Int j simul model 17 (2018) 1, 147-158
ISSN 1726-4529 Original scientific paper

OPTIMIZATION ALGORITHM SIMULATION FOR DUAL-RESOURCE CONSTRAINED JOB-SHOP SCHEDULING

Zhong, Q.[*,#]; Yang, H.[**] & Tang, T.[*]
[*] School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian 116025, China
[**] School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
E-Mail: zhongqi@dufe.edu.cn ([#] Corresponding author)

Abstract
This research aims to optimize the job-shop scheduling constrained by manpower and machine under complex manufacturing conditions. To this end, a branch population genetic algorithm was presented based on compressed time-window scheduling strategy, and optimized with elite evolution and fan-shaped roulette operator. Specifically, the compressed time-window scheduling strategy was proposed to meet the two optimization targets: the maximum makespan and the total processing cost. Then, the elite evolution and fan-shaped roulette operator were introduced to simplify the global and local search, enhance the capacity of branch population genetic algorithm, and suppress the early elimination of inferior solutions, thus preventing the algorithm from falling into the local optimal solution. Finally, the rationality and feasibility of the proposed algorithm were verified through a simulation test. The simulation results show that the proposed algorithm lowered the maximum makespan and total processing cost by 7.4 % and 4.7 %, respectively, from the level of the original branch population genetic algorithm. This means the compressed time-window scheduling strategy can significantly optimize the makespan and the cost, as well as the robustness and global search ability.
(Received, processed and accepted by the Chinese Representative Office.)

Key Words: Job-Shop Scheduling, Dual-Resource Constraints (DRCs), Compressed Time-Window Scheduling Strategy, Improved Branch Population Genetic Algorithm, Elite Evolution

1. INTRODUCTION

The dual-resource constrained (DRC) job-shop scheduling is developed by introducing two resource constraints, e.g. manpower and machine, to the traditional job-shop scheduling [1, 2]. With high flexibility and rich resources, the DRC approach represents the future trends of job-shop scheduling [3]. Of course, there are some defects in the new strategy, such as long processing time per job, path congestion, and the mismatch between the processing capacity and the makespan. What is worse, a mismatch may arise between the manpower and the number of machines, due to the high degree of automation and the limited number of operators [4, 5].

Relevant studies have proved that it is difficult to establish a DRC job-shop scheduling model by analytical methods [6-8]. Currently, the DRC job-shop scheduling is mainly solved by simulation algorithms, including but not limited to genetic algorithm [9], ant colony algorithm [10, 11], and neural network algorithm [12]. However, these algorithms fail to achieve sufficiently accurate results, despite their accuracy in handling simple job-shop scheduling problems [13, 14].

Recent years saw the emergence of artificial intelligence methods like metaheuristic algorithm [15, 16], local search algorithm [17], binary particle swarm optimization [18, 19], and multi-objective tabu search algorithm [20, 21]. In spite of some achievements, these methods are subject to certain limitations. For instance, the optimization results hinge on the quality of the initial population; the evolution and mutation operators lack autonomy and solely rely on the pre-set work steps; the existing optimization plans overlook the influencing factors of real production (e.g. job batch and waiting time).

This research aims to optimize the job-shop scheduling constrained by manpower and machine under complex manufacturing conditions. To this end, a branch population genetic algorithm was presented based on compressed time-window scheduling strategy, and optimized with elite evolution and fan-shaped roulette operator. Finally, the rationality and feasibility of the proposed algorithm were verified through a simulation test.

2. MATHEMATICAL OPTIMIZATION MODEL OF DRC JOB-SHOP SCHEDULING

Let $W_i = \{W_1, W_2, \ldots, W_w\}$ be the set of the working efficiency of the operators in the shop, $M_j = \{M_1, M_2, \ldots, M_m\}$ be the set of machines, and $P_q = \{P_1, P_2, \ldots, P_q\}$ be the set of jobs. Suppose each operator can control 3 or more machines, and each job has its unique processing steps and path. The operators and machines can be combined in different ways to process a job. In addition, the makespan is determined by machine performance, and the working efficiency and skill level of operators [22-24].

Assuming that the job-shop pursues the minimum makespan F_1 and the minimum cost F_2. Then, the objective functions can be expressed as:

$$\min F = [F_1, F_2] \tag{1}$$

$$F_1 = \min\left(Makespan\right) = \min\left(\max\left(T^{E}_{P_{ij}M_kW_l}\right)\right) \tag{2}$$

$$F_2 = \min\left(Cost\right) \tag{3}$$

Let us denote the start time of job processing as time 0. Hence, the cost equation of job processing can be expressed as:

$$\begin{aligned} Cost = {} & C_{year} \times \sum_{i=1}^{n}\left(C_{P_i} \times \left(\max\left(T^{PE}_{P_i}, T^{E}_{P_i}\right) - T^{S}_{P_{i1}}\right)\right) + C_{year} \times \sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(\left(C_{M_k} + C_{W_l}\right) \times \right. \\ & \left. T^{P}_{P_{ij}M_kW_l} \times \left(T^{S}_{P_{i(j+1)}M_{k'}W_{l'}} - T^{S}_{P_{ij}M_kW_l}\right)\right) + \sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(\left(C_{M_k} + C_{W_l}\right) \times T^{P}_{P_{ij}M_kW_l}\right) + \sum_{i=1}^{N}\left(C^{early}_{P_i} \times \right. \\ & \left. \max\left(0, T^{E}_{P_i} - T^{PE}_{P_i}\right) + C^{late}_{P_i} \times \max\left(0, T^{PE}_{P_i} - T^{E}_{P_i}\right)\right) \end{aligned} \tag{4}$$

Eq. (4) gives the gain brought by early completion and the loss resulted from delayed completion of job processing.

Eqs. (1) to (4) are constrained by the following conditions:

$$\begin{cases} T^{S}_{P_{ij}M_kW_l} \geq 0 \\ T^{P}_{P_{ij}M_kW_l} = t^{P}_{P_{ij}M_k} \big/ eW_lM_k \\ T^{E}_{P_{ij}M_kW_l} = T^{S}_{P_{ij}M_kW_l} + T^{P}_{P_{ij}M_kW_l} \\ T^{E}_{P_{ij}M_kW_l} \leq T^{S}_{P_{i(j+1)}M_qW_r} \end{cases} \tag{5}$$

$$\begin{cases} H_{P_{ij}M_kW_l} H_{P_{xy}M_kW_r} \left(T^{E}_{P_{ij}M_kW_l} - T^{S}_{P_{xy}M_kW_r}\right)\left(T^{S}_{P_{ij}M_kW_l} - T^{E}_{P_{xy}M_kW_r}\right) \geq 0 \\ H_{P_{ij}M_kW_l} H_{P_{xy}M_qW_r} \left(T^{E}_{P_{ij}M_kW_l} - T^{S}_{P_{xy}M_qW_r}\right)\left(T^{S}_{P_{ij}M_kW_l} - T^{E}_{P_{xy}M_qW_r}\right) \geq 0 \end{cases} \tag{6}$$

$$R_{M_k} \cap R_{W_l} \cap \left[T^{S}_{P_{ij}M_kW_l}, T^{E}_{P_{ij}M_kW_l}\right] \neq \varnothing \tag{7}$$

In Eqs. (1) to (7), CM_k, CW_l and CP_i are the hourly running cost of each machine, hourly wage of each operator, and the purchase cost of material, respectively; C_{pi}^{ear} and C_{pi}^{lat} are the reward for the early completion and the penalty for the delayed completion of P_i processing, respectively; T_{Pi}^{E} is the theoretical makespan. Eq. (5) stipulates that the job processing starts at time 0, the actual processing time depends on machine performance and operator efficiency, and the work steps are planned in advance; Eq. (6) requires that each machine only processes one job at a time, and each operator only controls one machine at a time; Eq. (7) means that a job can only be processed when both the machine and the operator are in the waiting state.

3. IMPROVED BRANCH POPULATION GENETIC ALGORITHM

3.1 Compressed time-window strategy

The following concepts were defined for the traditional time-window strategy.

Time-window TW [*tS TW*, *tE TW*]: The time-axis mapping results of the continuous processing capacity of the job-shop over a period of time. *tS TW* and *tE TW* are the start time and end time of the time-window, respectively.

Resource availability: The continuous processing capacity of machine and manpower in the discrete job-shop is divided into several independent time-windows over the time. The set of all time-windows is denoted as RM_k.

Resource combination capacity: The capacity to make machine and manpower available at the same time.

The traditional time-window scheduling strategy does well in the optimization of job-shop workspan. Nevertheless, the traditional strategy works poorly in cost optimization, and overlooks the multiple waiting time slots caused by the discreteness of job processing. To overcome these shortcomings, the traditional strategy was modified by compressing the time-windows, so that scheduling plan can be activated and hierarchized through the hierarchical optimization mechanism [25, 26].

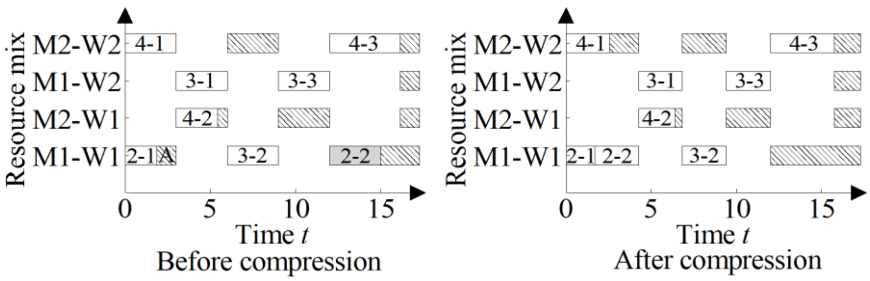


Figure 1: The scheduling plan of the traditional time-window strategy and that after compressing the time-windows.

Fig. 1 shows the scheduling plans of the traditional time-window strategy and the compressed time-window strategy. The slash-filled boxes stand for the schedulable time-windows. The numbers refer to the serial number of job and work step (e.g. "4-1" means the 1st work step of the 4th job). After time-window compression, the makespan extends from 16 to 18, which is unfavourable to cost control.

To optimize the maximum makespan, let the last work step of job processing P_{ij} obey the following equation:

$$\left\{P_{ij} \middle| P_{ij} \subset S_c \wedge T_{P_{ij}}^{E} = \max\left(\left\{T_{P_{xy}}^{E} \middle| \forall P_{xy} \subset S_c\right\}\right)\right\} \tag{8}$$

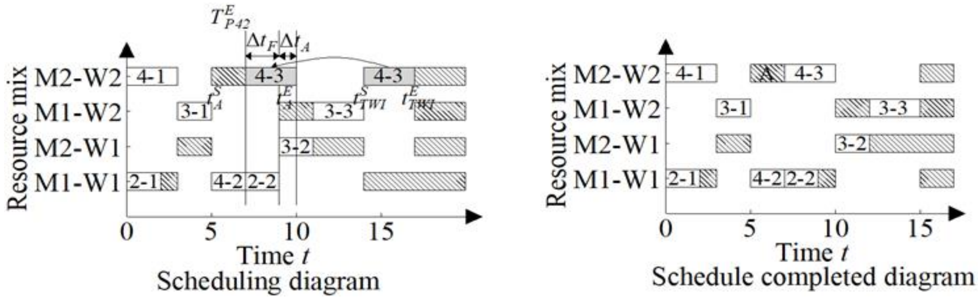


Figure 2: The scheduling plan and scheduled results of compressed time-window strategy.

The time-window A[*tS A*, *tR A*] in Fig. 1 was used to compress the time-window of P_{ij}. When the scheduling follows the plan in Fig. 2, the time-window usage Δt_F and the time delay Δt_A are expressed as:

$$\begin{cases} \Delta t_F = t_A^E - \max\left(t_A^S, T_{P_{i(j-1)}}^E\right) \\ \Delta t_A = T_{P_{ij}}^P - \Delta t_F = T_{P_{ij}}^P - t_A^E + \max\left(t_A^S, T_{P_{i(j-1)}}^E\right) \end{cases} \tag{9}$$

If and only if $\Delta t_F > \Delta t_A$, the maximum makespan is optimized by the plan in Fig. 2. The following equation can be obtained by substituting $\Delta t_F > \Delta t_A$ into Eq. (9):

$$2\times\left(t_A^E - \max\left(t_A^S, T_{P_{i(j-1)}}^E\right)\right) > T_{P_{ij}}^P \tag{10}$$

The job-shop cost consists of the material cost, the incremental material cost, the machine and manpower cost, additional cost of delayed completion. Among them, the incremental material cost is the most important part. The incremental cost of the job-shop can be expressed as:

$$\Delta C_{P_{ij}} = C_{year}\times\left(\left(C_{M_k}+C_{W_l}\right)\times T_{P_{ij}M_kW_l}^P \times \Delta t\right) \tag{11}$$

The total cost of the job-shop is affected by any local work step adjustment in the processing system. Suppose work step P_{ij} is compressed by time-window A[*tS A*, *tR A*]. Then, the variation in the total cost is:

$$\Delta C_F = C_{year}\times\left(\left(C_{M_k}+C_{W_l}\right)\times T_{P_{ij}M_kW_l}^P \times\left(\max\left(t_A^S, T_{P_{ij}}^E\right) - t_{TWI}^S\right)\right) \tag{12}$$

The other work steps are delayed due to the change in P_{ij}. The time delay can be expressed as:

$$\Delta t_A = T_{P_{ij}}^P - \Delta t_F = T_{P_{ij}}^P - t_A^E + \max\left(t_A^S, T_{P_{i(j-1)}}^E\right) \tag{13}$$

Hence, the upper bound ΔC_A of the incremental cost incurred by the delay in all work steps is:

$$\Delta C_A = \sum_{P_{ij}\subset P_A}\left(C_{year}\times\left(\left(C_{M_k}+C_{W_l}\right)\times T_{P_{ij}M_kW_l}^P \times \Delta t_A\right)\right) \tag{14}$$

The ΔC_A and ΔC_F must satisfy the following condition:

$$\Delta C_F + \Delta C_A < 0 \tag{15}$$

3.2 Design of branch population genetic algorithm

The optimization of DRC job-shop scheduling can be split into 2 sub-problems: job processing sequence, and processing capacity allocation of machines and operators. Here, the branch population genetic algorithm is modified, and introduced to solve the job-shop scheduling optimization model. The improved branch population genetic algorithm retains the global search ability of the original genetic algorithm. Meanwhile, the ant colony optimization algorithm was adopted to enhance the ability to find the local optimal solution and curb the loss of elite offspring. In this way, the evolutionary population becomes more diverse and better in quality, and the algorithm supports a greater search scope.

The branch population genetic algorithm approximates the different job processing methods as multiple branch paths, and divides the original population into multiple branch populations according to the path information. To improve the traditional algorithm, the elite evolution strategy was introduced to select operators and build local search ability.

Eq. (16) shows the branch population algorithm generated by the ant traffic method. In the equation, $\tau OOz\ PxyP_{ij}$ and $\tau ROz\ P_{ij}M_kW_l$ are the sequence pheromone and the resource pheromone, respectively. The two pheromones were constructed as per time and cost indices. After each iteration, the viable population and adjoint population were counted, and used to update the global or local pheromones.

$$P_{P_{xy}P_{ij}} = \begin{cases} 1, P \le q \wedge \arg\max\limits_{(i,j,k,l)\in P_s} \left\{ \left(\tau^{OO_1}_{P_{xy}P_{ij}} \cdot \tau^{OO_1}_{P_{xy}P_{ij}} \cdot \tau^{RO_1}_{P_{ij}M_kW_l} \cdot \tau^{RO_2}_{P_{ij}M_kW_l} \right)^{\alpha} \cdot \eta^{\beta}_{P_{ij}M_kW_l} \cdot \left(n_{P_{xy}P_{ij}} \cdot n_{P_{ij}M_kW_l} \right)^{\gamma} \right\} \\ \dfrac{\left(\tau^{OO_1}_{P_{xy}P_{ij}} \cdot \tau^{OO_1}_{P_{xy}P_{ij}} \cdot \tau^{RO_1}_{P_{ij}M_kW_l} \cdot \tau^{RO_2}_{P_{ij}M_kW_l} \right)^{\alpha} \cdot \eta^{\beta}_{P_{ij}M_kW_l} \cdot \left(n_{P_{xy}P_{ij}} \cdot n_{P_{ij}M_kW_l} \right)^{\gamma}}{\sum\limits_{(i,j,k,l)\in P_s} \left(\left(\tau^{OO_1}_{P_{xy}P_{ij}} \cdot \tau^{OO_1}_{P_{xy}P_{ij}} \cdot \tau^{RO_1}_{P_{ij}M_kW_l} \cdot \tau^{RO_2}_{P_{ij}M_kW_l} \right)^{\alpha} \cdot \eta^{\beta}_{P_{ij}M_kW_l} \cdot \left(n_{P_{xy}P_{ij}} \cdot n_{P_{ij}M_kW_l} \right)^{\gamma} \right)}, P > q \end{cases} \tag{16}$$

The update was implemented in the following steps:

(a) The elite offspring evolution plan was designed for the population: From the previous population, the chromosomes whose scheduling indices are above the target mean values were extracted, forming an “elite population”. The other individuals in the previous population constitute the “non-inferior solution population”. Then, the two populations were iterated separated, and the evolution was optimized by the fan-shaped roulette method.

(b) From the elite population and the non-inferior solution population, the chromosomes whose scheduling indices are above the target mean values were extracted, forming the “second generation elite population”. The other individuals in the two populations constitute the “second generation non-inferior solution population”.

(c) The chromosomes in the second-generation elite population and the second-generation non-inferior solution population were cross-evolved, and the elite individuals were selected to form the King chromosome population. The probability of non-inferior solution was enhanced for the King chromosome population, to prevent the premature convergence to the local optimal solution.

Let the N_e, N_a and N_p be the number of evolutionary population, branch population and non-inferior solution population, respectively. Then, the number of King chromosome population and that of elite population are $0.25N_e$ and $0.5N_e$, respectively. The cross-evolution complexity of the elite population is $O(N_a \times 0.5N_e) + O(N_e)$. Hence, the evolution of the elite population is far less time-consuming and complex than the evolution of the entire population (hereinafter referred to as the overall evolution).

After adding the compression time-window scheduling strategy, the improved branch population genetic algorithm consists of the following steps:

(1) Generation of the initial population: initialize the parameters of population size, iteration number and pheromone, and set the crossover probability and mutation probability.

(2) Formation of the target population: after iterating t times, generate the branch population according to Eq. (16), and combine the brand population with the non-elite individuals after the $(t\text{-}1)^{\text{th}}$ iteration into the target population for further evolution.

(3) Creation of the elite and general populations: decode the target population based on the compressed time-window scheduling strategy, extract the chromosomes with excellent

scheduling indices to form the elite population, and allocate the remaining chromosomes to the general population.

(4) Elite mutation operation: from the elite population and the non-inferior solution population, separately select one king chromosome, two elite chromosomes and two non-inferior solution chromosomes for cross evolution.

(5) Global update of pheromone and ant traffic: determine the set of non-inferior solution populations after the t^{th} iteration by the compressive time-window scheduling strategy, create the viable population with the elite chromosomes from the set by the fan-shaped roulette algorithm, and update the pheromone and ant traffic of the set of non-inferior solution populations and the set of viable populations.

(6) Termination condition: repeat the iterations from Step 3 to Step 5 until the optimal solution is obtained.

4. SIMULATION TEST AND ANALYSIS

4.1 Comparison of optimization effects before and after the algorithm optimization

This section attempts to analyse the optimization effect of branch population, elite evolution and fan-shaped roulette operator on the DRC job-shop scheduling model.

(1) Branch population

The optimal feasible solution for job-shop scheduling is the point set distributed over the time index (X-axis) and the cost index (Y-axis). Let S_x and S_y be the standard deviations of the X-axis and the Y-axis, respectively. Let $(S_x \times S_y)^{1/2}$ be the population discreteness. The value of $(S_x \times S_y)^{1/2}$ is positively correlated with the population discreteness. Figs. 3 and 4 present the effect of branch population on population discreteness and the total cost.

As shown in Fig. 3, the value of $(S_x \times S_y)^{1/2}$ plunged after introducing branch population. This means the branch population suppressed the local premature convergence of the scheduling model. According to Fig. 4, the branch population effectively lowered the total processing cost of the job-shop.

(2) Elite population evolution

The following evaluation indices were defined: the average size of the solution area S_I; the distribution range of solution space A_{range}; the evenness of optimal solution distribution ΔA. The mixed index C_H can be expressed as:

$$C_H = S_I \times \Delta A \big/ \left(A_{range} + 1 \right) \tag{17}$$

The value of S_I is positively correlated with the distance between the set of non-inferior solutions and the Pareto front; the value of A_{range} is negatively correlated with the value of ΔA, and the ability to find non-inferior solutions. Therefore, the overall advantage of the algorithm is better at low levels of C_H.

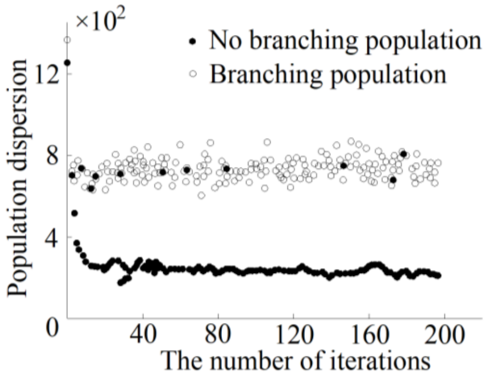


Figure 3: Branch population vs. population discreteness.

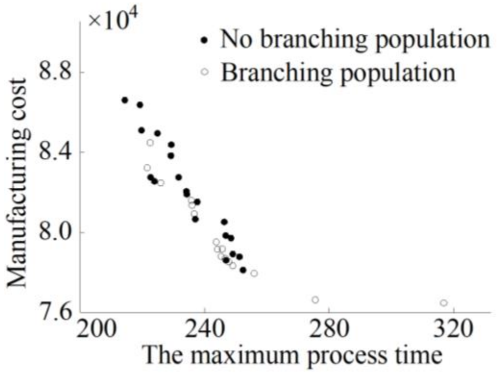
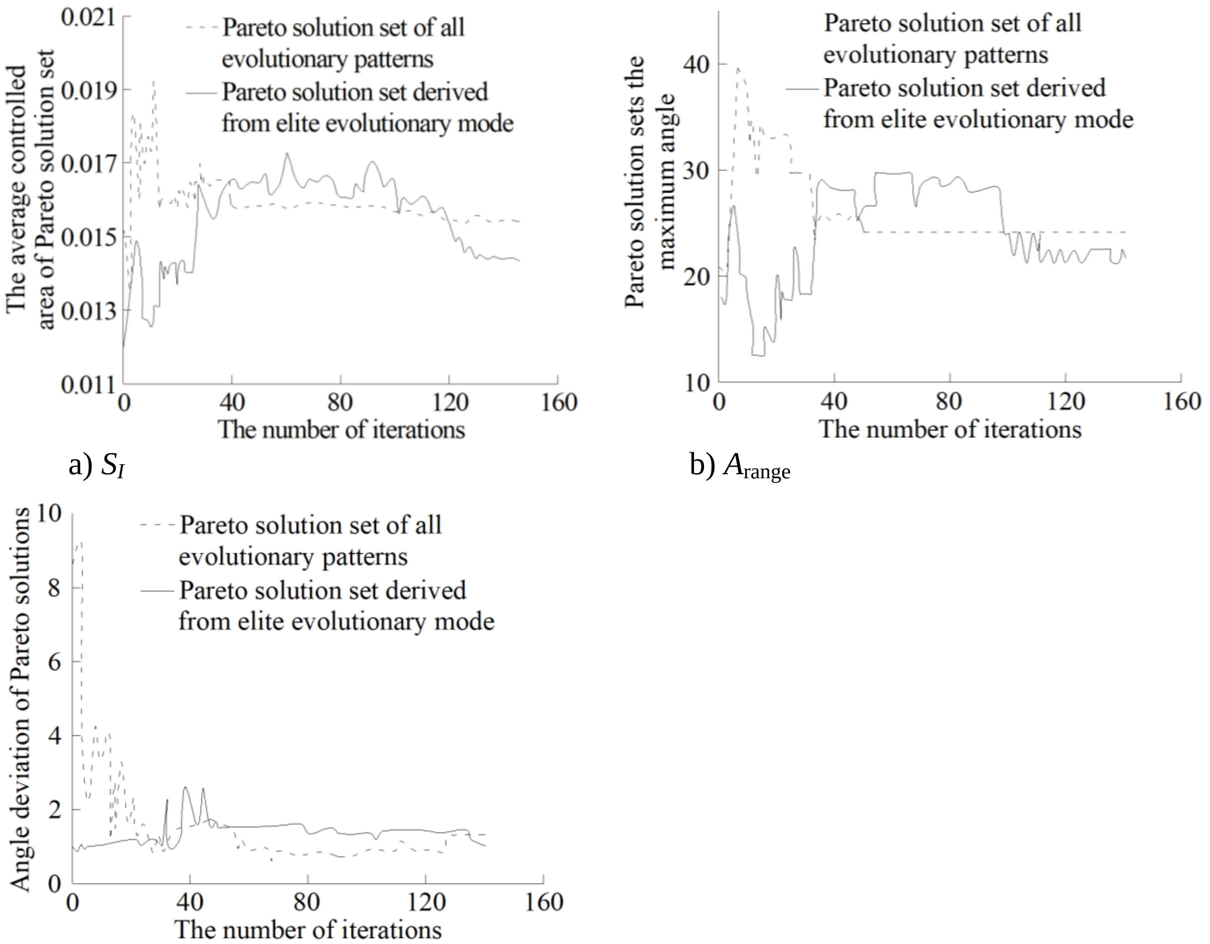


Figure 4: Branch population vs. total processing cost.

Fig. 5 compares the values of S_I, A_{range} and ΔA under the overall evolution mode and the elite evolution mode. The S_I values in Fig. 5 a show that the non-inferior solutions of the elite evolution algorithm are closer to the Pareto front than those of the overall evolution algorithm; the A_{range} values in Fig. 5 b reveal that the elite evolution algorithm is more likely to converge to the set of elite solutions, and weaker in global search ability than the overall evolution algorithm; the ΔA values in Fig. 5 c show that the indices of the elite evolution algorithm are more stable than those of the overall evolution algorithm; moreover, the overall evolution algorithm fluctuates more violently in the early phase of convergence, and generates more discrete solutions than the elite evolution algorithm.



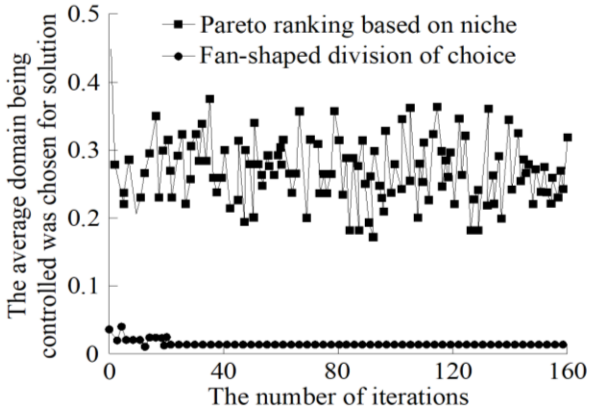
a) S_I b) A_{range} c) ΔA

Figure 5: Comparison of S_I, A_{range} and ΔA under the overall evolution mode and the elite evolution mode.

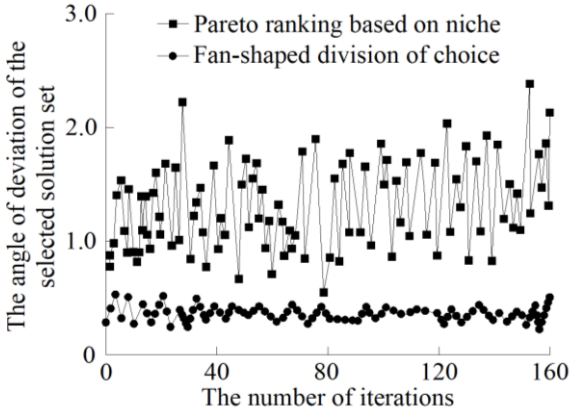
(3) Fan-shaped roulette operator

Fig. 6 compares the values of S_I, ΔA and C_H of the typical niche sorting operator and the fan-shaped roulette operator. It can be seen that the niche sorting operator considers the evenness of solution distribution while selecting the optimal solution. The elite solution will be eliminated if it is surrounded by many elite solutions. Thus, the niche sorting operator may lead to a high error. By contrast, the fan-shaped roulette operator acquires the population distribution features from the structural level, and optimizes the search scope without considering the solution distribution. In light of the curves of S_I, ΔA and C_H, the niche sorting operator brings about better solutions than the niche sorting operator.

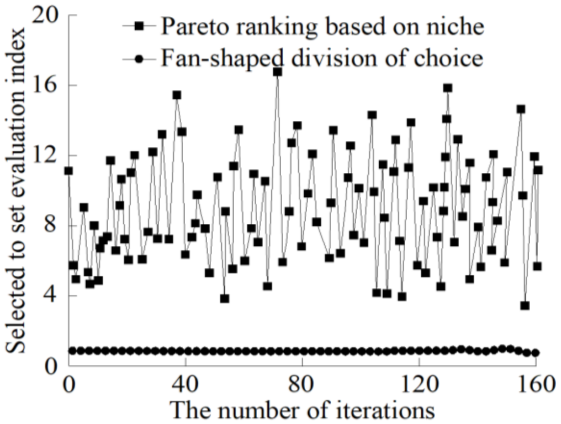
Next, the NSGA II algorithm was compared with the improved branch population genetic algorithm. The comparison results are shown in Table 1. Ten sets of examples are included in the table. It is clear that the proposed algorithm outperformed the NSGA II algorithm in all indices. Therefore, the elite evolution operator optimized the local search ability of the proposed algorithm. The enhanced local search mechanism pushed the overall search scope towards the Pareto front, and elevated the search ability. The fan-shaped roulette operator, however, solved the uneven distribution of the solutions of the original branch population genetic algorithm, and increased both the robustness and global search ability.



a) S_I



b) A_{range}



c) ΔA

Figure 6: Comparison of S_I, ΔA and C_H of the typical niche sorting operator and the fan-shaped roulette operator

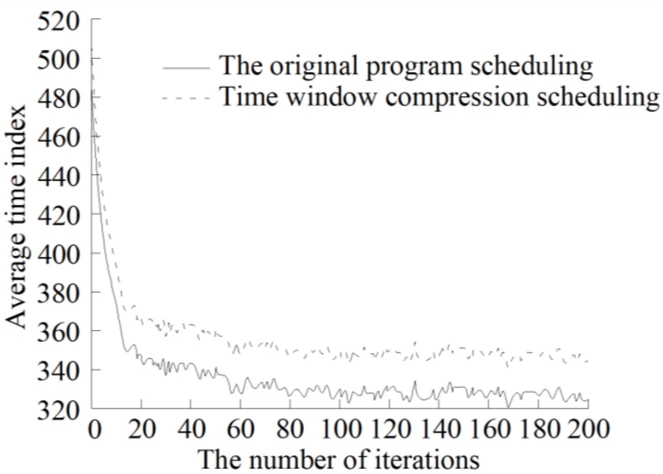
4.2 Comparison of resource constraint indices

Fig. 7 compares the maxim um makespan and the total processing cost of the original scheduling plan and the compressed time-window scheduling plan. It is observed that the latter shortened the maximum makespan and lowered the total processing cost by 7.4 % and

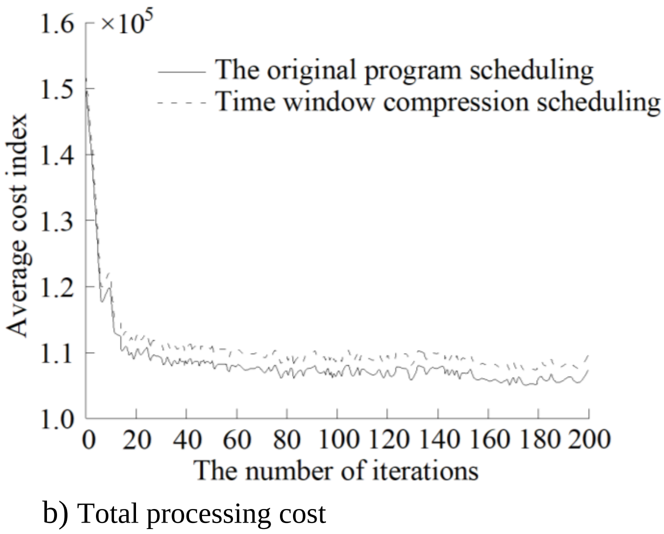
4.7 %, respectively, from the level of the original scheduling plan. This means the compressed time-window scheduling strategy can significantly improve the makespan and the total processing cost.

Table I: Comparison between the NSGA II algorithm and the improved branch population genetic algorithm.

Examples	BPGA				NSGA II			
	$\overline{S}_I$	A_{range}	$S_{\Delta A}$	C_H	$\overline{S}_I$	A_{range}	$S_{\Delta A}$	C_H
1	0.004	20.48	48.30	1.20	0.038	55.46	56.28	3.79
2	0.003	35.12	46.58	0.51	0.050	60.73	50.96	3.88
3	0.012	22.23	12.54	0.76	0.019	32.15	23.78	12.80
4	0.011	45.75	33.26	0.95	0.042	54.84	57.42	4.37
5	0.007	28.09	39.65	1.21	0.042	33.66	60.92	7.70
6	0.003	37.24	83.91	0.88	0.049	18.35	77.05	24.87
7	0.021	66.18	6.73	0.35	0.123	48.02	7.99	2.73
8	0.015	20.88	55.35	4.72	0.145	38.57	43.84	15.06
9	0.010	22.49	38.36	1.93	0.035	46.57	59.49	4.77
10	0.007	21.37	44.59	1.70	0.039	44.39	57.34	5.39



a) Maximum makespan



b) Total processing cost

Figure 7: Comparison of maximum makespan and total processing cost.

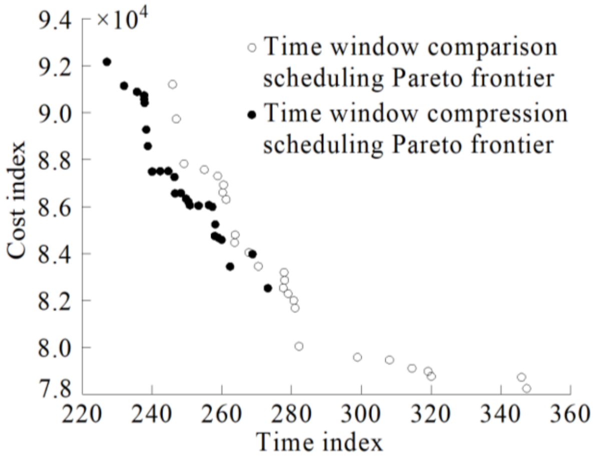


Figure 8: Non-inferior solutions of the original scheduling plan and the branch population genetic algorithm enhanced by compressed time-window scheduling

Fig. 8 presents the scatter diagrams of the non-inferior solutions acquired by the original branch population genetic algorithm and the branch population genetic algorithm enhanced by compressed time-window scheduling. It can be seen that, the enhanced algorithm produced solutions close to Pareto front, and evenly distributed non-inferior solutions in the global scope. Fig. 9 compares the solutions of the proposed algorithm and the HGA algorithm. It is clear that our algorithm had an edge over the HGA.

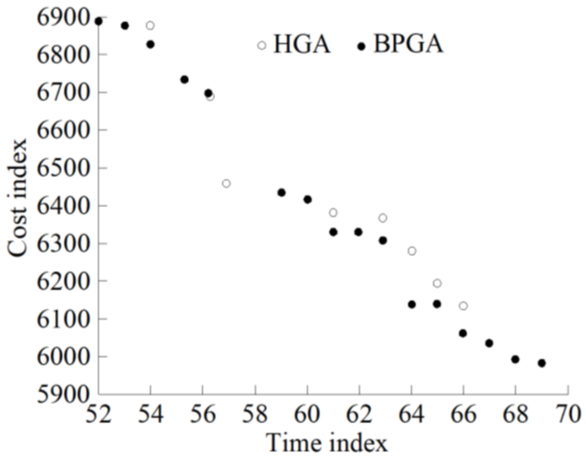


Figure 9: Non-inferior solutions of the HGA and the branch population genetic algorithm enhanced by compressed time-window scheduling

5. CONCLUSION

This research aims to optimize the job-shop scheduling constrained by manpower and machine under complex manufacturing conditions. To this end, a branch population genetic algorithm was presented based on compressed time-window scheduling strategy, and optimized with elite evolution and fan-shaped roulette operator. Finally, the simulation test was performed to verify the rationality and advantage of our algorithm.

The compressed time-window scheduling strategy was proposed to meet the two optimization targets: the maximum makespan and the total processing cost. Then, the elite evolution and fan-shaped roulette operator were introduced to simplify the global and local search, enhance the capacity of branch population genetic algorithm, and suppress the early elimination of inferior solutions, thus preventing the algorithm from falling into the local optimal solution.

Finally, the rationality and feasibility of the proposed algorithm were verified through a simulation test. The simulation results show that the proposed algorithm lowered the maximum makespan and total processing cost by 7.4 % and 4.7 %, respectively, from the level of the original branch population genetic algorithm. This means the compressed time-window scheduling strategy can significantly optimize the makespan and the cost, as well as the robustness and global search ability.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant No.: 71472029, The MOE (Ministry of Education in China) Project of Humanities and Social Science under Grant No.: 15YJC630193, Scientific research project of Liaoning Education Department under Grant No.: LN2017QN016, and Social Science Planning Fund Project of Liaoning under Grant No.: L15CJY003.

REFERENCES

[1] Jauhar, S. K.; Pant, M. (2015). Genetic algorithms, a nature-inspired tool: review of applications in supply chain management, *Proceedings of 4th International Conference on Soft Computing for Problem Solving*, 71-86, doi:10.1007/978-81-322-2217-0_7

[2] ElMaraghy, H.; Patel, V.; Abdallah, I. B. (2000). Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms, *Journal of Manufacturing Systems*, Vol. 19, No. 3, 186-201, doi:10.1016/S0278-6125(00)80011-4

[3] Kanit, R.; Gunduz, M.; Ozkan, O. (2009). Investigating the effectiveness of certain priority rules on resource scheduling of housing estate projects, *Journal of Construction Engineering & Management*, Vol. 135, No. 7, 609-613, doi:10.1061/(asce)co.1943-7862.0000021

[4] Browning, T. R.; Yassine, A. A. (2010). Resource-constrained multi-project scheduling: priority rule performance revisited, *International Journal of Production Economics*, Vol. 126, No. 2, 212-228, doi:10.1016/j.ijpe.2010.03.009

[5] Chaudhry, I. A.; Drake, P. R. (2009). Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms, *International Journal of Advanced Manufacturing Technology*, Vol. 42, No. 5-6, Paper 581, doi:10.1007/s00170-008-1617-z

[6] Liu, S. Q.; Kozan, E. (2012). A hybrid shifting bottleneck procedure algorithm for the parallel-machine job-shop scheduling problem, *Journal of the Operational Research Society*, Vol. 63, No. 2, 168-182, doi:10.1057/jors.2011.4

[7] Afzalirad, M.; Shafipour, M. (2018). Design of an efficient genetic algorithm for resource-constrained unrelated parallel machine scheduling problem with machine eligibility restrictions, *Journal of Intelligent Manufacturing*, Vol. 29, No. 2, 423-437, doi:10.1007/s10845-015-1117-6

[8] Hu, P.-C. (2006). Further study of minimizing total flowtime for the worker assignment scheduling problem in the identical parallel-machine models, *International Journal of Advanced Manufacturing Technology*, Vol. 29, No. 7-8, 753-757, doi:10.1007/s00170-005-2555-7

[9] Wang, T. C.; Xie, Y. Z. (2016). BP-GA data fusion algorithm studies oriented to smart home, *Mathematical Modelling of Engineering Problems*, Vol. 3, No. 3, 135-140, doi:10.18280/mmep.030304

[10] Amiri, M.; Zandieh, M.; Yazdani, M.; Bagheri, A. (2010). A variable neighbourhood search algorithm for the flexible job-shop scheduling problem, *International Journal of Production Research*, Vol. 48, No. 19, 5671-5689, doi:10.1080/00207540903055743

[11] Liu, H. H.; Li, N. (2017). Application of self-adaptive ant colony algorithm in machine technology manufacturing process route optimization, *Academic Journal of Manufacturing Engineering*, Vol. 15, No. 1, 11-16

[12] Baykasoglu, A. (2002). Linguistic-based meta-heuristic optimization model for flexible job shop scheduling, *International Journal of Production Research*, Vol. 40, No. 17, 4523-4543, doi:10.1080/00207540210147043

[13] Qi, Y. N.; Tang, M. C.; Zhang, M. (2014). Mass customization in flat organization: The mediating role of supply chain planning and corporation coordination, *Journal of Applied Research and Technology*, Vol. 12, No. 2, 171-181, doi:10.1016/S1665-6423(14)72333-8
[14] Tang, M.; Gong, D.; Liu, S.; Zhang, H. (2016). Applying multi-phase particle swarm optimization to solve bulk cargo port scheduling problem, *Advances in Production Engineering & Management*, Vol. 11, No. 4, 299-310, doi:10.14743/apem2016.4.228
[15] Huang, S.; Tian, N.; Wang, Y.; Ji, Z. (2016). Multi-objective flexible job-shop scheduling problem using modified discrete particle swarm optimization, *SpringerPlus*, Vol. 5, Paper 1432, 22 pages, doi:10.1186/s40064-016-3054-z
[16] Eskandari, H.; Hosseinzadeh, A. (2014). A variable neighbourhood search for hybrid flow-shop scheduling problem with rework and set-up times, *Journal of the Operational Research Society*, Vol. 65, No. 8, 1221-1231, doi:10.1057/jors.2013.70
[17] Fleszar, K.; Hindi, K. S. (2004). Solving the resource-constrained project scheduling problem by a variable neighbourhood search, *European Journal of Operational Research*, Vol. 155, No. 2, 402-413, doi:10.1016/s0377-2217(02)00884-6
[18] Kesavan, E.; Gowthaman, N.; Tharani, S.; Manoharan, S.; Arunkumar, E. (2016). Design and implementation of internal model control and particle swarm optimization based PID for heat exchanger system, *International Journal of Heat and Technology*, Vol. 34, No. 3, 386-390, doi:10.18280/ijht.340306
[19] Liang, C. H.; Zeng, S.; Li, Z. X.; Yang, D. G.; Sherif, S. A. (2016). Optimal design of plate-fin heat sink under natural convection using a particle swarm optimization algorithm, *International Journal of Heat and Technology*, Vol. 34, No. 2, 275-280, doi:10.18280/ijht.340217
[20] Zhang, J.; Yang, J. (2016). Flexible job-shop scheduling with flexible workdays, preemption, overlapping in operations and satisfaction criteria: an industrial application, *International Journal of Production Research*, Vol. 54, No. 16, 4894-4918, doi:10.1080/00207543.2015.1134839
[21] Zandieh, M.; Adibi, M. A. (2010). Dynamic job shop scheduling using variable neighbourhood search, *International Journal of Production Research*, Vol. 48, No. 8, 2449-2458, doi:10.1080/00207540802662896
[22] Li, J.; Huang, Y.; Niu, X. (2016). A branch population genetic algorithm for dual-resource constrained job shop scheduling problem, *Computers & Industrial Engineering*, Vol. 102, 113-131, doi:10.1016/j.cie.2016.10.012
[23] Li, J.; Huang, Y. (2016). A hybrid genetic algorithm for dual-resource constrained job shop scheduling problem, *International Conference on Intelligent Computing (ICIC 2016)*, *Intelligent Computing Theories and Application*, 463-475, doi:10.1007/978-3-319-42291-6_47
[24] Li, J. Y.; Sun, S. D.; Huang, Y. (2011). Adaptive hybrid ant colony optimization for solving dual resource constrained job shop scheduling problem, *Journal of Software*, Vol. 6, No. 4, 584-594, doi:10.4304/jsw.6.4.584-594
[25] Huang, Y.; Yan, X. T.; Li, J. Y. (2014). Dual resource constrained job shop scheduling based on compressed time window, *Advanced Materials Research*, Vol. 905, 731-735, doi:10.4028/www.scientific.net/amr.905.731
[26] Gavareshki, M. H. K.; Zarandi, M. H. F. (2008). A heuristic approach for large scale job shop scheduling problems, *Journal of Applied Sciences*, Vol. 8, No. 6, 992-999, doi:10.3923/jas.2008.992.999