


Neuro-Evolution of Augmenting Topologies for Dynamic Scheduling of Hybrid Flow Shop Problem [†]

Junjie Zhang , Yarong Chen ^{*}, Jabir Mumtaz ^{*}  and Shengwei Zhou

School of Mechanical and Electrical Engineering, Wenzhou University, Wenzhou 325035, China; zhangjunjie6232@163.com (J.Z.); zhousw_zsw@126.com (S.Z.)

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

[†] Presented at the Third International Conference on Advances in Mechanical Engineering 2023 (ICAME-23), Islamabad, Pakistan, 24 August 2023.

Abstract: In this paper, the Neuro-Evolution of Augmenting Topologies (NEAT) algorithm is proposed to minimize the maximum completion time in a dynamic scheduling problem of hybrid flow shops. In hybrid flow shops, machines require flexible preventive maintenance and jobs arrive randomly with uncertain processing times. The NEAT-based approach is experimentally compared with the SPT and FIFO scheduling rules by designing problem instances. The results show that the NEAT-based scheduling method can obtain solutions with better convergence while responding quickly compared to the scheduling rules.

Keywords: hybrid flow shop; reinforcement learning; Neuro-Evolution of Augmenting Topologies; makespan; dynamic scheduling



Citation: Zhang, J.; Chen, Y.; Mumtaz, J.; Zhou, S. Neuro-Evolution of Augmenting Topologies for Dynamic Scheduling of Hybrid Flow Shop Problem. *Eng. Proc.* **2023**, *45*, 25. <https://doi.org/10.3390/engproc2023045025>

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

Published: 11 September 2023



Copyright: © 2023 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 is a kind of flow shop containing more than two stages and at least one stage with multiple parallel machines, also known as a flexible flow shop. The hybrid flow shop scheduling problem (HFSP) is of great theoretical significance and practical value, as it is widely applied in the chemical, textile, steel and semiconductor industries. In the context of intelligent manufacturing, dynamic scheduling based on reinforcement learning has become a research trend. Many research studies have been conducted in the literature. Han et al. first proposed a reinforcement learning method for HFSP [1]. Gil and Lee studied the use of the deep reinforcement learning approach to solve the material scheduling problem of many machines in a hybrid flow shop environment [2]. Cai et al. proposed a new shuffle frog-learning algorithm with Q-learning to solve a distributed assembly hybrid flow shop scheduling problem with fabrication, transportation and assembly [3]. Wang, J.J. and Wang, L. studied an energy-aware distributed hybrid flow shop scheduling method based on reinforcement learning [4]. Lang et al. presented a dynamic scheduling method based on the NEAT algorithm for a two-stage hybrid flow shop scheduling problem with family setup times [5].

The problem of dynamic scheduling in hybrid flow shops where machines require flexible preventive maintenance and where jobs arrive randomly and processing times are uncertain has not yet been identified, so this paper studies the HFSP scheduling problem with the objective of minimizing the makespan and designs a dynamic scheduling method based on NEAT reinforcement learning and compares it with scheduling rules.

2. Problem Description

The dynamic scheduling problem of the hybrid flow shop can be described as follows: n dynamically arriving jobs $J_i (i = 1, 2, \dots, n)$ are processed continuously in $K (K \geq 2)$ stages, stage k has $m_k (m_k \geq 1; k = 1, 2, \dots, K)$ machines, and there exists at least one stage

with multiple machines. Flexible preventive maintenance of the machine is considered in accordance with production reality, i.e., the machine cannot be continuously used for more than the maintenance threshold UT , and the maintenance time is t^m . The scheduling objective is to minimize the makespan. The decision moment is triggered by the job being processed at a certain stage or by the arrival of a new job, which is then scheduled to the machine to optimize the objective value.

The problem studied in this paper considering the following assumptions: (1) Each job can be processed on any one of the machines at stage k . (2) Job processing time is the same for all machines at the same stage. (3) A machine can only process one job at a time. (4) All jobs are processed according to the same process sequence. (5) After determining the sequence of processing of the jobs in the first stage, the other stages are processed in this order.

3. Dynamic Scheduling Method for Hybrid Flow Shop Based on NEAT Algorithm

The interaction flow of the hybrid flow shop dynamic scheduling system based on reinforcement learning is shown in Figure 1.

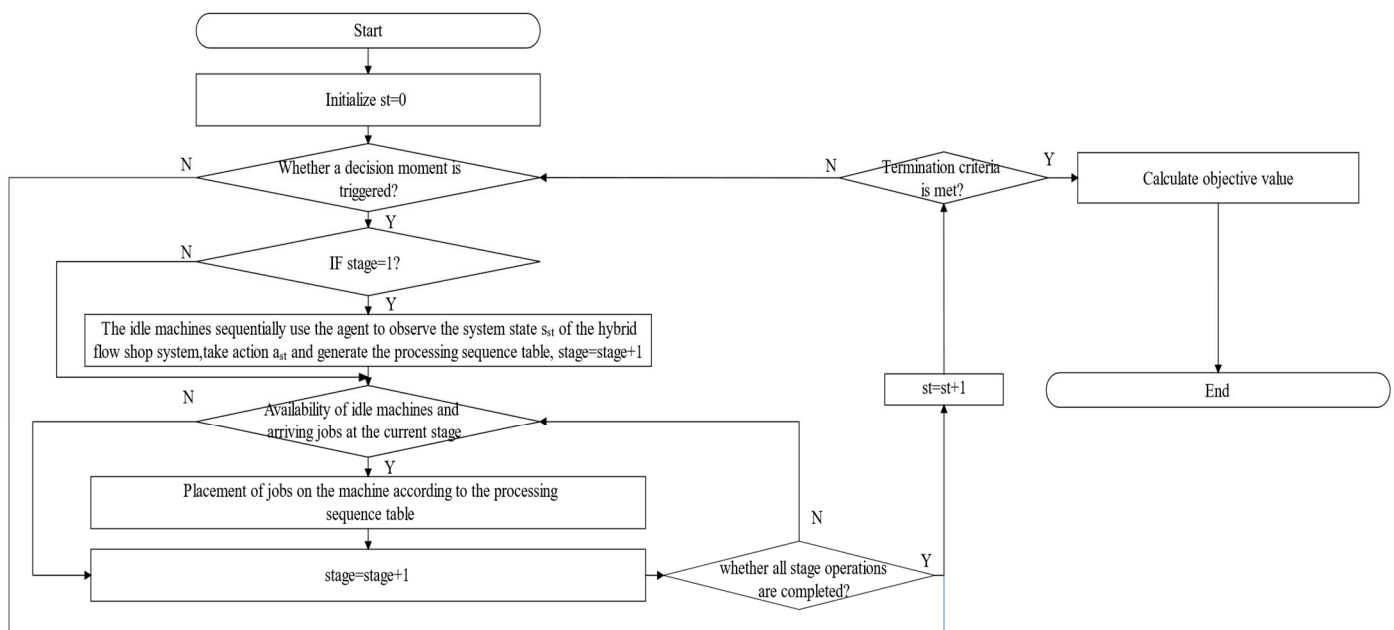


Figure 1. Interaction flow of hybrid flow shop system based on reinforcement learning.

3.1. State Space

Setting a suitable state space can reflect the change in the scheduling system in real time. For the characteristics of the dynamic scheduling problem of the hybrid flow shop, the environment is described in three dimensions (jobs, machines and stages) and the state space vector $S = [p_{ik}, T_i^r, T_{kc}, T_{kc}^a, U_{kc}^r]$ is defined. p_{ik} is the processing time corresponding to job J_i at stage k . T_i^r is the time interval between the current moment and the arrival time of job J_i . T_{kc} is the processing time of the job being processed by machine M_{kc} ($c = 1, 2, \dots, m_k$) at stage k , increasing the maintenance time t^m if maintenance is required. T_{kc}^a is the processing time of the job being processed by machine M_{kc} at stage k . U_{kc}^r is the maintenance threshold remaining at the current moment for machine M_{kc} at stage k .

3.2. Action Space

In the shop scheduling model, an action $a_t \in A_t$ is taken to determine a machine M_{kc} for processing job J_i at decision time t in state S_t . Based on the characteristics and objectives of the dynamic scheduling problem of a hybrid flow shop considering flexible preventive maintenance, the following three actions are designed.

Action 1: Selecting a job based on the SPT rule. The job J_i with the shortest processing time is selected from the waiting queue based on the SPT rule. One of the jobs J_i is selected at random if more than one job can be selected.

Action 2: Selecting a job based on FIFO rule. The job J_i with the earliest arrival time is selected from the waiting queue based on the FIFO rule. One of the jobs J_i is selected at random if more than one job can be selected.

Action 3: Waiting. No job is selected for machining.

The following constraints are set for several special cases: (1) When there are no jobs in the waiting queue, the agent can only select action 3. (2) When a new job arrives but there are no idle machines, action 3 is selected. (3) When there are jobs in the waiting queue and the machines are all idle, action 3 cannot be selected.

3.3. Fitness Function

The NEAT algorithm searches the optimal scheduling strategy through the optimization of the fitness function, so the fitness function of NEAT is designed to be $G = \frac{1}{C_{\max}}$. Maximizing the fitness function is equal to minimizing the makespan. C_{\max} can be described using Equation (1):

$$C_{\max} = \max_i C_i \quad (1)$$

where C_i is the completion time of all stages of processing of job J_i .

4. Numerical Example and Analysis

4.1. Parameter Settings

According to the pre-experimentation, we set the evolutionary generations in the NEAT algorithm to 10 and the population size to 100. The job processing time p_{ik} , the arrival time r_i and the number of machines m_k at each stage were generated based on a uniform distribution. The problem was divided into three instances according to the number of stages K and the number of jobs n . Each type of instance was run three times for the experiments, and the parameters of the training set are shown in Table 1.

Table 1. The range of parameter values for the training set problem.

K	n	p_{ik}	r_i	UT	t^m	m_k
2	10	U (1,5)	U (1,5)	10	2	U (1,2)
3	20	U (1,5)	U (1,5)	10	2	U (1,2)
4	30	U (1,5)	U (1,5)	10	2	U (1,2)

4.2. Results Analysis

In this paper, the performance of the NEAT scheduling method and the two scheduling rules are evaluated based on the makespan C_{\max} , running time T_t^R and response time T_s^R . The experimental results of the different algorithms are shown in Table 2.

Table 2. Experimental results of performance indicators for different scheduling methods.

Instance	NEAT			SPT			FIFO		
	C_{\max}	T_t^R/s	T_s^R/s	C_{\max}	T_t^R/ms	T_s^R/ms	C_{\max}	T_t^R/ms	T_s^R/ms
$K = 2, n = 10$	25	1.06	0.05	25	6.37	0.32	26	6.68	0.33
	45	1.16	0.06	47	9.87	0.49	47	10.29	0.51
	51	1.21	0.06	51	10.72	0.54	51	10.30	0.51
$K = 4, n = 30$	128	11.66	0.10	129	59.60	0.50	131	62.31	0.52
	122	11.82	0.09	122	62.20	0.52	124	58.60	0.49
	98	10.21	0.08	100	48.57	0.40	105	58.40	0.49
$K = 6, n = 100$	392	177.16	0.29	398	530.40	0.88	406	601.94	1.00
	384	164.61	0.27	404	485.38	0.81	396	461.69	0.77
	416	190.16	0.32	434	493.33	0.82	432	478.78	0.80

It can be seen from Table 2 that the NEAT-based approach outperforms the scheduling rules for all three instances with the objective C_{\max} . For the computation time, the SPT and FIFO rules have less time than the NEAT-based approach, because the NEAT reinforcement learning algorithm requires decisions to be made through interaction with the environment, while it is also able to respond the dynamic events quickly.

5. Conclusions

A NEAT-based approach is proposed for a dynamic scheduling problem of hybrid flow shops with machine preventive maintenance and the dynamic arrival of jobs to minimize maximum completion time. Our experimental results show that the NEAT algorithm is able to obtain better objective values while responding quickly. In future research, the proposed NEAT method can be used to solve dynamic scheduling problems considering some more realistic emergency events such as the insertion and withdrawal of orders.

Author Contributions: Conceptualization, J.Z. and Y.C.; methodology, J.Z. and S.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. and S.Z.; visualization, S.Z.; supervision, J.Z.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number [No.51705370].

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 conflict of interest.

References

1. Han, W.; Guo, F.; Su, X.C. A Reinforcement Learning Method for a Hybrid Flow-Shop Scheduling Problem. *Algorithms* **2019**, *12*, 222. [\[CrossRef\]](#)
2. Gil, C.B.; Lee, J.H. Deep Reinforcement Learning Approach for Material Scheduling Considering High-Dimensional Environment of Hybrid Flow-Shop Problem. *Appl. Sci.* **2022**, *12*, 9332. [\[CrossRef\]](#)
3. Cai, J.C.; Lei, D.M.; Wang, J.; Wang, L. A novel shuffled frog-leaping algorithm with reinforcement learning for distributed assembly hybrid flow shop scheduling. *Int. J. Prod. Res.* **2022**, *61*, 1233–1251. [\[CrossRef\]](#)
4. Wang, J.J.; Wang, L. A Cooperative Memetic Algorithm With Learning-Based Agent for Energy-Aware Distributed Hybrid Flow-Shop Scheduling. *IEEE Trans. Evol. Comput.* **2022**, *26*, 461–475. [\[CrossRef\]](#)
5. Lang, S.; Reggelin, T.; Schmidt, J.; Muller, M.; Nahhas, A. NeuroEvolution of augmenting topologies for solving a two-stage hybrid flow shop scheduling problem: A comparison of different solution strategies. *Expert Syst. Appl.* **2021**, *172*, 114666. [\[CrossRef\]](#)

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.