engineering proceedings

MDPI

Proceeding Paper

Improved Evolutionary Strategy Reinforcement Learning for Multi-Objective Dynamic Scheduling of Hybrid Flow Shop Problem †

Junjie Zhang, Yarong Chen * and Jabir Mumtaz *

School of Mechanical and Electrical Engineering, Wenzhou University, Wenzhou 325035, China; zhangjunjie6232@163.com

* Correspondence: yarongchen@126.com (Y.C.); jabirmumtaz@live.com (J.M.)

† Presented at the 4th International Conference on Advances in Mechanical Engineering (ICAME-24), Islamabad, Pakistan, 8 August 2024.

Abstract: This paper introduces the Improved Evolution Strategy Reinforcement Learning (I-ES) algorithm. The I-ES algorithm is designed to minimize the makespan and total energy consumption (TEC) in a multi-objective dynamic scheduling problem within a hybrid flow shop. It addresses key challenges such as flexible preventive maintenance for machines, random job arrivals, uncertain processing times, and the setup time. An experimental comparison of the I-ES-based approach with Evolution Strategy Reinforcement Learning (ES) algorithms and scheduling rules, which are a combination of job selection rules and machine selection rules, has been carried out by designing problem examples. The Generational Distance (GD) and Inverted Generational Distance (IGD) metrics of the I-ES algorithm are obtained by averaging all the instances as 389.14 and 1476.25, respectively, which are smaller than the other compared algorithms, so the results show that the I-ES algorithm can obtain solutions with superior convergence compared to that of the ES algorithm and dispatching rules.

Keywords: flexible flow shop; smart manufacturing; evolution strategy; dynamic scheduling; setup time

check for updates

Citation: Zhang, J.; Chen, Y.; Mumtaz, J. Improved Evolutionary Strategy Reinforcement Learning for Multi-Objective Dynamic Scheduling of Hybrid Flow Shop Problem. *Eng. Proc.* **2024**, *75*, 22. https://doi.org/10.3390/engproc2024075022

Academic Editors: Muhammad Mahabat Khan, Muhammad Irfan, Mohammad Javed Hyder and Manzar Masud

Published: 24 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A hybrid flow shop, also referred to as a flexible flow shop, comprises multiple stages where at least one stage includes several parallel machines. These hybrid flow shops are prevalent in various industries, such as equipment manufacturing, semiconductor fabrication, and chemical processing. Therefore, investigating Hybrid Flow Shop Scheduling Problems (HFSPs) is both theoretically significant and practically valuable. Within the scope of smart manufacturing, real-time scheduling applying adaptive learning methods has become a significant focus of research. Current studies on HFSP dynamic scheduling problems leveraging reinforcement learning (RL) have yielded promising outcomes. Chen et al. introduced an RL method for stochastic flexible flow shop scheduling, incorporating Monte Carlo tree search to improve the training efficacy and sample utilization rates, successfully validating their approach [1]. Heger et al. applied reinforcement learning to adaptively modify the k-value of Apparent Tardiness Cost with Setups (ATCS) ordering rules in a complex manufacturing setting, reducing the average tardiness rate by 5% [2]. Jia et al. developed a Multi-Population Memetic Algorithm combined with Q-learning to address Distributed Assembly Hybrid Flow Shop Scheduling with flexible preventive maintenance, demonstrating through their results that MPMA-QL offers superior solutions [3]. Zhao et al. introduced a deep reinforcement learning (DRL) architecture utilizing heterogeneous graph neural networks for solving hybrid flow shop issues, with the experimental results highlighting its excellent generalization performance and solution efficiency [4].

Liu et al. integrated Genetic Algorithms (GAs) with RL to tackle HFSPs, experimentally confirming the feasibility and effectiveness of their method [5]. Additionally, Liu et al. explored a deep multi-agent reinforcement learning strategy to address dynamic job shop scheduling issues [6]. They further introduced a two-depth Q-network algorithm with a hierarchical distributed framework to handle dynamic flexible job shop scheduling [7]. Su et al. examined a framework combining graph neural networks and evolutionary strategy RL for dynamic job shop scheduling, accommodating machine failures and random processing times [8].

To date, a comprehensive study addressing multi-objective hybrid flow shop dynamic scheduling, accounting for flexible preventive maintenance, random job arrivals, uncertain processing times, and setup times, remains unexplored. This paper addresses this gap by exploring HFSPs, aiming to reduce the makespan as well as the TEC. It proposes a real-time scheduling approach founded on I-ES RL, comparing its performance with the ES RL algorithm and various job sequencing policies.

2. Overview of the Problem

The multi-objective dynamic scheduling problem for the HFSP with setup times (MDSPHF-S) can be outlined as follows: n jobs J_i $(i = 1, 2, \ldots, n)$, arriving dynamically, undergo processing through K stages. These n jobs can be categorized into f families, where setup times are required between different families. Each stage k has m_k $(m_k \geq 1; k = 1, 2, \ldots, K)$ machines, with one or more stages including multiple machines. Flexible preventive maintenance (FPM) is considered in line with production requirements, meaning that a machine's continuous operating time must not surpass the upkeep limit UT, and the duration of maintenance is t^m. The primary goals of the scheduling are to minimize the makespan and the overall energy consumption. Decision points occur when a job is processed at a stage or when a new job arrives, at which point, the job is allocated to a machine to optimize the objective function.

The problem is framed with the following assumptions: (1) Each machine handles only one job at a time. (2) The setup times for machines and job transport times between stages are considered negligible. (3) All the jobs at stage k are processed in the same duration across all machines. (4) If two successive jobs on a machine are from the same family, the setup time is zero; otherwise, the setup is performed after the preceding job's completion. (5) The setup times are taken into account at the initial state.

3. Scheduling Strategy for MDSPHF-S Derived from I-ES Algorithm

To tackle the problem addressed in this study, we employ a partially connected neural network model to enhance the ES algorithm, referred to as the I-ES algorithm. Figure 1 illustrates the interaction sequence within the dynamic scheduling system of the HFSP implemented with the I-ES algorithm. Figure 1 illustrates the structure of an HFSP dynamic scheduling system based on the I-ES algorithm. The system uses a simulation module to model factors such as machine maintenance, setup times, and uncertain processing times in production, and feeds these data into a non-fully connected neural network model for processing, outputting optimized scheduling decisions. A feedback mechanism allows the simulation results to be used to adjust and optimize the neural network model, thus improving the stability and robustness of the scheduling system.

3.1. State Space

Selecting an appropriate state can capture real-time changes in the system state. For the MDSPHF-S characteristics, the environment is defined across three dimensions: job, machine, and stage. The state space vector SSS is represented as $[Q, k, N_k, U_{kc}^r, f_{kc}, p_{ik}^{min}, p_{ik}^{max}, p_{ik}^{later_min}, p_{ik}^{later_max}]$. Here, Q represents the total number of jobs in the shop; N_k represents the total number of jobs in stage k; U_{kc}^r represents the residual maintenance threshold of the machines in the shop; f_{kc} represents the family of jobs processed by the machine c in stage

k; and the last four represent the minimum and maximum processing time as well as the minimum and maximum residual processing time of all the jobs in stage *k*, respectively.

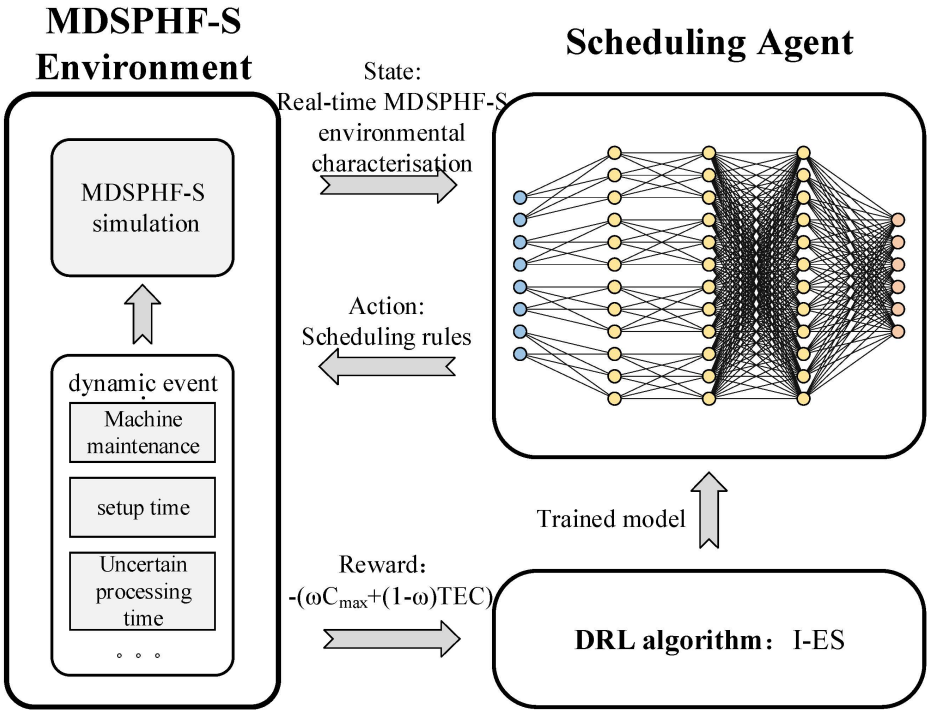


Figure 1. Interaction flow chart of MDSPHF-S dynamic scheduling system utilizing the I-ES.

3.2. Space of Available Actions

Within shop scheduling frameworks, an action $a_t \in A_t$ at decision moment t involves selecting a machine M_{kc} for job J_i to be processed at state S_t. Considering the characteristics and objectives of MDSPHF-S, the following four job sequencing rules and one machine selection rule are designed.

Job selection rules: select the job with the shortest/longest processing time at the current stage k (SPT/LPT), and select the job with the shortest/longest remaining total processing time at the current stage k (SRM/LRM).

Machine selection rule: choose the machine with the smallest sum of processing energy and setup energy (SPE).

3.3. Fitness Function

This paper aims to minimize both the makespan and TEC. These objectives are combined into a single optimization goal using a weighting method. Given that both the makespan and TEC are inversely related to the fitness function, the fitness function for the ES algorithm is expressed as Equation (1). C_{max} represents the maximum completion time after all the jobs are completed, the TEC is the total energy consumption, and ω is the weight, ω = 0.7.

$$Fitness = -(\omega C_{max} + (1 - \omega)\text{TEC}) \quad (1)$$

4. Quantitative Illustration and Assessment

4.1. Configuration of Parameters

Table 1 lists the parameters used to define the problem. P_{st} is the range of the setup time, EC_{mp} is the range of process energy consumption, EC_{st} is the range of setup energy consumption, EC_m is the range of maintenance energy consumption, and EC_i is the range of idle energy consumption. As an example, U (1, 10) means that it is uniformly generated in the range of 1 to 10.

Table 1. Parameter value ranges for the issue.

Instance	P_{st}	n	m_k	K	p_{ik}	EC_{mp}	EC_{st}	EC_m	EC_i
Small	U (1, 10)								
Medium	U (1, 50)	100	U (1, 4)	4	U (1, 99)	U (4, 15)	U (2, 4)	U (5, 18)	U (1, 2)
Large	U (1, 100)								

4.2. Analysis of Findings

This study evaluates the effectiveness of the I-ES scheduling approach and ES algorithms, along with four scheduling rules, by using the performance metrics GD and IGD. The GD measures how close the solution set is to the true Pareto front, while the IGD assesses the extent to which the solution set covers the true Pareto front. The calculation results for the different algorithms are presented in Figure 2 and Table 2.

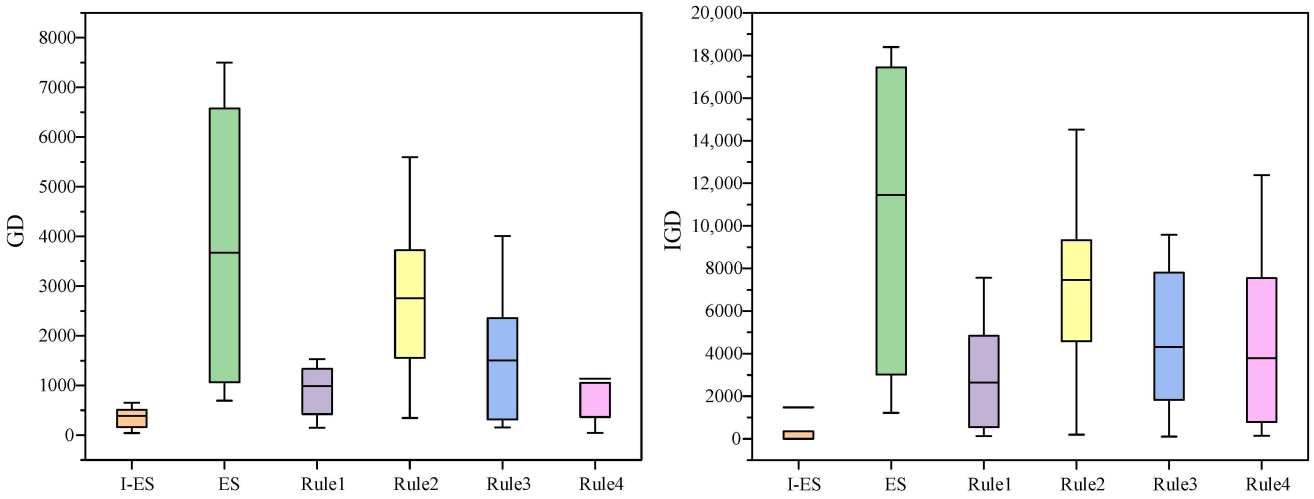


Figure 2. Boxplots of GD and IGD performance indicators.

Table 2. Comparative analysis of performance metrics for various scheduling methods.

Instance	GD						IGD					
	I-ES	ES	Rule1	Rule2	Rule3	Rule4	I-ES	ES	Rule1	Rule2	Rule3	Rule4
Small	152.68	843.13	196.54	1730.29	262.87	339.88	7310.72	11,514.5	7565.27	9321.38	7801.76	7984.68
	304.29	6574.85	657.3	1553.22	315.45	532.22	0.0	17,440.39	868.92	3764.25	770.4	925.29
	42.26	1064.78	150.61	344.88	155.74	43.99	68.18	3021.71	130.86	194.73	105.75	138.08
Medium	508.76	7235.0	690.12	1406.81	1529.78	860.47	0.0	18,396.85	2150.18	4586.95	3896.29	793.25
	289.38	695.22	423.01	2881.07	670.26	621.7	70.95	1222.33	1849.65	9146.78	2485.37	1942.01
	295.48	6521.05	1179.53	3721.76	1875.35	363.99	0.0	18,234.65	551.95	7122.61	4422.33	470.74
Large	1096.58	1425.79	1529.25	2168.61	2357.81	2440.61	5362.47	14,581.94	5470.35	7406.88	7996.56	7543.1
	161.31	1163.27	2729.57	5593.6	4005.96	3979.68	118.33	2179.99	4836.17	14,516.3	9581.24	12,377.79
	651.48	7495.49	1332.83	5402.44	2354.08	1049.78	355.58	16,440.16	355.58	11,078.65	1826.98	1942.4
Average	389.14	3668.73	987.64	2755.85	1503.03	1136.92	1476.25	11,448.06	2642.10	7459.84	4320.74	3790.82

Rule1: SPT-SPE; Rule2: LPT-SPE; Rule3: SRM-SPE; Rule4: LRM-SPE.

As shown in the box-and-line plot in Figure 2, it can be seen that the I-ES algorithm has better robustness. The box-and-line plot shows how multiple algorithms perform in different test instances. Specifically, the I-ES algorithm has a lower mean number position, indicating that it outperforms the other algorithms in most test instances. In addition, the I-ES algorithm has a relatively narrow box, which means that its performance is less volatile and the results are more stable. In contrast, the other algorithms have wider bins, indicating that their performance fluctuates more between instances and is less robust. The reason for this is that reducing redundant connections in non-fully connected neural networks can minimize these interfering factors. This enhances the model's capability to learn crucial features, thereby boosting the stability and robustness of the I-ES algorithm across different environments.

As observed in Table 2, the I-ES-based approach, when averaged across all cases, achieves GD and IGD values of 389.14 and 1476.25, respectively, indicating superior convergence compared to the other algorithms evaluated. This improved performance is attributed to the use of non-fully connected neural networks, which reduce the model complexity by limiting the number of connections, thereby decreasing the risk of overfitting. Additionally, the selective connectivity inherent in non-fully connected structures allows the model to focus on significant features while disregarding irrelevant ones, enhancing the model's ability to extract key information and thereby improving the overall performance.

5. Conclusions

To address the dynamic scheduling challenges in an HFSP, this paper proposes an I-ES-based method that optimizes objectives through a weighted combination of bi-objective optimization and an enhanced neural network model. The enhanced neural network model leverages advanced learning techniques to improve the scheduling accuracy and adaptability. The experimental findings, as observed in Table 2, demonstrate that the I-ES algorithm consistently achieves superior objective values compared to the other algorithms evaluated. Specifically, the I-ES-based approach, when averaged across all the cases, achieves GD and IGD values of 389.14 and 1476.25, respectively. These results indicate superior convergence and diversity. The lower GD value signifies that the solutions obtained by the I-ES algorithm are closer to the true Pareto front, while the lower IGD value indicates that the solutions cover the Pareto front more comprehensively. This performance demonstrates the efficacy of the I-ES method in handling dynamic scheduling challenges in the HFSP. The use of a weighted combination of bi-objective optimization and an enhanced neural network model allows for a more robust and adaptable approach to scheduling.

Future research could explore the extension of this method to handle more realistic dynamic events, such as order insertions and withdrawals, which are common in practical scheduling environments. Additionally, further enhancements to the neural network model and the optimization strategy could be investigated to improve the efficiency and applicability of the method in more complex and dynamic scheduling scenarios.

Author Contributions: Conceptualization, J.Z. and Y.C.; methodology, J.Z.; software, J.Z. and J.M.; validation, J.Z. and J.M.; formal analysis, J.Z. and Y.C.; writing—review and editing, Y.C.; visualization, J.Z.; supervision, J.Z.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: Basic scientific research project of Wenzhou City (G2023036 & G20240020).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Chen, L.; Liu, H.; Jia, N.; Ren, N.L.; Cui, R.B.; Wei, W. Real-time stochastic flexible flow shop scheduling in a credit factory with model-based reinforcement learning. *Int. J. Prod. Res.* **2024**. [CrossRef]
2. Heger, J.; Voss, T. Dynamically adjusting the k-values of the ATCS rule in a flexible flow shop scenario with reinforcement learning. *Int. J. Prod. Res.* **2023**, *61*, 147–161. [CrossRef]
3. Jia, Y.; Yan, Q.; Wang, H. Q-learning driven multi-population memetic algorithm for distributed three-stage assembly hybrid flow shop scheduling with flexible preventive maintenance. *Expert Syst. Appl.* **2023**, *232*, 120837. [CrossRef]
4. Zhao, Y.; Luo, X.; Zhang, Y. The application of heterogeneous graph neural network and deep reinforcement learning in hybrid flow shop scheduling problem. *Comput. Ind. Eng.* **2024**, *187*, 109802. [CrossRef]
5. Liu, Y.; Shen, W.; Zhang, C.; Sun, X.Y. Agent-based simulation and optimization of hybrid flow shop considering multi-skilled workers and fatigue factors. *Robot. Comput.-Integr. Manuf.* **2023**, *80*, 102478. [CrossRef]
6. Liu, R.; Piplani, R.; Toro, C. A deep multi-agent reinforcement learning approach to solve dynamic job shop scheduling problem. *Comput. Oper. Res.* **2023**, *159*, 106294. [CrossRef]

7. Liu, R.; Piplani, R.; Toro, C. Deep reinforcement learning for dynamic scheduling of a flexible job shop. *Int. J. Prod. Res.* **2022**, *60*, 4049–4069. [CrossRef]
8. Su, C.; Zhang, C.; Xia, D.; Han, B.; Wang, C.; Chen, G.; Xie, L. Evolution strategies-based optimized graph reinforcement learning for solving dynamic job shop scheduling problem. *Appl. Soft Comput.* **2023**, *145*, 110596. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.