International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 7382-7390

https://doi.org/10.70517/ijhsa463633

Particle swarm optimization based integrated process planning and scheduling problems

Fang Han[1,*] and Lijun Liu[1]
[1] School of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an, Shaanxi, 710021, China
Corresponding authors: (e-mail: littlefang2025@163.com).

Abstract The integrated process planning and scheduling problem is a key aspect in manufacturing systems. This paper investigates the integrated process planning and scheduling integration problem based on nonlinear process planning. The study adopts particle swarm optimization algorithm, designs a new flexible scheduling method based on "cursor", and uses the particle coding method integrated with the process to realize the simultaneous optimization of the integrated process planning route and scheduling. The experimental results of the algorithm show that, compared with the genetic algorithm, the proposed particle swarm optimization algorithm completes the convergence of the two objective functions of completion time and makespan in 59 and 55 iterations, respectively, with the convergence values of 355.32 s and 620.75. In the tests of 10 problems of different sizes, the average value of the IGD of this paper's algorithm is always within 300, which proves that the nondominated frontier obtained by it is closer to the true frontier. The rescheduling experiments under dynamic events show that the Best makespan results sought by the particle swarm algorithm are reduced by 2.32%-10.26% and the average makespan is reduced by 4.40%-10.38% compared with that of the genetic algorithm. It is shown that the integrated process planning and scheduling integration method based on particle swarm optimization proposed in this paper has better convergence in solving the two sub-problems of process route planning and batch scheduling sequencing, and is able to optimize the production process of the plant more effectively.

Index Terms Particle swarm optimization algorithm, Integrated process planning, Scheduling integration, Flexible process, Multi-objective optimization, AND/OR network diagrams

I. Introduction

Process planning and shop floor scheduling are two important components of flexible manufacturing systems. The former plans the production process based on the processing characteristics of the product, identifies the manufacturing resources available for production, and determines the processing sequence based on the cost-effectiveness metrics so that raw materials can be converted into products [1]-[3]. The latter is used to determine how to allocate the operations of all jobs to shop floor production resources under constraints such as time feasibility and resource availability [4]-[6]. Process planning and shop floor scheduling have different functions in a manufacturing system, and although they are usually executed in a sequential manner, they influence and constrain each other [7], [8]. Since the real-time feasibility of scheduling the shop floor is not considered in the process planning stage, the objective functions of the two are not optimized in a coordinated manner. Therefore, the "optimal" process route from process planning is not "optimal" in production, and the allocation of machines and resources by this process route often fails to achieve the desired optimal allocation [9]-[12]. It can be seen that the metrics of the process planning system are not effectively optimized due to the resource constraints of the scheduling system [13]. In addition, scheduling production for the information generated after the implementation of the process planning strategy cannot be fed back to the process planning system for improvement [14]. Therefore, the integrated production process planning and scheduling problem allows process planning to design all feasible process routes for each workpiece, and scheduling to prefer and develop an optimal integrated scheduling plan for each designed process route by weighing the constraints and based on the real-time state of the shop floor [15]-[17]. This enables the selection of appropriate production routes according to the actual situation of the production process, thus effectively eliminating resource conflicts and improving the performance of the manufacturing system [18].

In the modern manufacturing environment, integrated process planning and scheduling integrated optimization of processes is an important means to improve the overall efficiency of the manufacturing system. The key problem faced in current factory production is how to find a suitable machining method for the workpiece to optimize the specific performance of process machining. The effective solution of the process planning and scheduling problem is of great significance to improve the production efficiency of the enterprise and to respond to changes in market demand. The integrated process planning and scheduling optimization mode of traditional processes mostly focuses on the local optimization of the process planning system, with a low degree of overall integration and a lack of in-depth integration of intrinsic functions. This separated optimization method cannot fully consider the mutual influence between process planning and scheduling, which limits the overall effect of optimization. The nonlinear process planning model, as a production model based on static manufacturing, is conducive to finding the global optimal solution by generating all possible process routes and selecting the most suitable process route according to specific workshop resources. Flexible process route planning is able to provide a variety of alternative machining choices, such as process routes with the same machining characteristics and different equipment that can accomplish uniform machining tasks, given the resources, which plays a very important role in improving the flexibility of the workpiece production process and the ability to deal with unexpected events. In process route planning, AND/OR network diagram can visually express process flexibility, clearly distinguish "or" relationship and "with" relationship, and effectively describe complex process structure. As an intelligent optimization method, particle swarm optimization algorithm has the advantages of simple implementation and high computational efficiency, and shows better performance in solving the integrated problem of process planning and scheduling. Based on the above understanding, the study in this paper focuses on how to realize the integrated optimization of integrated process planning and scheduling by particle swarm optimization algorithm.

In this study, the integrated process planning and scheduling integration problem model is first established, and the constraints and optimization objectives of the problem are clarified. Then, an integrated process planning and scheduling optimization algorithm based on particle swarm optimization is proposed, which adopts a particle coding method that integrates the process planning route composition and batch scheduling sequencing, and improves the traditional particle velocity expression. In order to improve the decoding efficiency, a gap insertion decoding method suitable for multi-objective scheduling is designed. The advantages of the proposed algorithm in terms of convergence and decoding quality are verified through comparison experiments with the genetic algorithm. Meanwhile, a series of dynamic events are constructed to test the rescheduling results and evaluate the performance of the algorithm in complex dynamic environments.

II. Study on the integration of integrated process planning and scheduling

Traditional integrated optimization of integrated process planning and scheduling for processes is mostly just some degree of optimization of the process planning system, with a low total integration scale and lack of much intrinsic functionality fusion. This paper investigates the integrated process planning and scheduling problem based on nonlinear process planning.

II. A. Integrated Planning and Scheduling Integration Problem Description

The main purpose of integrated process planning and scheduling integration is to optimize the processing through the integration of production processes. The nonlinear process planning model is a production model based on static manufacturing. It is defined as follows: n workpieces will be processed on m machines, where each workpiece contains different processes, there will be a certain processing sequence of these processes, and each process can be processed on different machines and equipment with different processing times. The objective of the integrated planning and scheduling problem is to find a suitable processing method for each workpiece that optimizes some specific performance of the process.

Non-linear process planning is characterized by generating all possible process routes, and then selecting the most suitable process route based on specific shop floor resources, this method is conducive to finding the optimal process route.

Flexible process route planning refers to the fact that in the case of a given resource, many parts of the product processing process can be replaced, such as process routes with the same processing characteristics and different equipment that can accomplish uniform processing tasks. Process route and processing equipment flexibility, to improve the flexibility of the workpiece production process and the ability of production staff to deal with emergencies has a very important role.

In this paper, we use AND/OR network diagram, which is an intuitive expression of process flexibility, to describe the flexible process. Among them, OR is the meaning of "or", meaning that the parallel processes under the node are equivalent and replaceable, and any branch can complete the task at the node. A parallel process route branch node that does not have the word OR is an AND node, i.e., "with", meaning that different process routes under the node are executed, but there is no strict prioritization structure between the processes on the different branches.

II. B.Modeling the process planning and scheduling integration problem

Suppose that the factory produces a batch of products with a total number of orders (tasks, workpieces) of n, and the set of orders is $J = J_1, J_2, ..., J_n$. Each order can be completed by a different process, if the number of processes to complete the order J_i is R_i, then let $Pr_{i,j} (i = 1, 2, ..., n; j = 1, 2, ..., R_i)$ denotes the j th process of the order J_i. Each process consists of a number of different processes, noting that the k process of $Pr_{i,j}$ is denoted as $O^k_{i,j} (k = 1, 2, ..., JN_{i,j})$, $JN_{i,j}$ being the number of processes contained in the process $Pr_{i,j}$. The same process can be processed on different machines and equipment, but it takes different times to process the flowers on different machines and equipment. Denote the set of machines and equipments to complete the process $O^k_{i,j}$ as $M^k_{i,j}$, the set of machines and equipments contained in $M^k_{i,j}$ is $m^k_{i,j}$, and the set of times corresponding to $M^k_{i,j}$ is $P^k_{i,j}$. The plant facility layout, material handling and pre-processing are not taken into account. The goal of the algorithm is to provide the decision maker with a complete process plan and scheduling solution (including the sequence of process operations, operating equipment, and operating times in the process plan) based on the task list, with the shortest possible time to complete the task.

The following assumptions need to be made in order to easily describe the system model for process planning and integrated scheduling:

(1) The different workpieces are independent of each other, are not interconnected, and have no priority.
(2) Each machine and equipment can only process and handle one workpiece in its work.
(3) Different processes of the same workpiece cannot be processed at the same time.
(4) No interruptions are allowed once any process has begun.
(5) Machines and equipment are idle and initially, all workpiece tasks are executable.
(6) The transfer time between processes is zero, i.e., there is no transfer operation.
(7) There is no correlation between the preparation time of the processes and the machining sequence, and it is assumed that the preparation time of the processes is included in the machining time of the processes.

III. Integrated process planning and scheduling model based on particle swarm optimization

III. A. Particle Swarm Optimization Algorithm

The standard particle swarm optimization algorithm [19], [20] is an intelligent optimization algorithm inspired by the foraging behavior of a flock of birds, which may forage individually or in groups, such that the foraging behavior is completely random. The position of each individual bird in the flock is also random, some individuals are in a better position, while some individuals are in a less than ideal position. Therefore, the individuals in the flock of birds can be regarded as each individual particle, and containing m individual particles is a population, and the whole particle swarm algorithm can be regarded as operating in a D-dimensional search space. Each individual particle can also be regarded as a D-dimensional vector, which can be used as $X^t_i = (X^t_{i,1}, X^t_{i,2}, ..., X^t_{i,D})$ to represent the position of the i th individual particle at the t th moment, which can be represented by $v^t_i = (v^t_{i,1}, v^t_{i,2}, ..., v^t_{i,D})$ to denote the velocity of the individual particle, $(i = 1, 2, ..., m)$. Denote the historical best position of the i th particle individual in the population as p^*_i, and the historical best position of all particle individuals within the population as g_b. The positions and velocities of particle individuals in each generation are represented as follows:

$$\begin{cases} v_{t+1} = \omega v_t + c_1 r_1 (p_b - X_i) + c_2 r_2 (g_b - X_t) \\ X_{t+1} = X_t + v_{t+1} \end{cases} \tag{1}$$

Where ω is the inertia factor, r_1 and r_2 are random numbers between [0, 1], c_1 is the cognitive learning factor, c_2 is the social learning factor, both of which are constants, and $c_1 + c_2 = c$.

Particle swarm algorithm is an algorithm with simple and understandable steps, and its operation process can be roughly divided into 7 steps as follows:

Step 1: Initialization. Randomly generate a swarm of particles, each representing a possible solution in the solution space. Each particle has its own position and velocity.

Step 2: Calculate particle fitness. Calculate the fitness (or known as the objective function value) of each particle and evaluate its performance at the current position.

Step 3: Update the individual optimum. For each particle, compare the particle fitness and take the current position as the individual optimum.

Step 4: Update the population optimum. Select the global optimum from the individual optimums of all particles.

Step 5: Update particle velocity and position according to Eq.

Step 6: Update the individual optimum and population optimum. Recalculate the fitness of each particle based on the new position and update the individual optimum and population optimum.

Step 7: Terminate. If the stopping condition is satisfied (e.g., the maximum number of iterations is reached), the algorithm ends, otherwise return to step 5.

III. B. Integrated Planning and Scheduling Optimization Algorithm Flow

This paper adopts a multi-objective optimization algorithm based on particle swarm algorithm, in which a new flexible scheduling method based on “cursor” is adopted, and a particle coding method integrated with the process is used. The algorithm can optimize the process integrated planning route and scheduling at the same time. Integrated planning and scheduling optimization algorithm flow is shown in Figure 1.

(1) Coding

The appropriate particle coding method can not only visualize the solution problem, but also facilitate decoding and easy to realize the update of particle velocity and position. Flexible process integrated planning and scheduling contains two subproblems: route planning and batch scheduling sequencing. In order to accomplish the two subproblems solving at the same time, this paper adopts a particle coding method that integrates the two subproblems. The encoding consists of two parts, the first part indicates the composition of the process planning route, and the second part indicates the scheduling order of each batch.

(2) Particle velocity

In order to solve the problem in this paper, the traditional particle velocity expression is improved as follows:

$$v_{inpot_{(t+1)}} = rand(c_1)(rand_{inpot} - x_{inpot}) + rand(c_2)(p_{inpot} - x_{inpot}) \\ + rand(c_3)(arc_{(\forall i)npot} - x_{inpot}) \tag{2}$$

$$v_{i(t+1)} = \text{int}(v_{it}) + \text{int}(p_{it} - d_{it}) + \text{int}(arc_{(\forall i)t} - d_{it}) \tag{3}$$

where p_{inpot}, p_{it} are respectively an uncontrolled position of the variables x, d in the particle i, which is selected from the archive of the particle during the calculation. $arc_{(\forall i)npot}, arc_{(\forall i)t}$ denote the global optimal position of x, d in particle i, respectively, selected from the global archive. $rand(c_1), rand(c_2)$ and $rand(c_3)$ are 0, 1 integers randomly generated by the influence of c_1, c_2 and c_3, respectively, and int() is a -1, 1 integer affected by the value of the symbols in the parentheses, if the value greater than 0 then int() is 1, if less than 0 then -1.

(3) Decoding

Decoding is performed separately for the front and back parts, and for the front part, the batch size of the integrated planning path for each flexible process can be calculated. When decoding the latter part, the decoding is performed according to the gap-insertion decoding method. Processing sorting according to the batch, when calculating the completion time of the process, if it is the same type of workpiece as the previous process of a machine, the batch start time of the process is ignored, otherwise the batch start time of the process must be taken into account.

In order to improve the efficiency of decoding and reduce the search space of the problem, this paper designs a gap-insertion decoding method suitable for multiobjective scheduling. Assuming that a multi-objective scheduling problem has n optimal scheduling objectives, the particles adopt the process-based encoding method, and the weight coefficients of each objective of particle i are w_{in}. The process o in particle i can be machined on m machine tools, and the time required is t_i respectively. The p_m processes have been scheduled on machine m, and the starting machining time of each process is st_{pm} and the ending time is ct_{pm}. When decoding, for each machine, from the moment when process o can be processed to the moment when the last process in the machine queue is completed, it is judged whether process o can be inserted into each idle gap in turn. If it can, the process is inserted in the idle. Otherwise, the process will be placed at the end of the machine queue. The condition that process o can be inserted into idle p is $\max(ct_{pm}, pt_o) + t_i \le st_{(p+1)m}$ where pt_o is the machinable moment of process o. After obtaining the insertion position of process o on machine m, the start and completion times of process o if machine m is selected for machining are calculated, and each optimization objective corresponding to the scheduling scheme is calculated. Using the formula $F_m = \sum_{a=1}^{N} w_{in}(f_{n\_\max} - f_{mn}) / |f_{n\_\max} - f_{n\_\min}|$ Calculate the adaptability of the machining scheme of process o on machine m, where $f_{n\_\max}$ is the maximum value of machining target n on different machines and $f_{n\_\min}$ is the minimum value of machining target n on different machines. Finally, the machine with the largest fitness value is selected as the actual machining machine, and the machining pairs of each machine are updated. Decode the next process until the decoding is complete.

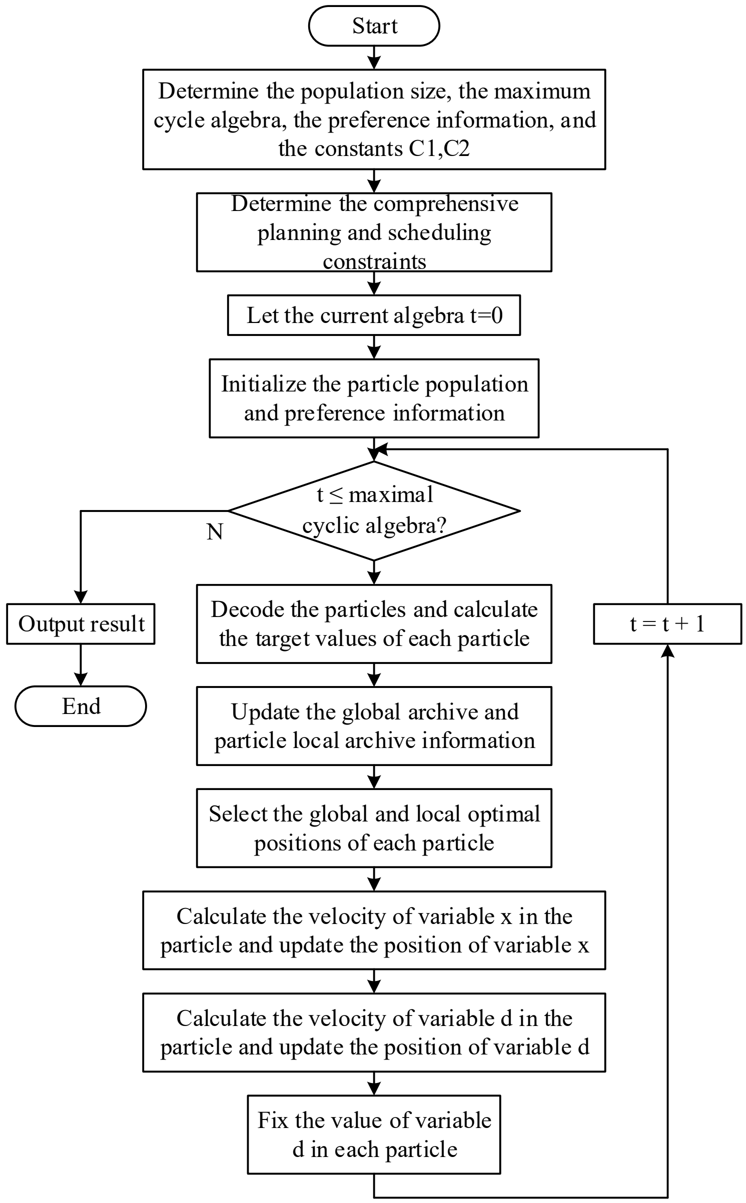

Figure 1: Comprehensive planning and scheduling optimization algorithm process

IV. Analysis of examples

IV. A. Verification of the convergence of the objective function of the PSO algorithm

In order to verify the effectiveness of the particle swarm optimization (PSO) algorithm, genetic algorithm (GA) [21] was chosen to simulate and optimize the integrated problem of process planning and scheduling integration of the process in MATLAB programming software. The parameters of particle swarm optimization algorithm are set as follows: the initial population size is 100, the inertia weight is 0.8, the acceleration constant c1=c2=1, and the maximum number of iterations is 200. The parameters of the genetic algorithm are set as follows: the initial population size is 100, the crossover probability is 0.90, the variance probability is 0.001, and the maximum number of iterations is 200.

After simulation experiments, the convergence curves of the two optimization objective functions of the integrated process planning and scheduling integration problems are shown in Figs. 2 and 3, respectively. Among them, the integrated process planning problem selects the completion time as the objective function, while the scheduling problem selects the scheduling efficiency index makespan as the objective function. As can be seen from the figure, the PSO algorithm for the completion time and makespan two objective functions, respectively, in the iteration of 59 times and 55 times to complete the convergence, the convergence value of 355.32s, 620.75, while the GA algorithm iterated to 70 times later, the completion time and makespan only converge one after another. The PSO algorithm selected in this paper has better convergence and the algorithm converges to a better solution, proving that the algorithm is feasible for solving the optimal integrated process planning and scheduling integration problem.

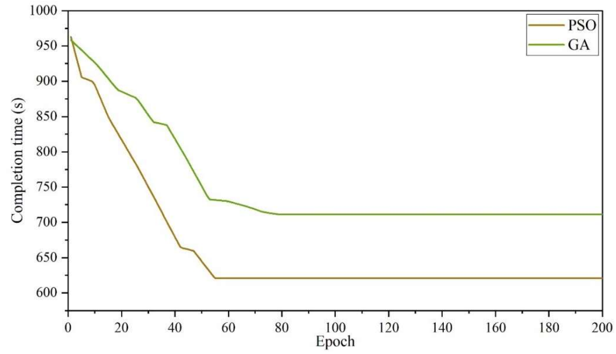

Figure 2: The integrated process planning target function is solved

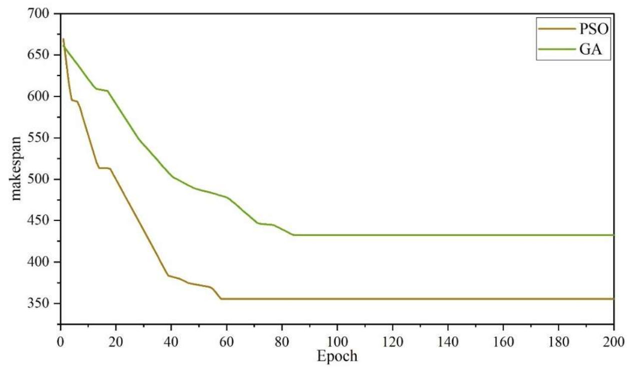

Figure 3: The scheduling target function is solved

In order to verify the effectiveness and superiority of the present PSO algorithm, it is compared with the GA algorithm, and the results of the Pareto frontier distribution are obtained as shown in Fig. 4. Due to the instance size limitation, intuitively, the distribution of non-dominated solutions derived by the PSO algorithm occupies a better position in the lower left corner of the GA algorithm, and the shortest completion time and makespan obtained are lower than those of the GA algorithm, which indicates that its search space is larger. Meanwhile, there are more infeasible solutions in the Pareto solution set of the GA algorithm that do not satisfy the processing order constraints, whereas the solution set of the PSO algorithm is full of feasible solutions that have been constrained to be tested and corrected.

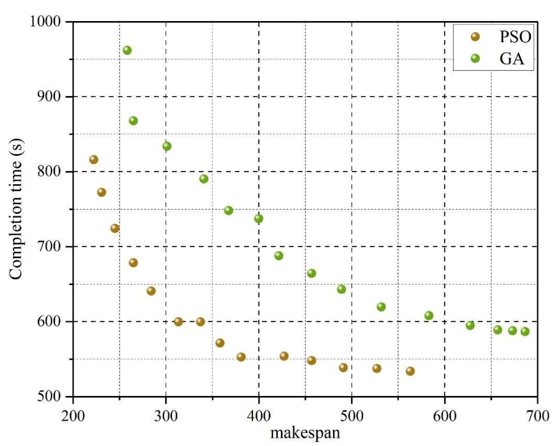

Figure 4: The Pareto frontier distribution results

IV. B. Evaluation of the effectiveness of the algorithm under different problem sizes

In order to test the effectiveness of this paper's algorithm on problems of different sizes, artifacts are randomly selected from 8 artifacts to form 10 problems of different sizes, and the same artifacts appear in problems 2 to 10. The algorithm of this paper is compared with the GA algorithm on 10 problems of different sizes. Each size of problem is tested 10 times, and the optimization performance of the algorithm is measured by Inverse Generation Distance (IGD), which is calculated as follows:

$$IGD(P^*, P) = \sum_{v \in P^*} d(v, p) / |P^*| \tag{4}$$

where, $P*$ is the PF found by the algorithm, P is the real PF of the algorithm, $d(v,p)$ is the smallest Euclidean distance between one of the solutions in the $P*$, v and P, and $|P*|$ is the number of $P*$. In this paper, smaller IGD values represent better results.

The average IGD values for 10 test algorithm comparisons are shown in Figure 5. As can be seen from the figure, the average IGD values of the experimental results of this paper's algorithm on Problems 1 to 10 all exhibit the smallest values, staying within 300. It indicates that the nondominated frontier obtained by this paper's algorithm is closer to the real nondominated frontier. As the problem size increases, the gap between the algorithm and the GA algorithm is more obvious, indicating that the algorithm proposed in this paper is more effective in solving the problem of integrated process planning and scheduling problems with a large problem size. The nondominated solutions obtained by the algorithm in this paper are feasible solutions and can guide the actual production.

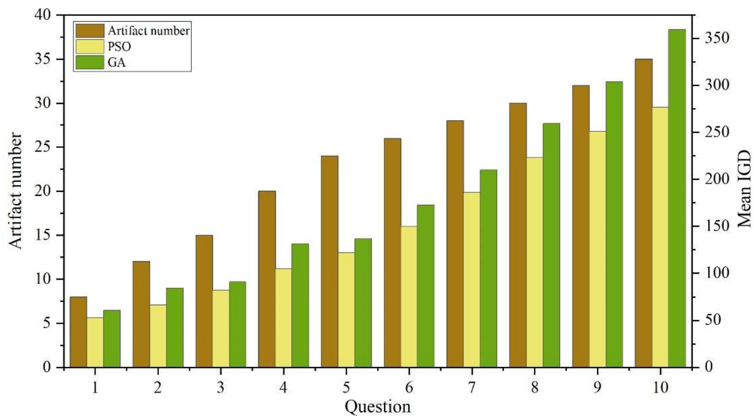

Figure 5: The IGD average compared to 10 test algorithms

IV. C. Example Analysis of Integrated Process Planning and Scheduling Problems

In order to verify the performance of the method proposed in this paper for solving the process integrated process planning and scheduling problem, a series of dynamic events are constructed to test the rescheduling results. The constructed dynamic events are as follows:

(1) Workpiece 1 arrives at 30 moments.
(2) Workpiece 2 arrives urgently at 90 moments.
(3) Machine 1 fails to be available at 100-150 moments.
(4) Machine 2 fails to be available at time period 150-200.
(5) Workpiece 3 arrives at 150 moments.
(6) Machine 3 is fault unavailable in the 200-250 time period.

In order to highlight the advantages of the algorithm selected in this paper, the GA algorithm is chosen to compare with the PSO algorithm, and the evaluation index of scheduling efficiency used is makespan. The simulation program is run for five times, and the optimization results of the process planning and scheduling of the process synthesis obtained by the calculation are shown in Table 1.

With makespan as the target, the Best makespan results obtained by applying the PSO algorithm are reduced by 2.32%-10.26% compared to GA. the average makespan results of the 10 calculations, obtained with the PSO algorithm, are reduced by 4.40%-10.38% compared to GA. In addition, the average completion time obtained under different dynamic events is reduced by 0.71%-9.47% for the results obtained with the PSO algorithm compared to GA. The above results analyze that the PSO algorithm chosen in this paper has better search performance than GA algorithm.

Table 1: Process scheduling efficiency

Algorithm		PSO			GA		
		Best makespan	Average makespan	Mean completion time (s)	Best makespan	Average makespan	Mean completion time (s)
Rescheduling frequency	0	421	424.1	514.7	431	443.6	518.4
	1	434	449.3	538.4	457	473.3	547.7
	2	450	457.8	544.5	479	499.8	579.5
	3	516	519.1	545.6	531	556.7	582.9
	4	542	550.4	547.9	571	582.5	594.1
	5	583	591.1	547.9	611	624.6	602.7
	6	612	622.2	547.9	682	694.3	605.2

The Gantt chart results for the integrated process planning and scheduling problem are shown in Figure 6. The horizontal axis in the figure represents the time span and the vertical axis represents the process machine task items. From the figure, it can be seen that the PSO algorithm can reasonably optimize the process planning of each machine task item and schedule the corresponding span of time to ensure the timely completion of the task.

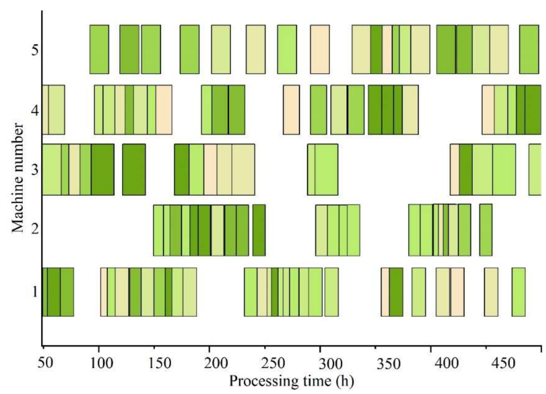

Figure 6: Gantt chart of the comprehensive process planning and scheduling

V. Conclusion

The particle swarm optimization algorithm shows superior performance in the integrated process planning and scheduling integration problem for process synthesis. Experiments show that the particle swarm algorithm can converge to a better solution more quickly than the genetic algorithm, reaching convergence in 59 and 55 iterations for the completion time and makespan objective functions, respectively, while the genetic algorithm requires more than 70 iterations to achieve convergence. Evaluating the optimization performance of different scale problems by IGD index, the particle swarm algorithm maintains the minimum IGD value in all 10 test problems, which is stably controlled within 300, proving that the non-dominated frontier it obtains is closer to the real non-dominated frontier. Especially in large-scale problems, the advantage of particle swarm algorithm is more obvious. In the dynamic event test, the average completion time of the particle swarm algorithm is 0.71%-9.47% lower than that of the genetic algorithm, and it maintains a better stability in the case of increased rescheduling frequency. The flexible scheduling method based on “cursor” and the particle coding strategy integrated with the process proposed in this paper effectively solves the two sub-problems of process route planning and batch scheduling sequencing, and the improved gap-insertion decoding method improves the decoding efficiency. The results of the Pareto frontier distribution verifies that the nondominated solution set obtained by the particle swarm algorithm is of high quality and feasible, which is able to guide the actual production effectively. The integrated process planning and scheduling Gantt chart results visually demonstrate the ability of the method to reasonably optimize the process machine task items.

Acknowledgements

The authors acknowledge the Shaanxi Province Key R&D Plan Project (NO.2019GY-024).

References

[1] Garrett, C. R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L. P., & Lozano-Pérez, T. (2021). Integrated task and motion planning. Annual review of control, robotics, and autonomous systems, 4(1), 265-293.

[2] Li, X., Gao, L., Li, X., & Gao, L. (2020). A hybrid genetic algorithm with variable neighborhood search for dynamic IPPS. Effective Methods for Integrated Process Planning and Scheduling, 429-453.

[3] Morganti, L., Benderbal, H. H., Benyoucef, L., Bortolini, M., & Galizia, F. G. (2020). A new process quality-based multi-objective multi-part approach for the integrated process planning and scheduling (IPPS) problem in reconfigurable manufacturing environment. IFAC-PapersOnLine, 53(2), 10755-10760.

[4] Moons, S., Ramaekers, K., Caris, A., & Arda, Y. (2017). Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion. Computers & Industrial Engineering, 104, 224-245.

[5] Zhang, L., Yu, C., & Wong, T. N. (2021). A graph-based constraint programming approach for the integrated process planning and scheduling problem. Computers & Operations Research, 131, 105282.

[6] Wu, X., & Li, J. (2021). Two layered approaches integrating harmony search with genetic algorithm for the integrated process planning and scheduling problem. Computers & Industrial Engineering, 155, 107194.

[7] Ba, L., Li, Y., Yang, M., Wu, X., Liu, Y., Gao, X., & Miao, Z. (2018). A mathematical model for multiworkshop IPPS problem in batch production. Mathematical Problems in Engineering, 2018(1), 7948693.

[8] Liu, Q., Li, X., Gao, L., & Fan, J. (2022). A multi-MILP model collaborative optimization method for integrated process planning and scheduling problem. IEEE Transactions on Engineering Management, 71, 4574-4586.

[9] Zhang, S., & Wong, T. N. (2018). Integrated process planning and scheduling: an enhanced ant colony optimization heuristic with parameter tuning. Journal of Intelligent Manufacturing, 29, 585-601.

[10] Wen, X., Qian, Y., Lian, X., Zhang, Y., Wang, H., & Li, H. (2023). Improved genetic algorithm based on multi-layer encoding approach for integrated process planning and scheduling problem. Robotics and Computer-Integrated Manufacturing, 84, 102593.

[11] Shi, G., Yang, Z., Xu, Y., & Quan, Y. (2022). Solving the integrated process planning and scheduling problem using an enhanced constraint programming-based approach. International Journal of Production Research, 60(18), 5505-5522.

[12] Zhu, X., Guo, X., Wang, W., & Wu, J. (2021). A genetic programming-based iterative approach for the integrated process planning and scheduling problem. IEEE Transactions on Automation Science and Engineering, 19(3), 2566-2580.

[13] Er, M., Arsad, N., Astuti, H. M., Kusumawardani, R. P., & Utami, R. A. (2018). Analysis of production planning in a global manufacturing company with process mining. Journal of Enterprise Information Management, 31(2), 317-337.

[14] Lee, D. K., Shin, J. H., & Lee, D. H. (2020). Operations scheduling for an advanced flexible manufacturing system with multi-fixturing pallets. Computers & Industrial Engineering, 144, 106496.

[15] Catthoor, F., Basten, T., Zompakis, N., Geilen, M., Kjeldsberg, P. G., Basten, T., ... & Voeten, J. P. (2020). Scenarios in the design of flexible manufacturing systems. System-scenario-based design principles and applications, 181-224.

[16] Sim, D. G., & Lee, D. H. (2024). A two-stage solution approach for scheduling flexible manufacturing systems with multi-fixturing pallets. Flexible Services and Manufacturing Journal, 1-25.

[17] López, A., Estévez, E., Casquero, O., & Marcos, M. (2020, June). Using industrial standards for modeling flexible manufacturing systems. In 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS) (Vol. 1, pp. 41-46). IEEE.

[18] Zhang, Z., Benyoucef, L., & Sialat, A. (2022). Sustainable integrated process planning and scheduling (IPPS) in RMS: Past, Present and Future. IFAC-PapersOnLine, 55(10), 791-797.

[19] Xinyue Li,Yu Zhang & Wang Hu. (2025). Ensemble learning training strategy based on multi-objective particle swarm optimization and chasing method. Expert Systems With Applications,281,127777-127777.

[20] Zhihao Zhu,Yu Guo & Zhi Gao. (2025). Optimization control of spacecraft proximation based on r-domination adaptive bare-bones particle swarm optimization algorithm. Expert Systems With Applications,278,127269-127269.

[21] Yunhao Li,Jinquan Zheng,Lili Kong,Long Chang,Lingyue Kong,Guiyue Kou... & Mingfei Mu. (2025). Numerical investigation of GaN MMIC PA thermal management system and multi-objective genetic algorithm optimization of heat sink parameters. Applied Thermal Engineering,272,126446-126446.