Research on Mixed Flow Workshop Scheduling Problem Based on Improved Subtractive Average Optimizer Algorithm

Changxing Li, Shuaihang Wang, Pengliang Zhao

***Abstract*—The Hybrid Flow-Shop Scheduling Problem (HFSP) is a pivotal challenge in the intelligent transformation of manuf acturing, where efficient solutions are critical for enhancing pro duction efficiency and resource utilization. However, traditional swarm intelligence algorithms for solving HFSP often exhibit weaknesses in global search capabilities, susceptibility to local o ptima, and insufficient convergence efficiency to meet practical scheduling demands. Furthermore, complex constraints in real industrial scenarios—such as dynamic variations in manual ope ration times and split orders—have not been adequately model ed, resulting in a significant gap between theoretical research a nd engineering applications. To address these limitations, this st udy conducts research from both theoretical optimization and e ngineering adaptation perspectives.For the classic HFSP with t he objective of minimizing makespan, an Improved Subtraction -Average-Based Optimizer (ISABO) is proposed. This algorith m integrates a simulated annealing perturbation mechanism an d a subgroup learning strategy. The simulated annealing pertur bation enhances global search capabilities by probabilistically a ccepting inferior solutions during iterations, thereby avoiding l ocal optima. Concurrently, the subgroup learning strategy guid es partial populations toward optimal solutions, improving solu tion quality. The research follows a progression from "basic pr oblem optimization" to "real-world scenario adaptation." Expe riments are conducted using production data from a paint rolle r manufacturing plant, with three widely-used HFSP algorithm s—Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), and Grey Wolf Optimizer (GWO)—selected as benchm arks. For the classic HFSP, ISABO achieves shorter makespan than all baseline algorithms in 100% of test instances and demo nstrates the highest stability (measured by standard deviation) i n 61.11% of cases .**

***Index Terms*—Hybrid flow shop scheduling, subtraction-aver age-based optimizer, single-objective optimization, local search.**

I. INTRODUCTION

Under the drive of the 4th Industrial Revolution, global manufacturing is undergoing a paradigm shift from rigid production to intelligent manufacturing. The paradigm of intelligent manufacturing is characterized by reconstructing the self-organizing ability of production systems through dynamic optimization algorithms, with its core being breaking through the linear constraint framework of traditional scheduling models. Industry research data shows that in the product life cycle of discrete manufacturing fields, non-value-added time (including waiting time between processes and material transportation, etc.) accounts for over 85% of the total production cycle, while t he effective working time of direct machining accounts for less than 15% of the total production cycle. This phenomenon reveals the inherent defect of traditional flow shop scheduling models in process coordination. The scheduling logic based on the "single process-single machine" rigid assumption leads to dynamic mismatching of equipment capacity and process requirements, thus causing systematic efficiency loss. The emergence of the Hybrid Flow-Shop Scheduling Problem (HFSP) provides a new theoretical paradigm for breaking this manufacturing efficiency bottleneck. By constructing a three-dimensional decision space of "processes-equipment-time sequence", it enables production systems to dynamically adjust process paths, offers quantifiable technological approaches for improving equipment effectiveness, and is one of the core aspects of modern manufacturing production system optimization. Its research and practice have irreplaceable strategic significance in promoting enterprises' adaptation to the transformation of intelligent manufacturing.

The HFSP is the most common workshop form and can be applied to many industries, such as textiles, electronics, chemicals, and steel, making it a research hotspot. Research on the HFSP can not only promote the development of related theories but also provide solutions for intelligent industrial production and improve industrial production efficiency. However, as the scale of the HFSP becomes increasingly large, traditional exact algorithms often cannot provide effective solutions within a limited time. Especially when various production constraints are considered, the problem becomes even more difficult to solve. Therefore, researchers are more inclined to use heuristic rules and meta-heuristic algorithms to find feasible solutions. These algorithms can avoid ineffective exploration through intelligent search and focus on areas most likely to yield optimal solutions, thereby obtaining high-quality solutions within a reasonable time.

Among many meta-heuristic algorithms, the Subtraction Average Optimizer (SAO) has gained attention due to its simple structure and high efficiency, with few parameters to set and the ability to handle complex optimization problems. By balancing exploration and exploitation in the solution space, SAO can quickly converge near the global optimal solution. In this study, we explore the application of SAO in solving the HFSP to address scheduling challenges in complex production environments and provide an efficient and feasible solution.

The Improved Subtraction-Average Based Optimizer (ISABO) is proposed based on a hybrid strategy, establishing a mathematical model for the Hybrid Flow Shop Problem (HFSP) with makespan as the objective function. The main steps are as follows：

- In the early stages of the algorithm, simulated annealing perturbation is used to achieve a wide-range search. In the later stages, it enables more refined search.
- A group learning strategy is employed, allowing part of

the population to move towards the optimal solution direction. This enhances the quality of the solution and effectively prevents the algorithm from falling into local optimum.

II. RELATED WORKS

In the realm of the constrained hybrid flow shop scheduling problem, Wei et al. [1] established a mathematical model with makespan and total tardiness as the objective functions, considering the constraint of operator setup requirements, and proposed an improved Salp Swarm Algorithm to address the hybrid flow shop scheduling problem with operator setup constraints. Tang et al.built a mathematical model aiming to minimize makespan and production resource utilization, taking into account a single batch coupling order and resource constraints. Xing et al. [2] designed a mathematical model targeting makespan under the constraint of limited transportation resources and put forward a population optimization algorithm based on a pyramid hierarchical strategy and local search. Ziadlou et al. [3] presented a two - objective mixed - integer linear programming model considering different order configurations for two - tier factories, with the objective of minimizing the maximum completion time of orders and transportation costs between the two - tier factories, and employed the ε - constraint method to handle this model. Amirteimoori et al. [4] proposed a two - stage parallel heuristic algorithm and a mixed - integer linear programming model, considering collision - free path planning constraints for transporters to solve the synchronous scheduling problem of orders and transporters in hybrid flow shop systems. Wu et al. [5] introduced a mixed - integer programming model for the dual - constrained hybrid flow shop scheduling problem, which simultaneously considered sequence - dependent setup time and transportation time constraints. Shen et al. [6] built a permutation flow shop scheduling model with processing interruptions, considering that machines might be disturbed by various events during processing, leading to deviations from the planned results, with makespan as the objective function. Han et al. [7] proposed a human - constrained hybrid flow shop scheduling model with the objective of maximizing makespan, considering the significant impact of human factors on production efficiency.

The Subtraction-Average-Based Optimizer (SABO), introduced by Trojovský and Dehghani [8], is a novel evolutionary algorithm designed to solve scheduling optimization problems. It updates the positions of population members in the search space using the subtraction average of searcher agents. Comparative experiments using standard benchmark functions and the CEC 2017 test suite demonstrated that SABO effectively balances exploration and exploitation, outperforming twelve other meta - heuristic algorithms on most benchmark functions.

Due to its strong global search and fast convergence, SABO is widely used in various optimization problems, such as energy conservation and emission reduction scheduling, fault diagnosis, and prediction. In the field of energy conservation and emission reduction scheduling, Moustafa et al. [9] proposed a SABO algorithm based on group learning strategies to solve engineering optimization problems. It incorporates cooperative learning with leader solutions to reduce power network losses through Thyristor Controlled Series Compensation (TCSC) in power systems. Wang et al. [10] introduced an opposition - based SABO algorithm that enhances performance by introducing multiple solutions to avoid local optima. It was applied to hybrid renewable energy systems composed of wind, solar photovoltaic, and fuel cells to provide reliable and efficient energy supply while reducing costs. Smaili et al. [11] designed a cooperative learning strategy to improve SABO's search ability and minimize energy losses during power supply. Xiao et al. [12] proposed a log - tent - pyramid SABO algorithm combining composite chaotic mapping and the golden sine algorithm. This algorithm expands the search space and avoids local optima by using composite chaotic mapping for population initialization. Benzaoui et al. [13] optimized the parameters of controllers and back - stepping controllers in a dual - stator induction generator wind power system using the SABO algorithm, improving energy consumption efficiency.

In fault diagnosis, Song et al. [14] proposed a multi - channel network based on the SABO algorithm for fault diagnosis of high - voltage circuit breakers. Yang et al. [15] presented a new gear fault diagnosis method that uses SABO to quickly and efficiently optimize the parameters of Variational Mode Decomposition (VMD) for signal preprocessing, addressing the limitations of single - channel vibration signals and diagnostic networks. Guo et al. [16] proposed a parameter adaptive re - constraint method based on the SABO algorithm to solve VMD - based fault - bearing vibration signal problems. Lu Fan [17] introduced a fault diagnosis method that combines VMD and wavelet threshold denoising, optimized by the SABO algorithm, to address the difficulty of extracting and identifying fault types in rolling bearings during operation.

III. MATH

This chapter uses the HFSP as the research object, constructs a problem model with the objective function of minimizing the maximum makespan, and proposes an Improved Subtraction-Average-Based Optimizer (ISABO) based on hybrid strategies to solve the problem.

This chapter tackles the hybrid flow-shop scheduling problem (HFSP) with a linear encoding method. Based on real-number representation, this approach partitions the solution vector into regions, each corresponding to different scheduling elements such as operation sequences and machine assignments. The encoding structure is formed by linearly dividing the entire encoding vector **E** into multiple intervals. Specifically, the interval [1, n×p] denotes the processing sequence of orders, where **n** is the number of orders and **p** is the number of operations per order. The interval [n×p+1, 2×n×p+1] indicates the machine IDs for each operation. This linear encoding method provides a strong representation capability, comprehensively covering all key elements of the HFSP. It allows for intuitive representation and manipulation of scheduling schemes while maintaining high computational efficiency. As a result, it offers a convenient approach for subsequent optimization algorithms, enhancing the optimization process's effectiveness and reducing computational complexity during problem-solving.

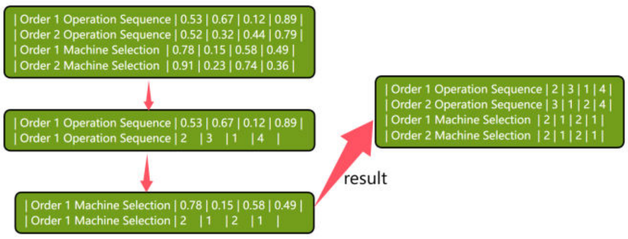
This chapter employs a random initialization strategy for the initialization method. Random initialization, a prevalent technique for generating initial solutions in optimization algorithms, involves randomly selecting several solutions within the feasible solution space defined by the problem to form an initial population. Specifically, each value in the population is randomly generated within a given range, typically [0, 1] or another range set according to specific problem requirements. This method offers the advantage of providing diverse initial solutions. It prevents the algorithm from becoming trapped in local optimal solutions at the initial stage and provides a broad search space for the algorithm. The fundamental concept of random initialization is to ensure population diversity by randomly selecting initial solutions, thereby establishing a foundation for the subsequent evolutionary process. Unlike initialization strategies based on heuristic methods, random initialization does not depend on specific domain knowledge, giving it strong generality and adaptability. It is particularly suitable for large-scale problems or problems without explicit heuristic rules. Especially in complex problems, randomly generated initial populations can enhance the algorithm's global search ability.

The encoding process in this chapter consists of two main parts: order processing sequence and machine selection. The order processing sequence uses a fully real - number encoding scheme. During decoding, the real numbers are sorted, and the resulting ranks determine the processing sequence of the operations. This encoding method enhances flexibility and effectively represents the sequence of operations, providing greater freedom for subsequent scheduling optimization.

For machine selection, a partitioning method is employed. Suppose an operation has m machines assigned to it, each with a partition value of 1/m. If the encoding value lies between 1/m and 2/m, the second machine is selected, and so on. During iteration, machines are scheduled on a first - come, first - served basis. The system dynamically maintains a machine idle - time table. By sorting the idle times, the system prioritizes machines with the shortest idle time that can meet the operation's processing requirements.

This encoding process boosts machine utilization and reduces operation waiting time, thus improving overall production efficiency. By real - time maintenance of the machine idle - time table, the system accurately tracks machine availability, enabling precise scheduling decisions and ensuring smooth and efficient operation processing. A decoding example is shown in Figure 1. Suppose there are four orders, each with two operations, and two machines for processing each operation.

Fig. 1. Decoding Example.



The Subtraction-Average-Based Optimizer (SABO) is improved in two ways. First, simulated annealing perturbation enables the algorithm to conduct a broad search in the early stages and a fine - tuned search in the later stages. Second, a group learning strategy guides part of the population toward the optimal solution, enhancing solution quality and helping the algorithm avoid local optima. These improvements allow the algorithm to better balance exploration and exploitation, thereby improving its overall performance in solving optimization problems.

In the simulated annealing algorithm, temperature variation adheres to a cooling strategy (usually a gradual decrease). The probability of accepting a new solution in each iteration depends on the current solution's quality and the current temperature. The formula for this probability is as follows:

$$P(E, E_{new}, T) = \begin{cases} 1, & if\ E_{new} < E \\ e^{-\frac{(E_{new}-E)}{T}}, & if\ E_{new} \geq E \end{cases}$$

Here, E denotes the energy (or fitness) of the current solution, E_{new} is the energy (or fitness) of the new solution, and T represents the current temperature.

Temperature updates follow a cooling strategy with exponential decay. Tk is the temperature at iteration k, α is the cooling coefficient, and T_{k+1} is the temperature at iteration k+1. The formula is:

$$T_{k+1} = a \cdot T_k$$

Figure 2 illustrates the simulated annealing algorithm, where the temperature starts from an initial value and gradually decreases with iterations. This temperature reduction is central to the algorithm, helping it to gradually converge during the search process and enhance the quality of the solution.

Fig. 2. Temperature Cooling Schedule for the Simulated Annealing Algorithm.

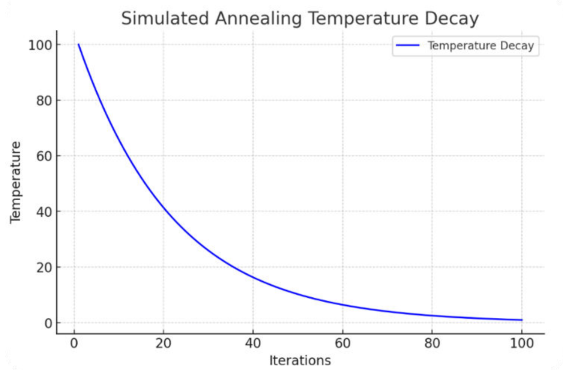
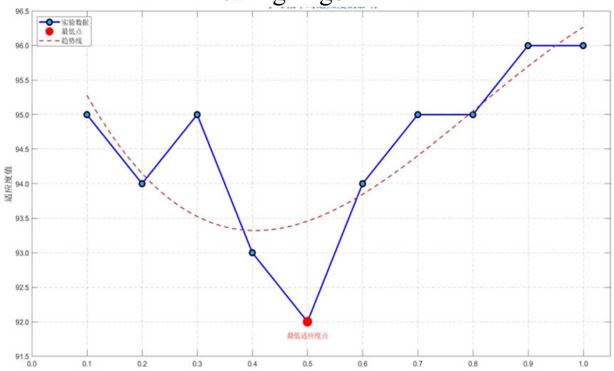


Figure 3 shows how learning probability affects fitness. Via parameter sensitivity analysis, it explores the impact of learning probability on algorithmic performance. The x-axis is learning probability (0.1 to 1.0 in steps of 0.1), and the y-axis is the minimum fitness from 30 independent experiments. Results indicate that a 0.5 learning probability yields the lowest fitness of 92, showing moderate learning intensity helps balance exploration and exploitation. When learning probability rises to 0.8 - 1.0, fitness stabilizes at 95 - 96, indicating enhanced convergence but a higher risk of optimal solutions.

Fig.3.Temperature Decay Chart of the Simulated

Annealing Algorithm



IV. The Subtraction-Average-Based Optimizer Algorithm Based on Hybrid Strategies

In the process of generating difference vectors, the Subtraction-Average-Based Optimizer (SABO) algorithm suffers from insufficient stability due to global averaging and random scaling. To address this issue, this chapter proposes an improved method that integrates a group learning mechanism and a simulated annealing perturbation strategy. After the “-v” subtraction operation, SABO calculates the arithmetic mean of all generated difference vectors to obtain a comprehensive average difference vector. Subsequently, the average difference vector is subjected to random scaling, which introduces significant randomness and leads to unstable solution quality. At this point, a leadership learning strategy is introduced to replace the random step movement, guiding some solutions towards the optimal direction. Furthermore, during the calculation of the difference matrix DX, simulated annealing perturbation is incorporated. This allows the algorithm to conduct extensive searches in the early stages and precise searches in the later stages, thereby enhancing its global search capability in the early phase and enabling refined searches in the later phase.

V. EXPERIMENTATION

A. Datasets

The test data used in the experiment referred to the production data of an actual paintbrush factory. The processing time for each stage of the order is a random integer within [10,50]. The number of production stages for each order is selected as 3, 4, and 5 respectively. The number of machines for each stage is selected as 2, 3, and 4 respectively. The number of orders (n) is selected as 5 and 10. A total of 18 sets of data (n × s × m) were simulated, as detailed in Table 3-4. Three common algorithms for solving the hybrid flow shop scheduling problem were selected for comparison. The population size of all algorithms was set to 50, and the number of iterations was set to a maximum of 100.

Table I

Statistics of the datasets

Problem size	orders	operations	machines
1	5	3	2
2	5	4	2
3	5	5	2
4	5	3	3
5	5	4	3
6	5	5	3
7	5	3	4
8	5	4	4
9	5	5	4
10	10	3	2
11	10	4	2
12	10	5	2
13	10	3	3
14	10	4	3
15	10	5	3
16	10	3	4
17	10	4	4
18	10	5	4

B. Experimental Results

This chapter addresses the hybrid flow shop scheduling optimization problem and verifies the effectiveness of the proposed Subtraction Average Optimizer based on hybrid strategies through comparative experiments. The Jaya algorithm, Whale Optimization Algorithm (WOA), and Grey Wolf Optimizer (GWO) were selected as benchmark algorithms for comparison, with the objective of minimizing the makespan. To ensure the reliability of the experimental results, each standard test instance was independently run 10 times. A dual-dimensional evaluation system was employed to comprehensively assess the performance of the algorithms: the Minimum Value (MIN) indicator to reflect the optimization capability of the algorithm, and the Special Standard Deviation (SSD) indicator to evaluate the stability of the algorithm. The MIN indicator takes the minimum value obtained from the 10 runs as the basis for evaluation, with a smaller value indicating a higher degree of proximity to the optimal solution. The SSD indicator is calculated using the formula , where N=10 is the number of experimental runs, C_i is the makespan of the i-th experiment, and Cmin is the minimum value from the 10 experiments. A smaller SSD value indicates a higher concentration of solution quality, meaning that the algorithm has a stronger ability to maintain a stable optimal solution across multiple runs. The dual-indicator evaluation system not only reflects the exploration capability of the algorithm in the solution space but also quantifies its robustness in escaping local optima. The MIN indicator compares the minimum values obtained from multiple runs of different algorithms, intuitively showing which algorithm can approach the global optimal solution more effectively. The SSD indicator assesses the stability and consistency of the algorithm's results by calculating the square root of the mean of the squared relative deviations between each experimental result and the minimum value. The experimental results demonstrate that the proposed algorithm performs remarkably well in both the MIN and SSD indicators, indicating that it not only effectively approaches the global optimal solution but also maintains high stability and robustness across multiple runs, showing significant advantages in solving the hybrid flow shop scheduling optimization problem.

$$SSD=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(C_{(i)}-C_{MIN})^2}$$

Table Ⅱ

Result

size	Indicator	ISABO	JAYA	WOA	GWO
1	MIN	**92**	95	92	95
	MEAN	**92. 2**	96. 6	93	96. 4
	SSD	**0. 47**	2. 45	1. 56	2. 26
2	MIN	**139**	142	146	145
	MEAN	**141**	144. 3	149. 2	147. 1
	SSD	**2. 49**	2. 85	3. 92	2. 52
3	MIN	**165**	166	177	176
	MEAN	**166**	180. 3	180. 3	179. 4
	SSD	**2. 00**	5. 74	4. 48	4. 59
4	MIN	**69**	69	74	79
	MEAN	**69**	73. 1	75. 7	81. 5
	SSD	**0**	4. 77	2. 43	2. 92
5	MIN	**79**	90	81	99
	MEAN	**79**	95	89. 6	102. 8
	SSD	**0**	6. 24	10. 91	4. 69
6	MIN	**114**	129	126	136
	MEAN	**117**	132. 9	132. 3	137. 8
	SSD	5. 83	5. 57	7. 92	2. 49
7	MIN	**56**	59	57	60
	MEAN	**56**	59. 6	58. 7	61. 4
	SSD	**0**	1. 41	2. 56	1. 94
8	MIN	**78**	87	84	88
	MEAN	**78. 4**	91. 8	90	90. 9
	SSD	**0. 67**	5. 77	6. 96	4. 10
9	MIN	**101**	121	115	120
	MEAN	**105**	125. 3	122	123. 4
	SSD	4. 67	5. 11	8. 04	3. 94
10	MIN	**136**	140	149	162
	MEAN	**139. 4**	148. 3	154. 8	175
	SSD	**3. 94**	9. 36	6. 68	14. 88
11	MIN	**198**	208	230	266
	MEAN	**201. 5**	215	239	270. 2
	SSD	**4. 51**	8. 04	10. 77	4. 94
12	MIN	**220**	241	269	311
	MEAN	**232. 5**	248. 6	279. 7	324. 8
	SSD	14. 29	8. 73	12. 32	15. 92
13	MIN	**115**	135	137	150
	MEAN	**123**	144. 4	142. 6	156. 3
	SSD	9. 1	10. 84	6. 8	7. 51
14	MIN	**139**	161	174	187
	MEAN	**148. 2**	182. 6	179. 2	194. 1
	SSD	10. 28	24. 54	7. 24	8. 24
15	MIN	**178**	208	208	237
	MEAN	**183. 7**	224. 2	225. 7	246. 9
	SSD	7. 23	18. 21	20. 11	11. 46
16	MIN	**93**	112	107	113
	MEAN	**95. 7**	117. 8	115. 3	118. 7
	SSD	**3. 38**	6. 94	9. 59	6. 92
17	MIN	**119**	151	158	157
	MEAN	**124**	163. 1	165. 1	166. 1
	SSD	**6. 07**	13. 83	9. 34	10. 62
18	MIN	**140**	184	188	189
	MEAN	**151. 2**	200. 5	200. 2	195. 7
	SSD	14. 18	19. 24	15. 11	8. 35

C. Analysis of Experimental Results

In the 3-operation scenario (Problem Sizes 1, 4, and 7), ISABO achieved the global optimal MIN and maintained the lowest SSD value among the test cases, demonstrating significantly superior robustness compared to the benchmark algorithms. The Grey Wolf Optimizer (GWO) exhibited marked limitations, with its MIN value consistently being the highest within the group.

In the 4-operation scenario (Problem sizes 2, 5, and 8), ISABO continued to hold the record for the optimal MIN value and maintained the lowest SSD level among the test instances. The Grey Wolf Optimizer obtained the maximum MIN value in 66.67% of the test cases, while the Jaya algorithm achieved the maximum MIN value in 33.33% of the cases.

In the 5-operation scenario (Problem sizes 3, 6, and 9), ISABO consistently maintained the smallest MIN value. The Grey Wolf Optimizer still had a 66.67% probability of obtaining the maximum MIN value in the test instances, indicating the weakest optimization capability.

When the problem size was extended to 10 orders, in the 3-operation scenario (Problem sizes 10, 13, and 16), ISABO achieved the optimal MIN value in the test instances and obtained the optimal SSD value in 66.67% of the test cases, with a significant improvement in solution accuracy compared to the benchmark algorithms. The MIN value of the Grey Wolf Optimizer remained the highest.

In the 4-operation scenario (Problem sizes 11, 14, and 17), ISABO maintained the optimal MIN value throughout and continued to expand its solution accuracy advantage. The Grey Wolf Optimizer obtained the maximum MIN value in 66.67% of the test cases.

In the 5-operation scenario (Problem sizes 12, 15, and 18), ISABO continued to hold the optimal MIN value record, while the Grey Wolf Optimizer maintained the maximum MIN value.

VI. CONCLUSION

First, the mathematical model of the problem was established based on the hybrid flow shop scheduling model introduced in Chapter 2, and an encoding method for solving the HFSP problem was designed. This method employs a real-number encoding scheme, where different encoding intervals are set to represent the operation scheduling sequence and the operation machine ID, thereby comprehensively covering all key elements of the scheduling problem. Second, the ISABO method introduces a simulated annealing perturbation strategy to simulate the random jumps of particles in the search space and allows the acceptance of solutions that are worse than the current solution, effectively avoiding the trap of local optima. Meanwhile, a leader group learning strategy is adopted to guide individuals to converge towards the current optimal solution, thereby accelerating the search process and ultimately achieving the global optimal solution. Finally, multiple comparative experiments have shown that the improved algorithm has a MIN value that is 100% smaller than the benchmark algorithms in the test cases, and the SSD indicator remains the lowest in 66.67% of the cases, demonstrating significantly superior performance in solution accuracy, convergence speed, and robustness compared to the GWO, WOA, and Jaya algorithms.

REFERENCES

[1] Yiping Huang, Libao Deng, Jianlei Wang, Weiwei Qiu, Jinfeng Liu, Modeling and solution for hybrid flow-shop scheduling problem by two-stage stochastic programming, Expert Systems with Applications, Volume 233, 2023, 120846, ISSN 0957-4174.

[2] Xing S, Shao Z, Shao W, Chen J, Pi D. Joint scheduling of hybrid flow-shop with limited automatic guided vehicles: A hierarchical learning-based swarm optimizer[J]. Computers & Industrial Engineering, 2024, 198: 110686. DOI: 10.1016/j.cie.2024.110686.

[3] Ziadlou G, Emami S, Asadi-Gangraj E. Network configuration distributed production scheduling problem: A constraint programming approach[J]. Computers & Industrial Engineering, 2024, 188:109916.

[4] Amirteimoori A, Mohamed M, Kia R. Simultaneous scheduling of jobs and transporters in a hybrid flow shop with collision-free transporter routing: A novel parallel heuristic[J]. *Computers & Operations Research*, 168, 2024:106687. DOI: 10.1016/j.cor.2024.106687.

[5] Wu, S., & Liu, L. (2023). Green Hybrid Flow Shop Scheduling Problem Considering Sequence Dependent Setup Times and Transportation Times. IEEE Access, 11, 39726-39737.

[6] Shen, J., Shi, Y., Shi, J., Dai, Y., & Li, W. (2023). An uncertain permutation flow shop predictive scheduling problem with processing interruption. Physica A: Statistical Mechanics and its Applications, 611, 128457.

[7] Han, W., Deng, Q., Gong, G., Zhang, L., & Luo, Q. (2021). Multi-objective evolutionary algorithms with heuristic decoding for hybrid flow shop scheduling problem with worker constraint. Expert Systems with Applications, 168, 114282.

[8] Trojovský P, Dehghani M. Subtraction-Average-Based Optimizer: A New Swarm-Inspired Metaheuristic Algorithm for Solving Optimization Problems[J]. *Biomimetics*, 2023, 8(2):149. DOI:10.3390/biomimetics8020149.

[9] Moustafa G, Tolba MA, El-Rifaie AM, Ginidi A, Shaheen AM, Abid S. A Subtraction-Average-Based Optimizer for Solving Engineering Problems with Applications on TCSC Allocation in Power Systems[J]. *Biomimetics*, 2023, 8:332. DOI:10.3390/biomimetics8040332.

[10] Wang Y, He X, Liu Q, Razmjooy S. Economic and technical analysis of an HRES (Hybrid Renewable Energy System) comprising wind, PV, and fuel cells using an improved subtraction-average-based optimizer[J].

[11] Smaili I H, Almalawi D R, Shaheen A M, Mansour H S E. Optimizing PV Sources and Shunt Capacitors for Energy Efficiency Improvement in Distribution Systems Using Subtraction-Average Algorithm[J]. *Mathematics*, 2024, 12:625. DOI:10.3390/math12050625 (Journal Online Sources style) K. Author. (year, month). Title. *Journal* [Type of medium]. Volume(issue), paging if given. Available: http://www.(URL).

[12] NAIR V, HINTON G E. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair[C]. International Conference on International Conference on Machine Learning.Omnipress, 2010: 807-814.

[13] Benzaoui, K., Bouguerra, A., Zeghlache, S., et al. (2024). SABO optimization algorithm-based backstepping controller for DSIG within a wind turbine system. Electr Eng. https://doi.org/10.1007/s00202-024-02839-1.

[14] Song Q, Wang J, Song Q, 等. Fault diagnosis of HVCB via the subtraction average based optimizer algorithm optimized multi channel CNN-SABO-SVM network[J]. *Scientific Reports*,2024,14:29507. DOI:10.1038/s41598-024-80954-6.

[15] Yang N, Liu J, Zhao W-Q, Tan Y. Fault Diagnosis of Gear Based on Multichannel Feature Fusion and DropKey-Vision Transformer[J]. *IEEE Sensors Journal*, 2024, 24(4):4758-4770. DOI:10.1109/JSEN.2023.3344999.

[16] Guo J, Zhang T, Xue K, Liu J, Wu J, Zhao Y. Parameter-adaptive re-constrained VMD method based on subtraction average-based optimizer for fault-bearing vibration signal processing[J]. *Measurement Science and Technology*, 2024, 36(1):016174. DOI:10.1088/1361-6501/ad962d.

[17] Liu, X., Li, J., Liu, J., Huang, C., & Liu, L. (2024). Prediction of permanent deformation of subgrade soils under F-T cycles using SABO-optimized CNN-BiLSTM network. Case Studies in Construction Materials, 21, e03807.