time and transportation time. The processing time is the total processing time of a certain batch of workpieces on a certain machine. The queuing time is when a workpiece arrives at the machine to be processed until the machine. The waiting time for starting the machining of the workpiece, which is the transportation time between the machines that are transported to the next machine to be processed at the end of processing of a workpiece;
7. The same kind of workpiece is the minimum batch, which cannot be batched. That is, one workpiece cannot be processed on multiple machines at the same time. The number of workpiece batches of different workpieces is defined from the beginning, and all processed workpieces are regarded as one. Overall, it needs to be processed together on one machine;
8. Each machine can only process one workpiece at the same time. When the machine starts to process a workpiece until all the batches of the workpiece are completed, this machine is occupied by one workpiece, and other workpieces that need this machine are processed can only wait;
9. Once a workpiece is processed, the machining process is not interrupted, which ensures that there is no machine failure and an emergency in which the workpiece is inserted midway;
10. At the beginning, all workpieces can be machined.

3.2 Establishment of mathematical model

For the job shop scheduling model, the following mathematical descriptions are established.

3.2.1 Scheduling target

Minimum total production time of the workpiece: $\min\{C_{\max}\}$.

Total workpiece queuing time, total workpiece transport time and total machine idle minimum: $\min\{Q_{\max} + H_{\max} + W_{\max}\}$.

3.2.2 Conditional constraints

$t_{ijk} - T_{ijk} > t_{ij-1l}, i = 1, 2\ldots, n; k, l = 1, 2, \ldots, m$: If the process j of the workpiece i is to be processed, it must be processed after the previous process $j - 1$ is completed. This is the processing order determination constraint of the workpiece, as shown in Fig. 2a. $t_{ijk} - t_{hxk} > T_{ijk}, i, h = 1, 2, \ldots, n; k = 1, 2, \ldots, m$: If the workpiece i wants the machining process j to be processed after the machine k has finished the process x of the existing machining workpiece h, this is the machine occupation constraint and the non-batch constraint of the workpiece machining, as shown in Fig. 2b. Where $t_{ijk} > 0, i = 1, 2, \ldots, n; k = 1, 2, \ldots, m$; $T_{ijk} > 0, i = 1, 2, \ldots, n; k = 1, 2, \ldots, m$.

3.2.3 Processing situation

One machine processes two parts, the latter part needs to wait in line; different machines process different parts of the same part, the latter machine needs to wait for the previous part of the part to be processed, and there is transportation time between the machine and the machine. Therefore, processing time and queuing time are divided into six cases:

Case 1: The workpiece to be processed has no prior process, and the machining machine has no workpiece machining.

Case 2: The workpiece to be processed has no prior process and the machining machine has workpiece machining, but the machining workpiece of the machining machine has been processed.

Case 3: The workpiece to be processed has no prior process and the machining machine has workpiece machining, but the machining workpiece of the machining machine has not been processed.

Case 4: The workpiece to be processed has a tight pre-process, and the pre-process completion time of the workpiece to be processed is greater than the current

machine's time to complete the machining of the previous workpiece.

Case 5: The workpiece to be processed has a tight pre-process, and the pre-process completion time of the workpiece to be processed is smaller than the completion time of the previous workpiece before machining. The transport time of the workpiece to be processed coincides with the processing time of the previous workpiece.

Case 6: The workpiece to be processed has a pre-process, and the completion time of the workpiece to be processed is less than the completion time of the workpiece before the current machine is processed. The transportation time of the workpiece to be machined partially coincides with the processing time of the previous workpiece (Table 1).

4 Discussion

4.1 Genetic algorithm optimization analysis

The optimization experiments are carried out for the adaptive genetic algorithm based on the number of iterations. The results show that the traditional genetic algorithm adopts fixed crossover and mutation probability values, adaptive crossover and mutation probability values, improved adaptive crossover and mutation probability values and the dynamic crossover and mutation probability values based on the number of iterations proposed in this paper (Fig. 3).

From the experimental results, the traditional genetic algorithm has the slowest convergence speed. It requires more than 60 iterations and more than 20 iterations to obtain the global optimal solution. The adaptive genetic algorithm has a slow convergence in the early stage and can preserve the characteristics of excellent individuals in the later stage. Compared with the traditional genetic algorithm, the improved adaptive genetic algorithm converges faster in the early stage, and the higher crossover and mutation probability tend to destroy the good structure. The convergence speed is still better than the adaptive genetic algorithm. The dynamic genetic algorithm has a
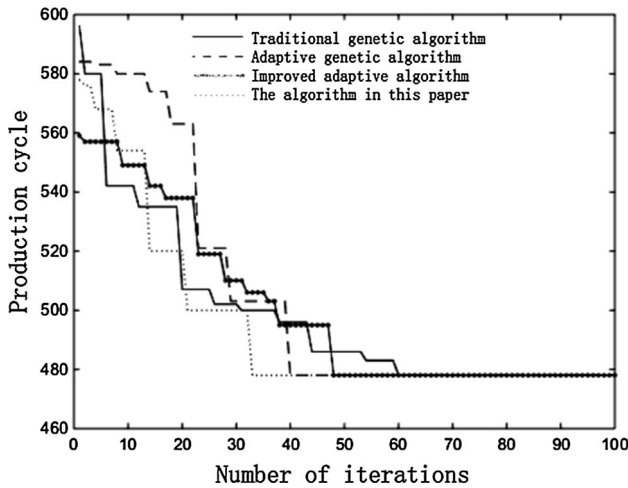
Fig. 2 Processing schematic

Processing schematic diagram (a)   BProcessing schematic diagram (b)
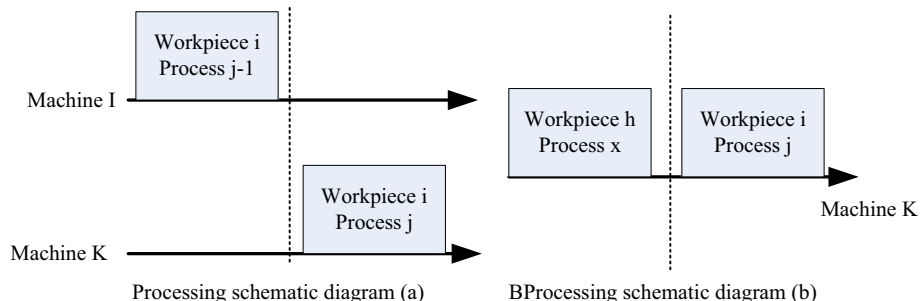
Table 1 Processing time of different workpieces on each machine (min)

Workpiece–machine	1	2	3	4	5	6	7	8	9	10	11	12	13
1	50	45	0	0	0	20	15	0	0	0	0	50	0
2	0	0	30	25	35	0	0	35	30	30	25	0	0
3	35	30	35	38	25	0	0	0	0	0	0	0	0
4	38	32	0	0	0	0	0	0	0	0	0	30	0
5	0	0	0	0	0	25	20	0	35	25	35	0	20
6	30	25	0	0	0	22	18	0	0	0	0	28	0

Fig. 3 Comparison of various genetic algorithms

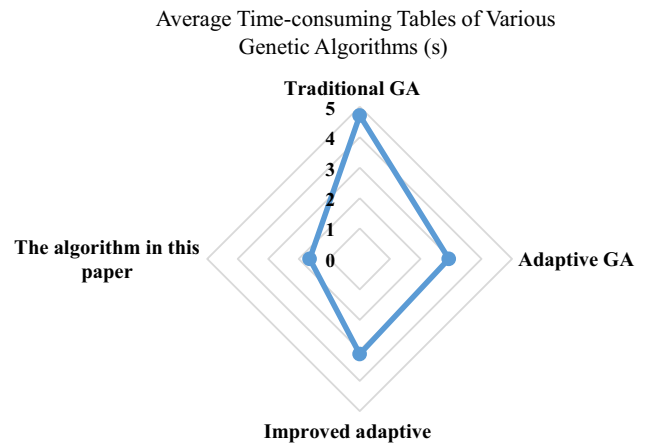Fig. 4 Average time consumption of each genetic algorithm

fast convergence performance in the early stage and can ensure that the elite individuals of the population are not destroyed in the later stage. Compared with GA, Adaptive GA and improved Adaptive GA have the fastest convergence speed and the best optimization performance. The average time of convergence of the four methods to the optimal solution is shown in Fig. 4.

The optimization of each genetic algorithm is shown in Fig. 5.

The dynamic genetic algorithm based on iteration number proposed in this paper has faster convergence speed than the other three methods. The highest efficiency error is the smallest when converges to the global optimal solution, and the dynamic adjustment of the crossover and mutation probability makes the algorithm not easy to fall into the local maximum, excellent, easier to find the global optimal solution.

4.2 Hybrid analysis of genetic algorithm and ant colony algorithm

The simulation model is simulated using actual data. There are 13 machines in total, and there are 6 kinds of workpieces. Each batch of workpieces is 5, 15, 12, 6, 10, 3, and each workpiece has up to 3 processes, and the number of iterations is 100. The cross-probability of the first encoding
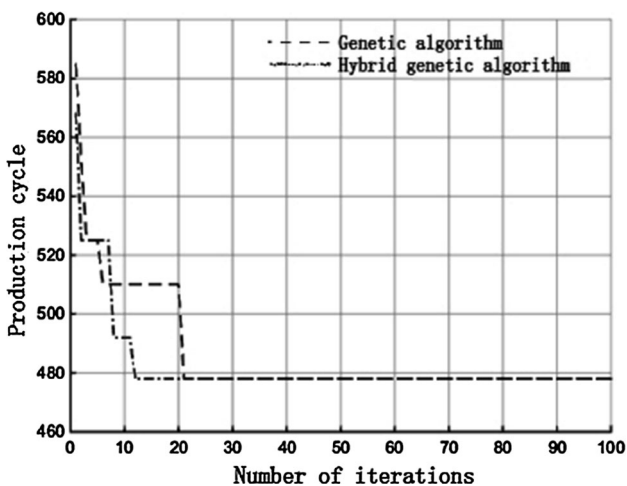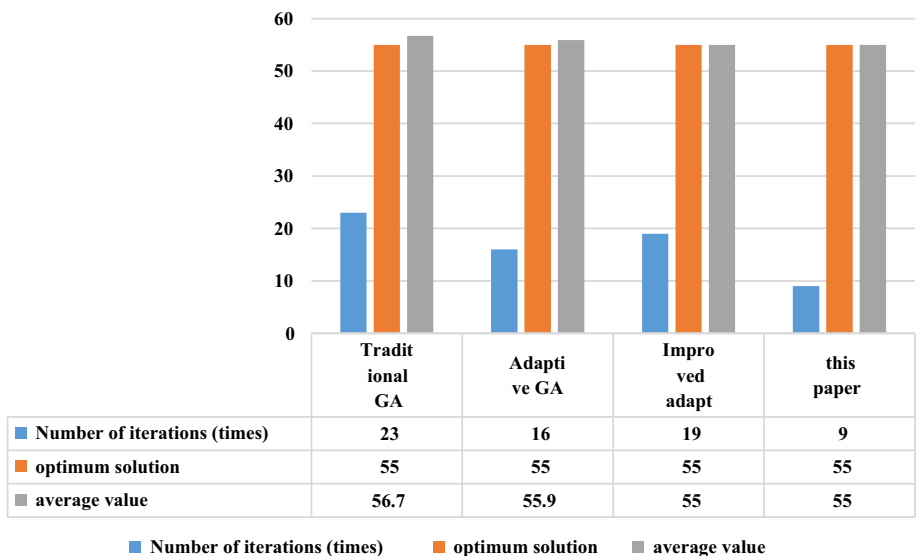
is 0.6. If the setting is too small, it will inhibit the speed of the population to generate new individuals, and it is easy to fall into the local optimum. Excessive destruction of the good individuals in the population will make the search tend to be disordered; the probability of variation is 0.001. Too small will inhibit the ability to produce new individuals, and the superior individuals generated by the destruction of the assembly. The genetic algorithm for determining the processing path adopts an adaptive genetic algorithm based on the number of iterations. The relative importance of the pheromone algorithm is 1, and the path visibility is relative. The degree of importance is 2, the initial pheromone concentration is 1, the pheromone evaporation rate is 0.7, and the number of ants is 10. The hybrid genetic and traditional genetic algorithms are compared as shown in Fig. 6. The time performance is shown in Fig. 7.
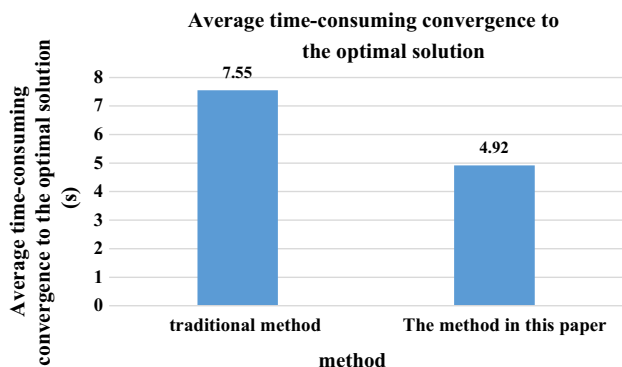
The optimal solution corresponding to each path is different. For different optimal solutions of different optimization paths, the deviation values of the two optimization modes are shown in Table 2.

As can be seen from the above table, the improved optimization method has smaller deviation values, and the efficiency of finding the optimal solution has also been greatly improved.

Fig. 5 Optimization results of
various genetic algorithms

	Tradit ional GA	Adapti ve GA	Impro ved adapt	this paper
■ Number of iterations (times)	23	16	19	9
■ optimum solution	55	55	55	55
■ average value	56.7	55.9	55	55

■ Number of iterations (times)   ■ optimum solution   ■ average value

Fig. 6 Comparison of mixed genetic and traditional genetic
algorithms

Fig. 7 Comparison of time performance between mixed genetic and
traditional genetic algorithms

5 Conclusions

1. Based on the production scheduling problem, this paper abstracts the mathematical model. According to the different parts of the same machine and the different parts of the same part, it is divided into several cases, and the corresponding processing time and waiting time are obtained, respectively. At the same time, the genetic algorithm is improved by genetic algorithm. A dynamic genetic operator based on the number of iterations is proposed, which further enhances the convergence performance and search ability of the genetic algorithm.

2. In this paper, the genetic algorithm and the ant colony algorithm are mixed. The genetic algorithm generates the initial population, and the improved coding conversion makes the information carried by the initial population more perfect. The initial pheromone is provided for the ant colony algorithm. The smaller granularity of fusion solves the most appropriate time combination, guiding each other to update the global optimal solution, so that the total production cycle is minimal. The two algorithms are fused to overcome the problem that the genetic algorithm has low search efficiency, easy to fall into local optimum, and the ant colony algorithm is difficult to grasp the global information.

3. Through the simulation of MATLAB simulation program, combined with the scheduling standard example, the performance analysis of different algorithms is carried out, the search efficiency of hybrid algorithm is improved, the convergence performance of hybrid algorithm is improved, and different time weights are obtained for different time weights. Optimize your

Table 2 Relative deviation table for optimization of two ant colony algorithms

Determining processing path	Optimum solution	Hybrid genetic algorithm		Traditional genetic algorithm	
		Relative optimal solution	Relative deviation (%)	Relative optimal solution	Relative deviation (%)
01	466	467	0.208	470	0.858
02	481	481	0	491	2.079
03	512	514	0.390	519	1.367
04	499	504	1.00	510	2.204
05	503	510	1.390	514	2.187

choice. The operation results of the system meet the requirements of production scheduling, which proves the feasibility and practicability of the hybrid genetic algorithm.

Compliance with ethical standards

Conflict of interest The authors declared that they have no conflict of interest.

References

1. Su N, Yi M, Zhang M (2017) Genetic programming for production scheduling: a survey with a unified framework. Complex & Intelligent Systems 3(1):41–66
2. Lei X, Song S, Chen X et al (2016) Joint optimization of production scheduling and machine group preventive maintenance. Reliability Engineering & System Safety 146:68–78
3. Rodammer FA, White KP (2015) A recent survey of production scheduling. IEEE Transactions on Systems Man & Cybernetics 18(6):841–851
4. Branke J, Nguyen S, Pickardt CW et al (2016) Automated design of production scheduling heuristics: a review. IEEE Transactions on Evolutionary Computation 20(1):110–124
5. Gonzalez J, Reeves G (2017) Master production scheduling: a multiple-objective linear programming approach. International Journal of Production Research 21(4):553–562
6. Shang Jianren, Tian Yunnan, Liu Yi, Liu Runlong (2018) Production scheduling optimization method based on hybrid particle swarm optimization algorithm. Journal of Intelligent and Fuzzy Systems 34(2):955–964
7. Shrouf F, Ordieres-Meré J, García-Sánchez A et al (2014) Optimizing the production scheduling of a single machine to minimize total energy consumption costs. Journal of Cleaner Production 67(6):197–207
8. Li Ming-Wei, Hong Wei-Chiang, Geng Jing, Wang Jianlun (2017) Berth and quay crane coordinated scheduling using multiobjective chaos cloud particle swarm optimization algorithm. Neural Computing and Applications 28(11):3163–3182
9. Lamghari A, Dimitrakopoulos R, Ferland JA (2014) A variable neighbourhood descent algorithm for the open-pit mine production scheduling problem with metal uncertainty. Journal of the Operational Research Society 65(9):1305–1314
10. Koo J, Kim BI (2016) Some comments on "Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency". International Journal of Advanced Manufacturing Technology 86(9):1–4
11. Gharaei Ali, Jolai Fariborz (2018) A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain. Applied Soft Computing 65:577–589
12. Chergui Akram, Hadj-Hamou Khaled, Vignat Frédéric (2018) Production scheduling and nesting in additive manufacturing. Computer and Industrial Engineering 126:292–301
13. Low C, Chang CM, Li RK et al (2014) Coordination of production scheduling and delivery problems with heterogeneous fleet. International Journal of Production Economics 153(7):139–148
14. Cui WW, Lu Z, Pan E (2014) Integrated production scheduling and maintenance policy for robustness in a single machine. Computers & Operations Research 47(7):81–91
15. Bakar MRA, Abbas IT, Kalal MA et al (2017) Solution for multiobjective optimisation master production scheduling problems based on swarm intelligence algorithms. Journal of Computational and Theoretical Nanoscience 14(11):5184–5194
16. Lamghari A, Dimitrakopoulos R (2016) Network-flow based algorithms for scheduling production in multi-processor open-pit mines accounting for metal uncertainty. European Journal of Operational Research 250(1):273–290
17. Zeng L, Liang X, Yan OU et al (2016) Energy integrated scheduling based on multiobjective-constrained optimization evolutionary algorithm. Computer Integrated Manufacturing Systems 76(2):230–267
18. Jiang SL, Liu M, Lin JH et al (2016) A prediction-based online soft scheduling algorithm for the real-world steelmaking-continuous casting production. Knowledge-Based Systems 111:159–172
19. Bierwirth C, Mattfeld DC (2014) Production scheduling and rescheduling with genetic algorithms. Evolutionary Computation 7(1):1–17
20. Wen HJ, Hou SW (2014) Application of dimensional information sharing-based particle swarm optimization algorithm for production scheduling. Industrial Engineering Journal 44(8):2769–2771
21. Jiang T, Deng G (2018) Optimizing the low-carbon flexible job shop scheduling problem considering energy consumption. IEEE Access 6:46346–46355
22. Yan HS, Wan XQ, Xiong FL (2015) Integrated production planning and scheduling for a mixed batch job-shop based on alternant iterative genetic algorithm. Journal of the Operational Research Society 66(8):1250–1258

23. Zhang YX, Xu G (2015) A research of collateral production scheduling based on genetic algorithms. In: International conference on computer modeling and simulation

24. Hai-Jun W, Shi-Wang H (2014) Application of dimensional information sharing-based particle swarm optimization algorithm for production scheduling. Ind Eng J 17(1):30–35

25. Soares MM, Vieira GE (2014) A new multi-objective optimization method for master production scheduling problems based on genetic algorithm. International Journal of Advanced Manufacturing Technology 41(5–6):549–567