International Journal of Mathematical, Engineering and Management Sciences
Vol. 10, No. 3, 754-776, 2025
https://doi.org/10.33889/IJMEMS.2024.10.3.038

Ram Arti Publishers

Development of Dispatching Rule based Heuristic Algorithms for Real-Time Dynamic Scheduling of Non-identical Parallel Burn-in Ovens with Machine Eligibility Restriction

Dhaval Pujara
Brij Disa Centre for Data Science and Artificial Intelligence,
Indian Institute of Management, Ahmedabad, 380015, Gujarat, India.
E-mail: dhavalp@iima.ac.in

M. Mathirajan
Department of Management Studies,
Indian Institute of Science, Bangalore, 560012, Karnataka, India.
Corresponding author: msdmathi@iisc.ac.in

(Received on July 20, 2024; Revised on September 3, 2024; Accepted on September 22, 2024)

Abstract
This study addresses a new problem, close to real life on scheduling of non-identical parallel Burn-in ovens (BO) in Semiconductor Manufacturing (SM) industry to minimize Total Weighted Tardiness (TWT) of the jobs. Due to computational intractability in using mathematical programming approach, many researchers considered Dispatching Rules (DR) based heuristic algorithm in SM industry for scheduling of various batch processing machines, including BO. However, there is no study that compares the various DR considered for scheduling of BO in general, particularly in (a) Dynamic Scheduling (DS) of Non-identical Parallel Burn-in Ovens (NPBO) with Machine Eligibility Restriction (MER) and (b) Real-Time DS (RTDS) of NPBO with MER. To address this research gap, this study proposed 25 variants of DR based Heuristic Algorithm (DR-HA). Empirical and statistical performance analyses, carried out with 240 test data, revealed that variants of DR-HA that outperform during DS-NPBO with MER also outperform in RTDS-NPBO with MER. Furthermore, this study gives an important inference that whenever any RTE occurs in RTDS-NPBO, there is no need to modify existing efficient algorithm or no need to develop new algorithm for RTDS-NPBO and updating input data related to type of occurred RTE is sufficient before running existing efficient DR-HA for DS-NPBO.

Keywords- Scheduling, Burn-in ovens, Dispatching rules, Total weighted tardiness, Real-time events.

1. Introduction

In discrete product manufacturing process, the products are produced either by serial or batch processing. In serial processing, jobs are processed in series, i.e., only one job is produced at a time, whereas in batch processing, multiple jobs are processed simultaneously in a batch by a machine called Batch Processing Machine (BPM) (Mathirajan et al., 2010). The scheduling of BPMs is more complex than scheduling of discrete machine as it requires three interrelated decisions: BPM selection, batch formation, and batch sequencing to be taken efficiently. From the various BPMs mentioned in **Table 1**, this study aims to schedule Burn-in Oven (BO), a BPM in the testing stage of Semiconductor Manufacturing (SM) industry to perform the Burn-in operation. In Burn-in operation, jobs (Integrated Circuits - ICs) are subjected to high thermal and electrical stresses over a prolonged time (up to 240 hours) for the reliability test (Uzsoy, 1994). Based on the customer requirement, each job requires different processing time in Burn-in operation. Due to high processing time, Burn-in operation acts as a bottleneck operation in the testing stage of SM process. Hence, developing efficient methodologies for scheduling of BO can improve the overall production rate of the SM process (Mönch et al., 2011).

BO contains multiple parallel boards to hold the jobs and the capacity of BO is specified by the number of boards it contains. A job available for burn-in operation contains multiple chips (ICs) of a single product and it occupies some space [in terms of number of boards] in BO. Hence, the size of a job is determined by the number of boards it occupies in BO. BO can process different types of jobs in a single batch. Hence, multiple jobs are included in a batch till the capacity of BO is utilized up to the maximum possible extent. However, the processing time of batch is decided based on the longest processing time of the jobs included in the same batch.

Most of the earlier studies on scheduling of BO consider static scheduling scenario where the jobs already available in the system are only considered while scheduling the Burn-in operation. However, in practice, along with the jobs available in the system, jobs that are expected to arrive in the future from the upstream operation for the burn-in operation should also be considered while scheduling the Burn-in operation, which is called a Dynamic Scheduling (DS) in the literature. Also, in general, SM shop floor contains multiple parallel Burn-in ovens with different capacities and unique technical specifications, i.e., Non-identical Parallel Burn-in Ovens (NPBO). Thus, this study considers real-life scenario on DS-NPBO with Machine Eligibility Restriction (MER).

Since Burn-in operation is the last operation in SM process, delay in Burn-in operation does not provide enough opportunities to cover up and can cause late delivery to customer, which affects the on-time delivery performance such as tardy jobs, lower customer satisfaction, loss of goodwill, etc. To avoid these situations and considering the importance of on-time delivery performance, this study considers Total Weighted Tardiness (TWT) scheduling objective, a due date based objective, which penalizes late jobs with the lateness penalty and improves the customer satisfaction by delivering most of the jobs on or before its due-date.

The problem on scheduling of a single Burn-in oven to minimize TWT is proved to be NP-hard in the literature (Brucker et al., 1998). Hence, the problem on DS-NPBO, a super set to scheduling of a single Burn-in oven, is considered as NP-hard. Therefore, mathematical programming approach is computationally intractable in solving real-life large-scale problems on DS-NPBO. Hence, this study considers various dispatching rules (DR) used in the SM industry for scheduling of BPM and proposes multiple variants of DR based Heuristic Algorithm (DR-HA) for DS-NPBO, as dispatching rules are widely used in scheduling various operations in SM industry.

In practical environment, different kinds of unanticipated events (called as Real-Time Events [RTE]) related to jobs and burn-in oven occur while operating Burn-in oven. Therefore, apart from the future arrival of jobs, the occurrence of RTE must be considered in dynamic scheduling scenario, which is called a Real Time Dynamic Scheduling (RTDS). Accordingly, this study triggers randomly the occurrences of RTEs while performing RTDS-NPBO using the developed DR-HA, evaluates the robustness of the proposed variants of DR-HA in terms of the performance ranking and identifies the efficient variants of DR-HA for RTDS-NPBO environment.

This article is structured as follows: problem statement and the underlying assumptions of this research article are provided in Section 2. Section 3 highlights the relevant previous literatures. Section 4 deals with development of 25 variants of DR-HA for DS-NPBO. Later, the performance of 25 variants of DR-HA is evaluated empirically and statistically in Section 5 and identified the efficient variant(s) of DR-HA for DS-NPBO. Section 6 deals with the study on impact of RTE on the performance of each of the 25 variants of DR-HA and identification of the efficient variant(s) of DR-HA for RTDS-NPBO. Finally, a summary and future research scope of the current study are discussed in Section 7.

Table 1. Sample reference studies related to BPM in discrete parts manufacturing industries.

Batch operation / Processor	Industry	Sample reference study
Oven for Hardening Synthetic Parts	Aircraft Industry	Zee et al. (2001)
Heat-Treatment Furnace (HTF)	Automobile Industry	Mathirajan et al. (2014)
Batch Distillation Process	Chemical Industry	Tang and Yan (2009)
Dyeing Machine	Clothing Industry	Zhang et al. (2017)
Stress Screening Chamber	Electronics Industry	Alipour et al. (2020)
Dry Kiln	Furniture Manufacturing	Marier et al. (2021)
Annealing Kiln	Glass Container Industry	Fachini et al. (2016)
Tissue Processors	Hospital laboratory	Leeftink et al. (2020)
Washer for Reusable Medical Devices	Hospital Sterilization Service	Tsai and Chou (2016)
IP Machine used in Clock Industry	Ion Plating (IP) Industry	Chan et al. (2007)
Heat-Treatment Furnace	Metalworking Industry	Dupont and Flipo (2002)
Bake-out / Box-Oven	Ceramic Industry	Koh et al. (2005)
Pottery Kiln	Pottery Manufacturing	Jia et al. (2020)
Diffusion Furnace	Semiconductor Manufacturing	Rani and Mathirajan (2020)
Burn-in Oven	Semiconductor Manufacturing	Keshavarz (2021)
Hole Punching	Sheet Metal Industry	Oulamara (2007)
Carousel	Shoe Manufacturing Factory	Fanti et al. (1996)
Heat-Treatment Furnace	Steel Casting Industry	Ravindra and Mathirajan (2014)
Soaking Bit Furnace	Steel Ingot Production	Li et al. (2011)
Heat-Treatment Furnace	Steel Manufacturing	Zheng and Li (2009)
Aging Test Operation	LCD Screen Manufacturing	Chung et al. (2009)
Tyre Curing / Mold	Tyre Manufacturing	Bellanger and Oulamara (2009)

Source: Rani and Mathirajan (2020)

2. Problem Description and Assumptions

There are n jobs to be scheduled among the m non-identical parallel Burn-in ovens (BO). Each BO has finite Batch Capacity (BC_m) and unique technical specifications. Due to that, certain machines may be not eligible to process all job types, i.e., some jobs may require specific BO for the operation, which is called the Machine Eligibility Restriction (*MER*) in scheduling environment. Every job 'j' is characterised by distinct - release date (r_j), processing time (p_j), due date (d_j), size (s_j), and lateness penalty (l_j). Also, job release dates and due dates are non-agreeable [i.e., for any two jobs Job-A & Job-B: $r_A \leq r_B$ not-implied $d_A \leq d_B$].

Due to very long processing time required for burn-in operation, (a) if any job related RTE (such as change in job- due date, release time, lateness penalty, new job addition, job cancellation) occurs then the data related to job-characteristic will be updated appropriately before making the decision on constructing a batch for scheduling on BO, (b) if any RTE related to resource (such as BO breakdown, operator illness, tool failure, shortage of material, defective material) occurs then the data on next availability time of BO for scheduling will be updated appropriately before making the decision on a choice of a BO for scheduling, and (c) if RTE related to job and RTE related to resource occur simultaneously, then all the relevant input data will be updated w.r.t. the type of the RTE before constructing a batch and before choosing a BO for scheduling.

With the above description on RTDS-NPBO problem considered in this study, the following assumptions are made for developing a scheduling algorithm:

- Each job is independent and all related data are deterministic and known beforehand.
- The size of each job is smaller than the capacity of the BO that has the least capacity among the non-identical burn-in ovens considered for scheduling (that is, job splitting is not permitted while constructing a batch for scheduling BO).
- Burn-in operation cannot be interrupted once the batch operation begins.

As per the standard three-field notation scheme developed by (Graham et al., 1979), the current research problem on *RTDS-NPBO* can be represented as: P_m | *p-batch*, r_j, p_j, d_j, s_j, l_j, *non-agreeable release times and due-dates, real time events, machine eligibility restriction* | *TWT*.

3. A Closely Related Work

In scheduling literature, (Ikura and Gimple, 1986) were the first to report a study on scheduling of BPM. (Mathirajan and Sivakumar, 2006) performed a detailed survey of the studies related to scheduling of BPM in SM process. As this study deals with scheduling of parallel BO to minimize a due-date based scheduling objective, only the relevant studies are identified from the literature and the same are summarised based on the various problem parameters in **Table 2**. From this table, this study observed that none of the existing studies in the literature has performed both DS-NBPO and RTDS-NPBO with considering the machine eligibility restriction and non-agreeable release time and due date scenario to optimize due date based objective. Hence, to fulfil this literature gap, the current study proposes a heuristic algorithm to solve the problem on DS-NPBO and RTDS-NPBO, described in the previous section.

Further, based on the problem-solving approach (methodology), the studies mentioned in **Table 2** are broadly classified into two classes: (a) exact approach (mathematical model) and (b) simple heuristic and / or meta-heuristic algorithms. Ventura and Kim (2000, 2003), Xu and Bean (2015), Hulett et al. (2017) and Pujara and Mathirajan (2020) proposed mathematical model to address parallel BO scheduling problem with different problem configurations. Ventura and Kim (2000, 2003) schedule identical BO in static and dynamic scheduling environments, respectively. Hulett et al. (2017), Pujara and Mathirajan, (2020) and Xu and Bean (2015) perform scheduling of non-identical BO with considering lateness penalty parameter. Further, as per **Table 2**, except for Pujara and Mathirajan (2020), no study has considered the machine eligibility restriction.

Due to NP-hard nature of the problem, all the studies in **Table 2**, except for Pujara and Mathirajan (2020), have considered the development of simple heuristic or meta-heuristic algorithms for addressing the problem on scheduling of parallel BO. Further, it is observed that most of the simple heuristic-based studies use dispatching rules (DRs) to perform batching and sequencing tasks in scheduling of BO. DRs are widely used in scheduling of BPM in SM industry as they (a) are easy to implement and take less computational time, (b) perform effectively across a broad spectrum of machine environments, and (c) can be easily modified to incorporate any dynamic changes (Nguyen et al., 2013). Despite the significant benefits and wide applications of DRs in SM industry, it is observed that no study in the literature has conducted the performance analysis of existing DRs, particularly in the context of scheduling of burn-in ovens with problem configuration considered in this study. Hence, to identify the efficient DRs for DS-NPBO, this study considers various DRs used in studies related to scheduling of BO (Lee et al., 1992; Kim et al., 2011; Chou and Wang, 2012; Li and Chen, 2014; Li et al., 2019). In addition, this study also considers the DRs used in the literature for scheduling of other types of BPM such as E-beam writer (Hung, 1998), Etching tank (Sung and Kim, 2002), diffusion furnace (Rani, 2018) in the SM industry, as mentioned in **Table 3**.

Thus, by considering all the DRs mentioned in **Table 3**, this study proposes 25 distinct dispatching-rule based heuristic algorithm (DR-HA) for DS-NPBO and development of the same is discussed in the next section.

4. Development of Dispatching Rule based Heuristic Algorithm (DR-HA)

There are three interrelated decisions: BO selection, batch construction, and batch selection involved at every decision making time during DS-NPBO. Accordingly, a decision making framework for DS-NPBO is presented in **Figure 1**. Further, a step-by-step procedure of DR-HA for implementing the decision making framework presented in **Figure 1** is as follows:

Table 2. A summary of the studies on parallel Burn-in ovens scheduling problem to minimize the due-date based scheduling objective.

Study	Parallel burn-in ovens		Nature of scheduling		Release times & due-dates		Machine eligibility restriction (MER)	Real-time events (RTE)	Weightage	Solution methodology		Objective function
	I	NI	ST	DY	A	NA				Exact approach	H and/or MH	
Lee et al. (1992)	√			√	√						√	L_{max}, #TJ
Ventura and Kim (2000)	√		√		√					√	√	TE/L
Ventura and Kim (2003)	√			√		√				√	√	TE/L
Mönch and Unbehaun (2007)	√		√		√						√	TE/L
Condotta et al. (2010)	√			√		√					√	L_{max}
Chou and Wang (2012)		√		√		√			√		√	TWT
Xu and Bean (2015)		√		√		√			√	√	√	TWT
Hulett et al. (2017)		√	√		√				√	√	√	TWT
Pujara and Mathirajan (2020)		√		√		√	√		√	√		TWE/L
This Study		√		√		√	√	√	√		√	**TWT**

I- Identical, **NI**- Non-identical, **ST**- Static, **DY**- Dynamic, **A** – Agreeable, **NA**- Non-Agreeable, **H** – Heuristic, **MH**- Meta-heuristic, $\mathbf{L_{max}}$ - Maximum Lateness, **TE/L**- Total Earliness/Lateness, **TWE/L**- Total Weighted Earliness/Lateness, **#TJ**- Number of Tardy Jobs, **TWT**- Total Weighted Tardiness

Table 3. A summary of various dispatching rule used by various studies on scheduling of BPM in SM process with due-date based scheduling objectives.

Dispatching rule	Sample studies
Earliest Due-Date (EDD)	Lee et al. (1992), Hung (1998), Chou and Wang (2012), Rani (2018), Li et al. (2019)
Longest Processing Time (LPT)	Lee et al. (1992), Mönch and Unbehaun (2007), Li and Chen (2014), Parsa et al. (2017), Li et al. (2019)
Shortest Processing Time (SPT)	Sung and Kim (2002), Mönch and Unbehaun (2007), Parsa et al. (2017)
Earliest Release Date (ERD)	Chou and Wang (2012), Li et al. (2013)
Flow Due-Date (FDD)	Jayamohan and Rajendran (2000), Rani (2018)
Operational Due-Date (ODD)	Khalouli et al. (2010), Rani (2018)
Modified Operational Due-Date (MOD)	Rani (2018)
Critical Ratio (CR)	Rose (2002), Rani (2018)
Minimum Slackness first (MS)	Rani (2018)
Minimum Modified Due-Date (MMDD)	Mazzini and Armentano (2001), Kim et al. (2011)
Minimum Modified Slackness (MMS)	Pearn et al. (2002), Kim et al. (2011)
Largest Weight first (LW)	Li and Chen (2014), Chen et al. (2016)
Decreasing order of Size (DECR-S)	Parsa et al. (2017)
Cost OVER Time (COVERT)	Mönch and Zimmermann (2004), Rani (2018)
Apparent Tardiness Cost (ATC)	Park et al. (2000), Monch et al. (2006), Rani (2018)

Step 1: At the time of decision making, ***form a Work-In-Process (WIP)*** list with all the unprocessed jobs. ***Capture all the input data*** related to job characteristics of the unprocessed jobs (p_j, r_j, s_j, d_j, l_j) and the available time of each Burn-in oven (BO) (say t_m, $t_m > 0$; $m = 1,\dots,M$; M= number of BO) along with the capacity.

Step 2: ***Select the BO*** available at the earliest:

i. If all the burn-in ovens have the *same available time*, select the one with the greatest capacity.
ii. If there is still a tie after considering *available time and capacity*, choose the BO with machine eligibility restriction.
iii. If a tie persists across *available time, capacity, and machine eligibility*, select the BO randomly.

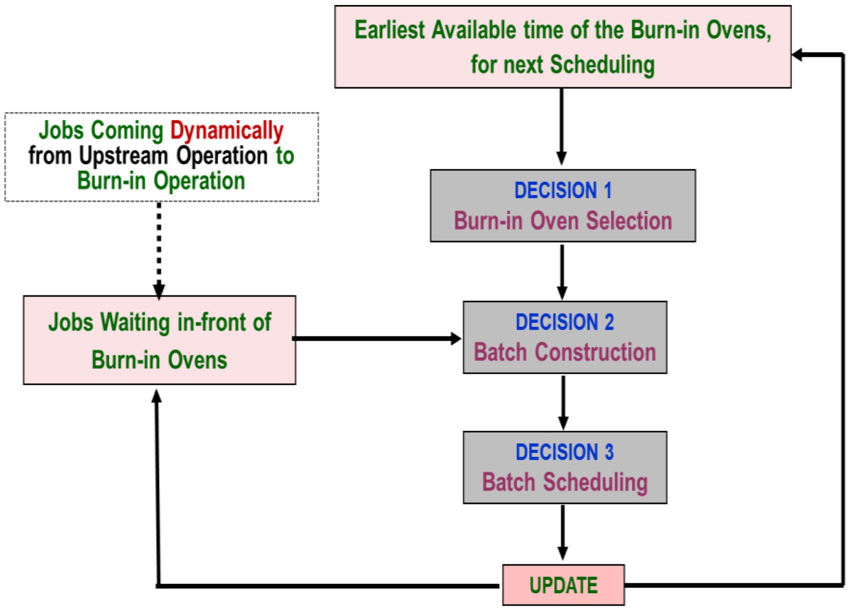

Figure 1. A decision making framework for DS-NPBO problem.

Step 3: ***Store*** the ***available time (t)*** and ***batch capacity (BC)*** of selected BO.
Step 4: ***Form a list of available unprocessed jobs*** for batch formation as follows:
If *selected BO is without MER*, then *except for 'jobs with MER'*, consider all unprocessed jobs available for batch formation.
Else, *when selected BO is with MER*, then consider all unprocessed jobs (*all jobs including jobs with MER*) available for batch formation.
Step 5: ***Compute the Job-Priority-Index (JPI)*** of each available unprocessed job as per the considered dispatching rule (DR).
Step 6: ***Sort all the jobs*** in ascending or descending order (as per DR) based on their JPI and make the *Sorted List of Available Unprocessed Jobs (SLAUJ)*.
Step 7: Considering the SLAUJ, ***form the temporary batches*** using the full batch strategy (Mönch et al., 2006).
Step 8: ***Set the release time of each temporary batch*** as *max* (*t*, highest release time of all the jobs in given temporary batch).
Step 9: ***Compute the Batch-Priority-Index (BPI)*** of each temporary batch as per the considered dispatching rule (DR).
Step 10: For each temporary batch, ***calculate the starting time and compare the same***,
If the completion time of any temporary batch is less than the starting time of all other temporary batches, then select such batch.
Else identify a *batch with the highest or lowest BPI (as per DR)* among all the temporary batches and select such batch.

Step 11: ***Allocate the selected batch***, identified in step 10, to the selected BO.

Step 12: Compute the value of the output parameters related to scheduled batch (such as starting time of the batch, processing time of batch, completion time of batch and job completion time) and based on that, ***compute the weighted tardiness of each job in scheduled batch***.

Step 13: ***Update the work in process*** list by excluding the jobs processed in the latest batch. Also, ***update the available time of selected BO*** (‘t’) as completion time of batch assigned to selected BO.

Step 14: Follow the procedure from Step 1 repeatedly as long as there are unscheduled jobs.

In the above step-by-step procedure for DS-NPBO, there are (i) ***batch construction for selected BO-*** where the proposed DR-HA computes the JPI of jobs using a dispatching rule and prioritizes the jobs based on their respective JPI value; later, the proposed DR-HA forms a batch by selecting the high priority jobs in succession till BO capacity is utilized up to the maximum extent; and (ii) ***batch selection for selected BO-*** where the proposed DR-HA computes the BPI of batches using a dispatching rule and selects a batch with the highest (or lowest) BPI value to process on selected BO.

Considering the proposed DR-HA, multiple variants of the same DR-HA are proposed with changes in dispatching rule considered for computing (a) JPI as well as order of sorting jobs after computing JPI in batch formation stage, and (b) BPI as well as order of sorting after computing BPI in batch selection stage. Accordingly, Step 5, Step 6, Step 9, and Step 10 are changed in the proposed DR-HA with different dispatching rules and sorting criteria, and thus 25 variants of DR-HA for DS-NPBO are obtained. The summary details of proposed 25 variants of DR-HA are presented in **Table 4**. All variants of DR-HA are coded in Python. Further, the formulas and the detailed explanation of all the dispatching rules, considered in proposed variants of DR-HA, are provided in Annexure-I.

5. Performance Evaluation of Proposed 25 Variants of DR-HA

For performance evaluation, we need to perform computational experiment and performance analysis. The specific details of these are as follows:

5.1 Computational Experiment

Any computational experiment for performance evaluation requires (a) test data, (b) a benchmark procedure and its solution to compare the proposed variants of DR-HA, and (c) performance measure(s), and the details of each of these are presented as follows:

Test data - Experimental Design: In the absence of real-life data sets, a good research process is to define a suitable experimental design for randomly generating test data. Accordingly, this study proposes an experimental design (**Table 5**) by considering the job and machine parameters mentioned in the research problem configuration and the existing experimental designs in the literature. Using the experimental design presented in **Table 5**, one can randomly generate medium to large-scale test data of problem sizes varying from 25 to 100 jobs. Further, the values of all job parameters (r_j, s_j, p_j, d_j, l_j) are generated using the uniform distribution as it is a high variance distribution that allows the variants of DR-HA to be evaluated in both favorable and unfavorable conditions. Thus, using the code developed for the proposed experimental design in Python, we randomly generated 240 test data, representing 5 instances for each of the 48 problem configurations.

Benchmark Procedure and its Solution**:** This study considers one of the estimated optimal solution procedures presented in (Rardin and Uzsoy, 2001) as benchmark procedure. To obtain the estimated optimal solution, which is the benchmark solution, each of the 240 test data is solved using each of the 25 variants of DR-HA, resulting in [240 × 25] TWT score matrix. Later, the estimated optimal value (EOV) for each of the 240 test data is computed using the Weibull distribution based estimated optimal solution procedure presented in (Rardin and Uzsoy, 2001), which considers the TWT score matrix to compute EOV for each test data. Thus, for each of the 240 randomly generated test data, EOV is obtained as a benchmark solution.

Table 4. A summary of the proposed 25 variants of DR-HA.

S. No.	Proposed variants of DR-HA	Step 5: computation of JPI is based on	Step 6: sorting order of JPI	Step 9: computation of BPI is based on	Step 10: sorting order of BPI
1.	DR-HA1	ERD	Ascending	Batch-ERD	Ascending
2.	DR-HA2	LPT	Descending	Batch-LPT	Descending
3.	DR-HA3	LPT	Descending	Batch-EDD	Ascending
4.	DR-HA4	EDD	Ascending	Batch-EDD	Ascending
5.	DR-HA5	MDD	Ascending	Batch-MDD	Ascending
6.	DR-HA6	MDD	Ascending	Batch-MS	Ascending
7.	DR-HA7	Job Size	Descending	WTB	Descending
8.	DR-HA8	Largest Weight	Descending	WTB	Descending
9.	DR-HA9	SPT	Ascending	WTB	Descending
10.	DR-HA10	LPT	Descending	WTB	Descending
11.	DR-HA11	FDD	Ascending	WTB	Descending
12.	DR-HA12	ODD	Ascending	WTB	Descending
13.	DR-HA13	MOD	Ascending	WTB	Descending
14.	DR-HA14	MS First	Ascending	WTB	Descending
15.	DR-HA15	COVERT	Descending	WTB	Descending
16.	DR-HA16	ATC- Vepsalainen	Descending	WTB	Descending
17.	DR-HA17	ATC-Farhad 1	Descending	WTB	Descending
18.	DR-HA18	ATC-Farhad 2	Descending	WTB	Descending
19.	DR-HA19	ATC-Farhad 3	Descending	WTB	Descending
20.	DR-HA20	ATC-Farhad 4	Descending	WTB	Descending
21.	DR-HA21	ATC-Bala	Descending	BATC-Bala	Descending
22.	DR-HA22	ATC-Monch	Descending	BATC-Monch	Descending
23.	DR-HA23	ATC-Li	Descending	BATC-Li	Descending
24.	DR-HA24	ATC-Li	Descending	BATC-Monch	Descending
25.	DR-HA25	ATC-Vimala	Descending	BATC-Vimala	Descending

Table 5. A summary of the experimental design considered to generate the test data.

Parameters	Number of levels	Level wise values
Number of Burn-in ovens (M)	2	2, 4
Burn-in oven Capacity (BC)	1	[20, 26]; if $M = 2$ [20, 26, 24, 22]; if $M = 4$
Available time of Burn-in oven (AT)	1	[2, 4]; if $M = 2$ [2, 4, 5, 7]; if $M = 4$
Machine that handles the Jobs with MER	1	Burn-in oven 2
No. of jobs (n)	3	25, 50, 100
Size of jobs (s_j)	2	U[4,10], U[4,14]
Release date of jobs (r_j)	2	U[1,20], U[1,30]
Processing time of jobs (p_j)	1	U[1,10]
Due-date of jobs (d_j)	2	$r_j + p_j$ + U[1,30], U[1,45]
Lateness penalty of jobs (l_j)	1	U[1,10]
Number of problem configurations		3 × 2 × 2 × 2 × 2 × 1 × 1 × 1 × 1 × 1 = 48
Number of test instances per configuration		5
Total numerical instances or test instances		**48 × 5 = 240**

Performance Measure for Empirical Analysis**:** The standard performance measure: Average Relative Percentage Deviation (ARPD), which measures the average case performance of each variant of DR-HA, is considered. To compute the ARPD score for any variant of DR-HA, the first step is to compute the

Relative Percentage Deviation (RPD_{ij}) value for every pair of test data ($i \epsilon$ [1,240]) and the proposed DR-HA variant ($j \epsilon$ [1,25]), as per Equation (1).

$$RPD_{ij} = \left[\left(Z_{ij} - EOV_i\right)/EOV_i\right] \times 100 \tag{1}$$

where, Z_{ij} = TWT value obtained by solving test data 'i' using the proposed DR-HA variant 'j'.
EOV_i = estimated optimal value for test data 'i'.

For each variant of DR-HA, after computing RPD_{ij} scores, ARPD score is computed over 240 test data, as per the following Equation (2):

$$ARPD_j = \Sigma_{i=1}^{N} RPD_{ij}/(N = 240) \tag{2}$$

where, $ARPD_j$: average relative percentage deviation of DR-HA variant 'j'.

Performance Measures for Statistical Analyses**:** This study considers descriptive statistics such as mean and median, and 95% confidence interval for all variants of DR-HA. In addition, the randomized complete block design (RCBD) and Tukey's multiple comparison tests are performed for statistical performance analysis.

5.2 Performance Analysis

As stated earlier, in this study, empirical and statistical analysis of performance of 25 variants of DR-HA for DS-NPBO is discussed. The details of these performance analyses are as follows:

Empirical Analysis**:** Each of the 240 test data is solved using 25 variants of DR-HA, and a [240 × 25] TWT score matrix is obtained. Using this [240 × 25] TWT score matrix and 240 EOV obtained earlier for the same 240 test data using the benchmark procedure considered in this study, compute the ARPD score for each of the 25 proposed variants of DR-HA as per Equations (1) & (2) and presented in **Table 6**. Considering the ARPD values in **Table 6**, this study observed that the ATC and BATC dispatching rule-based variants of DR-HA: DR-HA21 to DR-HA25, are performing well among all the proposed 25 variants of DR-HA. Particularly, the variant: DR-HA24 is the top performer. The possible reasons for the better performance of the variants: DR-HA21 to DR-HA25 could be due to the following:

- ATC rules take various job/batch parameters (release date, processing time, due-date, lateness penalty) into consideration while evaluating the JPI/BPI. With this, the better performing ATC dispatching rules combine the Weighted Shortest Processing Time (WSPT) and Minimum Slack first (MS first) dispatching rules and efficiently make the balance between two rules (WSPT and MS first) by evaluating the slackness of the jobs/batches. Further, the ATC rules use an exponential look-ahead feature to scale the slackness of job/batch, which has been found to improve the efficiency of ATC rules (Jayamohan and Rajendran, 2000).

Statistical Analyses**:** This study considers both Excel and Python for all statistical analyses. Accordingly, for each variant of DR-HA, the mean, median, and 95% interval estimation are computed by considering the [240 × 25] TWT values obtained earlier and provided in **Table 7**. Based on the statistical data provided in **Table 7**, five proposed variants of DR-HA: DR-HA21 to DR-HA25 are found performing well among all the proposed 25 variants of DR-HA and DR-HA24 is the top performer, which endorses the result obtained from the empirical analysis performed earlier.

In addition to the statistical performance analysis using the descriptive statistics, this study performs the randomized complete block design (RCBD) experiment to check if there is any significant difference in the performance of all the proposed 25 variants of DR-HA. This additional statistical performance analysis evaluates if the mean TWT of all 25 variants of DR-HA (provided in **Table 7**) are statistically identical or different from each other.

As per the terminology of RCBD experiment, every test data is considered as a block and each proposed variant of DR-HA is considered as a treatment. Further, in RCBD, multiple treatments (variants of DR-HA) are applied over each block (test data). The performance of each treatment (variant of DR-HA) is evaluated by analysing the effect of respective treatment on all the blocks (TWT of test data). Accordingly, the model of the RCBD experiment is as follows:

$$Y_{ij} = \mu + \alpha_j + \beta_i + \epsilon_{ij} \tag{3}$$

Here, Y_{ij} is a dependant variable shows the effect of treatment 'j' on block 'i' (i.e., TWT_{ij}), μ is the *overall mean TWT* (average of all TWT_{ij}; $i\epsilon[1,240]$ and $j\epsilon[1,25]$), α_j is the effect of treatment 'j' (i.e., it describes how mean TWT of variant of DR-HA 'j' differs from the overall mean TWT), β_i is the effect on block 'i' (i.e., it describes how mean TWT of test data 'i' differs from the overall mean TWT), and ϵ_{ij} is the random error.

Table 6. ARPD score for the 25 variants of DR-HA for DS-NPBO.

Proposed variants of DR-HA	ARPD score
DR-HA1	460.91
DR-HA2	2907.61
DR-HA3	1126.69
DR-HA4	525.54
DR-HA5	839.84
DR-HA6	617.60
DR-HA7	2289.91
DR-HA8	2418.11
DR-HA9	1713.59
DR-HA10	1826.70
DR-HA11	1930.06
DR-HA12	1774.92
DR-HA13	1633.75
DR-HA14	1023.57
DR-HA15	1986.31
DR-HA16	926.85
DR-HA17	1032.86
DR-HA18	1067.21
DR-HA19	962.88
DR-HA20	1038.2
DR-HA21	**386.62**
DR-HA22	**109.28**
DR-HA23	**105.62**
DR-HA24	**67.14**
DR-HA25	**121.88**

After applying all the treatments (variants of DR-HA) on each block (test data), RCBD experiment concludes if effect (mean TWT) of all the treatments (variants of DR-HA) are statistically identical or not, based on the following hypothesis:

H_0: There is no treatment effect (that is, mean TWT of all 25 variants of DR-HA are identical).

H_a: There is a treatment effect (that is, at least one variant of DR-HA has different mean TWT).

The above hypothesis is evaluated at 5% significance level in MS Excel, and the hypothesis result (analysis of variance- ANOVA) is provided in **Table 8**. Based on the analysis of the results presented in **Table 8**, the null hypothesis is rejected, as the *p*-values are strictly less than 0.05. Hence, it is concluded that mean TWT of at least one variant of DR-HA is statistically different from the mean TWT of other variants of DR-HA.

The result of RCBD experiment states that the performance (mean TWT) of few/all variants of DR-HA are different, i.e., some variants of DR-HA perform well compared to others. Hence, Tukey's multiple comparison (TMC) test is performed to identify the variants of DR-HA that are equally/more efficient compared to other variants of DR-HA. TMC test compares all the pairs of the mean TWT values. In case of 25 variants of DR-HA, TMC test performs 300 [(25×24)/2] pairwise comparisons. Accordingly, the hypothesis for the TMC test is as follows:

H_0: There is no difference between DR-HA$_x$ and DR-HA$_y$ in terms of the performance.
H_a: There is a difference between DR-HA$_x$ and DR-HA$_y$ in terms of the performance.
(here, $x = 1, 2,\ldots,25$; $y = 2, 3,\ldots,25$; $x < y$ *and* $x \neq y$).

Table 7. Result of the statistical evaluation of the proposed 25 variants of DR-HA.

Proposed variants of DR-HA	Mean	Median	95% Confidence interval
DR-HA1	4554.72	748.5	(3618.37, 5491.09)
DR-HA2	8277.39	3229	(6924.71, 9630.07)
DR-HA3	5101.00	1759	(4237.64, 5964.36)
DR-HA4	4550.33	793	(3646.55, 5454.11)
DR-HA5	4640.98	1182.5	(3759.09, 5522.88)
DR-HA6	3979.18	785.5	(3190.2, 4768.17)
DR-HA7	6552.10	2789.5	(5545.21, 7558.98)
DR-HA8	5857.26	2566	(4957.85, 6756.68)
DR-HA9	5251.76	2038.5	(4415.69, 6087.83)
DR-HA10	5073.67	2106.5	(4267.25, 5880.09)
DR-HA11	5852.25	2313.5	(4880.53, 6823.96)
DR-HA12	5536.03	2137.5	(4644.22, 6427.84)
DR-HA13	5064.12	2090.5	(4263.58, 5864.66)
DR-HA14	5355.63	1941	(4415.46, 6295.81)
DR-HA15	5181.71	2194	(4380.49, 5982.93)
DR-HA16	4131.35	1397.5	(3395.37, 4867.33)
DR-HA17	4373.03	1219	(3599, 5147.07)
DR-HA18	4637.03	1254	(3807.95, 5467.01)
DR-HA19	4392.13	1354	(3606.29, 5177.98)
DR-HA20	4624.53	1042.5	(3795.6, 5453.47)
DR-HA21	**2410.44**	**567.5**	**(1942.62, 2878.28)**
DR-HA22	**2597.18**	**360**	**(2065.3, 3129.07)**
DR-HA23	**2165.81**	**394**	**(1709.51, 2622.12)**
DR-HA24	**2025.82**	**303.5**	**(1586.47, 2465.18)**
DR-HA25	**2503.40**	**362.5**	**(1974.03, 3032.77)**

The result of TMC test is provided in **Table 9**, in terms of *p*-value for each of the 300 pairs of the variants of DR-HA. The *p*-values in **Table 9** shows that the performances of the variants of: DR-HA21 to DR-HA25, are statistically different from the other variants of DR-HA (i.e., DR-HA1 to DR-HA20), as the respective *p*-values in the columns corresponding to DR-HA21 to DR-HA25 in **Table 9** are strictly less than 0.10. Also, the *p*-value for each of the following ten pairs of the variants of DR-HA: (DR-HA$_x$, DR-HA$_y$); ($x, y = 21,\ldots, 25$); $x < y$, $x \neq y$] is 0.9 or nearly 1, which indicates that the performances (mean TWT) of these variants of DR-HA: DR-HA21 to DR-HA25 are statistically identical. Thus, based on the **Table 9**, this study concludes that the variants of DR-HA: DR-HA21 to DR-HA25 are the topmost heuristic algorithms among the proposed 25 variants of DR-HA for DS-NPBO. This inference supports the result obtained during the empirical and statistical analyses.

Table 8. ANOVA result for the randomised complete block design (RCBD) experiment.

Source of variation	Degree of freedom	Sum of squares	Mean square	F-Value	*p*-Value
Variants of DR-HA (Treatment)	24	$1.2E^{+10}$	$5.00E^{+8}$	138.69	0
Block (Instances)	239	$2.3E^{+11}$	$9.80E^{+8}$	272.652	0
Error	5736	$2.1E^{+10}$	3593674		
Total	5999	$2.7E^{+11}$			

6. Efficient Proposed Variant(s) of DR-HA for Real-time DS-NPBO (RTDS-NPBO)

In real life scenario, during DS-NPBO, there is a possibility of occurrences of real-time events (RTE) related to job (such as *due-date change, early/late arrival of jobs, change in job priority, new job addition, job cancellation) and/or* resource (such as *machine breakdown, operator illness, tool failure, shortage of material, defective material*). As no study in literature deals with Real-time Dynamic Scheduling (RTDS) of NPBO, this study reviewed the existing studies on RTDS of discrete processor. The review of studies on RTDS of discrete processor indicated that consideration of RTE while performing the dynamic scheduling of discrete processor reduces the efficiency of current algorithm and hence, there is a need for modifying the existing algorithm or developing rescheduling algorithm to handle RTE more efficiently (Vieira et al., 2003; Ouelhadj and Petrovic, 2009). However, due to long operation time of Burn-in process and the computerized tracking system on the SM shop floor, this study proposes the following research hypothesis:

"Appropriately modifying the data related to the jobs waiting in front of burn-in ovens for burn-in operation in case of the occurrence of job related RTE and modifying the BO available time in case of the occurrence of resource related RTE, is sufficient when there is an efficient algorithm for DS-NPBO, and developing the rescheduling algorithm/modifying the current algorithm whenever any types of RTE occur while scheduling the NPBO is not required."

The above research hypothesis is appropriately converted as measurable hypothesis and the same is given below:

H_o: *Performance ranking of each of the 25 variants of DR-HA obtained in DS-NPBO scenario is significantly different compared to the performance ranking of respective variant of DR-HA obtained in RTDS-NPBO scenario.*

H_a: *Performance ranking of each of the 25 variants of DR-HA obtained in DS-NPBO scenario is not significantly different compared to the performance ranking of respective variant of DR-HA obtained in RTDS-NPBO scenario.*

To test the above hypothesis on RTDS-NPBO, the decision making framework presented for DS-NPBO is modified for the problem on RTDS-NPBO and the same is presented in **Figure 2**. Further, the required computer code is developed for randomly generating either job or resource or both job and resource related RTE and incorporated in each of the proposed 25 variants of DR-HA for triggering the RTE while solving each of the 240 test data used in the earlier section. With this, the result in terms of TWT score (for each of the 240 test data solved using each of the proposed 25 variants of DR-HA) is recorded in [240 × 25] TWT score matrix for the RTDS-NPBO scenario. Further, using Equations (1) and (2), the ARPD score for each of the 25 proposed variants of DR-HA is obtained for RTDS-NPBO scenario and presented in **Table 10**. The ARPD score provided in **Table 10** shows that variants of DR-HA: DR-HA21 to DR-HA25 are performing well among 25 proposed variants of DR-HA considered for RTDS-NPBO, which highlights that variants of DR-HA that outperform during DS-NPBO also perform well in the case of RTDS-NPBO.

Table 9. Result of Tukey's pairwise multiple comparison test.

Variants	DR-HA2	DR-HA3	DR-HA4	DR-HA5	DR-HA6	DR-HA7	DR-HA8	DR-HA9	DR-HA10	DR-HA11	DR-HA12	DR-HA13	DR-HA14	DR-HA15	DR-HA16	DR-HA17	DR-HA18	DR-HA19	DR-HA20	DR-HA21	DR-HA22	DR-HA23	DR-HA24	DR-HA25
DR-HA1	0.001	0.9	0.9	0.9	0.9	0.12	0.12	0.9	0.9	0.88	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.01	0.01	0.01	0.01	0.01
DR-HA2		0.001	0.001	0.001	0.001	0.38	0.01	0.001	0.001	0.011	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
DR-HA3			0.9	0.9	0.9	0.7	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.001	0.004	0.001	0.001	0.001
DR-HA4				0.9	0.9	0.12	0.87	0.9	0.9	0.87	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.01	0.06	0.01	0.01	0.01
DR-HA5					0.9	0.19	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.009	0.01	0.008	0.008	0.01
DR-HA6						0.004	0.21	0.9	0.9	0.22	0.58	0.9	0.79	0.9	0.9	0.9	0.9	0.9	0.9	0.06	0.09	0.02	0.015	0.08
DR-HA7							0.9	0.87	0.68	0.9	0.9	0.67	0.9	0.8	0.01	0.05	0.18	0.05	0.17	0.001	0.001	0.001	0.001	0.001
DR-HA8								0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.4	0.67	0.9	0.69	0.9	0.001	0.001	0.001	0.001	0.001
DR-HA9									0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.001	0.004	0.001	0.001	0.001
DR-HA10										0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.002	0.004	0.001	0.001	0.004
DR-HA11											0.9	0.9	0.9	0.9	0.4	0.68	0.9	0.70	0.9	0.001	0.001	0.001	0.001	0.001
DR-HA12												0.9	0.9	0.9	0.76	0.9	0.9	0.9	0.9	0.001	0.001	0.001	0.001	0.001
DR-HA13													0.9	0.9	0.76	0.9	0.9	0.9	0.9	0.002	0.005	0.001	0.001	0.004
DR-HA14														0.9	0.9	0.9	0.9	0.9	0.9	0.001	0.001	0.001	0.001	0.001
DR-HA15															0.9	0.9	0.9	0.9	0.9	0.001	0.004	0.001	0.001	0.001
DR-HA16																0.9	0.9	0.9	0.9	0.04	0.06	0.015	0.015	0.05
DR-HA17																	0.9	0.9	0.9	0.01	0.06	0.01	0.01	0.02
DR-HA18																		0.9	0.9	0.01	0.01	0.008	0.008	0.01
DR-HA19																			0.9	0.01	0.06	0.01	0.01	0.012
DR-HA20																				0.01	0.01	0.008	0.008	0.01
DR-HA21																					0.9	0.9	0.9	0.9
DR-HA22																						0.9	0.9	0.9
DR-HA23																							0.9	0.9
DR-HA24																								0.9

Table 10. ARPD score for each of the proposed 25 variants of DR-HA for RTDS-NPBO.

Proposed variants of DR-HA	ARPD score
DR-HA1	371.49
DR-HA2	2434.92
DR-HA3	935.86
DR-HA4	310.83
DR-HA5	447.00
DR-HA6	340.00
DR-HA7	1927.51
DR-HA8	1892.96
DR-HA9	1520.33
DR-HA10	1485.92
DR-HA11	1415.54
DR-HA12	1224.01
DR-HA13	1450.34
DR-HA14	767.64
DR-HA15	1526.04
DR-HA16	681.84
DR-HA17	664.55
DR-HA18	777.97
DR-HA19	761.68
DR-HA20	917.73
DR-HA21	**252.43**
DR-HA22	**116.91**
DR-HA23	**103.16**
DR-HA24	**54.99**
DR-HA25	**121.68**

Table 11. Spearman's correlation test for statistical significance of the performances of the proposed 25 variants of DR-HA with respect to DR-NPBO and RTDS-NPBO scenarios.

Variant of DR-HA	***p*-value**	**Correlation coefficient between performances of variant of DR-HA in DS-NPBO and RTDS-NPBO scenarios**
DR-HA1	$<1e^{-10}$	0.98622
DR-HA2	$<1e^{-10}$	0.99096
DR-HA3	$<1e^{-10}$	0.98488
DR-HA4	$<1e^{-10}$	0.95957
DR-HA5	$<1e^{-10}$	0.96131
DR-HA6	$<1e^{-10}$	0.95392
DR-HA7	$<1e^{-10}$	0.98232
DR-HA8	$<1e^{-10}$	0.98426
DR-HA9	$<1e^{-10}$	0.97784
DR-HA10	$<1e^{-10}$	0.98186
DR-HA11	$<1e^{-10}$	0.97357
DR-HA12	$<1e^{-10}$	0.97008
DR-HA13	$<1e^{-10}$	0.97088
DR-HA14	$<1e^{-10}$	0.95425
DR-HA15	$<1e^{-10}$	0.97838
DR-HA16	$<1e^{-10}$	0.95876
DR-HA17	$<1e^{-10}$	0.96216
DR-HA18	$<1e^{-10}$	0.96100
DR-HA19	$<1e^{-10}$	0.96002
DR-HA20	$<1e^{-10}$	0.95901
DR-HA21	$<1e^{-10}$	0.94644
DR-HA22	$<1e^{-10}$	0.96419
DR-HA23	$<1e^{-10}$	0.93997
DR-HA24	$<1e^{-10}$	0.94750
DR-HA25	$<1e^{-10}$	0.94241

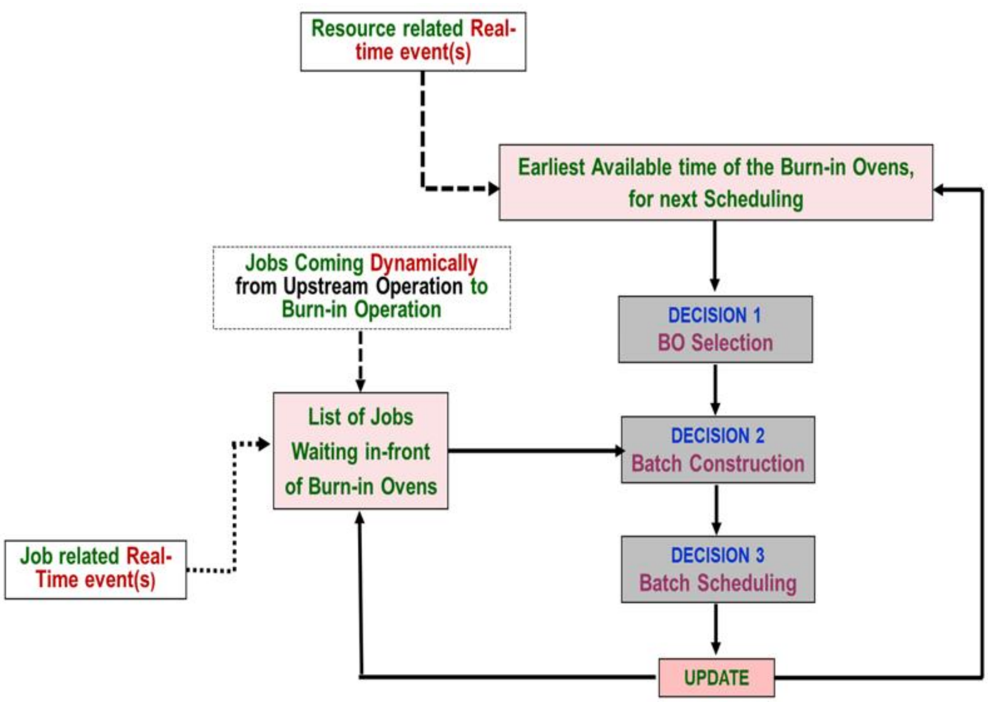

Figure 2. A decision making framework for RTDS-NPBO problem.

Using **Table 6** and **Table 10**, the performance ranking for each of the 25 variants of DR-HA is obtained w.r.t. DS-NPBO and RTDS-NPBO scenarios, and the same is presented in **Figure 3**. From **Figure 3**, it is observed that performance ranking of each variant of DR-HA remains almost same in both scenarios: DS-NPBO and RTDS-NPBO, which empirically supports the hypothesis provided in this study. Further, to support this observation statistically, the spearman's rank order correlation test is carried out, in which the correlation coefficient between the performance of each variant of DR-HA in DS-NPBO and RTDS-NPBO scenarios is computed and the same is provided in **Table 11**.

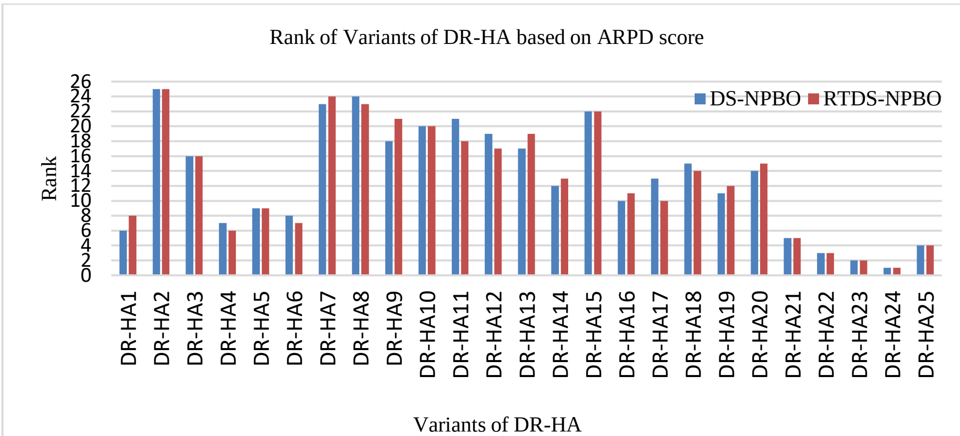

Figure 3. Performance ranking of the proposed 25 variants of DR-HA in DS-NPBO and RTDS-NPBO scenarios.

In **Table 11**, the *p*-value for each variant of DR-HA is < 0.05, which is strong evidence to reject H_0. Also, the high correlation coefficient for each variant of DR-HA shows a strong relation between the performance ranking of each variant of DR-HA in DS-NPBO and RTDS-NPBO scenarios. This highlights that variant of DR-HA that works efficiently in DS-NPBO scenario would work at the same efficiency in RTDS-NPBO scenario. This result is against the existing findings in the case of RTDS of discrete processor environment. Probably, this contracting result is due to long processing time required for burn-in operation and the presence of highly computerised shop floor in SM Industry.

7. Conclusion

This study addressed a new problem configuration, close to real life environment, on scheduling of non-identical parallel Burn-in ovens in the Semiconductor Manufacturing (SM) industry, by considering the real-life characteristics: occurrences of RTE, distinct job parameters (such as distinct- job sizes, processing times, due-dates, and release times), agreeable release times & due-dates, and Machine Eligibility Restriction (MER) to minimize the Total Weighted Tardiness (TWT) of the jobs. Due to NP-hard nature of the problem considered in this study, many studies in the literature have discussed Dispatching Rules (DR) based heuristic algorithm as DR are widely used in SM industry for scheduling of various batch processing machines, including burn-in ovens. However, to the best of our knowledge, there is no study that compares the various DR considered for scheduling of burn-in ovens in general, particularly in the context of Real Time Dynamic Scheduling (RTDS) of Non-identical Parallel Burn-in Ovens (NPBO) with MER problem defined in this study. To address this research gap, 25 variants of DR-HA is proposed. Initially, the proposed 25 variants of DR-HA were applied for Dynamic Scheduling (DS) of NPBO, considering 240 randomly generated test problems, and their performances were analysed empirically, in comparison with EOS obtained for each of the 240 test data, using the Average Relative Percentage Deviation (ARPD) score as an empirical performance measure.

In addition to the empirical analysis, this study carried out descriptive statistical measures, and several statistical tests such as randomized complete block design and Tukey's multiple comparison test. From both empirical and statistical performance analyses, it was observed that the proposed variants of DR-HA: DR-HA21 to DR-HA25 outperform the remaining proposed variants of DR-HA (DR-HA1 to DR-HA20). The possible reasons for these outperforming variants of DR-HA were also discussed.

To comprehend if the performance ranking of each of the 25 variants of DR-HA gets affected or not when the occurrences of RTE (real-life scenarios) are considered, an appropriate computer code is developed for randomly generating all defined RTE while performing RTDS-NPBO using each of the 25 variants of DR-HA. With this implementation for RTDS-NPBO, the same 240 test problems were solved using each of the 25 variants of DR-HA coupled with computer code for randomly generating all defined RTE, and ARPD score for each variant of DR-HA was computed. The ARPD scores obtained for each of the proposed 25 variants of DR-HA for the scenarios: (a) DS-NPBO and (b) RTDS-NPBO were compared empirically using performance ranking w.r.t. ARPD score in each scenario and statistically using Spearman's rank order correlation test. The empirical and statistical analyses clearly indicated that (a) in general, there is no change in the efficiency of each variant of DR-HA developed for DS-NPBO when it is applied for RTDS-NPBO scenario, and (b) the topmost efficient proposed variants of DR-HA: DR-HA21 to DR-HA25 remain robust and equally efficient in both DS-NPBO and RTDS-NPBO scenarios.

Though we have considered real time events to address RTDS-NPBO, the RTE considered in both cases of jobs and resources may not be exhaustive! The immediate future research work involves (a) identifying any left out RTE associated with jobs and resources to understand if the left out RTE has any impact on the performance ranking of the variants of DR-HA for RTDS-NPBO, and (b) developing meta-heuristic algorithms by considering efficient DR-HA as initial solution.

Conflict of Interest
The authors confirm that there is no conflict of interest to declare for this publication.

AI Disclosure
The author(s) declare that no assistance is taken from generative AI to write this article.

Acknowledgments
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor and anonymous reviewers for their comments that help improve the quality of this work.

ANNEXURE I: Dispatching Rules (DR) used for development of variants of DR-HA

As shown in **Table 4**, every variant of DR-HA considers two dispatching rules: one dispatching rule to compute the Job-Priority-Index (JPI) of jobs in batch formation stage and the second dispatching rule to compute the Batch-Priority-Index (BPI) of batches in batch selection stage. Accordingly, the formulas and the computational details of the DRs used for developing 25 variants of DR-HA are explained in this section.

Dispatching rules considered to compute the Job-Priority-Index (JPI) of jobs

Earliest Release-Date (*ERD*): ERD is function of a single job characteristic: 'Release time (r)'. As job release date is time independent job parameter, this rule comes under the static dispatching rule category.

ERD rule gives high priority to the job arrived early in the system. Accordingly, as per the ERD rule, the Job Priority Index (JPI) of job 'j' will be as follow:

$$JPI\ (j) = r_j.$$

Shortest Processing Time (*SPT*): SPT is a static DR. It considers processing time of job (p_j) to compute JPI and assigns high priority to the jobs with shorter processing time.

$$JPI\ (j) = p_j$$

Longest Processing Time (*LPT*): LPT is also a static DR that considers processing time of job (p_j) to compute JPI and assigns high priority to the jobs with higher processing time.

$$JPI\ (j) = p_j.$$

Largest Weight First (*LWF*): LWF considers lateness (l_j) penalty of the job to compute JPI and the jobs with higher JPI get more priority as per LWF rule.

$$JPI\ (j) = l_j.$$

Decreasing order of Size (*DECR-S*): DECR-S is a static DR and it is function of job size (s_j). Job occupying more space in BO gets high preference as per DECR-S rule.

$$JPI\ (j) = s_j.$$

Earliest Due Date (*EDD*): It is a simple and static DR considered by most of the studies in literature, and it considers single job characteristic: 'Due-date (d_j)' to compute JPI and assigns high priority to the jobs with shorter due-date.

$$JPI\ (j) = d_j.$$

Flow Due Date (*FDD*): It is a simple and static DR. FDD rule computes the JPI of job by adding two job characteristics: release date (r_j) and processing time (p_j), and job with lower JPI gets high priority.

$$JPI\ (j) = r_j + p_j.$$

Operational Due-Date (*ODD*): It is similar to the FDD dispatching rule, but it considers the due-date allowance factor (c= 3) in addition to the job characteristics considered in FDD.

$$JPI\ (j) = r_j + (c * p_j).$$

Modified Operational Due Date (*MOD*): It considers the time factor (τ): *a time instant at which the scheduling decision is made*. Hence, it is a dynamic/time dependant DR. Here, JPI of job can be computed as given below:

$$JPI\ (j) = max\{(r_j + c * p_j),\ \tau + p_j\}.$$

Modified Due-Date (*MDD*): It is a dynamic DR formed by combining two dispatching rules: EDD and FDD. This composite dispatching rule considers multiple job characteristics such as due-date, processing time, and job size.

$$JPI\ (j) = \max(d_j,\ \tau + p_j)/s_j.$$

Minimum Slack (*MS*) *First*: It is a simple and dynamic DR, which measures the urgency for completion of a specific job. MS First rule computes the JPI of job given below:

$$JPI\ (j) = \max(d_j - \tau - p_j\ , 0).$$

Cost OVER Time (*COVERT*): It is a composite & dynamic DR, which is formed by merging MS first and weighted shortest processing time. The equation for computing JPI using COVERT rule is given below:

$$\text{JPI}\ (j) = \left(\frac{w_j}{p_j}\right) * \left[1 - \left((d_j - \tau - p_j)^+ / \sum_{\mathrm{j=1}}^{\mathrm{n}} p_j\right)\right]^+.$$

Apparent Tardiness Cost (*ATC*): Like COVERT, ATC is also a composite & dynamic DR developed by combining two simple dispatching rules: MS first and WSPT. However, it differs from the COVERT rule in a way that it estimates the delay penalty using an exponential discounting formulation. **Table 12** shows the various ATC dispatching rules considered in this study.

Table 12. ATC rules considered for computing JPI of jobs.

ATC rule	**Job-priority-index(*j*) =**	**Source**
ATC-Vepsalainen	$\left(\frac{w_j}{p_j}\right) * \exp^{-\left[\max(0, d_j - p_j - \tau) / \left(k * \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right)\right]}$	Vepsalainen and Morton (1987)
ATC-Farhad 1	$\left(\frac{1}{p_j}\right) * \exp^{-\left[\max\left(0, d_j - (p_j + \tau)\right) / \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right]}$	Farhad and Laya (2007)
ATC-Farhad 2	$\left(\frac{1}{p_j * d_j}\right) * \exp^{-\left[\max\left(0, d_j - (p_j + \tau)\right) / \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right]}$	
ATC-Farhad 3	$\left(\frac{1}{p_j}\right) * \exp^{-\left[\max\left(0, d_j - \left(p_j + \max(p_j, \tau)\right)\right) / \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right]}$	
ATC-Farhad 4	$\left(\frac{1}{p_j * d_j}\right) * \exp^{-\left[\max\left(0, d_j - \left(p_j + \max(p_j, \tau)\right)\right) / \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right]}$	
ATC-Bala	$\left(\frac{w_j}{p_j}\right) * \exp^{-\left[\max(0, d_j - p_j - \tau) / \left(k * \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right)\right]}$	Mönch et al. (2004)
ATC-Monch	$\left(\frac{w_j}{p_j}\right) * \exp^{-\left[\max\left(0, d_j - p_j + (r_j - \tau)\right) / \left(k * \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right)\right]}$	Monch et al. (2006)
ATC-Li	$\left(\frac{w_j}{p_j}\right) * \exp^{-\left[\max\left(0, d_j - p_j - \tau + (r_j - \tau)^+\right) / \left(k * \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right)\right]}$	Li et al. (2009a)
ATC-Vimala	$\left(\frac{w_j}{p_j}\right) * f(w_j, p_j) * \exp^{-\left[\left(d_j - p_j - \tau + (r_j - \tau)^+\right)^+ / \left(k * \left(\frac{1}{n} * \sum_{j=1}^{n} p_j\right)\right)\right]}$ where, $f(w_j, p_j) = 1$ if $\left(\mathrm{RTB_b} + p_j - d_j\right) \leq 0$; $w_j / \left(\mathrm{RTB}_b + p_j - d_j\right)$ otherwise	Rani (2018)

Dispatching rules considered to compute the Batch-Priority-Index (BPI) of batches

Batch Earliest Release Date (*Batch-ERD*): Batch-ERD rule considers the release date of all jobs included in the selected batch and declares the maximum release date as the BPI of the batch. As per Batch-ERD, a batch with a lower BPI (early release date) gets high priority. Accordingly, BPI of batch '*b*' can be computed using Batch-ERD rule as follows:

$$\text{BPI}\ (b) = \max_{\mathrm{j} \in \mathrm{b}} \{r_j\}.$$

Batch Earliest Due Date (*Batch-EDD*): As per the Batch-EDD rule, the minimum due-date of the jobs included in the considered batch is declared as the BPI of respective batch, and a batch with a lower BPI (early due-date) gets high priority.

$$\text{BPI}\ (b) = \min_{j \in b} \{d_j\}.$$

Batch Longest Processing Time (*Batch-LPT*): As per the Batch-LPT rule, the longest processing time of the jobs included in the considered batch (or batch processing time) is taken as the BPI of respective batch, and a batch with a higher BPI gets high priority.

$$\text{BPI}\ (b) = \max_{j \in b} \{p_j\}.$$

Batch Minimum Slackness (*Batch-MS*): Batch-MS rule declares the minimum slackness of the jobs included in the considered batch as the BPI of respective batch, and a batch with a lower BPI gets high priority.

$$\text{BPI}\ (b) = \min_{j \in b} \{(d_j - \tau - p_j)^+\}.$$

Batch Modified Due-Date (*Batch-MDD*): As per the Batch-MDD rule, the minimum modified due-date of the jobs included in the considered batch is taken as the BPI of respective batch, and a batch with a lower BPI gets high priority.

$$\text{BPI}\ (b) = \min_{j \in b} \{\max(d_j,\ \tau + p_j)/s_j\}.$$

Weighted Tardiness of Batch (*WTB*): This rule computes the BPI of a batch by summing up the weighted tardiness of each job included in the given batch. The formula for computing WTB of batch '*b*' is as follows:

$$\text{BPI}\ (b) = \sum_{j=1}^{K} \{(\text{CTB}_b - d_j) * w_j\}.$$

where, K = No. of jobs included in the temporary batch '*b*'
CTB_b = completion time of batch '*b*'

Batch Apparent Tardiness Cost (*BATC*): BATC is a composite & dynamic DR. **Table 13** shows the various BATC dispatching rules considered in this study.

Table 13. BATC rules considered for computing BPI of batches.

BATC rule	**Batch-priority-index(*b*) =**	**Source**
BATC-Bala	$\Sigma_{j \epsilon b} \left\{\left(\frac{w_j}{p_j}\right) * \exp^{-\left[\max(0, d_j - p_j - \tau)/\left(k * \left(\frac{1}{n} * \Sigma_{j=1}^{n} p_j\right)\right)\right]}\right\}$	Mönch et al. (2004)
BATC-Monch	$\Sigma_{j \epsilon b} \left\{\left(\frac{w_j}{p_j}\right) * \exp^{-\left[\max\left(0, d_j - p_j + (RTB_b - \tau)\right)/\left(k * \left(\frac{1}{n} * \Sigma_{j=1}^{n} p_j\right)\right)\right]} * (\text{nbj}/\text{B})\right\}$	Monch et al. (2006)
BATC-Li	$\Sigma_{j \epsilon b} \left\{\left(\frac{w_j}{p_j}\right) * \exp^{-\left[\max\left(0, d_j - p_j - \tau + (r_j - \tau)^+\right)/\left(k * \left(\frac{1}{n} * \Sigma_{j=1}^{n} p_j\right)\right)\right]}\right\} * \min\left((\text{nbj}/\text{B}), 1\right)$	Li et al. (2009b)
BATC-Vimala	$\Sigma_{j \epsilon b} \left\{\left(\frac{w_j}{p_j}\right) * f(w_j, p_j) * \exp^{-\left[\left(d_j - p_j - \tau + (r_j - t)^+\right)^+/\left(k * \left(\frac{1}{n} * \Sigma_{j=1}^{n} p_j\right)\right)\right]}\right\} * (\text{nbj}/\text{B})$ where, $f(w_j, p_j)=1$ if $(\text{RTB}_b + p_j - d_j) \le 0$; $w_j/(\text{RTB}_b + p_j - d_j)$ otherwise	Rani (2018)

References

Alipour, P., Nguyen, C., & Damodaran, P. (2020). Using GRASP approach and path relinking to minimize total number of tardy jobs on a single batch processing machine. *International Journal of Industrial and Systems Engineering*, *35*(1), 1-22. https://doi.org/10.1504/ijise.2020.10017570.

Bellanger, A., & Oulamara, A. (2009). Scheduling hybrid flowshop with parallel batching machines and compatibilities. *Computers and Operations Research*, *36*(6), 1982-1992. https://doi.org/10.1016/j.cor.2008.06.011.

Brucker, P., Gladky, A., Hoogeveen, H., Kovalyov, M.Y., Potts, C.N., Tautenhahn, T., & Van De Velde, S.L. (1998). Scheduling a batching machine. *Journal of Scheduling*, *1*, 31-54. https://doi.org/10.1002/(SICI)1099-1425(199806)1:1<31::AID-JOS4>3.0.CO;2-R.

Chan, F.T.S., Au, K.C., Chan, L.Y., Lau, T.L., & Choy, K. (2007). A genetic algorithm approach to machine flexibility problems in an ion plating cell. *International Journal of Advanced Manufacturing Technology*, *31*, 1127-1134. https://doi.org/10.1007/s00170-005-0299-z.

Chen, J.C., Chen, T., Rezki, B., & Tu, Q. (2016). Capacity planning in thin film transistor – Liquid crystal display cell process. *Journal of Manufacturing Systems*, *39*, 63-78. https://doi.org/10.1016/j.jmsy.2016.03.003.

Chou, F., & Wang, H. (2012). Minimizing total weighted tardiness on parallel batch-processing machine scheduling problems with varying machine capacities. *Applied Mechanics and Materials*, *110*(116), 3906-3913. https://doi.org/10.1007/s00170-009-2342-y.

Chung, S.H., Tai, Y.T., & Pearn, W.L. (2009). Minimising makespan on parallel batch processing machines with non-identical ready time and arbitrary job sizes. *International Journal of Production Research*, *47*(18), 5109-5128. https://doi.org/10.1080/00207540802010807.

Condotta, A., Knust, S., & Shakhlevich, N.V. (2010). Parallel batch scheduling of equal-length jobs with release and due dates. *Journal of Scheduling*, *13*(5), 463-477. https://doi.org/10.1007/s10951-010-0176-y.

Dupont, L., & Flipo, C.D. (2002). Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure. *Computers and Operations Research*, *29*, 807-819. https://doi.org/10.1016/S0305-0548(00)00078-2.

Fachini, R.F., Esposto, K.F., Claudio, V., & Camargo, B. (2016). Glass container production planning with warm-ups and furnace extraction variation losses. *The International Journal of Advanced Manufacturing Technology*, *90*, 527-543. https://doi.org/10.1007/s00170-016-9369-7.

Fanti, M.P., Maione, B., Piscitelli, G., & Turchiano, B. (1996). Heuristic scheduling of jobs on a multi-product batch processing machine. *International Journal of Production Research*, *34*(8), 2163-2186. https://doi.org/10.1080/00207549608905019.

Farhad, G.T., & Laya, O. (2007). Development of a set of algorithms for the multi-project scheduling problems. *Joural of Industrial and System Engineering*, *1*(1), 23-36.

Graham, R.L., Lawler, E., Lenstra, J., & Kan, A.H.R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, *5*, 287-326.

Hulett, M., Damodaran, P., & Amouie, M. (2017). Scheduling non-identical parallel batch processing machines to minimize total weighted tardiness using particle swarm optimization. *Computers and Industrial Engineering*, *113*, 425-436. https://doi.org/10.1016/j.cie.2017.09.037.

Hung, Y. (1998). Scheduling of mask shop e-beam writers. *IEEE Transactions on Semiconductor Manufacturing*, *11*(1), 165-172.

Ikura, Y., & Gimple, M. (1986). Efficient scheduling algorithms for a single batch processing machine. *Operations Research Letters*, *5*(2), 61-65. https://doi.org/10.1016/0167-6377(86)90104-5.

Jayamohan, M.S., & Rajendran, C. (2000). New dispatching rules for shop scheduling: A step forward. *International Journal of Production Research*, *38*(3), 563-586. https://doi.org/10.1080/002075400189301.

Jia, Z.H., Li, Y.J., Li, K., & Chen, H.P. (2020). Weak-restriction bi-objective optimization algorithm for scheduling with rejection on non-identical batch processing machines. *Applied Soft Computing*, *86*, 105914. https://doi.org/10.1016/j.asoc.2019.105914.

Keshavarz, T. (2021). A lower bounding method for earliness and tardiness minimization on a single batch processing machine. *Journal of Industrial and Systems Engineering*, *13*(2), 9-26.

Khalouli, S., Ghedjati, F., & Hamzaoui, A. (2010). A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop. *Engineering Applications of Artificial Intelligence*, *23*(5), 765-771. https://doi.org/10.1016/j.engappai.2010.01.008.

Kim, Y.D., Kang, J.H., Lee, G.E., & Lim, S.K. (2011). Scheduling algorithms for minimizing tardiness of orders at the burn-in workstation in a semiconductor manufacturing system. *IEEE Transactions on Semiconductor Manufacturing*, *24*(1), 14-26. https://doi.org/10.1109/TSM.2010.2082470.

Koh, S., Koo, P., Kim, D., & Hur, W. (2005). Scheduling a single batch processing machine with arbitrary job sizes and incompatible job families. *International Journal of Production Economics*, *98*, 81-96. https://doi.org/10.1016/j.ijpe.2004.10.001.

Lee, C.Y., Uzsoy, R., & Martin-Vega, L.A. (1992). Efficient algorithms for scheduling semiconductor burn-in operations. *Operations Research*, *40*(4), 764-775. https://doi.org/10.1287/opre.40.4.764.

Leeftink, A.G., Bikker, I.A., Vliegen, I.M.H., & Boucherie, R.J. (2020). Multi-disciplinary planning in health care: a review. *Health Systems*, *9*(2), 95-118. https://doi.org/10.1080/20476965.2018.1436909.

Li, L., Qiao, F., & Wu, Q. D. (2009b). Aco-based scheduling of parallel batch processing machines to minimize the total weighted tardiness. In *2009 IEEE International Conference on Automation Science and Engineering* (pp. 280-285). IEEE. Bangalore, India.

Li, L., Qiao, F., & Wu, Q.D. (2009a). ACO-based multi-objective scheduling of parallel batch processing machines with advanced process control constraints. *International Journal of Advanced Manufacturing Technology*, *44*(9-10), 985-994. https://doi.org/10.1007/s00170-008-1904-8.

Li, S., Ng, C.T., Cheng, T.C.E., & Yuan, J. (2011). Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan. *European Journal of Operational Research*, *210*, 482-488. https://doi.org/10.1016/j.ejor.2010.11.021.

Li, S.S., & Chen, R.X. (2014). Single-machine parallel-batching scheduling with family jobs to minimize weighted number of tardy jobs. *Computers and Industrial Engineering*, *73*(1), 5-10. https://doi.org/10.1016/j.cie.2014.04.007.

Li, X., Chen, H., Du, B., & Tan, Q. (2013). Heuristics to schedule uniform parallel batch processing machines with dynamic job arrivals. *International Journal of Computer Integrated Manufacturing*, *26*(5), 474-486. https://doi.org/10.1080/0951192X.2012.731612.

Li, X.L., Li, Y.P., & Huang, Y.L. (2019). Heuristics and lower bound for minimizing maximum lateness on a batch processing machine with incompatible job families. *Computers and Operations Research*, *106*, 91-101. https://doi.org/10.1016/j.cor.2019.02.012.

Marier, P., Gaudreault, J., & Noguer, T. (2021). Kiln drying operations scheduling with dynamic composition of loading patterns. *Forest Products Journal*, *71*(2), 101-110. https://doi.org/10.13073/FPJ-D-20-00050.

Mathirajan, M., & Sivakumar, A.I. (2006). A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *International Journal of Advanced Manufacturing Technology*, *29*(9), 990-1001. https://doi.org/10.1080/07408179408966627.

Mathirajan, M., Bhargav, V., & Ramachandran, V. (2010). Minimizing total weighted tardiness on a batch-processing machine with non-agreeable release times and due dates. *International Journal of Advanced Manufacturing Technology*, *48*, 1133-1148. https://doi.org/10.1007/s00170-009-2342-y.

Mathirajan, M., Gokhale, R., Ramasubramaniam, M. (2014). Modeling of scheduling batch processor in discrete parts manufacturing. In: Ramanathan, U., Ramanathan, R. (eds) S*upply Chain Strategies, Issues and Models* (pp. 153-192). Springer, London. https://doi.org/10.1007/978-1-4471-5352-8_7.

Mazzini, R., & Armentano, V.A. (2001). Heuristic for single machine scheduling with early and tardy costs. *European Journal of Operational Research*, *128*(1), 129-146. https://doi.org/10.1016/S0377-2217(99)00345-8.

Mönch, L., & Unbehaun, R. (2007). Decomposition heuristics for minimizing earliness-tardiness on parallel burn-in ovens with a common due date. *Computers and Operations Research*, *34*(11), 3380-3396. https://doi.org/10.1016/j.cor.2006.02.003.

Monch, L., & Zimmermann, J. (2004). Improving the performance of dispatching rules in semiconductor manufacturing by iterative simulation. In *Proceedings of the 2004 Winter Simulation Conference, 2004.* (Vol. 2, pp. 1881-1887). IEEE. Washington, DC, USA.

Mönch, L., Balasubramanian, H., Fowler, J.W., Pfund, M.E. (2004). Minimizing total weighted tardiness on parallel batch process machines using genetic algorithms. In: Leopold-Wildburger, U., Rendl, F., Wäscher, G. (eds) *Operations Research Proceedings 2002* (Vol 2002). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-55537-4_37.

Mönch, L., Fowler, J.W., Dauzère-Pérès, S., Mason, S.J., & Rose, O. (2011). A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *Journal of Scheduling*, *14*(6), 583-599. https://doi.org/10.1007/s10951-010-0222-9.

Mönch, L., Unbehaun, R., & Choung, Y.I. (2006). Minimizing earliness-tardiness on a single burn-in oven with a common due date and maximum allowable tardiness constraint. *OR Spectrum*, *28*(2), 177-198. https://doi.org/10.1007/s00291-005-0013-4.

Monch, L., Zimmermann, J., & Otto, P. (2006). Machine learning techniques for scheduling jobs with incompatible families and unequal ready times on parallel batch machines. *Engineering Applications of Artificial Intelligence*, *19*(3), 235-245. https://doi.org/10.1016/j.engappai.2005.10.001.

Nguyen, S., Zhang, M., Johnston, M., & Tan, K.C. (2013). A computational study of representations in genetic programming to evolve dispatching rules for the job shop scheduling problem. *IEEE Transactions on Evolutionary Computation*, *17*(5), 621-639. https://doi.org/10.1109/TEVC.2012.2227326.

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, *12*(4), 417-431. https://doi.org/10.1007/s10951-008-0090-8.

Oulamara, A. (2007). Makespan minimization in a no-wait flow shop problem with two batching machines. *Computers and Operation Research*, *34*, 1033-1050. https://doi.org/10.1016/j.cor.2005.05.028.

Park, Y., Kim, S., & Lee, Y.H. (2000). Scheduling jobs on parallel machines applying neural network and heuristic rules. *Computers and Industrial Engineering*, *38*(1), 189-202. https://doi.org/10.1016/S0360-8352(00)00038-3.

Parsa, N.R., Karimi, B., & Husseini, S.M.M. (2017). Exact and heuristic algorithms for the just-in-time scheduling problem in a batch processing system. *Computers and Operations Research*, *80*, 173-183. https://doi.org/10.1016/j.cor.2016.12.001.

Pearn, W.L., Chung, S.H., & Yang, M.H. (2002). A case study on the wafer probing scheduling problem. *Production Planning and Control: The Management of Operations*, *13*(1), 66-75. https://doi.org/10.1080/09537280110061593.

Pujara, D.J., & Mathirajan, M. (2020). Mixed integer linear programming model for scheduling burn-in ovens to minimize total weighted earliness/lateness. In *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering* (pp. 1-5). IEEE. Jaipur, India. https://doi.org/10.1109/ICRAIE51050.2020.9358292.

Rani, M.V. (2018). *Impact of real time events on the relative efficiency of the proposed dynamic scheduling algorithms for diffusion furnace (s) in the semiconductor Manufacturing* (Doctoral dissertation).

Rani, M.V., & Mathirajan, M. (2020). Performance evaluation of due -date based dispatching rules in dynamic scheduling of diffusion furnace. *Operational Research Society of India*, *57*(2), 462-512. https://doi.org/10.1007/s12597-019-00434-8.

Rardin, R.L., & Uzsoy, R. (2001). Experimental evaluation of heuristic optimization algorithms : a tutorial. *Journal of Heuristics*, *304*(7), 261-304.

Ravindra, G., & Mathirajan, M. (2014). Minimizing total weighted tardiness on heterogeneous batch processors with incompatible job families. *International Journal of Advanced Manufacturing Technology*, *70*, 156-1578. https://doi.org/10.1007/s00170-013-5324-z.

Rose, O. (2002). Some issues of the critical ratio dispatch rule in semiconductor manufacturing. *Winter Simulation Conference Proceedings*, *2*, 1401-1405. https://doi.org/10.1109/wsc.2002.1166410.

Sung, C.S., & Kim, Y.H. (2002). Minimizing makespan in a two-machine flowshop with dynamic arrivals allowed. *Computers and Operations Research*, *29*, 275-294.

Tang, L., & Yan, P. (2009). Particle swarm optimization algorithm for a batching problem in the process industry. *Industrial and Engineering Chemistry Research*, *48*(20), 9186-9194. https://doi.org/10.1021/ie801742m.

Tsai, Y.C., & Chou, F.D. (2016). A Bi-criterion simulated annealing method for a single batch-processing machine scheduling problem with dynamic job release times. In *2016 International Conference on Artificial Intelligence and Computer Science* (pp.103-107). Guilin, China. https://doi.org/10.12783/dtcse/aics2016/8180.

Uzsoy, R. (1994). Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research*, *32*(7), 1615-1635. https://doi.org/10.1080/00207540050028034.

Ventura, J.A., & Kim, D. (2000). Parallel machine scheduling about an unrestricted due date and additional resource constraints. *Institute of Industrial Engineers Transactions*, *32*, 147-153.

Ventura, J.A., & Kim, D. (2003). Parallel machine scheduling with earliness – tardiness penalties and additional resource constraints. *Computers and Operation Research*, *30*, 1945-1958.

Vepsalainen, A.P.J., & Morton, T.E. (1987). Priority rules for job shops with weighted tardiness costs. *Management Science*, *33*(8), 1035-1047.

Vieira, G.E., Herrmann, J.W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling*, *6*(1), 39-62. https://doi.org/10.1023/A:1022235519958.

Xu, S., & Bean, J.C. (2015). Scheduling parallel-machine batch operations to maximize on-time delivery performance. *Journal of Scheduling*, *19*, 583-600. https://doi.org/10.1007/s10951-015-0449-6.

Zee, D.J., van der, van Harten, A., & Schuur, P. (2001). On-line scheduling of multi-server batch operations. *Institute of Industrial Engineers Transactions*, *33*(7), 569-586. https://doi.org/10.1080/07408170108936855.

Zhang, R., Chang, P.C., Song, S., & Wu, C. (2017). A multi-objective artificial bee colony algorithm for parallel batch-processing machine scheduling in fabric dyeing processes. *Knowledge-Based Systems*, *116*, 114-129. https://doi.org/10.1016/j.knosys.2016.10.026.

Zheng, R., & Li, H. (2009). Minimizing weighted number of tardy jobs on unbounded single batching machine with family jobs. In *2009 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 1718-1722). IEEE. Hong Kong, China.

Original content of this work is copyright © Ram Arti Publishers. Uses under the Creative Commons Attribution 4.0 International (CC BY 4.0) license at https://creativecommons.org/licenses/by/4.0/

Publisher's Note- Ram Arti Publishers remains neutral regarding jurisdictional claims in published maps and institutional affiliations.