Migration Letters

Volume: 21, No: 4, pp. 419-441

ISSN: 1741-8984 (Print) ISSN: 1741-8992 (Online)
www.migrationletters.com

Jobshop Scheduling Combining Trail and Regular Production Plan

Gorle Mangamma[1], M. Srinivasa Rao[2], V.V.S. Kesavarao[3]

Abstract

Because of globalization and diversification of markets, application of conventional production scheduling techniques will not yield better economic results. Production disturbances due to new order handlings and combined processing of different jobs is the requirement in flexible job shop scheduling problems. Condition of simultaneous presence of trail and regular product demands from customers is to be handled nowadays to sustain in the market. This paper proposes a reinforcement algorithm. The algorithm that has been proposed has provided satisfactory results. In this connection an empirical mathematical model was developed for choosing a dispatching rule (DR) which is optimal to thoroughly modify the strategies of scheduling during the manufacturing process.

Keywords: *job shop scheduling, make span, production disruptions, dispatch rules.*

1. Introduction

Due to financial globalization and market expansion, the necessities for a manufactured good are solution into much varied, custom specific, besides continuously changing. multi variety customized mixed-line production has become the order instead of production a single product in a large scale which is unavoidable to meet in determinate issues for example product variety augmentation, decrease in batch size, demand variation, including order fluctuations, all these factors make changes in the operating efficiency of organizations.

Job Shop Scheduling represents one among the various methods in production control for optimum utilization of production resources and its optimization can get better efficiency of mixed-line productivity system, decrease expenditure, as well as enhance production.

Flexible Job Shop Scheduling problem (FJSP) generally takes into restricting routing procedures and limitations of machines. Researchers presumed that a machine cannot perform more than one process concurrently at same time [1] and satisfies the conventional routing constraints. In reality additional constraints such as combined processing constraint usually exist, for making sure the correctness of assembly

In the job shop scheduling, which is mixed line in nature often gets orders for trial runs as well as batch manufacturing simultaneously. While manufacturing new products which are made for the first time in the trail production order which is

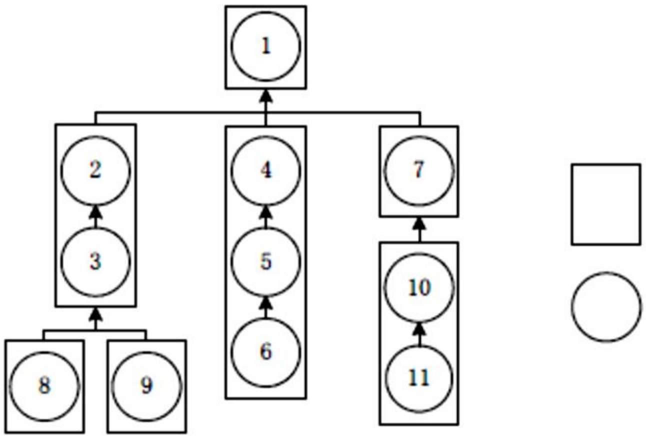
[1] Part-time Ph.D. Scholar in Department of Mechanical Engineering, College of Engineering, Andhra University, Visakhapatnam
[2] DGM (Maintenance), Department of MMSM, Visakhapatnam Steel Plant.
[3] Professor, Department of Mechanical Engineering, College of Engineering, Andhra University, Visakhapatnam.

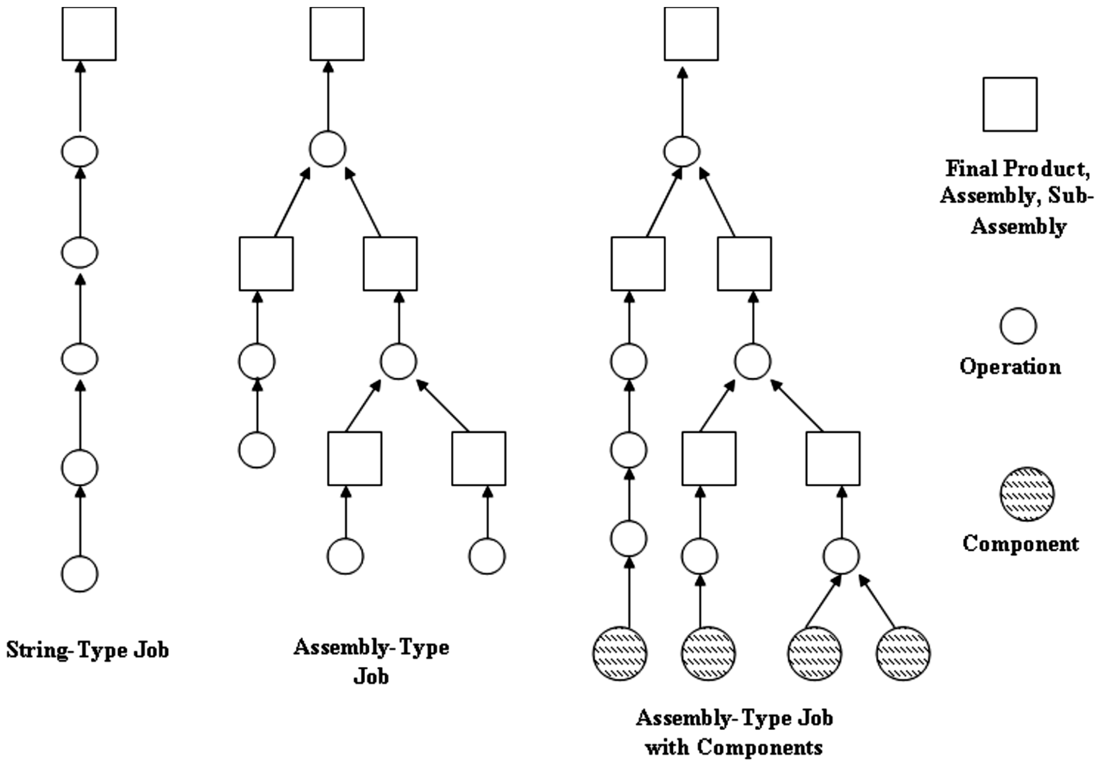
indeterminate, and the manufacturing time cannot be predicted efficiently. In fact the production orders which are batch in nature are somewhat steady to a certain level of production. The real manufacturing plan of the production jobs which are batch in nature cannot be executed as per pre fixed program due to the uncertainty in trail production. During batch production or development of new products development .it makes the scheduling issue is more difficult in the job shop environment. An assembly system is a complex series and parallel-series flow path. If any part of the series flow path becomes a bottleneck, the whole production rate decreases.

In the case of automated assembly equipment, it could be due to a number of factors, including software problems, communication problems, equipment issues, and more. Where humans are concerned, the Assembler must be up to the task before him. He must be qualified and reliable. If his throughput is slow or if he is error-prone, then you have an inordinate number of units going back for rework and a throughput bottleneck.

Also, floor layout is important. Work must always flow down the line and never double back or take a side path. The more steps involved, the more time is lost, and the fewer the units produced.

In assembly job shop (AJS) refers to a floor which considers both manufacturing and components assembly activities. Every Job possess certain set of components and subcomponents which are assembled together to prepare the final job. These subcomponents in turn have sub-sub-components and so on which can be said as multi level components jobs. Higher-level subcomponents cannot be assembled until all preceding lower-levels components and/or subassemblies are mixed together according product plan.   This makes the AJS problems quite challenging as end-items with single level, and multi -level assembly structures are considered. The efficiency of the shop is calculated using mean flow time and mean tardiness. In an assembly job shop, the problematic issues are basically job delays which are two types. The first type of delay is due to inherent machine capacity and the other one is due to non availability either assembly or sub assembly items which are essential in parallel before starting assembly operations.





Mathematical programming reaches global optimal schedules under various constraints [2]., metaheuristic methods with maximum variable space (For example, GA and swarm intelligence) are also being used [3, 4]. Recently, artificial intelligent techniques are applied on scheduling problems which are dynamic in nature by constantly optimizing scheduling efficiency. Lawrynowicz [5] suggested application of scheduling rules to improve efficiency, develop an effective framework with algorithms which are in nature heuristic to resolve issues. Liu [6] applied an analytic hierarchy technique to enhance the effectiveness of scheduling. Trentesaux et al. [7] applied in dynamic shop environment simulation-based four new dispatching rules. Liu [8] suggested the Q-learning algorithm an adaptive real-time scheduling technique. Qu et al. [9] developed a learning method (RL) which is reinforced with nature in the designing of dispatching rules. Chen et al. [10] explained the lean redesign optimization strategy in numerous dynamic scenarios. Chen et al. [11 ,12] suggested a way of scheduling which involves encapsulation for steel processing including dynamic rescheduling model and modeled genetic programming for the rescheduling problems under irregular events. In a dynamic job shop scheduling a mechanism which directs and learns on its own using multi-agent method is also being applied.

2. Literature Review

From above stated literature review it evident that the scheduling methods are of three types: the first one is algorithm-technique which heuristic based, second one is a technique which is based on dispatching rules, third one is a technique based on reinforcement learning. The heuristic algorithms solve by encoding and decoding techniques and results can be attained on the computation of fitness function and it needs rescheduling in case any problem in production process. The second method as stated above uses a set of dispatching scheduling rules such as shortest processing time and first come and first out and it reschedules production disturbance occurs in case occurs but the quality of its result is not high even though it addresses real time solution.

In an adaptive scheduling technique/reinforcement algorithm, which intelligently selects the optimum scheduling technique as per the context of operational environment.

2.1 Literature gap

Zhu et al [14] has not considered scheduling of the jobs in each manufacturing thing which is a common practice. Anran Zhao [15] has applied dispatch rules on single machine using machine learning while multi machine is not addressed, same is addressed here;

Hence motivation of this paper is application of reinforcement algorithm which gives quality solutions as well as real time solutions.

2.2 Objectives of the Paper

- Formulation of mathematical scheduling
- scheduling rules optimization for the jobs
- Proposal of scheduling method such as reinforcement algorithms which respond to disturbance events in order to get high-quality scheduling results.
- Application of mixed line job shop schedule concerns and integrated processing limitations to a flexible job shop scheduling model on the simultaneous occurrence of trial manufacturing as well as orders for batch production.
- A learning technique centred on a case is taken and it is linked with disturbance processing mechanism. The problems of conventional dispatching rule techniques as per real-time state after learning to address how to enhance the efficiency of algorithm
- Hence the contribution of the paper is focused on

1) A disturbance control technique

2) A dynamic and adaptive scheduling

3. Methodology

Formulation of simple job shop scheduling in batches with objectives

Notations:

i Notation for jobs, n.

h Notation for batches, b = l,...,K +1.

r Notation for positions, r =1,...,n.

Parameters:

No number of jobs that require processing at start.

p_i time of processing of a job i.

K No. of non availability time slots

D_i target date specified for a job i.

St_b starting time of b th unavailability slot

Fi_b the b^{th} unavailability slot's end time

M large number.

Variables:

$co_{[r]}$ end time of work positioned at r

x_{irb} binary integer with a value of 1 if task i is planned in location r and group b is scheduled in sequence, else 0.

e_i job earliness

E_{nux} Maximum earliness of jobs

t_i job lateness

T_{niax} Maximum lateness of job

y_{irb} finishing time of job i when it is positioned in slot r, batch b.

The funct(1) lowers sum of the maximum earliness as well as punctuality of jobs for make span

$$\text{koin } z = E_{koax} + T_{koax} \qquad (1)$$

Min c

C = max(V_{irb})

S.t. $$\sum_{r=1}^{n}\sum_{b=1}^{k+1} x_{irb} = 1 \ \forall i = 1,2,3,4,5,6,...,n$$

funct(2) ensures that each job assigned to only one batch along with one position

$$\sum_{i=1}^{n}\sum_{b=1}^{k+1} x_{irb} = 1 \qquad \forall i = (1,2,3,4,5,6,...,n\ )$$

func(3) ensures that only one job scheduled to position r

$$\sum_{i=1}^{n}\sum_{r=1}^{n} x_{irb} p_i \le (S_b - F_{b-1}) \qquad \forall b = (1,2,3,4,5,6,...,k+1)$$

func(4) processing time for each batch is restricted

$$\sum_{i=1}^{n}\sum_{r=1}^{n} x_{irb} \le M \times \sum_{i=1}^{n}\sum_{r=1}^{n} x_{irb(b-1)} \qquad \forall b = (2,...,k+1) \qquad (5)$$

because when no work is found in one group, no job can be assigned in the subsequent batch.

$$c_{[r]} \ge c_{[r-1]} + \sum_{i=1}^{n}\sum_{b=1}^{k+1} x_{irb} p_i \qquad \forall r = (1,2,3,4,5,6,...,n) \qquad (6)$$

$$c_{[r]} \ge \sum_{i=1}^{n}\sum_{b=1}^{k+1} x_{irb} \square (p_i + F_{b-1}) \qquad \forall r = (1,2,3,4,5,6,...,n) \qquad (7)$$

$$c_{[r]} \le \sum_{i=1}^{n}\sum_{b=1}^{k+1} x_{irb} \times p_i + \operatorname{Max}\left\{ c_{[r]}, \sum_{i=1}^{n}\sum_{b=1}^{k+1} x_{irb} \times F_{b-1} \right\} \qquad \forall r = 1,2,3,4,5,6,...,n \quad (8)$$

$$\sum_{r=1}^{n} c_{[r]} \times x_{irb} \le S_b \qquad \forall i = (1,2,3,4,5,6,...,n,\ b = 1,2,3,4,5,6,...,k+1) \quad (9)$$

$$t_i - e_i = \sum_{r=1}^{n}\sum_{b=1}^{k+1} c_{[r]} \times x_{irb} - d_i \qquad \forall i = (1,2,3,4,5,6,...,n) \qquad (10)$$

$$E_{koax} \ge e_i \qquad \forall i =(\ 1,2,3,4,5,6,...,n) \qquad (11)$$

$$T_{koax} \ge t_i \qquad \forall i = (1,2,3,4,5,6,...,n) \qquad (12)$$

$$x_{irb} \in \{0,\ 1\} \quad i = (1,2,3,4,5,6,...,N),\ r = 1,2,3,4,5,6,...,N,\ b = 1,2,3,4,5,6,...,k+1 \quad (13)$$

$c_{[r1]} \geq 0$ r = (12,344...,n) (14)

$e1_i \geq 0$ i = (1,...2345,n) (15)

$t_{i1} \geq 0$ i = (1,2345...,n) (16)

$$w_r = \left(c_{[r-1]} + \sum_{i=1}^{n}\sum_{b=1}^{k+1} x_{irb} p_i\right) - \left(\sum_{i=1}^{n}\sum_{b=1}^{k+1} x_{irb} (p_i + F_{b-1})\right) \quad \forall b = 1,2,;34...n + 1 (17)$$

wr is a variable with no sign. Then there should be:

$w_r + KO \times (1 - z_r) \geq 0$ $\forall r = 1,2,3,4,5,6,...,n$ (18a)

$w_r \leq KO \times z_r$ $\forall r = 1,2,3,4,5,6,...,n$ (19a)

Likewise, to revoke the nonlinear t erko in constraint (9) as well as (10) we specify:

$y_{irb} = x_{irb} \times p_i$ $\forall r = 1,2,3,4,5,6,...,n$, $r = 1,2,3,4,5,6,...,n$, $b = 1,2,3,4,5,6,...,k + 1$ (20)a

We get

$y_{irb} - KO \times (1 - x_{irb}) \leq c_{[r]}$ $\forall i = 1,2,3,4,5,6,...,n$, $r = 1,2,3,4,5,6,...,n$, $b = 1,2,3,4,5,6,...,k + 1$ (21a)

$y_{irb} + KO \times (1 - x_{irb}) \geq c_{[r]}$ $\forall i = 1,2,3,4,5,6,...,n$, $r = 1,2,3,4,5,6,...,n$, $b = 1,2,3,4,5,6,...,k + 1$ (22a)

$y_{irb} \leq KO \times x_{irb}$ $\forall i = 1,2,3,4,5,6,...,n$, $r = 1,2,3,4,5,6,...,n$, $b = 1,2,3,4,5,6,...,k + 1$ (23a)

Hence, constraints (9) as well as (10) will be rewritten in the form of (24) as well as (25).

$y_{irb} \leq S_b$ $\forall i = 1,2,3,4,5,6,...,n$, $r = 1,2,3,4,5,6,...,n$, $b = 1,2,3,4,5,6,...,k + 1$ (24a)

This only resolves the job shop problem

3.2 dispatch rules: The technique of enhancing the job shop scheduling problem/issue suggested needs awareness from past data to make a technique for choosing an best DISPATCHING RULES which can make scheduling tactics in existent time. The scheduling activities transform in response to the development of the manufacturing activity. Assuming to alter the scheduling policies' time is the theme of accomplishing the technique suggested. The latest time of scheduling policy is changing instant of DISPATCHING RULES. As job issue time as well as data in a manufacturing intervals are not known prior to being issued to the production system, technique of separating the sub-scheduling interval is fixed. Subsequent sub scheduling interval plans the jobs incoming with respect to preceding sub scheduling interval. To update the scheduling policy, the commencement timing of the sub-scheduling interval is used as the change point of its DISPATCHING RULES. The subsequent sub-scheduling interval is determined by the end timing of the work forthcoming in the interval, while sub-scheduling periods are sorted again enhancing the scheduling problem/issue using the prepared technique.

Before validating the performance of suggested technique, the following theories were suggested to check :

i. Without failure happening throughout machine activity;

ii. Every job is performed just on the machine once without any importance one job over other;

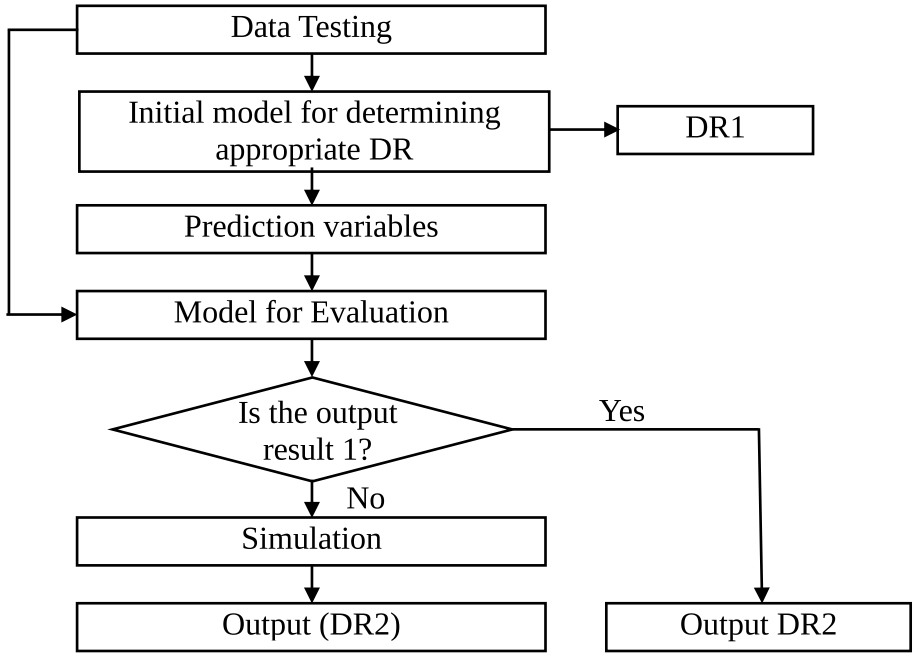
iii. only one activity of a job at a time on the machine performed;

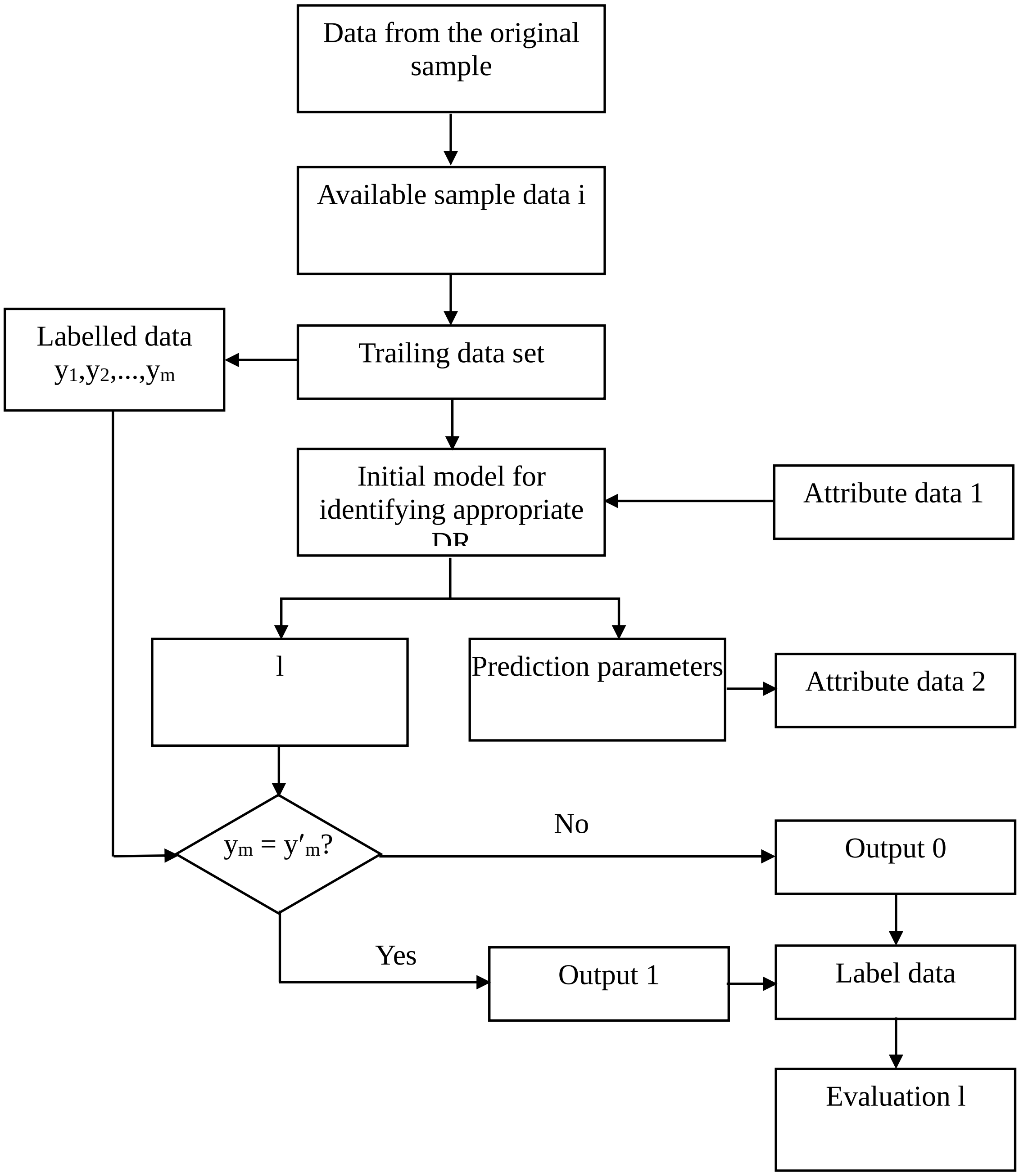
iv. The job's due date is ignored.

DISPATCHING RULES cannot acclimatize well to all production shops under the norms of every scheduling One -parameter DISPATCHING RULES along with mixed parameter DISPATCHING RULES to provide DISPATCHING RULES library rests on work specifications [15, 32]., the parameters of single parameter DISPATCHING RULES chosen comprises of setting time (ST) prior to any activity, operating time (PT) as well as the total time (TPT) of job, as mentioned :

$$TPT = ST + PT \tag{25}$$

$$R = \sum_{j=1}^{J} PT_j \tag{26}$$





3.3.1 Joint Processing Restraint in flexible job shop problem

It will prove challenging to guarantee the accuracy required for assembly if the jobs are executed effectively, hence, for meeting the necessities of assembling accuracy, separate tasks of more than 2 jobs need to be performed on the same machine at same instant and next task is performed only when the two jobs are complete, which is nothing but , combined processing.

Depicted in Fig.1, Jo_1 and Jo_2 are performed in 2 process routes, and the jth task of job J_i is labeled as Op_{ij}, while the third task Op_{12} and Op_{22}2 jobs requires combined processing, and Op_{13} and $O_{2p3}$are performed only after the joint processing of Op_{12} and Op_{22} is finished.

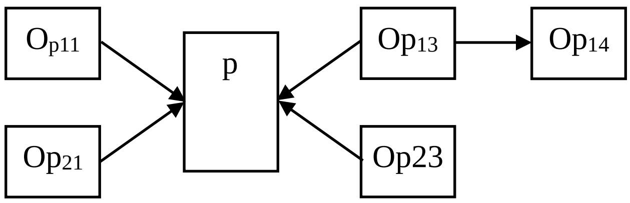


Figure 1: joint processing

3.3.2 Result for joint Processing restriction

2 tasks of dissimilar jobs requirement to be planned together when they require combined processing and satisfy the restriction. A virtual task is a strategy that comprises of tasks requiring combined processing that is viewed as a single activity. The main task is currently assigned by the job with a tiny job number and responsible for selecting type of virtual task scheduling. When parts and job number differ from one another, the task corresponding to the lower job number is designated as master task. The other task is auxiliary. The processing route with combined processing is depicted in Fig. 2.

1. For job scheduling , the requirement of combined processing is to be addressed.

2. If combined processing is not desirable, is planned as it is else further examine whether it satisfies the constraint for moving (3).

3. when all of the limitations are met, the 2 jobs are merged into a single virtual task for scheduling or wait till further tasks get released before proceeding to (3).

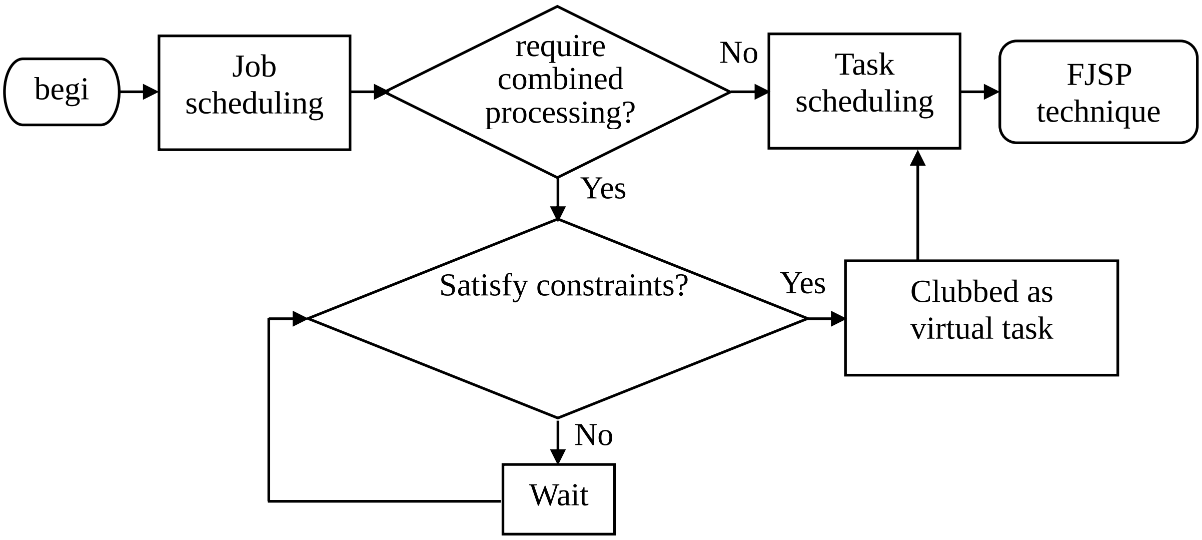


Figure 2: Task process route chart

3.3.3 formulation of model

The timeline is regarded as objective, so the underlying assumptions are adopted:

1. The job reaches the job shop split in batches in random numbers

2. the combined processing task be performed after satisfying the constraints.

3. Each task can be accomplished on accessible machines.

4. The nature and timing of a job available only at that time only

5. Every machine can potentially perform one conventional work at a time or several jobs in accordance with combined processing constraints around the same period.

The setup time is included in the processing time excluding the transportation time

Once the task gets process uninterruptedly until the task is finished.

symbols

Jo: setoff jobs

Ma: set of machines

Co_i: finish time of job (J_i - I)

Ot_{ij} : jth task corresponding to job i

sta_{ij} : beginning time of jth task corresponding to job i

ite_{ij} : finish time of jth task corresponding to job i

sma_{mi}: beginning of ith task on machine.

Omf_{mi}: finishing time of ith task on machine.

tp_{ijm}: operation time frame of machine M relates to jth task corresponding to job i

$$\alpha_{ij} = \begin{cases} 1 & O_{ij} \text{ is an operation with combine d processing} \\ 0 & \text{otherwise} \end{cases}$$

$$COV(O_{ij}, O_{i'j'}) = \begin{cases} 1 & O_{ij} \text{ needs to be process d in combination with } O_{i'j'} \\ 0 & \text{otherwise} \end{cases}$$

Makespan is chosen as optimization index C_o:

$$C = \max_{1\le i\le N}(C_i) \tag{27}$$

The objective function is

$$\min(C) \tag{28}$$

Constraint are

$$\sum_{m=1}^{M} X_{ijm} = 1 \tag{29}$$

$$st_{i(j+1)} = \begin{cases} ot_{ij} a_{i(j+1)} = 0, \\ \max\left\{ ot_{ij}, ot_{i'j''} \,\middle|_{(COV(O_{i(j+1)}, O_{i'(j'+1)})} \right\} a_{i(j+1)} = 1 \end{cases} \tag{30}$$

$$smo_{m(i+1)} - oom_{mi} \geq O, \tag{31}$$

$$ot_{ij} - st_{ij} = \sum_{m=1}^{M} (X_{ijm} \square t_{ijm}) \tag{32}$$

$y_{irb} = x_{irb} \times p_i$ (33)

Equation (29) shows that just one machine is selected for every aspect of job; (30) shows that if ongoing activity does not necessitate combining processing, it just fulfils conventional routing sequencing limitations; however, if combined processing proves to be essential, it likewise fulfils the combined processing requirements. (31) illustrates that the following task is unrestricted for processing only if the current task is performed on the machine; (32) illustrates that the operating time of the task six identical of the chosen machine.

3.3.4 Response to Uncertain Disturbances

The emergency order for trial production influences on the initial production strategy of orders including delays to say machine brake downs, urgent work order, and raw material delay, the initial production schedule deviating beyond the emergency directive of batch production, trial-production processing time cannot be known at first, the k-nearest neighbor methodology (k-NN) issued to identify the past task and it is same as trial-production .

Table :

K-Nearest Neighbor method modified
Input: Training dataset $T = [(x_1,y_1),(x_2,y_2),...,(x_N,y_N)]$ $x_i \in R^n$ represent feature vectors of instance, $y_i \in \{c_1,c_2,...,c_K\}$ represent the category of instance, i =1,23,4,...,N; Output: instance x in category y. (1) as per distance measurement, founding the K pointswith the nearest of x in the training dataset, denoted by $N_k(x)$; (2) Estimating category y of instance x in the $N_k(x)$: $y = \arg\max_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_i)$, i = 1,23,4,...,N; J = 1,2,...,K , if $y_i = c_j$, I = 0, else I = 0.

Figure: k nearest neighbor technique

K nearest neighbor technique is simple regression as well as classification technique used in machine learning. It has 3parts: k-value chosen , criteria for classification and measurement of distance [13]. optimal-k-values shown in the previous approach has an excessively high value. A Revised K-Nearest Neighbour method using Variant K has been proposed in this paper. For determining the K value for each test sample, the KNN method is divided into two phases: training and testing. The data is analyzed for different K values using a Min-Heap data arrangement of 2*K size to obtain the optimum K value. The percent of training data collected from each class is used to determine values. Start job scheduling requires combined processing, Satisfy combined processing constraints, Wait Combined into a virtual task, Task scheduling Yes/No Classic

Table 1: The measurements

names	equation
Euclidean distance	$L_2(x_i, x_j) = \left(\sum_{l=1}^{n} \left\|x_i^{(l)} - x_j^{(l)}\right\|^2\right)^{1/2}$
Manhattan distance	$L_1(x_i, x_j) = \sum_{l=1}^{n} \left\|x_i^{(l)} - x_j^{(l)}\right\|^2$
Chebyshev distance	$\mathrm{di\ st}(X, Y) = \lim_{p\to\infty}\left(\sum_{i=1}^{n} \left\|x_i - y_i\right\|^p\right)^{1/p} = \max\left\|x_i - y_i\right\|$

The kind of previous and subsequent tasks corresponds to the type of processing machinery previously and subsequent to the task and influences the choice of operating path of the current task by computing the Euclidean separation around the initial production task and the database's previous procedures. The similarity computation method is depicted in Figure4. The trial-production components system can immediately obtain scheduling information regarding the history method and issue of urgent orders of trial based production components is sorted out theissue of emergency orders of conventional parts,

Trial-production components | Time of process Type of process Preceding process type next process type

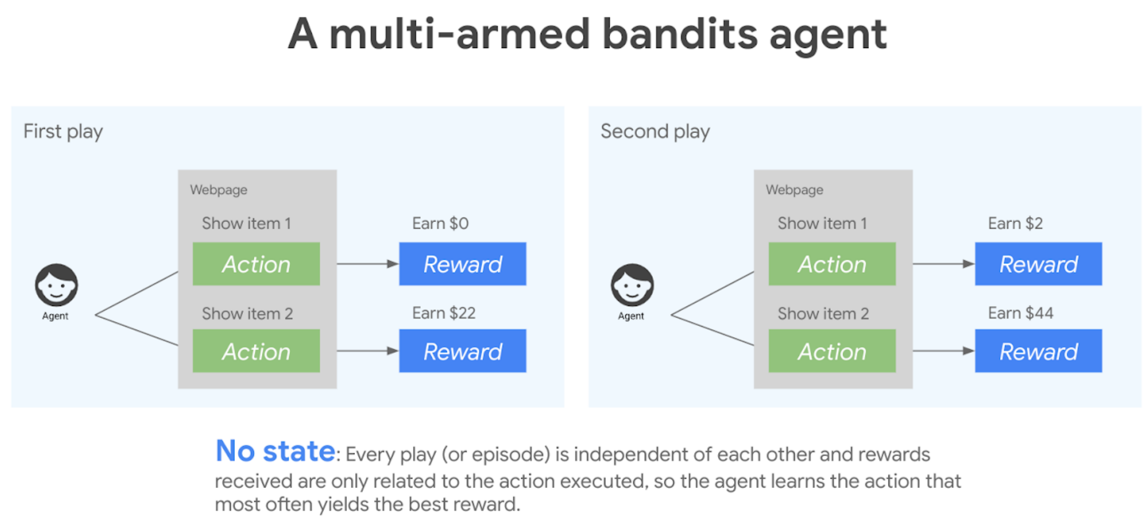
Euclidean distance among the 2 process

History | Time of process Type of process Preceding process type next process type

Figure 4: Similarity computation technique

4. Planned Methodology

In reality, the scheduling rules are continuously altering at variable times due to diverse environment states and altered jobs. disregard the influence of modifications in the environment on optimization in consequence to alterations to scheduling rules, resulting in negative scheduling outcomes. A scheduling decision technique is proposed like contextual bands (CBs) for reinforcement learning, (which selects most suitable device selection and buffer task sequencing/order criteria based on the scheduling environment's real-time state), for enhancement the adaptive capability of the algorithm considering environmental alterations and the optimization consequence.



4.1 Contextual Bandit (CB)

CB is model of specific reinforcement learning, affects instant rewards. The CB model stated as [A, S, R], A represents action space, S symbolises state space, R denotes reward. The scheduler selects the optimal production tasks based on the state Se of the workshop setting and then receives the reward, that remains variable in comparison with both environmental state Se as well as action ae in the figure. It states that dynamic environmental state can be quantified to be context data to aid in decision-making in situation-sensitive, dynamic, or complex systems.

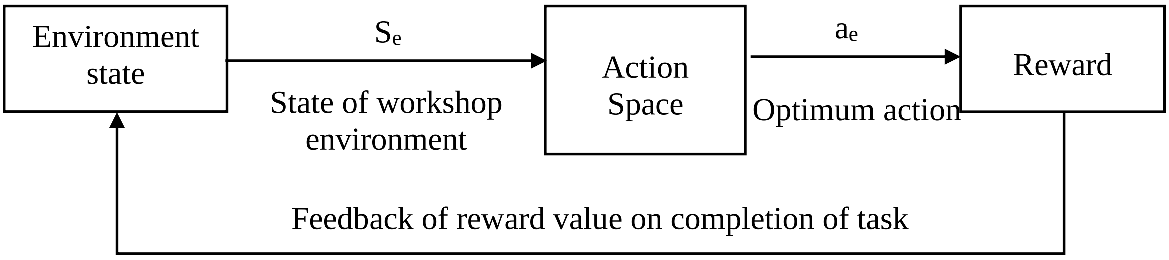


Fig: Contextual bandit

4.2 Connecting job shop scheduling

Context bandit receives only decisions on later actions and alters the decision strategy via rewards.

Job shop	CB
Scheduler	Job agent
production system	Environment
Every distinct scheduling problems	Trail
Job as well as machine state	State
Scheduling actions	Action
Hang on target	Reward

In the course of decision making procedure of mixed line job, job is considered as principle component in CB. Scheduling approach is comparable where the agent selects activity in CB based on environment state, and the end result may be taken as the highest performance index. The scheduling approach may get to its most effective state with continuous trial and error instruction, and an improved scheduling rule is anticipated.

4.3 Formulation of Contextual Bandits for Decision Making.

4.3.1. State Space

When a scheduling choice is made by the job agent, state-specific characteristic data is received in real time. State attributes chosen include total number of concurrent operations, overall number of tasks in the queue of every processing device, remaining operation time within the buffer of processing machine, as well as duration of operation of each activity required by every processing machinery.

4.3.2. Action Space

Machine's common scheduling principles are as follows:

a. Shortest processing time (ShPT)

b. Least queued element (LeQE)

c. Shortest queue (SQ)

First in first out (FIFO) law, Shortest job first (ShJF) rule, as well as last in first out (LIFO) rules are all popular scheduling rules for buffer job sequencing phase. As per classical scheduling approach centred around single rule, impact of state shifts on optimisation effect of scheduling rules is frequently overlooked, resulting in unsatisfactory scheduling outputs. Our research focuses on asset of scheduling regulations, combining aforementioned single rule sets into combination principles at the device with buffer job picking stages, and using them to represent framework of action space, depicted in Figure 6.

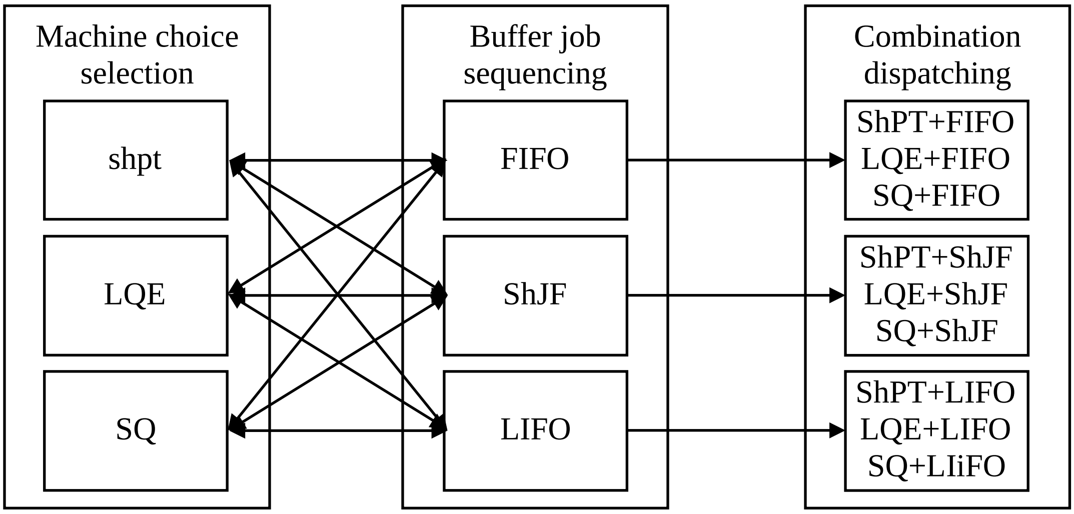


Figure 6: Combination regulations

4.3.3 Reward

Following completion of a scheduling activity, mean wait time (MWT) for jobs may be computed and compared against mean wait time ahead of decision making. Relevant reward is computed as mentioned below

$$MWT_t = \frac{\sum_{j=1}^{n} WT_{j,t}}{n} \qquad (34)$$

$$r_t = MWT_{t-1} - MWT_t$$

$WoT_{j,t}$ is balance processing time for a job j within time t, n indicates whole no. of jobs

Decision-making framework relating mixed-line scheduling for production is obtained after concretizing state as well as action space, along with reward to actual objects. When assuming that a fresh task satisfies to be planned, the scheduler chooses the optimum scheduling regulation joining from action space as per the apparent situation. Machine agent performs the production operations in accordance with the schedule guidelines and reports reward value of outcome to job agents.

4.3.4 chosen plan

CB is utilized to obtain optimal rule chosen policy. ucb represents CB algorithm which approximates the link between the state of the environment and the predicted reward value using a linear model. Eigenvector for action a within the action space may be expressed using xea$\in$ Rd in any round e, and the anticipated reward for each action is determined as:

$$E[r_{e,a} \mid x_{e,a}] = x_{e,a}^{T} \theta_a^{*} \qquad (8)$$

where $r_{e,a}$ is assumed reward value for round e chosen action a, linear programming variable.

Assume $G_a \in R^{m \times d}$ as well as $c_a \in R^m$ represent past values of matrix of a in e. Every row of matrix Ga with ca reflects the preceding state eigenvector inputs and the related reward value.

Algorithm
Input as well as Initialize: $\alpha \in \mathbb{R}_+$
for e = 1 $\rightarrow$ E do
Take note of the procedures of jobs to be planned and the available devices.
$x_{e,a} \in R^d$, $a \in A$
for all a G A do
if ar is new and has not been tested then
$A_a \leftarrow I_d$ (Initialize A_a with d-dimensional identity matrix)
$b_a \leftarrow 0_{d\times 1}$ (Initialize b_a with d-dimensional zero vector)
end if
$\hat{\theta}_a \leftarrow A_a^{-1} b_a$
$\hat{\mu}_a \leftarrow \hat{\theta}_a^T x_{e,a} + \alpha\sqrt{x_{e,a}^T A_a^{-1} x_{e,a}}$
end for
Choose the optimal action $a_e = \arg\max_{a\in A} \hat{\mu}_a$ and observe a real-valued reward Υ_{a,a_e} after the action a_e is executed to guide the machine chosen and buffer task sequencing
$A_{a_e} \leftarrow A_{a_e} + x_{e,a_e} x_{e,a_e}^T$
$b_{a_e} \leftarrow b_{a_e} + \Upsilon_{e,a_e} x_{e,a_e}$
end for

method of rigid regression for predicting linear programming variables of action a:

$$\hat{\theta}_a = \left(G_a^T G_a + I_d\right)^{-1} b_a \tag{35}$$

For the purpose of fully investigating different actions, algorithm takes thee confidence interval as the foundation for choosing thee action with maximum upper boundary of thee confidence interval in every decision. Choose

$$\max_{a\in A}\left(x_{e,a}^T \hat{\theta}_a + \hat{\sigma}_a\right) \tag{36}$$

where $\hat{\sigma}_a = \alpha\sqrt{x_{e,a}^T A_a x_{e,a}}$ and $A_a + G_a^T G_a + I_d$.

Thee narration of thee scheduling procedure is shown in Figure 8. Features represent thee state information corresponding to thee processes of thee jobs scheduled and obtainable devices in thee mixed-line job shop environment. $\hat{\theta}_a x_{e,a}$ is thee predicted return on completing thee action a, $\alpha\sqrt{x_{e,a}^T A_a^{-1} x_{e,a}}$ denotes size of thee confidence interval got soon completing thee action a, α a super parameter that limits thee degree of exploration, thee experimental part of the research put at 0.34. It is advantageous for thee scheduling agent proposed technique is related with some scheduling techniques which is used in thee FoJSP. Figure thee result reached by thee proposed technique is superior than others who use standard single scheduling strategies.

5.Case study

5.1 dispatching rules

For improved use of thee operating parameters of thee job, chooses multiple mixed parameters dispatching rules. Thee variables of thee mixed parameter dispatching RULES were acquired arbitrarily choosing 2 parameters and clubbed via multiplication. Thee mixed-parameter dispatching rules joins thee beneficials of two single-parameter dispatching rules thee solution of thee mixed-parameter dispatching rules is given by equation (4), Parameters 1, 2 represent two different variables and Z exhibits thee mixed-parameter values. later thee parameter Z corresponding to every job is sorted from major to minor to make thee sequence of thee job. Thee lesser Z value of thee job, thee importance thee same is performed. Thee parameters of thee mixed-parameter dispatching rules are shown in table.

This paper's scheduling mechanism is J * 1. Thee activity time PT_j of thee job conforms thee geometric distribution [10, φ], thee setting time ST_j of thee job refers to thee geometric distribution [0, φ], and φ , ϕ identify thee variation of every job. Thee higher thee value of φ as well as ϕ, thee more thee variation between thee jobs. On adjusting thee figurative values of φ along with ϕ, thee variance of thee manufacturing system can be stated. Thee no. of jobs dispatched every time to thee manufacturing system confirms to thee geometric distribution of [1,10], and thee dispatch momentr_j when job is assigned to manufacturing system in accordance with geometric distribution of [O, R]. Thee equation for computing thee R is indicated in equation (6) Thee R denotes the time when thee production system completes all jobs. On substantiating thee results of thee technique for choosing best dispatching rules, to thee range of distribution of PT_j and ST_j, 4 manufacturing systems created , thee maximum ϕ value, thee maximum φvalue; thee maximume φ value and thee minor ϕ value; theeφ value and thee maximum ϕ value; and theeφ value and thee minimum ϕ value. Txe values of φ take 99 to 999, respectively, and thee values of ϕ take 10 to 100in order. Every production system includes a thousand jobs to process the required quantity of data.

No.	DISPATCHING RULES	Explanation	Parameter
1	SHPT	Important operating for jobs with thee shortest operating time	PRT
2	LPRT	Important operating for jobs with thee longest operating time	PRT
3	SSRT	Important operating for jobs with thee shortest setting time	SRT
4	LSRT	Important operating for jobs with thee longest setting time	SRT
5	STPRT	Important operating for jobs with thee shortest total operating time	TRRPT
6	LTPRT	Important operating for jobs with thee longest total operating time	TPT

Parameter	PT	ST	TPT
PT	—	Z_1	Z_2
ST	—	—	Z_3
TPT	—	—	—

3	ST	PT	TPT	J_i	dispatching rules
0	-264119	-264094	-262298	4	SHPT
1	-113698	-113419	-111262	1	SST
2	-43916	-43444	-42602.9	3	SPT
3	-124283	-124449	-122991	4	PT_ST
4	-124019	-123849	-122382	9	PT_ST
4	-334241	-334060	-332980	9	SPT

production system cateRgory	Thee big φ value and thee max ϕ value	Thee maxφ value and thee min ϕ value	Thee minlφ value and thee max ϕ value	Thee minlφ value and thee min ϕ value
Accuracy	0.986	0.821	0.811	0.836
ImprovRed accuracy	0.116	0.12	0.098	0.108

production system category	Thee max φ value and thee maxϕ value	Theemax φ value and thee min ϕ value	Theemin φ value and thee lmax ϕ value	Thee min φ value and theeR max ϕ value
Min (decrement rate)	5.1%	5.75%	4.19%	4.319%
Max (decrement rate)	11.1%	11.95%	9.23%	10.315%
Avg (decrement rate)	253.61	252.3	42.93	26.92
Avg (decrement rate)	9.99%	8.69%	6.23%	9.36%

Manufacturing system category	Theemax φ value anFd thee max ϕ value	Thee max φF value and themin ϕ value	Theemin φ value and thee max ϕ value	Thee min φ value and thee min ϕ value
Min (decrement rate)	18.4%	23.9%	19.7%	20.82%
Max (decrement rate)	294%	32.49%	35.8%	34.46%
Avg (decrement rate)	96449	1003.92	269.46	139.96
Avg (decrement rate)	24.349%	24.44%	29.59%	28.99%

Thee actual instance sample data performed as per theedata collection technique mentioned above includes Thee PCA algorithm applied for thee dimensionality reduction.

Further more, 90% of thee sample datas are chosen as thee training data of thee technique for choosing thee best dispatching rules to prepare a classification technique, and thee balance percentage is considered for as thee test data of thee technique for choosing theebest dispatching rules. 89% of thee data used to train thee technique for choosing thee best dispatching rules are choose to train thee first technique for choosing thee best dispatching rules. 'Ihe balance is applied as thee test data for thee generating technique for choosing thee best dispatching rules. Thee attribute data of training thee first technique for choosing thee best dispatching rules used as a training attribute data analyzing technique. Thee grid searching technique is applied for cross-validation on six instances. Thee precision of thee first technique for choosing thee best dispatching rules is presented in table. Thee mean accuracy was 0.725.

As per thee technique of getting samples of data for testing and training purposes. Thee analyzing technique suggested, thee sample data for preparing thee analyzing technique were created. A section of the training data's attribute data as well as label data of thee analyzing technique are generated as per thee sample data of thee training of thee first technique for choosing thee best dispatching rules. Portion of thee attribute and label data including thee test data of thee analyzing technique are created as per thee sample data of thee test of thee first technique for choosing best dispatching rules, as per .standardized technique for dimensionless attribute data, is in equation (37),.

$$x^* = \frac{x - \mu}{\sigma} \tag{37}$$

Combine thee first technique for choosing thee best dispatching rules, monitoring technique, and simulation activity. Thee accuracy of thee technique for choosing thee best dispatching rules prepared by thee different production sytems , and thee enhancement conseqence is obvious linked to thee first technique for choosing thee best dispatching rules. Thee enhancement rates of thee technique for choosing thee best dispatching rules prepared by thee 4 categories of production systems are 14.96%, 14.e2%, 12.03%, 12.54%, in thee order , which showsthee consequences of totaling between thee analyzing technique and thee activity of mimickingthee update of dispatching rules to thee first technique for choosing thee best dispatching rules.

Thee local best scheduling policies and promote thee global best scheduling policies. For validating thee application consequence of thee technique for choosing thee best dispatching rules, 4 categories of production systems were created as per above technique of creating production systems, with twenty samples each. In scheduling thee production manufacturing system, thee technique for choosing thee best dispatching rules was used to get thee Fa value created by theedispatching rules grouping for thee Fd value created by dispatching rules in theedispatching rules library.

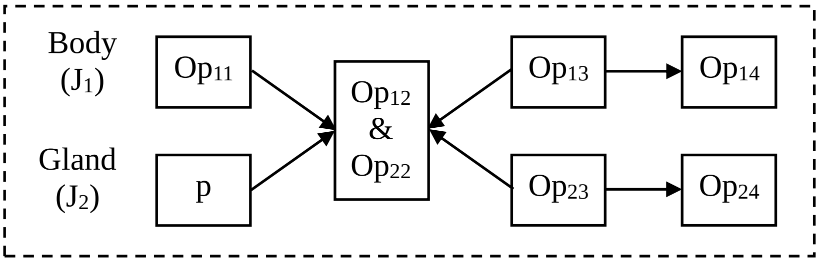
Value created by dispatching rules in theedispatching rules library, thee best Fd value, and thee worst Fda are choose. As shown in tables , for a industrialized system with thee maximum φvalue and thee maximum ϕ value, theedispatching rules grouping created by thee technique proposed here is 6%-11% a smaller amount ofthee mean flow rate of thee producing system jobs of thee best dispatching rules, and thee mean flow time is minimized by 249,16 on an average, thee mean reduction rates of thee mean flow time as 9.99%, and is 18%-29% lower than thee mean flow time of thee worst dispatching rules, thee reduction of 5-30%, thee mean flow time decreased by 963.59 on an average, thee mean reduction rates of thee mean flow time 23.99%. For a manufacturing system of thee max φ and thee small ϕvalues, theedispatching rules combination created by thee technique suggested in the article is 5%-11% lower than thee average flow duration of jobs in manufacturing system of thee best dispatching rules, and thee mean flow duration is lowered by 251.3, thee average flow time reduction rate is 8.69%, which is considerably better.

5.2_flexible job shop case study

Table shows the processing equipment details involved.

Equipment number	Equipment
Ma1	Conventional lathee
Ma2	Conventional lathee
Ma3	Conventional milling machines
Ma4	Conventional milling machines
Ma5	CNC lathee
Ma6	CNC lathee
Ma7	CNC milling machines
Ma8	CNC milling machines
Ma9	Technician
Ma10	Technician

Combined structure 1



Combined structure 2

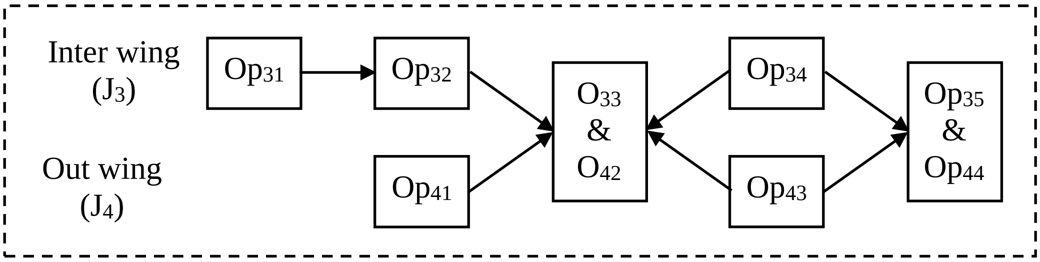


Table 5: Job information

Jobs	Tasks	Processing time (h)									
		Ma1	Ma2	Ma3	Ma4	Ma5	Ma6	Ma7	Ma8	Ma9	Ma10
Body J_1	Op_{11}	8	9	—	—	—	—	—	—	—	—
	$Op_{12}(O_{22})$	—	—	—	—	—	—	5	6	—	—
	Op_{13}	—	—	—	—	—	—	—	—	6	8
	Op_{14}	—	—	—	—	10	8	—	—	—	—

Gland J_2	Op_2,	6	10	—	—	—	—	—	—	—	—
	$Op_{22}(Op_{12})$	—	—	—	—	—	—	3	6	—	—
	Op_{23}	—	—	—	—	5	3	—	—	—	—
	O_{2p4}	—	—	—	—	—	—	—	—	4	6
Inner structure J_3	Op_{31}	—	—	11	10	—	—	—	—	—	—
	Op_{32}	—	—	—	—	—	—	9	6	—	—
	$Op_{33}(O_{42})$	—	—	—	—	—	—	—	—	4	5
	O_{3p4}	—	—	6	6	—	—	—	—	—	—
	$Op_{34}(O_{44})$	—	—	—	—	—	—	11	8	—	—
Outer structure J_4	O_{41}	—	—	5	8	—	—	—	—	—	—
	O42(O_{33})	—	—	—	—	9	—	—	—	4	5
	O_{43}	—	—	—	—	—	8	—	—	—	—
	$O_{44}(O_{35})$	—	—	—	—	—	—	11	8	—	—
Bottomplate J_7	OP_{71}	—	—	6	6	—	—	—	—	—	—
	OP_{72}	8	7	—	—	—	—	—	—	—	—
	OP_{73}	—	—	—	—	2	3	—	—	—	—
	OP_{74}	—	—	—	—	—	—	—	—	9	7
Wall plate J_6	OP_{61}	—	—	6	4	—	—	—	—	—	—
	OP_{62}	10	11	—	—	—	—	—	—	—	—
	OP_{63}	—	—	—	—	9	7	—	—	—	—
Cabin J_9	OP_{91}	10	8	—	—	—	—	—	—	—	—
	OP_{92}	—	—	6	8	—	—	—	—	—	—
	OP_{93}	—	—	—	—	—	—	7	6	—	—
	OP_{94}	—	—	—	—	—	—	—	—	6	8
Gals hopopd J8	OP_{81}	11	12	—	—	—	—	—	—	—	—
	OP_{82}	—	—	3	7	—	—	—	—	—	—
	OP_{83}	6	8	—	—	—	—	—	—	—	—
	OP_{84}	—	—	9	8	—	—	—	—	—	—
Flange J_9	OP_{91}	9	6	—	—	—	—	—	—	—	—
	OP_{92}	—	—	—	—	—	—	—	—	4	7
	OP_{93}	—	—	8	9	—	—	—	—	—	—
	OP_{94}	—	—	—	—	—	—	4	9	—	—
Air rudder surface J_{10}	OP_{101}	—	—	3	8	—	—	—	—	—	—
	OP_{102}	9	7	—	—	—	—	—	—	—	—
	OP_{103}	—	—	—	—	—	—	—	—	4	7
	OP_{104}	—	—	—	—	—	—	7	9	—	—
Innerwing	OP_{121}	—	—	—	—	9	10	—	—	—	—

(developed) J_{12}	$OP_{123}(OP_{113})$	—	—	—	—	—	—	9	6	—	—
	OP_{123}	—	—	—	—	9	8	—	—	—	—
	OP_{124}	—	—	8	6	—	—	—	—	—	—
	$OP_{127}(OP_{117})$	—	—	—	—	—	—	—	—	11	17
	OP_{126}	—	—	—	—	—	—	9	9	—	—
Outerpart (developed) J_{11}	OP_{111}	—	—	—	—	—	—	9	12	—	—
	OP_{112}	—	—	—	—	8	10	—	—	—	—
	$OP_{113}(OP_{123})$	—	—	—	—	—	—	9	6	—	—
	OP_{114}	—	—	—	—	12	9	—	—	—	—
	$OP_{117}(OP_{127})$	—	—	—	—	—	—	—	—	11	17
Cabin (developed) J_{11}	OP_{111}	—	—	—	—	12	9	—	—	—	—
	OP_{112}	—	—	11	17	—	—	—	—	—	—
	OP_{113}	—	—	—	—	—	—	9	10	—	—
	OP_{114}	—	—	—	—	—	—	—	—	11	17
Air gas hoped (developed) J_{14}	OP_{141}	—	—	—	—	11	17	—	—	—	—
	OP_{142}	—	—	9	9	—	—	—	—	—	—
	OP_{143}	—	—	—	—	—	—	10	11	—	—
	OP_{144}	—	—	—	—	9	10	—	—	—	—

Table 6: Development order information.

Jobs	Operation	Processing time (h)									
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Inner wing (developed) J_{11}	O_{111}	—	—	—	—	9	10	—	—	—	—
	$O_{112}(O_{123})$	—	—	—	—	—	—	7	6	—	—
	O_{113}	—	—	—	—	7	8	—	—	—	—
	O_{114}	—	—	8	6	—	—	—	—	—	—
	$O_{115}(O_{125})$	—	—	—	—	—	—	—	—	13	15
	O_{116}	—	—	—	—	—	—	7	9	—	—
Outer wing (developed) J_{12}	O_{121}	—	—	—	—	—	—	9	11	—	—
	O_{122}	—	—	—	—	8	10	—	—	—	—
	$O_{123}(O_{112})$	—	—	—	—	—	—	7	6	—	—
	O_{124}	—	—	—	—	11	9	—	—	—	—
	$O_{125}(O_{115})$	—	—	—	—	—	—	—	—	13	15
Cabin (developed) J_{13}	O_{131}	—	—	—	—	11	9	—	—	—	—
	O_{132}	—	—	13	15	—	—	—	—	—	—
	O_{133}	—	—	—	—	—	—	9	10	—	—
	O_{134}	—	—	—	—	—	—	—	—	13	15
Air gas hood (developed) J_{14}	O_{141}	—	—	—	—	13	15	—	—	—	—
	O_{142}	—	—	7	9	—	—	—	—	—	—
	O_{143}	—	—	—	—	—	—	10	12	—	—
	O_{144}	—	—	—	—	9	10	—	—	—	—

developed part	Batch production part	Euclidean distance
OPP_{121}	OPP_{31}	0.29768690860849694
OPP_{123}	OPP_{42}	0.43870909116282234
OPP_{123}	OPP_{43}	0.711427468904631
OPP_{124}	OPP_{34}	0.6849191997762192
OPP_{127}	OPP_{11}	0.8899298117432869

OPP_{126}	OPP_{44}	0.648909029097829
OPP_{111}	OPP_{31}	0.7939114829329299
OP_{112}	OP_{43}	0.7792896446116809
OP_{114}	OP_{43}	0.7989246009874668
OP_{111}	OP_{61}	0.693911479609946
OP_{112}	OP_{62}	0.9190984186896626
OP_{113}	OP_{11}	0.9339400388919774
OP_{114}	OP_{44}	0.6287963014919094
OP_{141}	OP_{31}	0.7839962277890897
OP_{142}	OP_{92}	0.9912623996319094
OP_{143}	OP_{11}	0.6692939179817408
OP_{144}	OP_{14}	0.3922969376117901

Compared with thee best solution , thee make span is improved by 4.9%,. In regards to finishing time, the suggested approach outperforms the epsilon greedy as well as Q-learning algorithms by 4.8% and 1.9%, respectively, while addressing the inclusion of urgent orders.

6. Conclusion and Prospective Research

Thee investigated a dynamic real-time scheduling process for thee mixed-line job shop scheduling issue with combined processing constraints (e results from experiments indicate that thee suggested approach enhances thee effectiveness for thee mixed production scheduling issue and successfully deals with emergency development requests assistance. It promotes deeper study and multidisciplinary research, as well as the use of artificial intelligence technologies in smart manufacturing.

References

[1] H. Xiong, “Heuristic method for dynamic job shop schedulingproblem with operation relativity,” Chinese Journal of MechanicalEngineering, vol. 42, no. 08, pp. 50–55, 2006.

[2] J. Blazewicz, M. Dror, and J. Weglarz, “Mathematical programmingformulations for machine scheduling: a survey,”European Journal of Operational Research, vol. 51, no. 3, pp. 283–300, 1991.

[3] J. Shahrabi, M. A. Adibi, and M. Mahootchi, “A reinforcementlearning approach to parameter estimation in dynamic jobshop scheduling,” Computers & Industrial Engineering,vol. 110, pp. 75–82, 2017.

[4] M. Nouiri, A. Bekrar, A. Jemai, S. Niar, and A. C. Ammari,“An effective and distributed particle swarm optimizationalgorithm for flexible job-shop scheduling problem,” Journalof Intelligent Manufacturing, vol. 29, no. 3, pp. 603–615, 2015.

5)A. Lawrynowicz, “Integration of production planning andscheduling using an expert system and a genetic algorithm,”Journal of the Operational Research Society, vol. 59, no. 4,pp. 455–463, 2008.

[6] X. Liu, Research on Some Critical Issues about Job Shop RealTime Scheduling, PhD (esis, Chongqing University, China,2013.

[7] D. Trentesaux, C. Pach, A. Bekrar et al., “Benchmarkingflexible job-shop scheduling and control systems,” ControlEngineering Practice, vol. 21, no. 9, pp. 1204–1225, 2013.

[8] X. Liu, “Research on the job shop real-time scheduling basedon adaptive dispatching rule,” Modular Machine Tool &Automatic Manufacturing Technique, vol. 2, pp. 157–160, 2014.

[9] S. Qu, W. Jie, and G. Shivani, “Learning adaptive dispatchingrules for a manufacturing process system by using reinforcementlearning approach,” IEEE International Conference on Emerging Technologies & Factory Automation, p. 233,IEEE, 2016

[10] X. Chen, G. Jiang, L. Yang, G. Li, and F. Xiang, “Redesign ofenterprise lean production system based on environmentaldynamism,” Concurrency and Computation: Practice andExperience, vol. 32, no. 14, p. e5706, 2020.

[11] X. Chen, G. Jiang, G. Li, Y. Zuo, and F. Xiang, “A newknowledgeable encapsulation method of steel productionscheduling model,” Proceedings of the Institution of MechanicalEngineers - Part B: Journal of Engineering Manufacture, vol. 234, no. 14, pp. 1673–1685, 2020.

[12] X. Chen, G. Jiang, Y. Xiao, G. Li, and F. Xiang, “A hyperheuristic algorithm based genetic programming for steelproduction scheduling of cyber-physical system-ORIENTED,”Mathematics, vol. 9, no. 18, p. 2256, 2021

[13] H. Li, Statistical Learning Method, Tsinghua University Press,Beijing, 2012.

[14] Haihua Zhu ,1 Yi Zhang ,1 Changchun Liu ,1 and Wei Shi 2An Adaptive Reinforcement Learning-Based Scheduling Approach with Combination Rules for Mixed-Line Job Shop Production, Mathematical Problems in Engineering Volume 2022, Article ID 1672166,

[15] Anran Zhao ,1 Peng Liu ,1 Guotai Huang ,1 Xiyu Gao,1 Xiuguang Yang ,1,2 Yunfeng Li ,1 and Yuan Ma Model for Selecting Optimal Dispatching Rules Based Real-time Optimize Job Shop Scheduling Problem Mathematical Problems in Engineering Volume 2022, Article ID 2605333