BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 70(4), 2022, Article number: e141724
DOI: 10.24425/bpasts.2022.141724

CONTROL AND INFORMATICS

Research on optimization of unrelated parallel machine scheduling based on IG–TS algorithm

Xinfu CHI*, Shijing LIU, and Ce LI

Dong Hua University, College of Mechanical Engineering, Shanghai 201620, China

Abstract. This issue is a typical NP-hard problem for an unrelated parallel machine scheduling problem with makespan minimization as the goal and no sequence-related preparation time. Based on the idea of tabu search (TS), this paper improves the iterative greedy algorithm (IG) and proposes an IG–TS algorithm with deconstruction, reconstruction, and neighborhood search operations as the main optimization process. This algorithm has the characteristics of the strong capability of global search and fast speed of convergence. The warp knitting workshop scheduling problem in the textile industry, which has the complex characteristics of a large scale, nonlinearity, uncertainty, and strong coupling, is a typical unrelated parallel machine scheduling problem. The IG–TS algorithm is applied to solve it, and three commonly used scheduling algorithms are set as a comparison, namely the GA–TS algorithm, ABC–TS algorithm, and PSO–TS algorithm. The outcome shows that the scheduling results of the IG–TS algorithm have the shortest manufacturing time and good robustness. In addition, the production comparison between the IG–TS algorithm scheduling scheme and the artificial experience scheduling scheme for the small-scale example problem shows that the IG–TS algorithm scheduling is slightly superior to the artificial experience scheduling in both planning and actual production. Experiments show that the IG–TS algorithm is feasible in warp knitting workshop scheduling problems, effectively realizing the reduction of energy and the increase in efficiency of a digital workshop in the textile industry.

Key words: warp knitting machine; parallel machine scheduling; iterative greedy algorithm; tabu search.

1. INTRODUCTION

The manufacturing industry is the pillar of the national economy. However, there are imbalances in the flow of material, energy, capital, and data in the production process of traditional manufacturing workshops. It has become a trend to build information-based, intelligent, and efficient digital workshops to achieve optimal scheduling. Production scheduling optimization is the core of the digital workshop and is the key to improving and increasing efficiency and enhancing the competitiveness of enterprises. Unrelated parallel machine scheduling problem (UPMSP) is typical for the production processes of the textile industry, electronic manufacturing, and machining [1]. Garey, Johnson, and other researchers [2] confirmed that the scheduling problem of the realization of makespan minimization on a parallel machine is an NP-hard complete problem. For the scheduling problem of warp knitting workshops in the textile industry, due to the different processing and production times of the same cloth on different warp knitting machines, the processing time depends on the matching relationship between the workpiece and the machine. Therefore, this kind of issue is an unrelated parallel machine scheduling problem. This problem studies the processing and distribution process of n cloth orders on m warp knitting machines. And only one processing procedure is required for each cloth order. The warp knitting workshop scheduling problem is characterized by large scale, strong constraints, nonlinearity, multiple minima, strong coupling, and uncertainty.

In recent years, many scholars at home and abroad have studied the unrelated parallel machine scheduling problem. Bożejko et al. [3] propose the introduction of new elimination block properties allowing for accelerating the operation of approximate algorithms of local searches, solving this problem, and improving the quality of solutions determined by them. Suresh et al. [4] proposed a new hybrid meta-heuristic algorithm called hybrid pathfinder algorithm (HPFA) to solve the optimal reactive power dispatch (ORPD) problem. Ghaith Rabadi et al. [5] proposed a meta-heuristic-RaPS algorithm for large-scale non-preemptive unrelated parallel machine scheduling problems, effectively reducing makespan. Considering the uncertainties of processing time and delivery time, Torabi et al. [6] proposed an effective multi-objective particle swarm optimization (MOPSO) algorithm to minimize makespan and total tardiness for the unrelated parallel machine scheduling problem. Ridvan Gedik et al. [7] proposed a new constraint programming (CP) model for this problem, proving that this model had a better solving ability than all other algorithms on small-scale problems. Luis Fanjul-Peyro et al. [8] suggested a new mixed-integer linear programming algorithm (MILP) and compared the results with the widely used CAM-PAGEN algorithm. They found that the relative deviation was less than 0.8 % in large-scale problems. Eva Vallada et al. [9] proposed a new search algorithm based on the improved discrete search algorithm and the iterative greedy algorithm. Its calculation results for the same historical problem were superior to the optimal solution in history. There is relatively less research on textile in-

*e-mail: xinfu.chi@dhu.edu.cn

Manuscript submitted 2022-01-20, initially accepted for publication 2022-05-31, published in August 2022.

dustry scheduling at home and abroad. Combining local search algorithms, Gao [10] proposed a diversity technology based on a vector group. And the simulation results of it were better than that of the vector immune genetic algorithm (VIGA). These scheduling algorithms are mostly based on heuristic algorithms and require a long solving time. In actual production, as the order production has strong uncertainty and randomness, the scheduling algorithm needs to respond quickly to the demand. Based on the idea of TS (tabu search) and IG (iterative greedy algorithm), this paper proposes an IG–TS hybrid algorithm, which can quickly schedule the production orders of textile enterprises, solve the problem of production scheduling difficulty caused by frequent orders, and optimize the production period and the utilization rate of the machine.

2. PROBLEM DESCRIPTION

2.1. Symbol definition

n number of cloth jobs to be produced;

m number of warp knitting machines;

$J = \{J_1, J_2, \dots, J_i, \dots, J_n\}$ set of cloth to be produced, J_i is the i-th cloth job;

$M = \{M_1, M_2, \dots, M_k, \dots, M_m\}$ set of warp knitting machines, M_k is the k-th warp knitting machine;

$C_{\max}$ maximum completion time of all cloth production;

C_{last} maximum completion time of the last scheduled cloth production;

C_i completion time of cloth J_i;

$T_{i,k}$ the i-th shift in the future, production capacity of warp knitting machine M_k;

$x_{i,j,k}$ if J_j is next to J_i and is processed on M_k, then the number is 1, otherwise 0;

V a large enough positive real number;

R_k release time of M_k;

π final feasible scheduling solution.

2.2. Mathematical model

The model of warp knitting workshop scheduling problem is described as follows: n cloth jobs are processed on m unrelated warp knitting machines, and each cloth job only needs to complete one process. In addition, the following assumptions are made: each job needs to be executed once, and only one job can be executed on one machine at the same time. The job release time is 0, and the job is not allowed to be interrupted or preemptible after its start. In this paper, the problem of minimizing makespan is studied. And the mixed mathematical model [11] is as follows:

$$\operatorname{Min} C_{\max} = \max\{C_i | i = 1, 2, \dots, n\}, \tag{1}$$

$$s.t. \sum_{i \neq j}^{n} i = 0 \sum_{k=1}^{m} x_{i,j,k} = 1, \forall j = 1, 2, \dots, n, \tag{2}$$

$$\sum_{\substack{i=0 \\ i \neq h}}^{n} x_{i,h,k} = \sum_{\substack{j=0 \\ j \neq h}}^{n} x_{h,j,k}, \forall h = 1, 2, \dots, n, \ \ \forall k = 1, 2, \dots, m, \tag{3}$$

$$C_j \geq C_i + \sum_{k=1}^{m} x_{i,j,k} \left(ST_{i,j,k} + t_{j,k}\right) + V \left(\sum_{k=1}^{m} x_{i,j,k} - 1\right), \quad \forall i, j = 0, 1, \dots, n, \tag{4}$$

$$\sum_{j=0}^{n} x_{0,j,k} = 1, \ \forall k = 1, 2, \dots, m, \tag{5}$$

$$x_{i,j,k} \in \{0, 1\}, \ \forall i, j = 0, 1, \dots, n; \ \forall k = 1, 2, \dots, m, \tag{6}$$

$$C_j \geq 0, \ \forall j = 1, 2, \dots, n. \tag{7}$$

Equation (1) is the objective function of the problem. Equation (2) means that each cloth job needs to be processed and can only be processed once by a warp knitting machine. Equation (3) indicates that each cloth job has at most one preorder job and one post-order job. Equation (4) means that each cloth job can only be processed after the completion of the preorder job, and V can ensure that the inequality is permanent. Equation (5) indicates the uniqueness of the first job of each warp knitting machine. Equation (6) represents the value range of decision variables. Equation (7) indicates that the completion time of all cloth jobs is non-negative.

3. SCHEDULING ALGORITHM DESIGN OF IG–TS

The iterative greedy (IG) algorithm is an algorithm with strong local search ability, which has been successfully applied to UPMSP scheduling problems without sequence-related preparation time [12]. The IG algorithm continuously searches in the neighborhood of the current solution based on the greedy idea. The algorithm is general and easy to implement, but this algorithm is easy to fall into a minimum and cannot guarantee global optimization. The tabu search (TS) algorithm is a global iterative optimization algorithm with strong local search ability.

However, the TS algorithm is mainly based on neighborhood search, which is highly dependent on the initial solution. An effective and reasonable initial solution helps the search achieve the optimal solution quickly.

The IG–TS scheduling algorithm proposed in this paper combines the global search ability of TS and the local search ability of IG to solve the highly feasible warp knitting workshop scheduling solution. Based on these two algorithms, the IG–TS algorithm can draw the advantages of both algorithms. First, the appropriate initial scheduling solution is calculated by the IG algorithm, and then the TS algorithm is used to search for the global optimal solution. The flow of the IG–TS scheduling algorithm is shown in Fig. 1.

The IG–TS algorithm uses iterative greedy search to find a feasible scheduling solution, which can avoid roundabout search and achieve fast global optimal convergence. The important processing steps of each iteration include the deconstruction stage, reconstruction stage, and neighborhood search stage. The IG–TS algorithm mainly includes the following 12 processes:

1. Production information initialization. The cloth job to be produced is n. The number of warp knitting machines available for processing is m. The production capacity of warp knitting machines is $T_{i,k}$ The tabu list is null set Φ.

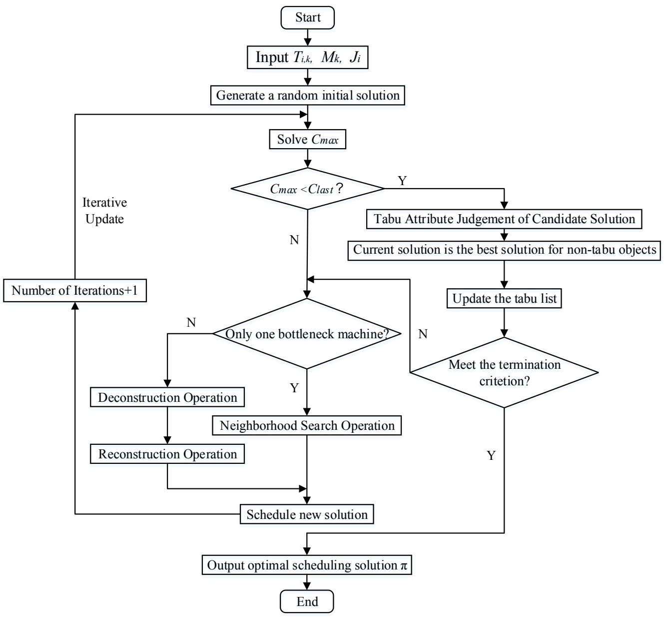

Fig. 1. IG–TS algorithm flow chart

2. Design of fitness function. The fitness function of the IG–TS algorithm is as follows:

$$C_{\text{diff}} = C_{\max} - C_{\min}.$$

   In this equation, $C_{\max}$ is the longest processing time in warp knitting machine; $C_{\min}$ is the shortest processing time in a warp knitting machine.
3. Design of termination criterion. The maximum number of iterations is set to 500, and the termination threshold of the fitness value is set to 100 min. When makespan remains unchanged for 200 consecutive times, the search is terminated when either of the above criteria is met.
4. Generates an initial solution. Randomly assign cloth production set J to warp knitting machine set M.
5. Update tabu list. If the fitness function value of the current solution is superior to that of the previous generation, the current solution is updated to the tabu list, and otherwise, the current solution is maintained.
6. Termination criterion judgment. If the termination criterion is reached, then jump to step 12, otherwise continue to execute.
7. Judgment of the number of bottleneck machines. When the scheduling solution falls into a key bottleneck machine, the neighborhood search is performed (jump to step 10). Otherwise, it will continue to execute.
8. Deconstruction stage. In order to achieve faster convergence, at each iteration, a job is randomly taken out on the machine with the longest processing time.
9. Reconstruction stage. Put each removed job on each machine and then select the machine that can make the makespan minimum into the operation.
10. Neighborhood search stage. Neighborhood search operations are mainly divided into intra-machine operations and inter-machine operations. Common neighborhood search methods include interchange, interpolation, and inversion.

This paper designs the following four search operations:

a. Intra-machine operations are as follows: without loss of generality, let $j > i$. The detailed operation steps are shown in Figs. 2 and 3:
   - $\text{SWS}(i, j, k)$: Exchange the location of the i-th and j-th jobs on the table of the machine M_k.
   - $\text{ISS}(i, j, k)$: Place the i-th job on the machine M_k after the j-th job on the same machine.

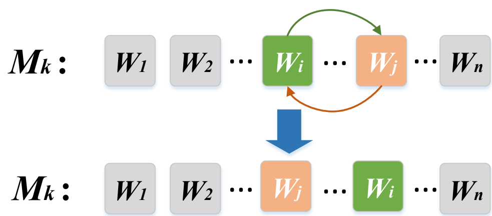

Fig. 2. "SWS" neighborhood search

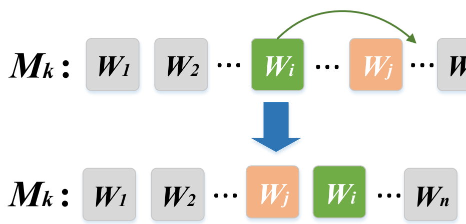

Fig. 3. "ISS" neighborhood search

b. Inter-machine operations are as follows: $k_1 \neq k_2$. The detailed operation steps are shown in Figs. 4 and 5:
   - $\text{SWD}(i, j, k_1, k_2)$: Exchange the i-th job on the table of the machine M_{k_1} with the j-th job on the table of the machine M_{k_2}.
   - $\text{ISD}(i, j, k_1, k_2)$: Put the i-th job on machine M_{k_1} after the j-th job on machine M_{k_2}.

11. Update the scheduling solution. Substitute the new scheduling solution to the program and return to step 5;
12. Output optimal scheduling solution π.

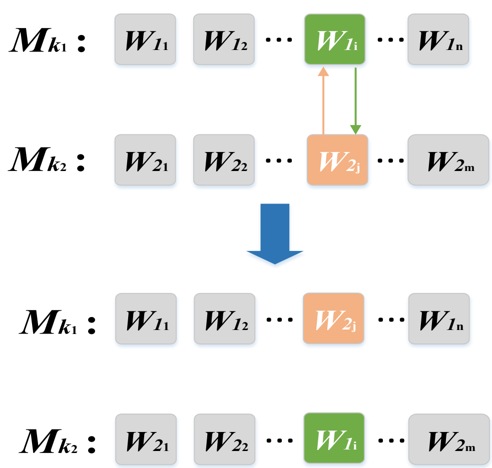

Fig. 4. "SWD" neighborhood search

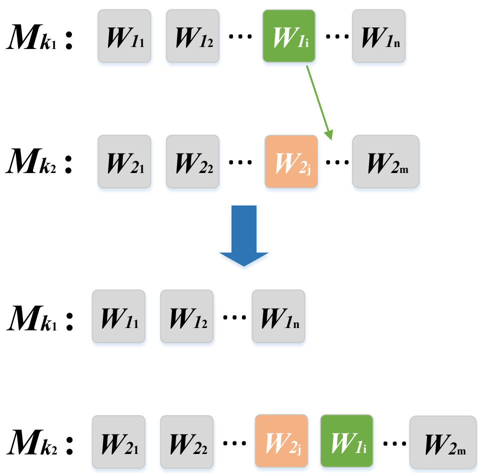

Fig. 5. "ISD" neighborhood search

Adopt O Representation to present the algorithm complexity of the IG–TS algorithm. The complexity of the deconstruction stage is $O(n)$. The complexity of the reconstruction stage is $O(n)$. The complexity of solving processing time is $O(mn)$. The complexity of the neighborhood search stage is $O(n^2)$. Therefore, the algorithm complexity of the IG–TS algorithm is $O(mn+n^2)$.

4. CONTRAST ALGORITHM DESIGN

4.1. GA–TS algorithm

A genetic algorithm is an intelligent bionic algorithm based on the biological idea of "survival of the fittest." This algorithm has been widely studied and applied in the field of job shop scheduling [13–15]. GA has the ability of parallel search and can quickly converge to the optimal or suboptimal solution from multiple points in the solution space, which is suitable for solving multiple combinatorial optimization problems. However, the local search ability of GA is poor, and it is easy for this algorithm to converge prematurely. As mentioned above, the TS algorithm has a unique memory function that enables GA to jump out of the current local optimal solution and achieve global optimal optimization. The GA–TS hybrid algorithm based on the two algorithms also achieves complementary advantages. In recent years, scholars and researchers at home and abroad have also made many studies and improvements to this algorithm. Zhang Huizhen et al. [16] proposed the EGA–TS hybrid algorithm based on an elite genetic algorithm and tabu search algorithm to solve the quadratic allocation problem (QAP), the classical combinatorial optimization problem, and compared it with the solutions of other mainstream algorithms. The results proved that the EGA–TS hybrid algorithm was feasible and competitive. Lin Boliang and other researchers [17] managed to use the GA–TS algorithm to solve the scheduling problem of railway congestion. They compared the solution of the GA–TS algorithm with the optimal solution obtained by the enumeration method and the optimal solution obtained by commercial optimization software, confirming the feasibility and advantages of this algorithm. Sukkerd [18] successfully applied the GA–TS algorithm to solve the flexible job-shop scheduling problem (FJSP). Thongwan et al. [19] combined the conditional genetic algorithm (CGA) and conditional tabu search algorithm (CTSA) and then proposed the hybrid algorithm, effectively reducing the frequency of multi-objective future flood disasters of reservoirs.

Based on the research results of the above scholars and researchers, this paper grasps their core ideas and designs the following GA–TS hybrid algorithm as one of the comparison algorithms of the IG–TS algorithm. The algorithm flow chart is shown in Fig. 6.

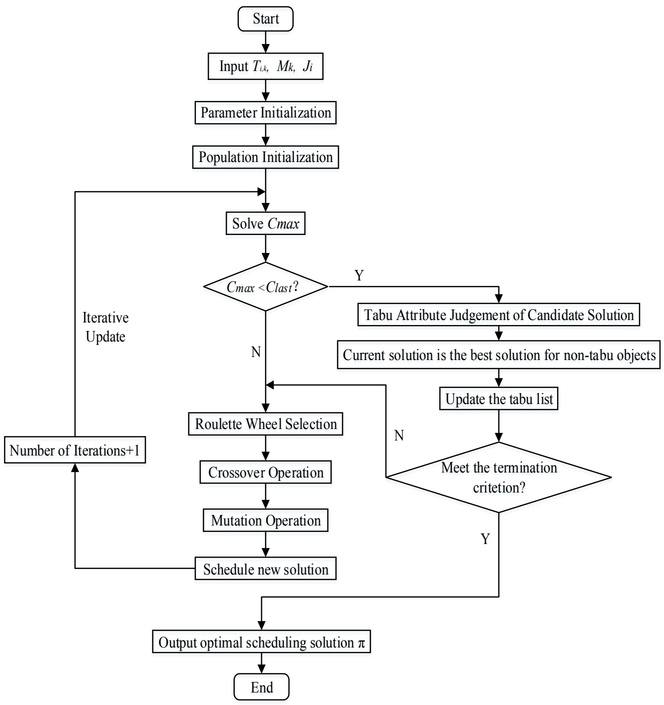

Fig. 6. GA–TS algorithm flow chart

4.2. ABC–TS algorithm

The artificial bee colony algorithm is a swarm intelligence optimization algorithm that imitates the honey-harvesting process of honeybees. This algorithm has also been widely studied and applied in the field of job shop scheduling [20,21]. Although the ABC algorithm has made some achievements in theoretical research and practical application, the neighborhood searchability of the standard ABC algorithm is not extraordinarily strong and lacks clarity in the search process. Although the scout bee designed in the algorithm can help the solution process jump out of the local optimum, the lead bee may repeat the search in the search process, which makes the artificial bee colony algorithm easy to fall into a local optimum when solving large-scale problems. Scholars at home and abroad have proposed using a tabu search algorithm to solve this problem by introducing a tabu list to store the optimal solution in the search process, which enables the ABC algorithm to have the function of memory and improve the efficiency, accuracy, and robustness of search. Su et al. [22] devised an artificial bee colony algorithm with variable neighborhood search and tabu list (ABC-VNS2019TL) to solve the scheduling problem of the urban carpooling system. Lu et al. [23] used the ABC–TS algorithm to solve the unrelated parallel machine scheduling problem to minimize makespan time, and made a comparison among ABC, TS, and PSO algorithms to prove the effectiveness and robustness of the ABC–TS algorithm.

Based on the research results of the above scholars and researchers, this paper grasps their core idea and designs the following ABC–TS hybrid algorithm as one of the comparison algorithms of the IG–TS algorithm. The algorithm flow chart is shown in Fig. 7.

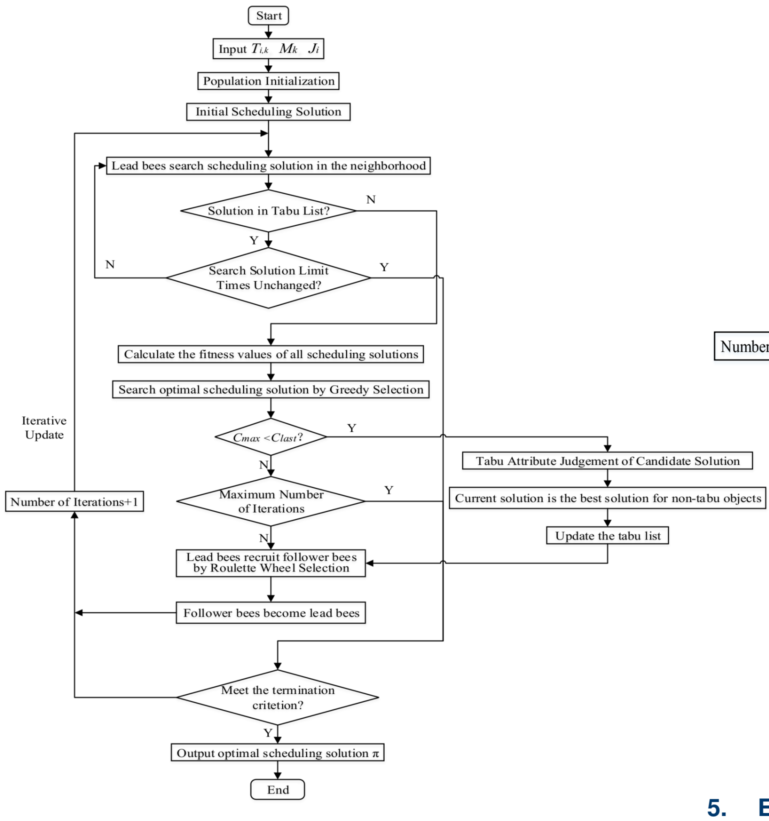

Fig. 7. ABC–TS algorithm flow chart

4.3. PSO–TS algorithm

The particle swarm optimization (PSO) algorithm is a bionic algorithm based on the predation of a group of birds, which searches for the optimal solution through cooperation and information sharing among individuals in the group. At present, this algorithm has also been widely studied and applied in the field of job shop scheduling [24–26]. However, this algorithm easily falls into the local optimal solution and its global optimization ability is poor. In view of this, scholars at home and abroad turned to the tabu search algorithm. They used a tabu list to store the collected local optimal solution, which can effectively avoid particle swarm optimization falling into a local optimal cycle, realize the diversity of the particle swarm optimization search path, and is beneficial to the global optimal convergence of the particle swarm optimization algorithm. Ibrahim Alharkan et al. [27] solved the scheduling problem of two unrelated parallel machines using tabu search algorithm and particle swarm optimization algorithm. Adopting hybrid binary particle swarm optimization with tabu search (HBPSO–TS), Lin Geng et al. [28] effectively solved the complex set-union knapsack problem (SUKP) and confirmed the feasibility of particle swarm optimization with tabu search.

Based on the research results of the above scholars and researchers, this paper grasps their core idea and designs the following PSO–TS hybrid algorithm as one of the comparison algorithms of the IG–TS algorithm. The algorithm flow chart is shown in Fig. 8.

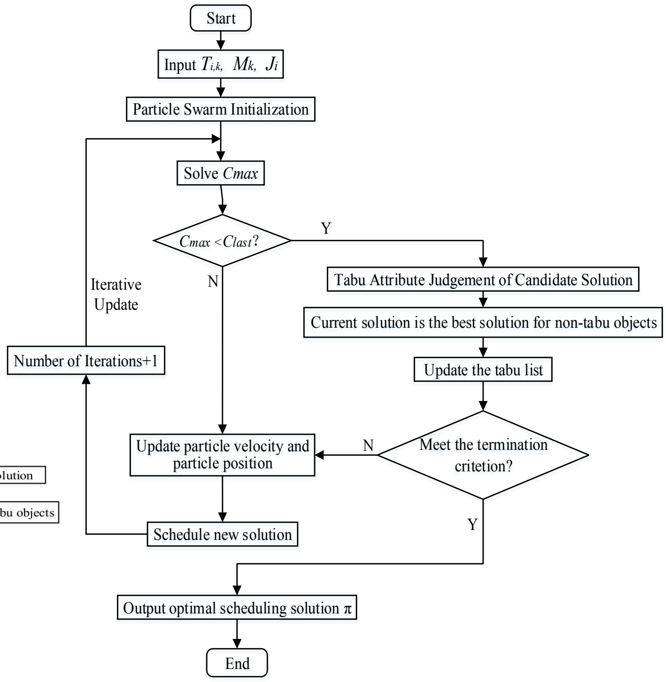

Fig. 8. PSO–TS algorithm flow chart

5. EXPERIMENTAL STUDY

5.1. Algorithm comparison tests

To sum up, algorithms of IG–TS, GA–TS, ABC–TS, and PSO–TS are implemented based on JavaScript. The runtime environment is Node.js, v10.16.3. And the processor is a 2.4 GHz CPU.

The number of warp knitting machines available for scheduling in warp knitting workshops is 50. The production capacity of these machines in the next five shifts is shown in Table 1. There are seven types of production operations to be scheduled. And the specific cloth length is shown in Table 2.

Algorithms of IG–TS, GA–TS, ABC–TS, and PSO–TS are used to solve the scheduling of these cloth orders to be produced. In order to ensure the fairness of the algorithm, the same iteration termination condition is set, which terminates when the fitness function value remains unchanged for 200 consecutive times. Each example of each algorithm is evaluated 10 times, and the optimal solution (Min), average manufacturing time (Ave), and standard deviation (S) of the results are compared and analyzed.

Table 1
The production capacity in the next five shifts

Machine ID	Production capacity in future shifts $T_{i,k}$ (min/m)					Machine ID	Production capacity in future shifts $T_{i,k}$ (min/m)				
	$T_{1,k}$	$T_{2,k}$	$T_{3,k}$	$T_{4,k}$	$T_{5,k}$		$T_{1,k}$	$T_{2,k}$	$T_{3,k}$	$T_{4,k}$	$T_{5,k}$
1	6	8	7	7	7	26	18	36	8	6	7
2	13	11	12	8	28	27	11	8	7	7	8
3	49	28	11	21	8	28	13	9	11	10	34
4	16	24	37	27	50	29	13	16	9	8	7
5	50	51	25	10	10	30	12	36	7	9	6
6	6	24	15	6	7	31	8	9	13	15	23
7	19	22	11	11	8	32	13	26	30	9	16
8	13	12	12	10	25	33	13	21	36	18	16
9	14	17	10	15	9	34	7	9	9	8	19
10	13	19	18	6	12	35	13	12	16	20	23
11	16	23	7	16	12	36	19	9	6	8	11
12	9	23	26	25	36	37	13	9	38	53	6
13	9	23	11	23	15	38	13	15	23	9	8
14	8	7	27	17	23	39	9	8	6	16	13
15	26	15	26	13	8	40	19	36	8	9	13
16	34	21	15	9	23	41	12	21	43	15	20
17	9	8	7	6	8	42	25	48	51	10	23
18	13	20	9	18	36	43	13	21	10	8	6
19	10	8	9	26	56	44	7	13	8	7	9
20	23	7	6	8	16	45	53	23	17	10	16
21	8	7	6	11	9	46	35	9	16	22	11
22	26	11	13	8	8	47	13	12	8	9	16
23	8	7	10	9	8	48	43	8	12	23	27
24	36	26	37	17	13	49	19	27	35	26	62
25	9	8	6	8	30	50	53	19	8	8	52

Table 2
Cloth job order information form

Number	Number of orders	Type A	Type B	Type C	Type D	The total length of cloth/m
1	20	5	6	5	4	800
2	30	6	8	10	6	1320
3	40	10	8	8	14	2060
4	50	15	9	10	16	2430
5	100	26	28	22	24	4320
6	150	40	36	40	34	6520
7	200	63	45	44	48	8530

Note: Orders of type A are 10 meters, type B 20 meters, type C 50 meters, and type D 100 meters.

In this paper, the relative percentage deviation (RPD) between the scheduling results of other algorithms and the scheduling results of the IG–TS algorithm is used to evaluate the performance of the algorithm. The calculation method of the RPD value of each test example is shown in equation (8):

$$\mathrm{RDP} = \frac{C_c - C_{\mathrm{IG}}}{C_{\mathrm{IG}}} \times 100\%. \tag{8}$$

In this equation, C_c is the optimal $C_{\max}$ solved by contrast algorithms and C_{IG} is the optimal $C_{\max}$ solved by the IG–TS hybrid algorithm.

As there are five shifts of the production capacity in the future, the acceptable initial maximum fitness function is set to be 3600 minutes. The iterative convergence curves of all examples of the four algorithms are shown in Fig. 9. It can be seen from the figure that these algorithms all converge at a faster

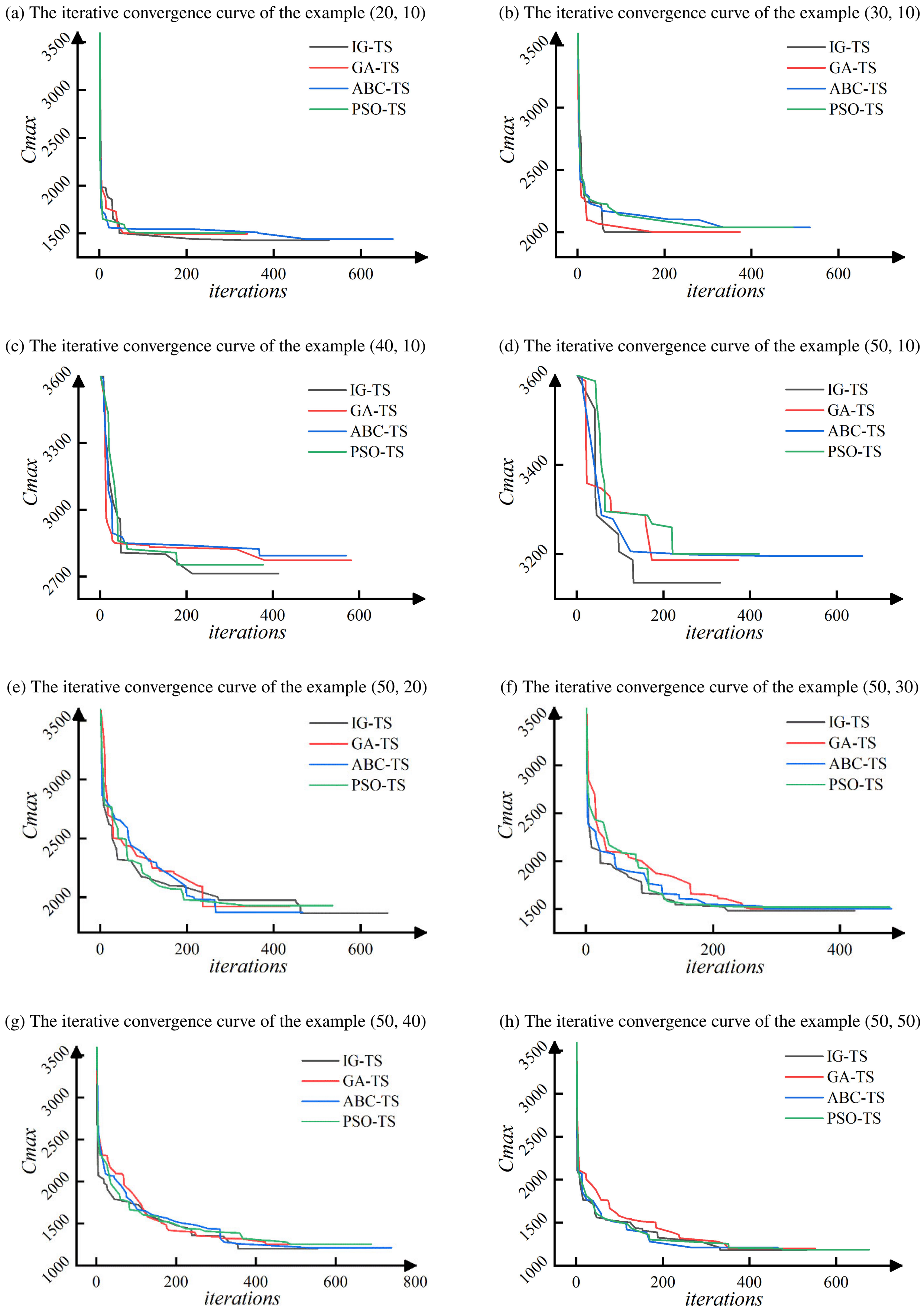
(a) The iterative convergence curve of the example (20, 10)
(b) The iterative convergence curve of the example (30, 10)
(c) The iterative convergence curve of the example (40, 10)
(d) The iterative convergence curve of the example (50, 10)
(e) The iterative convergence curve of the example (50, 20)
(f) The iterative convergence curve of the example (50, 30)
(g) The iterative convergence curve of the example (50, 40)
(h) The iterative convergence curve of the example (50, 50)
IG-TS
GA-TS
ABC-TS
PSO-TS
Cmax
iterations

Fig. 9 a–h

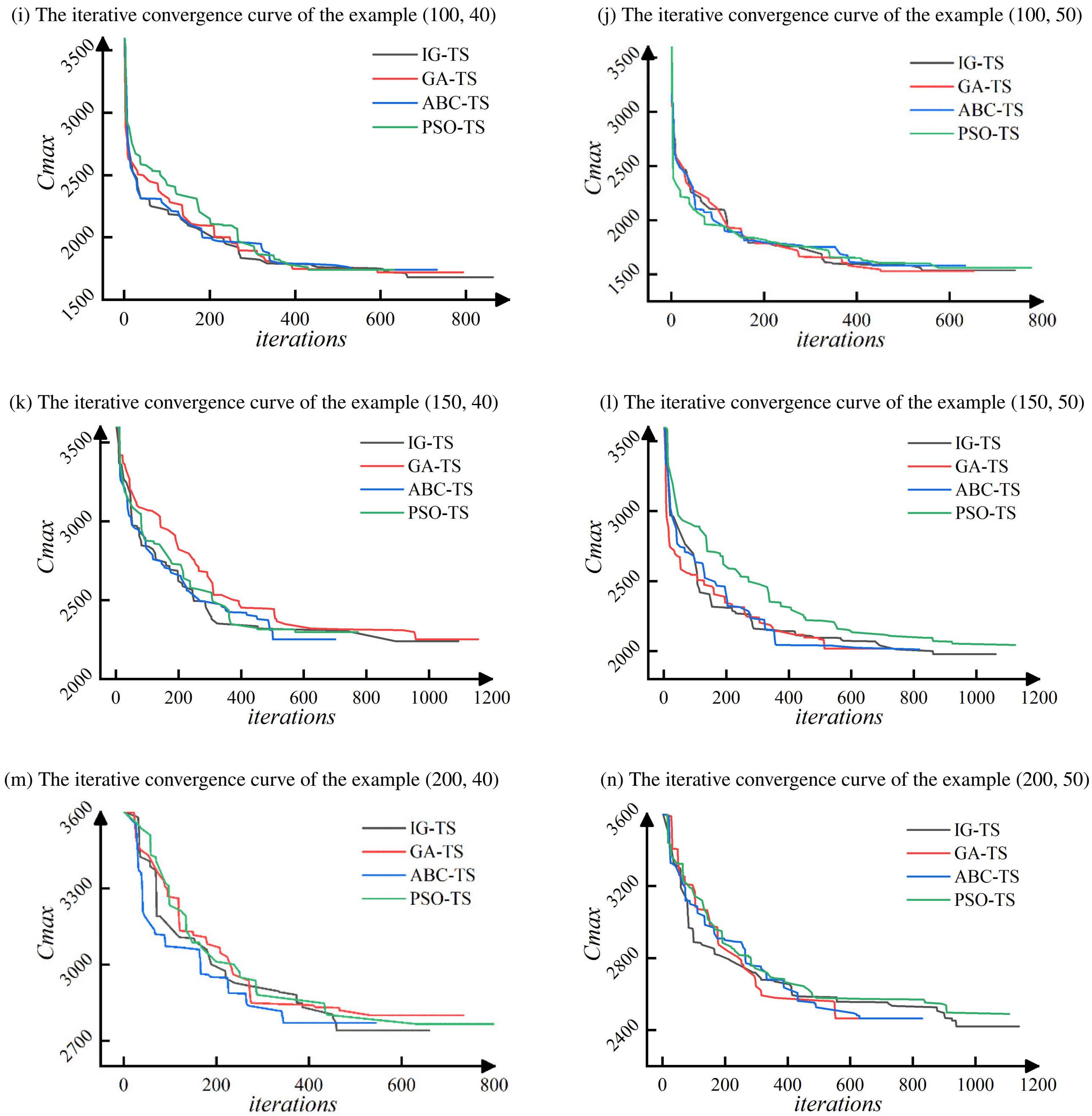

Fig. 9. Iterative convergence curve of four algorithms

rate, which results from the tabu search algorithm avoidance of roundabout search, whereas the IG–TS algorithm usually converges to a better $C_{\max}$.

The RPD values of the four algorithms are compared as shown in Fig. 10, and the results of the comparison of scheduling results of calculation examples are shown in Table 3. It can be seen from Fig. 10 that in almost all test examples, the RPD values of GA–TS, ABC–TS, and PSO–TS algorithms are all above IG–TS, which also means that the IG–TS algorithm can obtain better processing and manufacturing time. It can be seen from Table 3 that the standard deviation of the IG–TS algorithm is smaller than that of the other three algorithms and that this algorithm can converge to the global optimal solution more stably and has better robustness.

5.2. Production comparison tests

Figure 11 shows the Gantt chart of the scheduling results of the above small-scale example problems, which displays the scheduling information of orders. The scheduling solution is applied to the warp knitting workshop of a textile company in Fujian Province, China, to verify the effectiveness of scientific scheduling. At the same time, as a comparison, workshop production plan makers are required to schedule the production of small-scale examples based on the artificial experience.

Table 3
Comparison of scheduling results of calculation examples

Example scale	n, m	IG–TS			GA–TS			ABC–TS			PSO–TS		
		Min	Ave	S	Min	Ave	S	Min	Ave	S	Min	Ave	S
Small-scale examples	20,10	**1428.57**	1481.71	30.01	1495.38	1505.65	33.22	1440.00	1449.22	48.10	1502.18	1528.58	54.63
	30,10	**2002.18**	2058.07	45.73	2002.18	2019.03	46.22	2040.00	2135.84	57.51	2040.00	2065.54	46.69
	40,10	**2713.29**	2796.15	42.99	2772.00	2821.72	78.48	2793.29	2826.15	79.26	2753.16	2840.29	72.29
	50,10	**3136.50**	3217.09	48.52	3186.82	3203.33	57.77	3195.38	3268.92	60.40	3200.65	3238.08	72.18
Large-scale examples	50,20	**1865.65**	1937.51	38.21	1920.97	1978.34	50.66	1872.00	1880.30	75.82	1930.00	1952.15	44.14
	50,30	**1482.86**	1501.89	8.46	1502.18	1561.16	9.35	1502.18	1571.99	13.51	1517.54	1586.30	9.34
	50,40	**1200.87**	1245.55	19.82	1255.38	1351.19	20.24	1211.54	1258.69	24.08	1255.38	1273.04	31.38
	50,50	**1180.00**	1202.49	20.62	1200.87	1217.21	22.58	1210.77	1215.32	32.59	1185.38	1211.35	34.59
	100,40	**1680.00**	1714.83	24.46	1720.00	1817.91	39.42	1740.00	1815.67	25.73	1740.00	1741.58	31.67
	100,50	1536.84	1563.14	22.68	**1529.09**	1562.96	34.75	1580.00	1590.39	44.03	1560.00	1569.32	26.09
	150,40	**2240.00**	2277.09	28.43	2251.61	2317.99	48.96	2251.75	2314.48	38.18	2296.29	2308.47	48.29
	150,50	**1977.06**	2029.29	31.69	2016.84	2072.72	55.79	2011.43	2096.31	49.41	2044.29	2057.22	48.29
	200,40	**2740.32**	2790.14	28.98	2800.00	2830.88	36.05	2770.00	2827.50	53.77	2766.00	2826.56	40.46
	200,50	**2464.88**	2470.65	30.99	2477.43	2520.69	45.40	2464.88	2482.73	39.14	2490.00	2562.98	37.76

Note: Data in bold is the optimal scheduling solution.

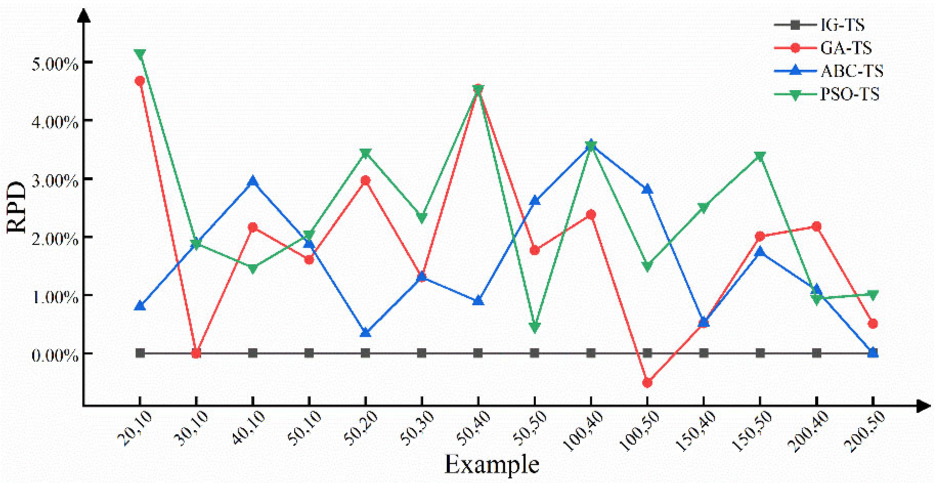

Fig. 10. Comparison of RPD values of the four algorithms

Table 4
Comparison of algorithm and experience scheduling results

n, m	IG–TS		Production based on experience	
	t_{plan}/min	t_{make}/min	t_{plan}/min	t_{make}/min
20, 10	0.50	1502.69	34.50	1658.08
30, 10	1.22	1980.85	48.59	2669.25
40, 10	1.34	2793.25	61.25	4160.58
50, 10	1.50	3085.68	80.20	5865.36

Note: t_{plan} is the planned consuming time of production plan, t_{make} is the consuming time of the production of examples.

After actual tests, the results shown in Table 4 are obtained. It can be found that the production scheduling based on the IG–TS algorithm consumes less than three minutes in terms of the time-consuming production planning, while the manual scheduling takes at least half an hour. The actual production completion time of the example is close to the scheduling results of the IG–TS algorithm, while the scheduling based on artificial experience takes more time. Thus, it can be concluded that scientific scheduling is particularly important for order-oriented enterprises such as textile enterprises that can effectively reduce energy consumption and costs, save resources, and optimize the production period with the adoption of scientific scheduling.

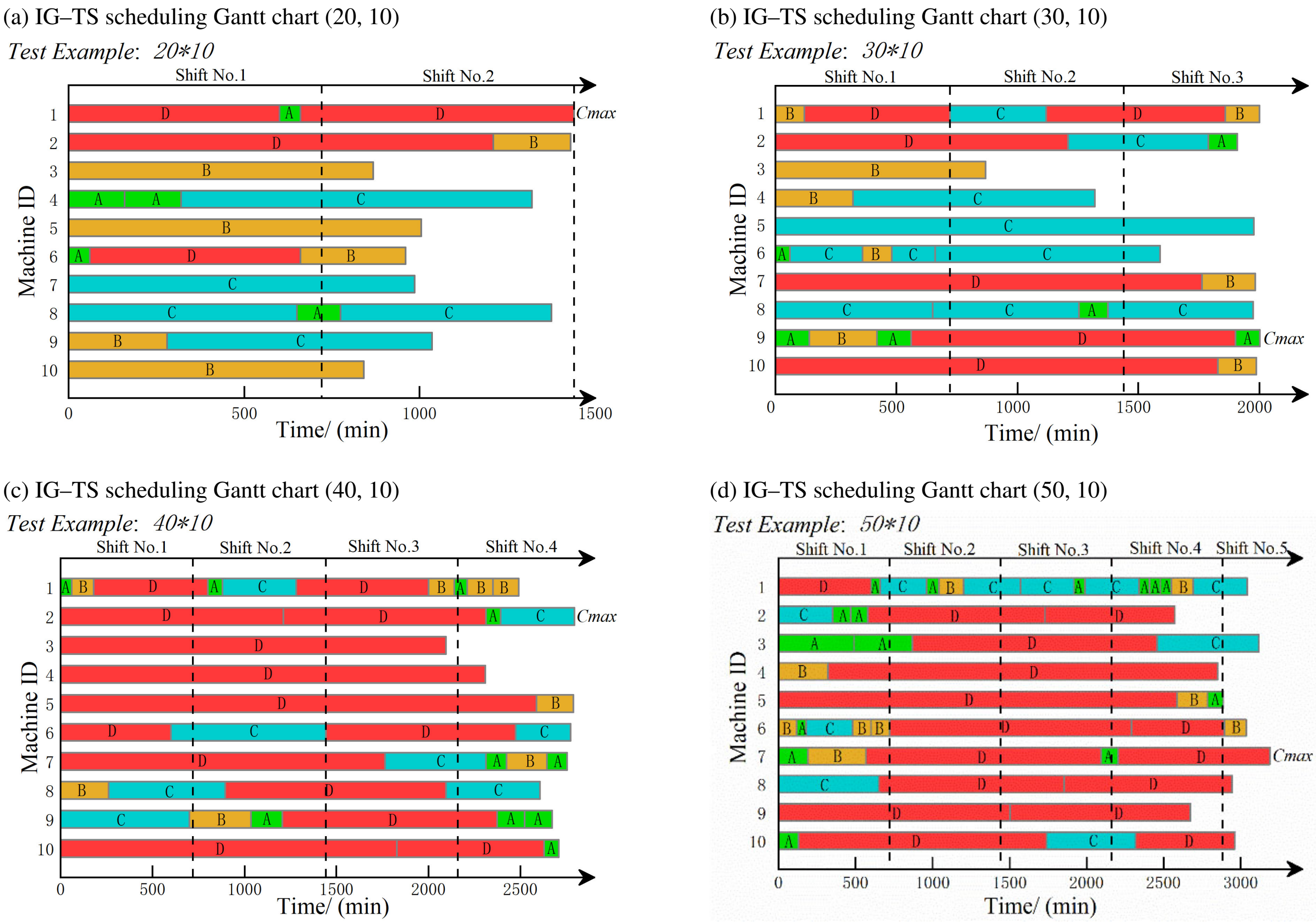

Fig. 11. Gantt chart for small-scale examples

6. CONCLUSIONS

In this paper, the iterative greedy (IG) algorithm is improved based on the idea of tabu search, and the IG–TS hybrid algorithm is proposed for solving the minimum makespan of the unrelated parallel machine scheduling problem without sequence-related preparation time. The IG–TS algorithm can avoid roundabout search and approach the global optimal solution with an amazingly fast convergence speed. The complexity of the algorithm is $O\left(mn+n^2\right)$. This algorithm is applied to the warp knitting workshop scheduling problem with complex characteristics such as large scale, strong constraints, nonlinearity, and uncertainty. Having scheduled seven kinds of cloth orders, the scheduling results are compared with those of the GA–TS algorithm, ABC–TS algorithm, and PSO–TS algorithm. The experimental results show that the IG–TS algorithm has the best scheduling results and more stable robustness in the production scheduling problem of warp knitting digital workshops in textile enterprises. In addition, in view of the small-scale example problem, the actual production is conducted according to the scheduling results of the IG–TS algorithm and the scheduling scheme based on artificial experience. The results show that the scheduling based on the IG–TS algorithm can effectively save the time of planning and production, improve production efficiency, and better optimize the production period. This can further promote the cost reduction, energy saving, and efficiency improvement of warp knitting digital workshops in textile enterprises. In addition, this algorithm is also applicable to the unrelated parallel machine scheduling problem in other industries.

CONFLICT OF INTEREST STATEMENT

All authors disclosed no relevant relationships.

DATA AVAILABILITY

Data are available on request to the authors.

ACKNOWLEDGEMENTS

This work was financially supported by the National Key R&D Program of China (Grant No. 2018YFB1308800) and Fundamental Research Funds for the Central Universities (Grant No. 2232020D-30).

REFERENCES

[1] S.-W. Lin and K.-C. Ying, "A multi-point simulated annealing heuristic for solving multiple objective unrelated parallel machine scheduling problems," *Int. J. Prod. Res.*, vol. 53, no. 4, pp. 1065–1076, 2015, doi: 10.1080/00207543.2014.942011.

[2] H.R. Lewis, Michael R., Garey and David S. Johnson. "Computers and intractability. A guide to the theory of NP-completeness." W.H. Freeman and Company, San Francisco 1979, x+338 pp.," *J. Symb. Log.*, vol. 48, no. 2, pp. 498–500, 1983, doi: 10.2307/2273574.
[3] W. Bożejko, M. Uchroński, and M. Wodecki, "Blocks for two-machines total weighted tardiness flow shop scheduling problem," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 68, no. 1, pp. 31–41, 2020, doi: 10.24425/bpasts.2020.131829.
[4] V. Suresh and S. Senthil Kumar, "Research on hybrid modified pathfinder algorithm for optimal reactive power dispatch," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 69, no. 4, pp. e137733, 2021 2021, doi: 10.24425/bpasts.2021.137733.
[5] G. Rabadi, R.J. Moraga, and A. Al-Salem, "Heuristics for the Unrelated Parallel Machine Scheduling Problem with Setup Times," *J. Intell. Manuf.*, vol. 17, no. 1, pp. 85–97, 2006, doi: 10.1007/s10845-005-5514-0.
[6] S.A. Torabi, N. Sahebjamnia, S.A. Mansouri, and M.A. Bajestani, "A particle swarm optimization for a fuzzy multi-objective unrelated parallel machines scheduling problem," *Appl. Soft Comput.*, vol. 13, no. 12, pp. 4750–4762, 2013, doi: 10.1016/j.asoc.2013.07.029.
[7] R. Gedik, D. Kalathia, G. Egilmez, and E. Kirac, "A constraint programming approach for solving unrelated parallel machine scheduling problem," *Comput. Ind. Eng.*, vol. 121, pp. 139–149, 2018, doi: 10.1016/j.cie.2018.05.014.
[8] L. Fanjul-Peyro, R. Ruiz, and F. Perea, "Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times," *Comput. Oper. Res.*, vol. 101, pp. 173–182, 2019, doi: 10.1016/j.cor.2018.07.007.
[9] E. Vallada, F. Villa, and L. Fanjul-Peyro, "Enriched metaheuristics for the resource constrained unrelated parallel machine scheduling problem," *Comput. Oper. Res.*, vol. 111, pp. 415–424, 2019, doi: 10.1016/j.cor.2019.07.016.
[10] J. Gao, "A novel artificial immune system for solving multi-objective scheduling problems subject to special process constraint," *Comput. Ind. Eng.*, vol. 58, no. 4, pp. 602–609, 2010, doi: 10.1016/j.cie.2009.12.009.
[11] A. Guinet, "Scheduling sequence-dependent jobs on identical parallel machines to minimize completion time criteria," *Int. J. Prod. Res.*, vol. 31, no. 7, pp. 1579–1594, 1993, doi: 10.1080/00207549308956810.
[12] R. Ruiz and T. Stützle, "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem," *Eur. J. Oper. Res.*, vol. 177, no. 3, pp. 2033–2049, 2007, doi: 10.1016/j.ejor.2005.12.009.
[13] C. Yu, Q. Semeraro, and A. Matta, "A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility," *Comput. Oper. Res.*, vol. 100, pp. 211–229, 2018, doi: 10.1016/j.cor.2018.07.025.
[14] H. Soleimani, H. Ghaderi, P.-W. Tsai, N. Zarbakhshnia, and M. Maleki, "Scheduling of unrelated parallel machines considering sequence-related setup time, start time-dependent deterioration, position-dependent learning and power consumption minimization," *J. Cleaner Prod.*, vol. 249, p. 119428, 2020, doi: 10.1016/j.jclepro.2019.119428.
[15] Y.-B. Woo, S. Jung, and B.S. Kim, "A rule-based genetic algorithm with an improvement heuristic for unrelated parallel machine scheduling problem with time-dependent deterioration and multiple rate-modifying activities," *Comput. Ind. Eng.*, vol. 109, pp. 179–190, 2017, doi: 10.1016/j.cie.2017.05.007.
[16] H. Zhang, F. Liu, Y. Zhou, and Z. Zhang, "A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem," *Inf. Sci.*, vol. 539, pp. 347–374, 2020, doi: 10.1016/j.ins.2020.06.036.
[17] J. Xiao, J. Pachl, B. Lin, and J. Wang, "Solving the block-to-train assignment problem using the heuristic approach based on the genetic algorithm and tabu search," *Transp. Res. Part B, Methodol.*, vol. 108, pp. 148–171, 2018, doi: 10.1016/j.trb.2017.12.014.
[18] W. Sukkerd and T. Wuttipornpun, "Hybrid genetic algorithm and tabu search for finite capacity material requirement planning system in flexible flow shop with assembly operations," *Comput. Ind. Eng.*, vol. 97, pp. 157–169, 2016, doi: 10.1016/j.cie.2016.05.006.
[19] T. Thongwan, A. Kangrang, and H. Prasanchum, "Multi-objective future rule curves using conditional tabu search algorithm and conditional genetic algorithm for reservoir operation," *Heliyon*, vol. 5, no. 9, p. e02401, 2019, doi: 10.1016/j.heliyon.2019.e02401.
[20] H. Tehzeeb ul, T. Alquthami, S.E. Butt, M.F. Tahir, and K. Mehmood, "Short-term optimal scheduling of hydro-thermal power plants using artificial bee colony algorithm," *Energy Rep.*, vol. 6, pp. 984–992, 2020, doi: 10.1016/j.egyr.2020.04.003.
[21] H. Li, X. Li, and L. Gao, "A discrete artificial bee colony algorithm for the distributed heterogeneous no-wait flowshop scheduling problem," *Appl. Soft Comput.*, vol. 100, p. 106946, 2021, doi: 10.1016/j.asoc.2020.106946.
[22] S. Su, F. Zhou, and H. Yu, "An artificial bee colony algorithm with variable neighborhood search and tabu list for long-term carpooling problem with time window," *Appl. Soft Comput.*, vol. 85, p. 105814, 2019, doi: 10.1016/j.asoc.2019.105814.
[23] S. Lu, X. Liu, J. Pei, M.T. Thai, and P.M. Pardalos, "A hybrid ABC–TS algorithm for the unrelated parallel-batching machines scheduling problem with deteriorating jobs and maintenance activity," *Appl. Soft Comput.*, vol. 66, pp. 168–182, 2018, doi: 10.1016/j.asoc.2018.02.018.
[24] D.C. Hop, N. Van Hop, and T.T.M. Anh, "Adaptive particle swarm optimization for integrated quay crane and yard truck scheduling problem," *Comput. Ind. Eng.*, vol. 153, p. 107075, 2021, doi: 10.1016/j.cie.2020.107075.
[25] H. Ding and X. Gu, "Hybrid of human learning optimization algorithm and particle swarm optimization algorithm with scheduling strategies for the flexible job-shop scheduling problem," *Neurocomputing*, vol. 414, pp. 313–332, 2020, doi: 10.1016/j.neucom.2020.07.004.
[26] M.K. Marichelvam, M. Geetha, and Ö. Tosun, "An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors – A case study," *Comput. Oper. Res.*, vol. 114, p. 104812, 2020, doi: 10.1016/j.cor.2019.104812.
[27] I. Alharkan, M. Saleh, M. A. Ghaleb, H. Kaid, A. Farhan, and A. Almarfadi, "Tabu search and particle swarm optimization algorithms for two identical parallel machines scheduling problem with a single server," *J. King Saud Univ. Eng. Sci.*, vol. 32, no. 5, pp. 330–338, 2020, doi: 10.1016/j.jksues.2019.03.006.
[28] G. Lin, J. Guan, Z. Li, and H. Feng, "A hybrid binary particle swarm optimization with tabu search for the set-union knapsack problem," *Expert Syst. Appl.*, vol. 135, pp. 201–211, 2019, doi: 10.1016/j.eswa.2019.06.007.