electronics

MDPI

Article

NSGA-III-Based Production Scheduling Optimization Algorithm for Pressure Sensor Calibration Workshop

Ying Zou, Zuguo Chen *, Shangyang Zhu and Yingcong Li

The School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China; 21010401001@mail.hnust.edu.cn (Y.Z.); 23020401004@mail.hnust.edu.cn (S.Z.); 23010401019@mail.hnust.edu.cn (Y.L.)

* Correspondence: zg.chen@hnust.edu.cn

Abstract: Although the NSGA-III algorithm is able to find the global optimal solution and has a good effect on the workshop scheduling optimization, the limitations in population diversity, convergence ability and local optimal solutions make it not applicable to certain situations. Thus, an improved NSGA-III workshop scheduling optimization algorithm is proposed in this work. It aims to address these limitations of the NSGA-III algorithm in processing workshop scheduling optimization. To solve the problem of individual elimination in the traditional NSGA-III algorithm, chaotic mapping is introduced in the improved NSGA-III algorithm to generate new offspring individuals and add the selected winning individuals to the offspring population as the parent population for the next iteration. The proposed algorithm was applied to a pressure sensor calibration workshop. A comparison with the traditional NSGA-III algorithm was conducted through a simulation analysis. The results show that the proposed algorithm can obtain a better convergence performance, improve the optimization ability and avoid falling into local optimal solutions.

Keywords: improved NSGA-III algorithm; chaotic mapping; scheduling optimization; pressure sensor calibration workshop

Citation: Zou, Y.; Chen, Z.; Zhu, S.; Li, Y. NSGA-III-Based Production Scheduling Optimization Algorithm for Pressure Sensor Calibration Workshop. *Electronics* **2024**, *13*, 2844. https://doi.org/10.3390/electronics13142844

Academic Editors: Jitao Li, Xu Fang, Chao Deng, Shankar A. Deka and Heling Yuan

Received: 27 June 2024
Revised: 11 July 2024
Accepted: 15 July 2024
Published: 19 July 2024

1. Introduction

With the development of automatic control technology and advanced sensor technology in manufacturing industry [1,2], effective production scheduling has become a key point for the efficient operation of the manufacturing industry in recent year. Production scheduling plays a key role in workshop management, as it may affect the execution of production plans, the utilization of equipment and the flow of material. Thus, a well-devised scheduling plan is essential for proficient workshop operations [3]. However, production scheduling is a typical optimization problem with complexity, multiple objectives [4–6] and constraints [7], which makes it difficult to obtain satisfactory performance by controlling the efficiency, cost, resource utilization and performance with intelligent scheduling strategies.

To address the above-mentioned challenges, a number of algorithms and planning models have been proposed to realize production scheduling. In [8], a decomposition-based multi-objective artificial bee colony algorithm is proposed to deal with multi-objective scheduling problems while considering stability and efficiency. Considering machine breakdowns and maintenance, particle swarm optimization was employed to deal with dynamic issues for multi-agent systems in [9]. Deep reinforcement learning was introduced to deal with scheduling problems for flexible job shops by taking completion time, energy conservation and total costs into consideration [10,11]. In [12], a rescheduling method is proposed for flexible job shops. It combines right-shift rescheduling and complete rescheduling to manage unexpected machine failures. In addition, it establishes three stability indicators to maintain consistency pre- and post-rescheduling. In [13], a backtracking search algorithm with a crossover feature is proposed to improve the boost search capability and avoid

premature convergence in dynamic scheduling environments. In [14], a multi-objective quantum-inspired seagull optimization algorithm was developed to optimize the convergence and distribution of the solution. Furthermore, considering preventive maintenance and transportation, it combines genetic algorithms and differential evolution to address flexible job shop scheduling and augment the search effectiveness of genetic algorithms [15]. Compared with the above-mentioned algorithms, the NSGA (non-dominated sorting genetic algorithm) has shown a better multi-objective optimization ability, as it maintains diversity and balances the solution set through non-dominated sorting and crowding distance measures, which makes it applicable for various optimization problems [16,17].

Therefore, extensive research has been conducted on numerous production scheduling issues utilizing the NSGA. In [18], competitive mechanisms were employed in NSGA-II. Furthermore, it utilized random mutation strategies and crossover methods tailored for processes and machines to promote the generation of new populations. In [19], NSGA-II was modified to address a low-carbon flexible job shop model, which demonstrates rapid convergence to local optimal Pareto solutions. In [20], a fuzzy membership function was introduced into NSGA-II for multi-objective optimization, which facilitates optimal trade-off solutions from the Pareto front. In taking working overtime into consideration, an improved NSGA-II is proposed for multi-objective job shop scheduling in [21]. In the proposed algorithm, an adaptive mechanism was developed to sustain global convergence stability and apply a local search process to enhance utilization. In [22], a combination of automatically constructed parallel algorithms is employed to promote the optimization performance of NSGA-II. Moreover, a detailed comparison of and NSGA-III is conducted in [23], showing that the solution accuracy and solution distribution of NSGA-III are better than those of NSGA-II. Thus, a modified NSGA-III that combines reinforcement learning is proposed in [24]. It employs dual Q-learning with an improved ε-greedy strategy to adaptively adjust the key parameters of NSGA-III. In [25], the limitation of the global search capability of NSGA-III is addressed by introducing cooperative evolutionary algorithms. In the proposed method, it utilizes the competition and cooperation among group members to enhance the convergence and search capabilities. However, this results in a heavier computation cost compared to using the traditional methods.

Motivated by the above discussion, an improved NSGA-III algorithm is proposed in this work to realize production scheduling optimization for a pressure sensor calibration workshop. In the proposed algorithm, a chaos mapping mechanism is introduced to prevent the traditional NSGA-III algorithm from settling into local optima. Technically, it combines and modifies genes in eliminated individuals and adds superior individuals to the offspring population for subsequent iterations, which enriches population diversity and mitigates the local optimum problem. The contributions of this work are summarized as follows:

1. A chaos mapping mechanism is introduced into the NSGA-III framework to enrich the parent population in subsequent generations.
2. The enhanced NSGA-III algorithm is applied to address the scheduling problem for pressure sensor calibration workshops.
3. The proposed algorithm takes multiple objectives, machine switching time reduction, total processing time minimization and excess product reduction into consideration for instance simulation.

The remainder of this paper is structured as follows. In Section 2, the model of the scheduling problem is presented. In Section 3, the proposed enhanced NSGA-III algorithm is described in detail. The effectiveness of the proposed algorithm is analyzed in Section 4. Conclusions and future work are given in Section 5.

2. Problem Description and Model Construction

2.1. Problem Description and Constraint Analysis

The problem of pressure sensor calibration production workshop scheduling can be generally described as follows: there are a types of products, which are processed through

the assembly line, and there are only b brackets in the assembly line for each round of processing. Each bracket can install six products at most, or not install any products. The b brackets correspond to b processing machines in the flow shop processing. Each processing machine needs different processing times for different types of products, and the maximum number of products processed per round is also different.

After analyzing the objective conditions and constraint conditions in the actual production scheduling process, an ideal production model is established to achieve the industry versatility of production scheduling. Workshop constraints and related assumptions are given below:

1. Different products cannot be placed on the same trestle at the same time.
2. In this work, equipment failures are not considered. If the product is processed within a given certain period of time, the process task is completed.
3. There exist some cases where a certain machine cannot process certain products; for example, machine b_1 cannot process product a_1 or a_2.
4. In considering the situation that neighboring machines are placed closely and the sizes of products to be produced not being uniform, there exists a constraint such that product a_1 cannot be processed after product $a_2(a_1, a_2 \in a)$.

2.2. Description of Workshop Specific Production Problems

In the product-making process, let $C_{\max}$ be the maximum completion time of the entire schedule (the time required from the first workpiece to the last workpiece to finish processing). And the shortest total processing time to process n products is defined as $\min\{C_{\max}\}$. Since the switching of processing machines and their trestles requires manual adjustment, it is necessary to reduce switching times. Thus, the objective function $\min\{D\}$ is established for minimizing the switching times. While satisfying the processing output, the processing of surplus products is reduced as much as possible, and an objective function for the surplus products is represented by the resource utilization rate $\max\{R\}$.

$$\min\{C_{\max}\} = \min\left(\max \sum_{t=1}^{m} c_i\right) \tag{1}$$

$$\min\{D\} = \min\left(\sum_{j=1}^{j} \sum_{k=1}^{K} SW_{jk}\right) \tag{2}$$

$$\max\{R\} = \max\left(\frac{\sum_{i=1}^{m} \sum_{k=1}^{K} T_{ik}}{\sum_{j=1}^{J} T_j}\right) \tag{3}$$

Here, c_i represents the completion time of the i-th batch, SW_{jk} is the number of product changes in the k-th process on the j-th production line, T_{ik} is the running time of the k-th process in the i-th batch, and T_j is the j-th production line at the end of the production time.

3. Design of Improved NSGA-III Algorithm

3.1. Introduction of the Traditional NSGA-III Algorithm

In 1994, the NSGA algorithm was first proposed [26]. Subsequently, to address the shortcomings of the NSGA algorithm, such as time complexity and the inability to quantify shared parameters, the NSGA-II algorithm was proposed [27]. In NSGA-II, crowding distance is used to select individuals at the same non-dominated level. Thus, to improve the NSGA-II algorithm, NSGA-III adopts an individual selection method based on reference points [28].

As shown in Figure 1, the operations of NSGA-III algorithm include population initialization, reference point generation, calculating the fitness value of each individual in the parent population, crossover and mutation to generate offspring from the population, calculating individual fitness values, combining the parent population and progeny population and rapid combination for non-dominated sorting through the choice of a reference point

operation with N better individual choices for the next generation of the population, as well as judging whether the maximum number of iterations is satisfied. If the maximum number of iterations is reached, the algorithm is terminated; if not, the quick non-dominated sorting of the population is processed to produce offspring populations. The algorithm is repeated according to the above steps.

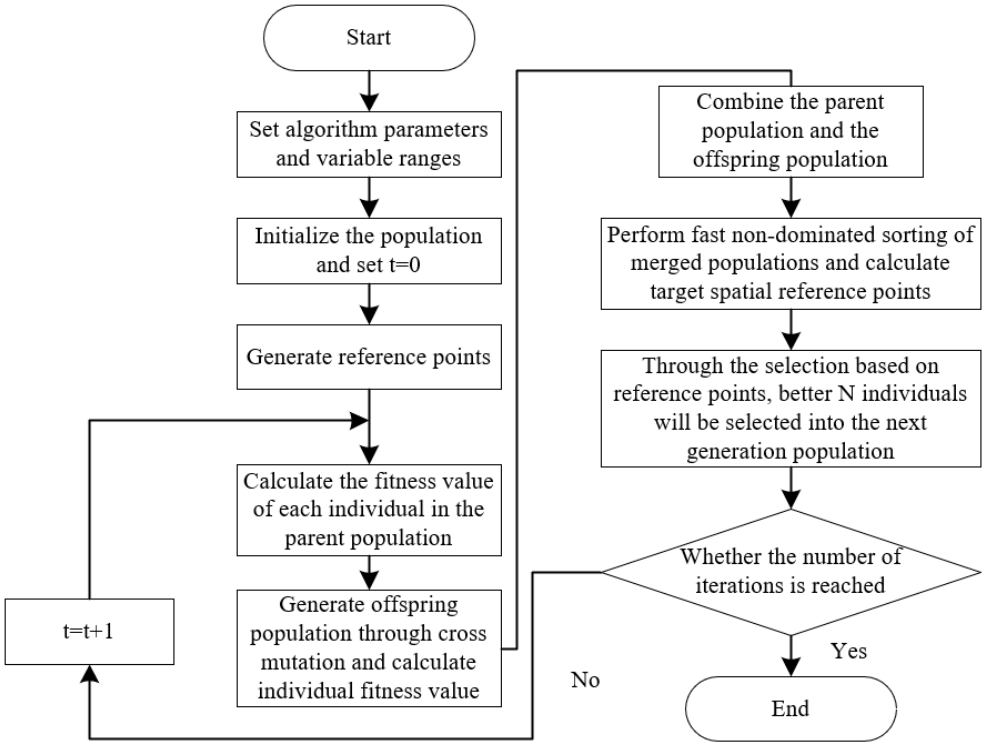


Figure 1. Flowchart of traditional NSGA-III algorithm.

3.2. The Genetic Algorithm Based on Chaotic Non-Dominated Sorting

In order to avoid falling into the local optimal solution prematurely and make the offspring population diversified, the improved NSGA-III algorithm is proposed to chaotically map the eliminated chromosome population to generate new individuals and select the best offspring individuals from the new individuals.

In chaotic mapping, logistic mapping is used to describe the changes of the population, as shown below:

$$X_{n+1} = X_n \times \mu \times (1 - X_n), \qquad \mu \in [0, 4], X \in [0, 1] \tag{4}$$

where μ is the logistic parameter, X_n is a parent chromosome, and X_{n+1} is the new chromosome individual obtained through chaotic calculation.

In the improved NSGA-III algorithm, the eliminated chromosome population is chaotically mapped to generate new individuals, and the superior offspring individuals are selected from the new individuals. The process of the improved generation of new offspring using chaotic maps for eliminated populations is shown in Figure 2.

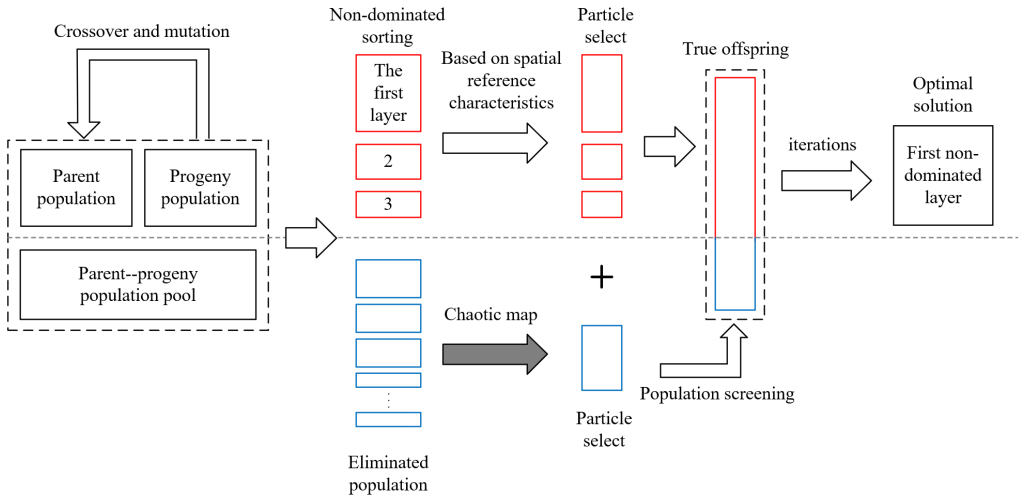


Figure 2. Improved NSGA-III algorithm principle based on chaotic mapping.

3.3. The Design Process of Improved NSGA-III Algorithm

3.3.1. Population Coding

In this work, the production information needs to be preprocessed. According to the production characteristics, the types and numbers of products arranged in the flow shop are taken as genes on the chromosomes. The chromosome model is given below:

$$F_a(T) = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1a} \\ T_{21} & T_{22} & \cdots & T_{2a} \\ \vdots & \vdots & & \vdots \\ T_{b1} & T_{b2} & \cdots & T_{ba} \end{bmatrix}_{(b \times a)} \tag{5}$$

$$G_b(N) = \begin{bmatrix} N_{11} & N_{12} & \cdots & N_{1a} \\ N_{21} & N_{22} & \cdots & N_{2a} \\ \vdots & \vdots & & \vdots \\ N_{b1} & N_{b2} & \cdots & N_{ba} \end{bmatrix}_{(b \times a)} \tag{6}$$

where T represents the time, N represents the number of products, T_{ba} represents the time taken by the b-th pressure sensor to process type a products, and N_{ba} represents the number of the b-th pressure sensor processing type a products.

3.3.2. Population Initialization

In terms of the production scheduling of the pressure sensor calibration workshop, the characteristics of diverse resource allocation and limited production capacity are considered. In this work, a random generation method is used to generate the initial population to ensure the diversity of the population. In this case, we can make sure that the generated scheduling plan will not exceed the production capacity of the machines or workers.

3.3.3. Selection Operation

As for selection operation, it performs non-dominated sorting on the new population, categorizing individuals within the population into different non-dominated levels based on their performance in the objective space. Then, it uses reference point sorting for the same non-dominated level, selects one or more representative reference points and sorts individuals based on their distance or similarity to these reference points. In this case, it not only ensures the diversity of the population but also increases the sharpness of the population.

3.3.4. Crossover Operation

Crossover operation refers to the pairwise pairing of selected parents, and the chromosomes of parents exchange genes in a certain way. According to the characteristics of actual production problems, the multi-point crossover method is introduced to set two or more crossover points in the chromosomes of parents and then perform partial gene exchange by means of interval exchange, as shown in Figure 3.

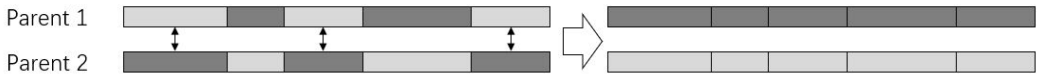


Figure 3. Multi-point crossover.

3.3.5. Mutation Operation

In this study, the random mutation method is used to realize the mutation operation as shown in Figure 4. This method replaces a parent population with the probability of mutation and randomly assigns the value of a certain or several loci on its chromosome, or it uniformly selects and replaces the genes in the range of the entire gene sequence within a certain range gene value.

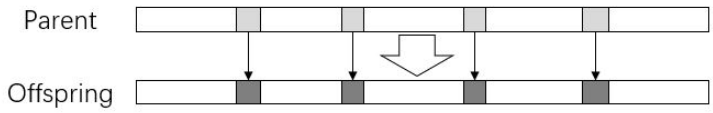


Figure 4. Schematic of uniform variation.

3.3.6. Chaotic Mapping

In this study, a chaos mapping method is utilized to generate a new population from individuals that have been eliminated. The eliminated individuals are the inputs of the chaos mapping. And new trait values are generated by the logistic mapping. These new trait values are then employed to construct new individuals. The newly created individuals are evaluated, and the superior ones are selected and incorporated into the population. An example of the logistic mapping is presented in Figure 5.

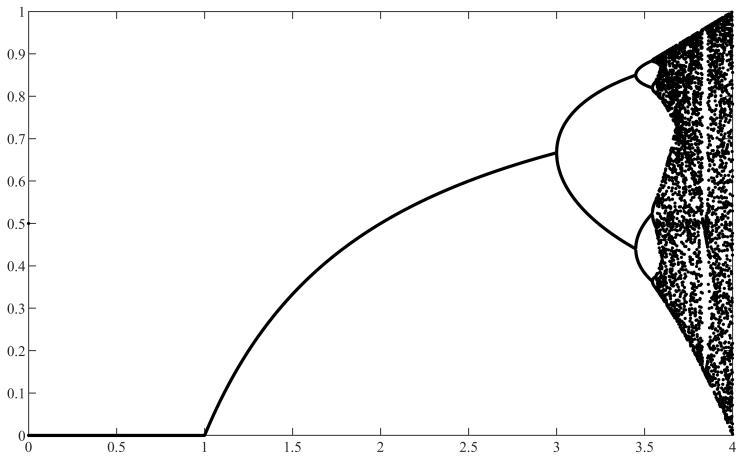


Figure 5. An example of logistic chaotic mapping.

3.4. The Steps and Flowchart of the Improved NSGA-III Algorithm

The steps of the improved NSGA-III algorithm are expressed in detail below. And the flow chart of the improved NSGA-III algorithm is shown in Figure 6.

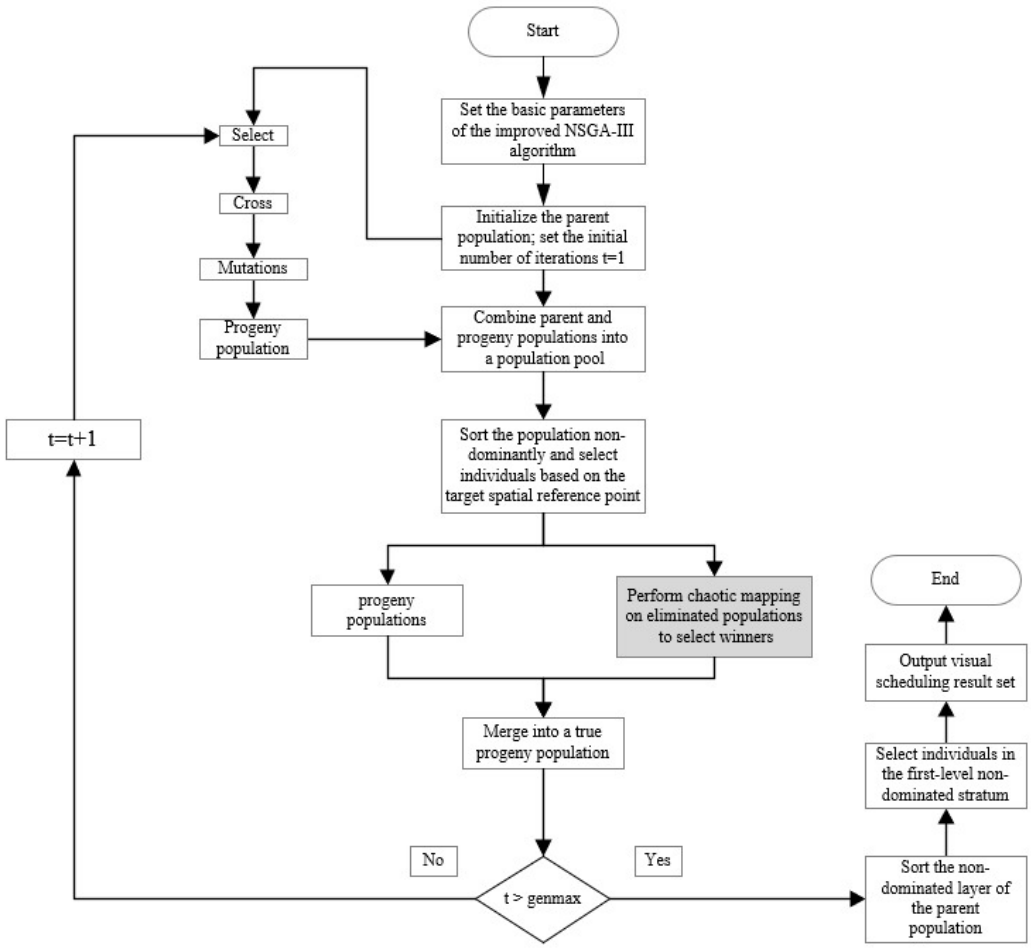


Figure 6. Flowchart of the improved NSGA-III algorithm.

Step 1: The basic parameters of the proposed algorithm include the population size N, the number of iterations *genmax* and the preprocessing product parameters. The crossover ratio is set as 0.9, and the mutation ratio is set as 0.1.

Step 2: The initial parent population of N individuals is generated, subject to the constraints, and the number of population iterations is initialized as $t = 1$.

Step 3: According to the crossover and mutation ratio, the initial progeny population of N individuals is obtained.

Step 4: Combine the initial parent population and the initial offspring population to form a new population with $2N$ individuals.

Step 5: Perform a non-dominated ranking of the new group. If two individuals do not dominate each other, they will be placed in the same non-dominated layer.

Step 6: The new population is constructed from the first non-dominated layer to the Nth layer. The creation of partial-progeny individuals stops when the number of partial-progeny individuals is equal to N or larger than N for the first time. The number of individuals of some offspring is defined as S_t. If S_t is larger than N, the last non-dominated layer is retained, and the individuals in the last non-dominated layer are eliminated based on the target spatial reference point such that the number of individuals in some offspring is equal to N.

In this step, N represents the number of populations, and the value is determined by tuning parameters. Here, the traditional NSGA-III algorithm is used to solve the problem. Through adjusting the parameters, it can be found that when N exceeds a certain threshold, the final result remains unchanged. In order to ensure the running time of the program, the smallest threshold is set as N.

Step 7: A new population is generated through the chaotic mapping of the eliminated $2N - N = N$ individuals, and N_1 is obtained by adjusting the parameters. This step aims to adjust the parameters on the basis of determining the number N of the population. The main adjustment parameter basis is the following: when the N_1 value exceeds a certain threshold, the winning individuals obtained by the screening remain unchanged). Winning progeny individuals are selected from the new population.

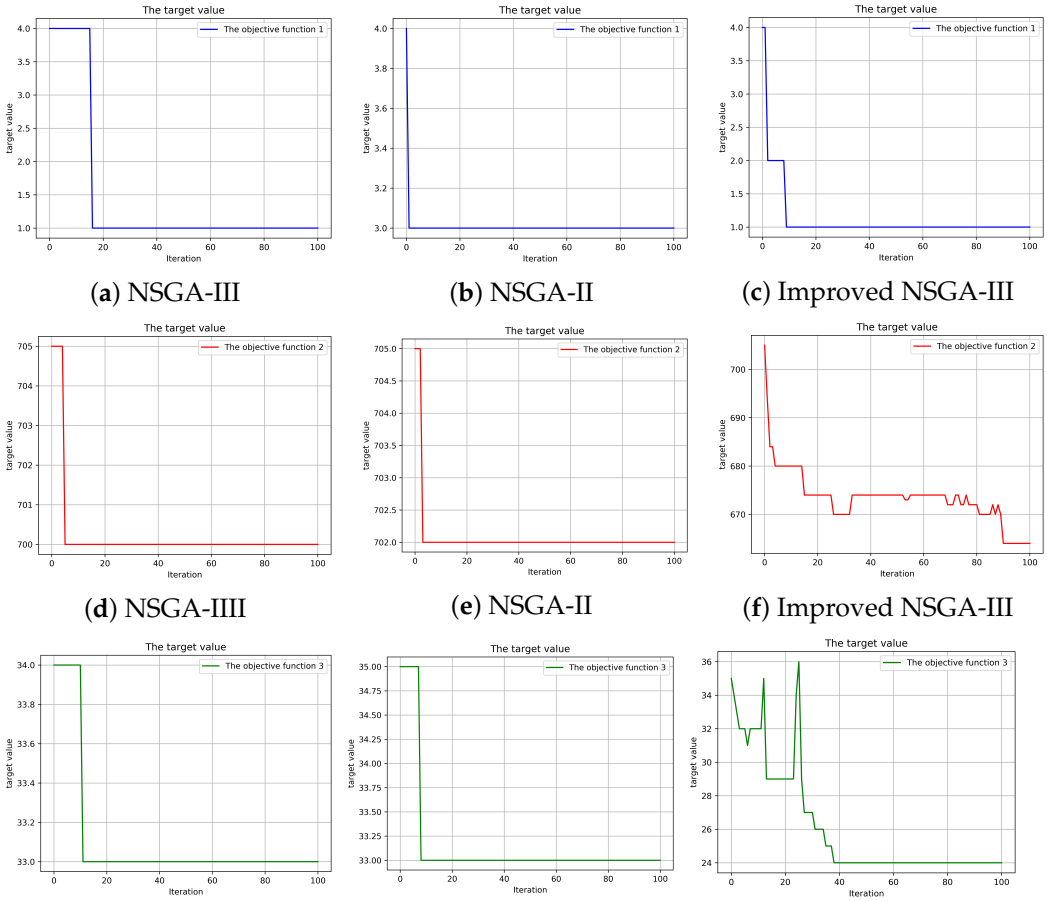
Step 8: The N partial offspring individuals and the N_1 superior offspring individuals are combined to form a true-offspring population with $N + N_1$ individuals, which is the parent population for the next iteration. The obtained true-progeny population is sorted in a non-dominated manner, and the set of individuals in the first non-dominated layer is the optimal solution set obtained in this iteration.

Step 9: The algorithm ends when the maximum number of iterations of the population is reached. If $t < genmax$, then $t = t + 1$; otherwise, it jumps out of the loop. In this way, it can obtain the global optimal solution.

4. An Example of the Improved NSGA-III Algorithm

In this study, a simulation study was conducted using Python 3.7 on a computer with a 2.20 GHz CPU and 8 G of memory. Actual workshop scheduling data of a company were used for simulation analysis. In the simulation experiment, the convergence performances of the traditional NSGA-III, the traditional NSGA-II and the improved NSGA-III algorithms were compared. The hyperparameters were defined as follows: the number of populations N, the number of winning individuals N_1, the cross ratio 0.9, the mutation ratio 0.1 and the maximum number of iterations 100. The reported value is the result after 100 iterations, and the result remained the same for each test. In Figure 7, (a), (d) and (h) represent the convergence performance of objective function 1 (given in Equation (1)), objective function 2 (given in Equation (2)) and objective function 3 (given in Equation (3)) with the traditional NSGA-III algorithm, respectively; (b), (e) and (g) represent the convergence performance of objective function 1, objective function 2 and objective function 3 with traditional NSGA-II algorithm, respectively; (c), (f) and (i) represent the convergence performance of objective function 1, objective function 2 and objective function 3 with the improved NSGA-III algorithm based on the chaotic mapping, respectively.

It can be seen from Figure 7 that with the traditional NSGA-III algorithm, objective function 1 converges to 1 at 16 iterations, objective function 2 converges to 700 at about 5 iterations, and objective function 3 converges to 33 at about 11 iterations. With the traditional NSGA-II algorithm, objective function 1 converges to 1 at about 1 iteration, objective function 2 converges to 700 at about 4 iterations, and objective function 3 converges to 33 at about 10 iterations. With the improved genetic algorithm, objective function 1 converges to 1 at about 10 iterations, objective function 2 converges to 663 at about 90 iterations, and objective function 3 converges to 24 at about 39 iterations. From the above observations, it can be concluded that with the improved NSGA-III algorithm, objective function 1 converges to 1, objective function 2 converges to 663, and objective function 3 converges to 24. This indicates that the proposed method yields the shortest processing time, the fewest transitions, and the minimum number of excess product processings. This implies that the proposed method meets the optimization requirements under different objectives and provides better optimization results than other methods.



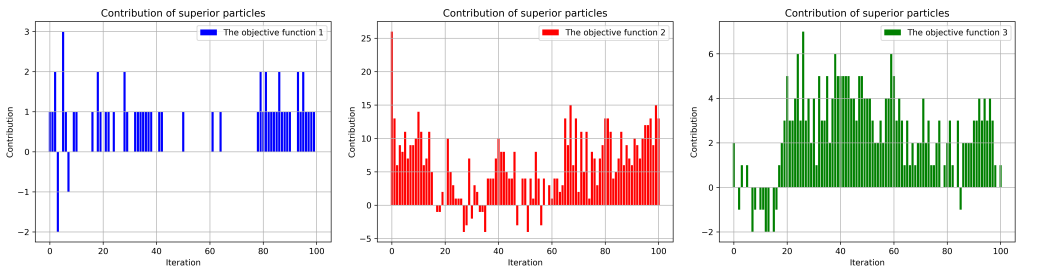
(**a**) NSGA-III (**b**) NSGA-II (**c**) Improved NSGA-III

(**d**) NSGA-IIII (**e**) NSGA-II (**f**) Improved NSGA-III

(**g**) NSGA-III (**h**) NSGA-II (**i**) Improved NSGA-III

Figure 7. Convergence performances of the algorithms.

In the simulation experiment, the chaotic mapping was used for the eliminated population to obtain a new population, and the winning population selected from the new population was merged into the offspring population and regarded as the parent population for the next iteration. This requires the consideration of whether the winning group obtained by the chaotic mapping is better than the offspring population. Thus, the difference between the optimal fitness of the winning group and the optimal fitness in the offspring population is defined as the contribution of the winning group. The result is shown in Figure 8, where (a) represents optimal objective function 1 in the winning group minus optimal objective function 1 in the offspring population; (b) represents optimal objective function 2 in the winning group minus optimal objective function 2 in the offspring population; (c) represents optimal objective function 3 in the winning group minus optimal objective function 3 in the offspring population.

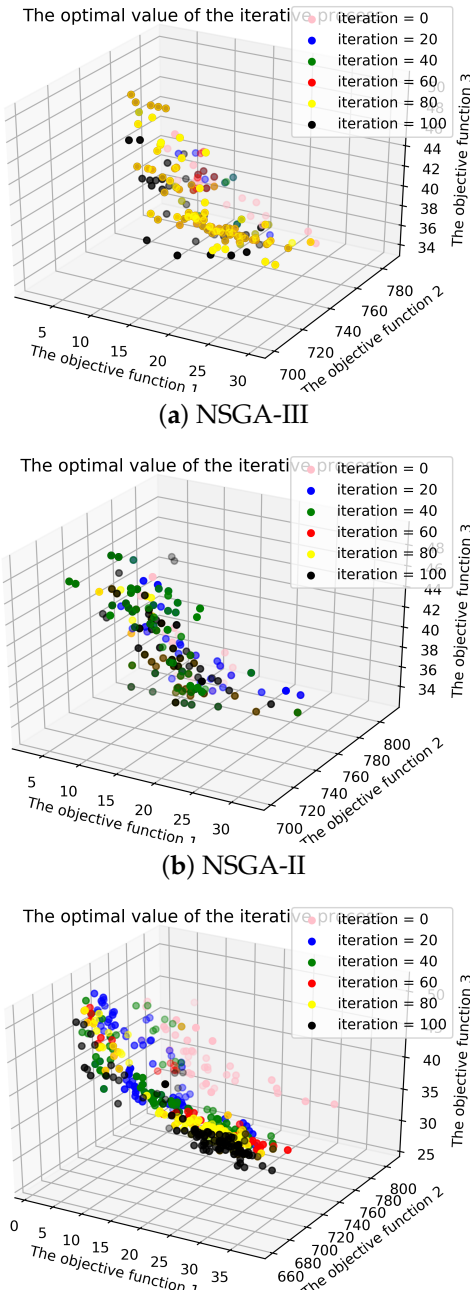


(**a**) Objective function 1 (**b**) Objective function 2 (**c**) Objective function 3

Figure 8. Contribution of the winning group.

The winning offspring individuals obtained through screening the eliminated population through chaotic mapping are illustrated in Figure 8. It can be seen that the winning population can contribute better solutions to a single objective function in most iterations. It can be concluded that the improved NSGA-III algorithm based on chaotic mapping provides more individuals to obtain the global optimal solution. In this way, it ensures the diversity of the offspring population during the convergence process and the avoidance of falling into the local optimal solution.

To obtain clear insight on the changing trend of the optimal solution in the convergence process of the algorithm, the optimal solution was recorded every 20 iterations in the process of 100 iterations. In Figure 9, the optimal solution of the traditional NSGA-III, NSGA-II and the improved NSGA-III algorithms in the convergence process are shown in (a), (b) and (c), respectively.



(**a**) NSGA-III

(**b**) NSGA-II

(**c**) Improved NSGA-III

Figure 9. The optimal solutions.

It can be seen from Figure 9 that the improved NSGA-III algorithm can obtain more optimal solutions and distribute more evenly in the target space, compared with the other two algorithms. Note that the traditional NSGA-III and NSGA-II iterative algorithms tend to fall into the local optimal solution prematurely in the iterative process. Furthermore, the solutions of the improved NSGA-III algorithm show more obvious stratification between different iterations than the other two algorithms. Meanwhile, the distribution of solutions in the target space is more uniform. And the obtained solutions are better than those of other two algorithms. Therefore, it can be concluded that the improved NSGA-III algorithm proposed in this work can achieve satisfactory performance in the field of production scheduling.

5. Conclusions

In this paper, the NSGA-III algorithm and its application in the production scheduling of a pressure sensor calibration workshop is studied. In considering the limitations of the traditional NSGA-III algorithm, the superior offspring individuals were selected through the chaotic mapping of the eliminated population in this work. In this way, the quality of the population is improved, and the diversity of the population is ensured. Meanwhile, this avoids the problem that the traditional NSGA-III algorithm may fall into the local optimal solution. The simulation results show that the improved NSGA-III algorithm can achieve better performances in convergence and the diversity of the population than the traditional algorithms. Furthermore, it can reduce production costs and improve production efficiency in the actual scheduling process.

Although the improved NSGA-III algorithm proposed in this work is a good solution to the problem of easily falling into local optimal solutions, its scheduling results still need further improvement for large-scale production scheduling problems and more target optimization problems. Thus, parallel computing technologies may be employed to accelerate the execution speed of the algorithm and enhance its capability to handle large-scale problems in our future work. Furthermore, it is also worthy to explore the application of chaos mapping in other evolutionary algorithms.

Author Contributions: Methodology, Y.Z. and Z.C.; validation, Y.Z., Z.C. and S.Z.; writing—original draft preparation, Z.C. and Y.L.; writing—review and editing, Y.Z. and S.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (62203164) and Scientific Research Fund of Hunan Provincial Education Department (Outstanding Young Project) (21B0499).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Fang, X.; Xie, L.; Li, X. Distributed localization in dynamic networks via complex laplacian. *Automatica* **2023**, *151*, 110915. [CrossRef]
2. Fang, X.; Xie, L. Distributed Formation Maneuver Control Using Complex Laplacian. *IEEE Trans. Autom. Control* **2023**, *69*, 1850–1857. [CrossRef]
3. Yazdani, M.; Amiri, M.; Zandieh, M. Flexible job-shop scheduling with parallel variable neighborhood search algorithm. *Expert Syst. Appl.* **2010**, *37*, 678–687. [CrossRef]
4. Luo, M. Study on Scheduling Optimization of Automatic Production Line in Production Shop Based on Multi-objective Optimization Algorithm. *Acad. J. Manuf. Eng.* **2018**, *16*, 147.
5. Del Gallo, M.; Mazzuto, G.; Ciarapica, F.E.; Bevilacqua, M. Artificial intelligence to solve production scheduling problems in real industrial settings: Systematic Literature Review. *Electronics* **2023**, *12*, 4732. [CrossRef]
6. Momenikorbekandi, A.; Abbod, M. Intelligent Scheduling Based on Reinforcement Learning Approaches: Applying Advanced Q-Learning and State–Action–Reward–State–Action Reinforcement Learning Models for the Optimisation of Job Shop Scheduling Problems. *Electronics* **2023**, *12*, 4752. [CrossRef]

7. Xixing, L.; Baigang, D.; Shunsheng, G.; Renhe, X. Production scheduling optimization method for textile machinery manufacturing enterprise based on improved bee algorithm. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 2805–2811.
8. Sassi, J.; Alaya, I.; Borne, P.; Tagina, M. A decomposition-based artificial bee colony algorithm for the multi-objective flexible jobshop scheduling problem. *Eng. Optim.* **2022**, *54*, 524–538. [CrossRef]
9. Nouiri, M.; Bekrar, A.; Jemai, A.; Niar, S.; Ammari, A.C. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *J. Intell. Manuf.* **2018**, *29*, 603–615. [CrossRef]
10. Yuan, E.; Wang, L.; Cheng, S.; Song, S.; Fan, W.; Li, Y. Solving flexible job shop scheduling problems via deep reinforcement learning. *Expert Syst. Appl.* **2024**, *245*, 123019. [CrossRef]
11. Gong, G.; Chiong, R.; Deng, Q.; Han, W.; Zhang, L.; Lin, W.; Li, K. Energy-efficient flexible flow shop scheduling with worker flexibility. *Expert Syst. Appl.* **2020**, *141*, 112902. [CrossRef]
12. Fan, C.; Wang, W.; Tian, J. Flexible job shop scheduling with stochastic machine breakdowns by an improved tuna swarm optimization algorithm. *J. Manuf. Syst.* **2024**, *74*, 180–197. [CrossRef]
13. Caldeira, R.H.; Gnanavelbabu, A.; Vaidyanathan, T. An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption. *Comput. Ind. Eng.* **2020**, *149*, 106863.
14. Wang, Y.; Wang, W.; Ahmad, I.; Tag-Eldin, E. Multi-objective quantum-inspired seagull optimization algorithm. *Electronics* **2022**, *11*, 1834. [CrossRef]
15. Wang, H.; Sheng, B.; Lu, Q.; Yin, X.; Zhao, F.; Lu, X.; Luo, R.; Fu, G. A novel multi-objective optimization algorithm for the integrated scheduling of flexible job shops considering preventive maintenance activities and transportation processes. *Soft Comput.* **2021**, *25*, 2863–2889. [CrossRef]
16. Liu, C.; Li, H.; Shi, Y. A unitary distributed subgradient method for multi-agent optimization with different coupling sources. *Automatica* **2020**, *114*, 108834. [CrossRef]
17. Liu, C.; Li, H.; Shi, Y. Resource-aware exact decentralized optimization using event-triggered broadcasting. *IEEE Trans. Autom. Control* **2020**, *66*, 2961–2974.
18. Yuan, M.; Li, Y.; Zhang, L.; Pei, F. Research on intelligent workshop resource scheduling method based on improved NSGA-II algorithm. *Robot. Comput.-Integr. Manuf.* **2021**, *71*, 102141. [CrossRef]
19. Seng, D.; Li, J.; Fang, X.; Zhang, X.; Chen, J. Low-carbon Flexible Job-shop Scheduling Based on Improved Nondominated Sorting Genetic Algorithm-II. *Int. J. Simul. Model. (IJSIMM)* **2018**, *17*, 712–723. [CrossRef]
20. Dissanayake, A.M.; Ekneligoda, N.C. Multiobjective optimization of droop-controlled distributed generators in DC microgrids. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2423–2435.
21. Shi, S.; Xiong, H. Solving the multi-objective job shop scheduling problems with overtime consideration by an enhanced NSGA-II. *Comput. Ind. Eng.* **2024**, *190*, 110001. [CrossRef]
22. Ma, X.; Liu, S.; Hong, W. Enhancing Multi-Objective Optimization with Automatic Construction of Parallel Algorithm Portfolios. *Electronics* **2023**, *12*, 4639. [CrossRef]
23. Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 3045–3052.
24. Tang, H.; Xiao, Y.; Zhang, W.; Lei, D.; Wang, J.; Xu, T. A DQL-NSGA-III algorithm for solving the flexible job shop dynamic scheduling problem. *Expert Syst. Appl.* **2024**, *237*, 121723. [CrossRef]
25. Tan, B.; Chen, H. Stochastic multi-objective optimized dispatch of combined cooling, heating, and power microgrids based on hybrid evolutionary optimization algorithm. *IEEE Access* **2019**, *7*, 176218–176232. [CrossRef]
26. Srinivas, N.; Deb, K. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **1994**, *2*, 221–248. [CrossRef]
27. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
28. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]