

A Novel Production Scheduling Approach Based on Improved Hybrid Genetic Algorithm

Authors: Lili Dai, He Lu, Dezheng Hua, Xinhua Liu, Hongming Chen, Adam Glowacz, Grzegorz Królczyk, Z Li, Z Li

Date: 2024-05-10T00:00:00+00:00

Abstract

Owing to the complexity of production workshops in the discrete manufacturing industry, traditional genetic algorithms (GA) cannot effectively solve production scheduling problems. To enhance GA-based approaches for addressing production scheduling issues, a simulated annealing algorithm (SAA) is employed to develop an improved hybrid genetic algorithm. First, the crossover and mutation probabilities of genetic operations are adjusted, and an elite replacement operation is adopted for the simulated annealing operator. Subsequently, a mutation method is utilized for the comparison and replacement of genetic operations to obtain the optimal value of the current state. Finally, the proposed hybrid genetic algorithm is compared with several scheduling algorithms, thereby verifying the superiority and efficiency of the proposed method in solving production scheduling problems.

Full Text

Preamble

A Novel Production Scheduling Approach Based on Improved Hybrid Genetic Algorithm

Lili Dai¹, He Lu^{1,2}, Dezheng Hua², Xinhua Liu², Hongming Chen², Adam Glowacz³, Grzegorz Królczyk⁴, Z Li^{4,*}

¹ Lianyungang Normal College, Lianyungang 222006, China

² School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou 211006, China

³ Department of Automatic Control and Robotics, AGH University of Science and Technology, 30-059 Kraków, Poland

⁴ Faculty of Mechanical Engineering, Opole University of Technology, Opole 45-758, Poland

*Correspondence: z.li@po.edu.pl

Abstract: Due to the complexity of production shops in the discrete manufacturing industry, traditional genetic algorithms (GA) cannot effectively solve production scheduling problems. To enhance GA-based methods for production scheduling, this work develops an improved hybrid genetic algorithm using simulated annealing algorithm (SAA). Firstly, the crossover and mutation probabilities of genetic operations are adjusted, and an elite replacement operation is adopted for the simulated annealing operator. Then, a mutation method is employed for comparing and replacing genetic operations to obtain the optimal value of the current state. Finally, the proposed hybrid genetic algorithm is compared with several scheduling algorithms, verifying its superiority and efficiency in solving production scheduling problems.

Keywords: production scheduling, hybrid genetic algorithm, artificial intelligence, sustainable design, discrete manufacturing

1. Introduction

Production scheduling represents one of the classic non-deterministic polynomial problems, encompassing aircraft carrier scheduling, port cargo scheduling, parts processing scheduling, and numerous other domains [1-2]. In production environments, the scheduling of workshops, equipment, personnel, and materials varies across different manufacturing enterprises, and production cycles also differ significantly [3-5]. With the emergence of large-scale production systems and the proposal of various intelligent algorithms, workshop production scheduling has attracted increasingly extensive attention from managers, yielding continuous remarkable results [6]. In the competitive environment of the new era, enterprises have placed greater emphasis on leveraging production scheduling to rapidly achieve resource allocation, manage production schedules, and improve production efficiency [7].

Although numerous intelligent scheduling algorithms exist, seeking more efficient and practical scheduling approaches remains an important objective in discrete manufacturing industry production scheduling [8-9]. Given the complexity and diversity of discrete production scheduling, two main algorithmic categories address these problems [10-13]. The first category comprises traditional unified research methods, primarily including Lagrangian relaxation [14], branch-and-bound [15], and mathematical programming [16], which currently demonstrate limited effectiveness in practical engineering applications. The second category encompasses heuristic algorithms, mainly including genetic algorithm, simulated annealing algorithm [17], particle swarm optimization [18], and ant colony algorithm [19]. For instance, Choi et al. [20] proposed a mixed integer programming model and local search algorithm to solve project scheduling across various manufacturing environments. Nie et al. [21] studied dynamic scheduling problems with job release dates, proposing a heuristic algorithm and reactive scheduling strategy based on gene expression coding for production

scheduling applications. These algorithms feature simple structures, are easy to implement, and can achieve satisfactory results in solving production scheduling problems [22].

Among many intelligent algorithms, genetic algorithm (GA) with its special optimization mode represents an effective approach for production scheduling problems and has achieved notable results [23-24]. Pezzella et al. [25] proposed a GA combining multiple strategies for initial population generation, selection, and reproduction to address production scheduling problems. Giovanni and Pezzella [26] proposed an improved GA for distributed production scheduling problems that determines workpiece processing paths through a greedy decoding process. Zhang et al. [27] employed a Pareto-optimization-based GA with two new objective functions based on setup and synergy costs to solve production scheduling problems. Chamnanlor et al. [28] proposed a hybrid GA based on ant colony algorithm for production scheduling. Meng Yue et al. [29] created a hybrid path-relinking algorithm for production scheduling problems, combining genetic algorithm, domain structure algorithm, and path-relinking algorithm to further enhance computational capability. Zhao et al. [30] proposed a hybrid genetic-simulated annealing algorithm for production scheduling that addresses algorithm prematurity, though the rapid and substantial temperature drop easily leads to incomplete search data and loss of good individuals. Wang et al. [31] applied the hybrid genetic-simulated annealing algorithm to flexible job shop scheduling. While simulated annealing factors were added to improve algorithm performance and enable escape from local optima, the influence of adaptive crossover and mutation probabilities on final algorithm convergence was not considered.

Based on the above literature analysis, to better and more effectively solve production scheduling problems, this work proposes an improved hybrid genetic algorithm (IHGA). Unlike other hybrid genetic algorithms in selecting individuals for crossover and mutation, the proposed algorithm adopts an adaptive strategy to adjust probability magnitude. In the early stages of iteration, higher probability is first used to select more individuals, expanding the scope of later optimization, increasing species diversity, and reducing the probability of premature phenomena. Premature algorithms initially select a large number of high-fitness individuals, resulting in quickly obtaining local rather than global optimal values. In the later stage of population iterative reproduction, excellent individuals can be preserved with low probability, enabling the algorithm to converge relative to the optimal value as soon as possible. In the simulated annealing factor, using the memory function, the optimal solution undergoes mutation by probability to obtain as many better individuals as possible. A heating strategy is added to avoid falling into local optima, finding new individuals compared with the fitness function of the original individual.

The remainder of this paper is organized as follows. Section 2 presents system modeling of production scheduling. Section 3 describes the operation process of the improved hybrid genetic algorithm. Section 4 carries out experimental anal-

ysis. Section 5 conducts application testing of the improved approach. Section 6 summarizes conclusions and future works.

2. Modelling Production Scheduling

Production scheduling problems in the discrete manufacturing industry typically refer to determining the processing sequence of each workpiece under various production requirements and processing constraints, considering relevant parts in the assembly plan. Given the characteristics of the discrete manufacturing industry and actual demands, time arrangement constitutes the primary factor. The production scheduling of discrete manufacturing industry can be described as: a batch of parts needs to be produced; the number of parts is n ; each part contains many working procedures; the batch of parts is completed using p sets of machines in the shortest possible time. The main constraint conditions for production scheduling are as follows: (1) At the beginning of production, each part can be randomly selected and processed on the designated machine. (2) Each piece of equipment used for production in the workshop can only process one part at any time. (3) Each workpiece can only be processed once or not at all on each piece of equipment. (4) Sudden interruption is forbidden after processing has started. (5) Any part in the first process is not in order, but the same part in production has certain sequence constraints that absolutely cannot be changed. (6) Part production must conform to the actual process line and needs to be practical. (7) The processing time of each part process has been determined and does not change with sorting. (8) Auxiliary time such as tool installation and part transportation is not considered.

According to the actual production situation of discrete manufacturing industry, each part has a certain entry point, but assembly can only occur after all relevant parts have been produced. To improve part production and assembly efficiency, it is necessary to minimize the overall production time of the batch of parts. Mathematical modeling aims to find the completion time of the entire batch of parts, that is, the latest time to find the completion of the last parts. The mathematical model is as follows:

In terms of the production characteristics of parts in discrete industry workshops, the calculation formula for the latest processing time of process P of part j is as follows:

The processing time of machine M is: The end time of part j for process P is:

The latest production time for finishing the last step of part is:

The meaning of each variable in the mathematical model is shown in Table 1. The mathematical model describes production scheduling; through the model, the production time of each plan can be calculated, and then the time consumption of each plan can be compared to select the best scheduling plan with the shortest time.

3.1. Chromosomal Coding

There are many encoding methods for chromosomes in genetic algorithms, and an appropriate encoding method can improve the efficiency and ability to find the global optimal solution. Due to the complexity and particularity of discrete production scheduling, a procedure-based coding method is chosen in this work, and the encoding mode of chromosomes is shown in Figure 1 [Figure 1: see original paper]. The length of chromosomes is related to the number of machines and parts. So, if there are m machines, n parts, and process i ($0 \leq i < n$) processing procedures of each machine, the length of the chromosome is $chSize \leq (n \times m)$.

Since the number of processes in the processed parts may not be equal, and the number of processes in some parts is less than the number of machines, the length of chromosome $1 \dots (0) \max() \dots (0)$, $== iiMjPjPMmWTPSEmWTPiMjPmWTST=+iijPjPEST=+0\max()iimmWT$ should be the sum of the number of processes in all parts to be processed. In specific chromosomes, the coding rule is that the number of times a part appears on a chromosome indicates the number of processes it needs to be processed. For example, the first occurrence of j_1 in a chromosome means the first process of j_1 part, and the fourth occurrence means the fourth process of j_1 , based on which the process of other workpieces can be deduced.

Figure 1. Discrete scheduling chromosome encoding scheme

Table 1. Variables in the mathematical model

Meaning of variable | Latest start time of process j for part i | Completion time of process j for part i | Number of the machine required for part i to perform process j | Time taken to complete process j for part i | Process of processing part i using machine j | Machine processing time with serial number i | Process of machining part with serial number i | The part i | Current operation P of j | Number of machines P | Processing time of P by machine M |

3.2. Design of Fitness Function

In the genetic algorithm optimization process, the fitness function is mainly used to evaluate the advantages and disadvantages of chromosomes and constitutes the only condition to identify individual quality in the population. Under the condition of minimizing completion time in discrete manufacturing industry production scheduling, the fitness function value is the maximum completion time obtained in accordance with the mathematical model created above, and the fitness function can be used to effectively determine chromosome quality. In this method, the larger the fitness function value, the longer the completion time of all parts, and the smaller the chromosome fitness, making it easier to be eliminated in the selection operation. Conversely, the smaller the fitness function value, the shorter the completion time of all workpieces, and the larger the chromosome fitness, making it easier to be selected and inherited to the next

generation population.

3.3. Selection Operation

When selecting individuals in a population, excellent individuals are selected with high probability, while those with low fitness are selected with low probability. This ensures the population evolves in a favorable direction at the beginning of iteration. Through comparative analysis, the tournament method is employed for individual selection in this work. The main operation method randomly selects Z individuals from the population, lets them compete for fitness, and then selects the best one. In this paper, all individuals participating in the championship constitute the whole population, and Y is set to 3, meaning 3 random individuals are selected from the population, and then the optimal individual j is selected. The steps are shown in Figure 2 [Figure 2: see original paper].

Figure 2. Tournament selection of excellent individuals

There are usually three main methods for selecting individuals in genetic algorithms. The first is the more commonly used tournament method, with operation method as described above. The second is the roulette selection method, which mainly determines individual selection probability through comparison of individual fitness values in the population. According to probability, the composition of the offspring population is determined. For solving the time minimization problem of production scheduling, it is necessary to transform the fitness function into a maximization problem, where the fitness value of each individual is obtained separately, and the fitness value of each individual is divided by the sum of fitness values of all individuals. The result is the probability of individual selection. The cumulative probability of all individuals constitutes a roulette wheel, and a new generation population is continuously obtained by generating random numbers between $[0, 1]$ for roulette selection. The third random traversal sampling method has the same individual selection probability as the second method. The difference is that to meet the requirement of equidistant individual selection—for example, if m individuals need to be selected, the distance of the selection pointer should be $1/m$, and the position of the first pointer can be determined by generating a random number in $[0, 1/m]$. Through analysis of the above methods, the advantages of tournament selection include small time complexity, easy parallel processing, and reduced premature phenomenon.

3.4. Crossover Operation

One of the key operations in the continuous evolution of population is crossover operation. Crossover operation changes the gene sequence of parts on the basis of preserving chromosome gene fragments, which increases population diversity, improves algorithm search ability, and increases the probability of excellent individuals. Different from previous crossover operations, this work adopts an

adaptive adjustment strategy to maintain individual diversity at the initial stage of population iteration and ensure convergence of optimization results as soon as possible at the later stage. An OX crossover method based on process coding is adopted in the hybrid algorithm. The operator crossover diagram is shown in Figure 3 [Figure 3: see original paper] and Figure 4 [Figure 4: see original paper]. In the crossover process, the starting and ending positions of chromosomes of the parent and mother are random, so the randomness of newly generated individuals is greater.

Figure 3. Parent chromosomes generated offspring

Step 1: During crossover, two chromosomes P1 and P2 are randomly selected each time according to the selection operation as the male and female parent.

Step 2: One chromosome is selected as the paternal parent, and then a gene fragment is intercepted in the paternal parent by randomly determining two positions in the chromosome. The intercepted gene fragment is used as a progeny chromosome prototype. The operation process is shown in Figure 3.

Step 3: The remaining chromosome is taken as the parent, and then the missing codes of the progeny prototype are completed from the mother, as shown in Figure 4.

Figure 4. Mother chromosome completes the progeny chromosome prototype

Step 4: The first child is generated through Steps 2 and 3 above, and then the second child is generated by repeating Steps 2 and 3.

The crossover probability is P_c , and the probability can be adjusted adaptively by introducing the number of iterations. The crossover probability is closely related to individual fitness and the number of iterations. The calculation formula of the probability is as follows: where f_{max} is the maximum fitness value in the population; f is the larger fitness value of the two chromosomes that need to be crossed; f_{avg} is the average value of fitness of all individuals in the population; gen is the number of iterations of the population so far; $k1$ is a fixed size adjustment parameter ranging from 0 to 1.

3.5. Mutation Operation

Mutation operation has a relatively small probability in genetic algorithms but significantly impacts population diversity. Therefore, the mutation probability should be adjusted slightly in the same way as crossover operation. The mutation probability has a great relationship with chromosome fitness values and population iteration numbers. The mutation operation is mainly based on the location mutation method, which randomly selects chromosomes in the population according to probability, randomly determines two locations on chromosomes, and carries out gene exchange to generate new chromosomes. In this

algorithm, two pairs of genes on chromosomes are exchanged, as shown in Figure 5 [Figure 5: see original paper].

Figure 5. Chromosomal mutations

The probability of mutation operation is also adjusted adaptively as the number of iterations increases, and the adjustment formula is shown as follows: where f is the fitness value of individuals that may be mutated at present; k_2 is the adjustment parameter of individual mutation, generally between (0, 1). Other variable names in the formula have the same meanings as described in (4).

3.6. Simulated Annealing Operator

To improve the local searching ability of genetic algorithm, a new hybrid genetic algorithm is proposed. A simulated annealing algorithm is created, which differs from previous genetic-simulated annealing algorithms in its adjusted simulated annealing operator. In past hybrid genetic algorithms, the simulated annealing operator lacked memory function. However, in this work, using memory function saves the optimal solution generated in the annealing process. The optimal value is determined by comparison in the simulated annealing operator, which is not applicable to the algorithm because the optimal value is highly likely to be ignored. For this IHGA, the optimal value is searched from the population multiple times at each temperature state, and the optimal value in this round is determined with a certain probability by reaching the specified optimization times. At temperature T_k (k is the number of cooling times), the optimal value in the population is compared with the randomly selected value. T_1 is the initial temperature, others have similar meanings in turn, and T_{min} is the end temperature, which is set to 0.001 in this work. The cooling formula added in this work is as follows: where α is the cooling coefficient.

In the simulated annealing process, to prevent the temperature from falling too fast at the initial cooling stage, an appropriate heating strategy is adopted in the annealing process, which is beneficial for increasing the acceptance probability of various chromosomes, ensuring more diverse chromosome selectivity, and avoiding local optimization. The search steps are as follows:

Step 1: The current state bit is set as S , the initial value of the cycle counter is $d = 1$, the initial value of the counter for the new individual of the generation population is $O = 1$, and the length of the Markov chain is L .

Step 2: Select v individuals with the lowest fitness value from the population generated after mutation operation as the initial solution in annealing operation, and make the current state bit $S = v$.

Step 3: Select an individual randomly from the population, determine this state bit as $S = v$, and calculate the increment of fitness value as $dE = f(v) - f(v)$.

4.1. Comparison of IHGA and Existing Popular Algorithms

For the discrete shop scheduling problem, the improved particle swarm optimization (IPSO) proposed by Liu et al. [32] and the Quantum Whale optimization algorithm (QWOA) proposed by Yan et al. [33] can deal with some production scheduling problems. However, their optimization performance obviously has some shortcomings compared with the hybrid algorithm proposed in this work. Based on two kinds of algorithm about case set scheduling, comparing with IHGA obtained case set scheduling results, comparison of the two aspects mainly is the average of the scheduling results of optimal solution and the solution of data comparison, as shown in Table 2. The $n \times m$ represents the total number of parts and machines; the product of for the optimal solution has been obtained, the Avg. runs the algorithm ten times to get the average of the solutions. According to the comparison results in Table 2, for scheduling case sets FT06, LA01 and LA06, the IHGA algorithm proposed in this work can obtain the known optimal solution due to the combination of the excellent local search ability of simulated annealing operator.

As shown in Table 2, although IHGA did not obtain the known optimal solution for algorithm case set FT10 and FT20, the relevant solutions obtained by IHGA are superior to the other two algorithms, with significantly stronger optimization capability.

Table 2. Comparison of IHGA, QWOA and IPSO

Numerical example The size is 1010 1010 The optimal The optimal The optimal solution solution solution

4.2. Comparison of IHGA and Traditional GA

To further confirm that the improved algorithm is superior to the standard genetic algorithm, the algorithm proposed in this work is compared with the genetic algorithm in searching results, as shown in Table 3.

Table 3. Comparison of IHGA and GA optimization results

Numerical example ($n \times m$) Improved effect /% Optimal solution Optimal solution 1010 Optimal solution 1010

Table 4. Details of LA03 datasets

Process Each process processing machine and processing time Process 0 Process 1 Process 2 Process 3 Process 4 Artifacts Artifacts 1 Artifacts 2 Artifacts 3 Artifacts 4 Artifacts 5 Artifacts 6 Artifacts 7 Artifacts 8 Artifacts 9 Artifacts 10

It can be seen from Table 3 that in FT06, LA01, LA03, LA06, LA08 and LA13 cases, both algorithms can finally obtain the optimal solution of the scheduling problem, but the average value obtained by IHGA is superior to GA. The IHGA did not find the optimal solution of scheduling cases in FT10, FT20 and LA18.

However, compared with GA, the optimization effect was greatly improved, with the improvement effect ranging from 4.2% to 6.9%, and the average improvement effect ranging from 5.7% to 8.4%. In the comparison of the two algorithms, the search result is greatly improved mainly because the simulated annealing operator enhances the local search ability of genetic algorithm, which substantially improves the search capability of the algorithm proposed in this work.

Taking LA03 in the test datasets as an example, the feasibility and superiority of the algorithm proposed in this work in solving discrete scheduling problems are explained in detail. The specific datasets of LA03 are shown in Table 4. The data in Table 4 represents that processing parts by a process uses the serial number of the machine and the processing time. For example, in the corresponding parts in Table 2, the 1/29 means in the processing of selecting the numbers for 1 second process machinery for processing, it takes time for 29, and other data has the same meaning.

Figure 7 [Figure 7: see original paper]. LA03 convergence curve

For example, it can be seen from the comparison of results in Table 3 that both IHGA and GA algorithms can find the optimal solution, but the average value obtained by IHGA is better than that obtained by GA, which means that IHGA has good robustness. After a period of comparative testing, the convergence of the iterative curves of the two algorithms in dealing with the LA03 scheduling problem is shown in Figure 7. As can be seen from Figure 7, GA not only has a slow convergence speed when searching for the optimal value, but also cannot find the optimal value, converging at about 630. The IHGA has a relatively fast convergence speed and can jump out of the local optimum in time to find the global optimum value of 597. This indicates that the IHGA proposed in this work is obviously superior to the genetic algorithm and can more effectively solve production scheduling problems in workshops.

Taking LA03 as an example, the IHGA can work out the currently known optimal fitness solution 597, select the chromosome with the best fitness, and obtain the corresponding scheduling Gantt chart according to the process arrangement results of each part in the chromosome, as shown in Figure 8 [Figure 8: see original paper]. The Gantt chart can prove that the IHGA proposed in this work can provide an excellent and effective solution for production scheduling in discrete industries.

Figure 8. Scheduling Gantt diagram of LA03 obtained by the IHGA

5. Industrial Field Test

A MES system is developed using C# object-oriented development language, and the corresponding database is established by SQL Server, where the production scheduling function adopts the proposed IHGA. To verify the feasibility of the MES system, industrial application tests were carried out with the help of a network environment and hardware platform of discrete manufacturing

industry. The reducer production project of the industry was selected as the basis, and the business process of the designed MES system was verified in the workshop. The specific steps are as follows:

Step 1: Production plan. According to order demands, the production planner makes a specific reducer production project, where 12 kinds of parts are set in the production process, and the project is distributed after the process and working hours of each part are modified. The system records the names of the employees who distribute and receive the production plan. The distributed production plan contains basic information.

Step 2: Production scheduling. The team leader of the production unit checks the distributed production plan to conduct scheduling optimization according to the IHGA, and the optimized results through Gantt chart are displayed as shown in Figure 10 [Figure 10: see original paper]. The processing sequence and corresponding processing time of different parts on different equipment are shown, and further optimization of the production plan has been realized, including specific production time and operating staff.

Step 3: Production and processing. Workshop operators receive the production plan distributed by the team leader, log in to the MES system according to their own authority, and obtain the parts to be produced. Then, the parts are produced according to the production process time and process drawings. The system records the name and production time of employees, and the production team leader using the system can see the production status of the process at any time, as shown in Figure 9 [Figure 9: see original paper].

Figure 9. Gear shop operation diagram

The industrial experiment proves that the IHGA can realize production scheduling for the industry, enabling the production of a variety of products to be completed in a relatively short time and greatly improving workshop production efficiency. According to the detailed production plan generated after scheduling, the on-site production situation of the workshop is shown in Figure 10. It can be further seen that when the operator uses drilling machine and lathe to process the reducer workpiece, the optimized software can run normally and record the production time, equipment name, and employee name to reflect the processing status of the parts and facilitate traceability of defective products. Real-time display is carried out on the large screen of the workshop to reflect staff working efficiency, equipment running status, and production situation, realizing transparent production in the workshop.

Figure 10. Production scene diagram of discrete manufacturing industry

5. Conclusions

Firstly, this paper introduces the basic situation of production scheduling in discrete industries. Considering that simulated annealing algorithm can improve

the shortcomings of genetic algorithm optimization, an improved hybrid genetic algorithm is proposed to solve production scheduling problems. In this work, the shortcomings of genetic algorithm are analyzed and studied, and an adaptive strategy is adopted to adjust the probability of crossover and mutation of genetic operators. The adjustment mainly depends on the number of iterations and fitness, ensuring high population diversity in the early stage and enabling convergence in the later stage. For the added simulated annealing operator, the elitist substitution strategy is adopted to save the optimal solution and avoid local optima in genetic algorithm. The results show that the improved hybrid genetic algorithm has better optimization ability compared with other scheduling algorithms. Compared with the genetic algorithm, although the optimal solution was not obtained in three cases, the improvement effect of the optimal solution was 4.2% ~ 6.9%, and the average improvement range was 5.7% ~ 8.4%. The simulated annealing operator played a good local optimization effect in IHGA, which greatly improved the optimization ability of the overall algorithm.

In the actual production process of discrete industries, production scheduling considers more complex indexes, so the next main task is to study flexible production scheduling with multiple indexes to improve workshop production efficiency and reduce resource waste.

Acknowledgements The research leading to these results has received funding from the Norwegian Financial Mechanism 2014-2021 under Project Contract No 2020/37/K/ST8/02748.

References

1. Wang P., Xiao X. C., Yuan L. H., Fang J.. Application and Research in Discrete Manufacturing Industry Based on MES, *China Science and Technology Information*, 2015 (18): 119-120.
2. Yang N.. Implementation and Application of Discrete Manufacturing MES System, *Equipment Manufacturing Technology*, 2020 (05): 273-275+278.
3. Jin B. F., Li B.. Multi Objective Job Shop Scheduling Based on Genetic Algorithm, *Modular Machine Tools and Automatic Processing Technology*, 2019 (4): 4.
4. Cao R., Hou X. P., Jin S. T.. Research on Flexible Job Shop Scheduling Problem Based on Improved Genetic Algorithm, *Computer and Digital Engineering*, 2019, 47 (2): 4.
5. Lu y., Feng Z. J., Hu Y.. Research on Scheduling Algorithm for Multiple Handling Requests of A Single AGV Car, *Modular Machine Tools and Automatic Processing Technology*, 2019 (2): 4.
6. Kumar N., Mishra A.. Comparative Study of Different Heuristics Algorithms in Solving Classical Job Shop Scheduling Problem, *Materials Today: Proceedings*, 2020, 22: 1796-1802.
7. Kong L., Wang L.M., Li F. i.. Green Production Scheduling of Hybrid Flow Shop Based on Machining Matching Characteristics of Machine Tool,

- Computer Integrated Manufacturing System*, 2019, 25 (5): 11.
8. Xing G., Nyberg T. R.. Comparison Between CIMS and CIPS, *IEEE International Conference on Emerging Technologies & Factory Automation*. IEEE, 1999: 409-416.
 9. Chen X. H.. Research on Reconfigurable Architecture and Key Technologies of Discrete Manufacturing Execution System Based on Semantic Gateway, Chongqing: Chongqing University, 2014.
 10. Chang J. T.. Research on Some Key Technologies in Production Management of Discrete Manufacturing Enterprises, Xi' an: Xi' an University of Electronic Science and technology, 2014.
 11. Zhong R. Y., Huang G. Q., Dai Q. Y., et al. RFID-enabled Real-time Manufacturing Execution System for Discrete Manufacturing: Software Design and Implementation, *IEEE International Conference on Networking*. IEEE Xplore, 2011: 311-316.
 12. Li F., Li X. H.. Application Practice of Manufacturing Execution System in Discrete Industries, *Intelligent Manufacturing*, 2008 (1): 24-27.
 13. Bian D. Z., Guo J. M., Wang M., Song F. F.. Research on MES System Based on Discrete Shipbuilding, *Manufacturing Technology and Machine Tools*, 2020 (01): 149-152.
 14. Mao K., Pan Q. K., Pang X. F.. Lagrangian Algorithm for Steelmaking Continuous Casting Production Scheduling Problem, *Journal of Systems Engineering*, 2014, 29 (2): 13.
 15. Bellabdaoui A, Teghem J. A Mixed-integer Linear Programming Model for The Continuous Casting Planning, *International Journal of Production Economics*, 2006, 104(2): 260-270.
 16. BELLMAN R. E., DREYFUS S. E.. *Applied Dynamic Programming* [M]. [S.1]: Princeton University Press, 2015.
 17. Wu D. W., Lu T. T., Liu X. B., et al. Parallel Simulated Annealing Algorithm for Job Shop Scheduling Problem, *Computer Integrated Manufacturing System*, 2005, 11 (6): 847-850.
 18. Zhang J., Wang W. L., Xu X.L.. et al. Hybrid Particle Swarm Optimization Algorithm for Multi-objective Flexible Job Shop Scheduling Problem, *Control Theory and Application*, 2012, 29 (6): 8.
 19. Wang W. L., Zhao C., Xing J.. A Method for Solving Flexible Job Shop Scheduling Problem Based on Improved ant Colony Algorithm, *Journal of System Simulation*, 2008, 20 (16): 4.
 20. Choi I. C., Choi D. S.. A Local Search Algorithm for Job Shop Scheduling Problems with Alternative Operations and Sequence- dependent Setups, *Computers & Industrial Engineering*, 2002, 42(1): 43-58.
 21. Nie L., Gao L., Li P., et al. A GEP-Based Reactive Scheduling Policies Constructing Approach for Dynamic Flexible Job Shop Scheduling Problem with Job Release Dates, *Journal of Intelligent Manufacturing*, 2013, 24(4): 763-774.
 22. Chen J. C., Chen K. H., Wu J. J., et al. A Study of The Flexible Job Shop Scheduling Problem with Parallel Machines and Reentrant Process, *The International Journal of Advanced Manufacturing Technology*, 2008,

- 39(3-4): 344-354.
23. Lu T. T., Chen P., Wan X. Y.. An Improved Cellular Genetic Algorithm for Solving Multi-objective Flexible Job Shop Scheduling Problem, *Modern Manufacturing Engineering*, 2016, 000 (011): 41-49.
 24. Gao J., Gen M., Sun L., et al. A Hybrid of Genetic Algorithm and Bottleneck Shifting for Multi-objective Flexible Job Shop Scheduling Problems, *Computers & Industrial Engineering*, 2007, 53(1): 149-162.
 25. Pezzella F., Morganti G., Ciaschetti G.. A Genetic Algorithm for The Flexible Job-shop Scheduling Problem, *Computers & Operations Research*, 2008, 35(10): 3202-3212.
 26. De Giovanni L., Pezzella F.. An Improved Genetic Algorithm for The Distributed and Flexible Job-shop Scheduling Problem, *European Journal of Operational Research*, 2010, 200(2): 395-408.
 27. Zhang R., Chang P. C., Wu C.. A Hybrid Genetic Algorithm for The Job Shop Scheduling Problem with Practical Considerations for Manufacturing Costs: Investigations Motivated by Vehicle Production, *International Journal of Production Economics*, 2013, 145(1): 38-52.
 28. Chamnanlor. Embedding Ant System in Genetic Algorithm for Re-entrant Hybrid Flow Shop Scheduling Problems with Time Window Constraints, *Journal of Intelligent Manufacturing*, 2015, 40(10): 1-17.
 29. Meng Y., Zhao S. H.. Solving Job Shop Scheduling Problem with Hybrid Algorithm Integrating Path Reconnection, *Mechanical Engineer*, 2019 (12): 32-36+39.
 30. Zhao W.. Application of Simulated Annealing Genetic Algorithm in Job Shop Scheduling, *Computer Simulation*, 2011, 28 (07): 31.
 31. Wang J. H., Lu C.. Application of A Genetic Algorithm Combined with Simulated Annealing in Flexible Job Shop Scheduling, *Digital Technology and Application*, 2019, 37 (01): 133-136.
 32. Liu H. M., Zeng H. Y., Zhou W., Wang T.. Optimization of Job Shop Scheduling Problem Based on Improved Particle Swarm Optimization, *Journal of Shandong University (Engineering Edition)*, 2019, 49 (01): 75-82.
 33. Yan X., Ye C. M., Yao Y.. Quantum Whale Optimization Algorithm for Job Shop Scheduling, *Computer Application Research*, 2019, 36 (04): 975-979.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv – Machine translation. Verify with original.