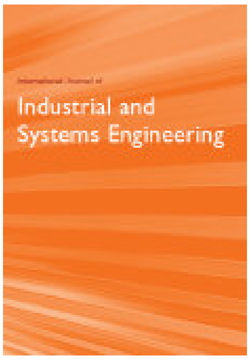
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research




**International Journal of Industrial and Systems Engineering**

ISSN online: 1748-5045 - ISSN print: 1748-5037
https://www.inderscience.com/ijise

# A machine learning algorithm for scheduling a burn-in oven problem

M. Mathirajan, Reddy Sujan, M. Vimala Rani, Pujara Dhaval

**DOI:** 10.1504/IJISE.2021.10042607

**Article History:**

| | |
|---|---|
| Received: | 21 April 2020 |
| Accepted: | 28 February 2021 |
| Published online: | 20 January 2023 |

# A machine learning algorithm for scheduling a burn-in oven problem


## M. Mathirajan*

Department of Management Studies,
Indian Institute of Science,
Bangalore, India
Email: msdmathi@iisc.ac.in
*Corresponding author

## Reddy Sujan

Department of Information Technology,
National Institute of Technology Suratkal,
Karnataka, India
Email: sujanreddy242@gmail.com

## M. Vimala Rani

Vinod Gupta School of Management,
Indian Institute of Technology-Kharagpur, India
Email: vimala@vgsom.iitkgp.ac.in

## Pujara Dhaval

Department of Management Studies,
Indian Institute of Science,
Bangalore, India
Email: dhavalpujara09@gmail.com


**Abstract:** This study applies artificial neural network (ANN) to achieve more accurate parameter estimations in calculating job-priority-data of jobs and the same is applied in a proposed dispatching rule-based greedy heuristic algorithm (DR-GHA) for efficiently scheduling a burn-in oven (BO) problem. The integration of ANN and DR-GHA is called as a hybrid neural network (HNN) algorithm. Accordingly, this study proposed eight variants of HNN algorithms by proposing eight variants of DR-GHA for scheduling a BO. The series of computational analyses (empirical and statistical) indicated that each of the variants of proposed HNN is significantly enhancing the performance of the respective proposed variants of DR-GHA for scheduling a BO. That is, more accurate parameter estimations in calculating job-priority-data for DR-GHA via back-propagation ANN leads to high-quality schedules w.r.t. total weighted tardiness. Further, proposed HNN variant: HNN-ODD is outperforming relatively with other HNN variants and provides very near optimal/estimated solution.

**Keywords:** dispatching rules; semiconductor manufacturing; greedy heuristic algorithm; GHA; artificial neural network; ANN; optimal solution; estimated optimal solution; dispatching rule-based greedy heuristic algorithm; DR-GHA; hybrid neural network; HNN.

**Reference** to this paper should be made as follows: Mathirajan, M., Sujan, R., Rani, M.V. and Dhaval, P. (2023) 'A machine learning algorithm for scheduling a burn-in oven problem', *Int. J. Industrial and Systems Engineering*, Vol. 43, No. 1, pp.20–58.

**Biographical notes:** M. Mathirajan obtained his PhD in Operations Management and MS degree by research in Applied Operations Research (OR) from the Indian Institute of Science (IISc), Bangalore. He also received his MSc in Mathematics from the Madurai Kamaraj University and Postgraduate Diploma in OR from the College of Engineering, Guindy. He has been working as a faculty of IISc Bangalore since 1986 and is serving as the Chief Research Scientist since 2013. His areas of interest include mathematical/heuristic optimisation and research methods for operations and supply chain management, sequencing and scheduling, personnel scheduling, routing and scheduling of logistics, urban road transport, and container terminal logistics problems.

Reddy Sujan is an undergraduate student in the Department of Information Technology at the National Institute of Technology Karnataka, Surathkal, and will be graduating in 2022. His research interests are machine learning, deep learning, and operations research. He is a recipient of the Kishore Vaigyanik Protsahan Yojana (KVPY) Fellowship Award. In April of 2020, he was awarded the 'Professor BG Raghavendra Internship Award in Operations Research' by the Indian Institute of Science.

M. Vimala Rani is currently working as an Assistant Professor in Operations at the VGSoM, Indian Institute of Technology-Kharagpur. She obtained her PhD in Operations Management from the Indian Institute of Science (IISc), Bangalore. She has received a Gold medal and the Best PhD thesis award by the Council of IISc. After her PhD, she has worked as 'IISc Research Associate' at the Department of Management Studies, IISc, Bangalore. She also worked as an Assistant Professor at the Amrita School of Business, Bangalore. Her research interest is on developing the mathematical model and heuristic algorithms for the problems related to industrial engineering and management.

Pujara Dhaval is currently a research scholar in the Department of Management Studies at Indian Institute of Science, Bengaluru. He received his MTech in Industrial Engineering and Management from the National Institute of Technology, Calicut, and BE in Production Engineering from the Gujarat Technological University, Gujarat.

## 1 Introduction

Scheduling of a batch processing machine (BPM) or batch processor (BP) problem has been continuously addressed by many researchers due to its complexity in terms of problem parameters and their attributes (Table 1). The BPM is a processor which processes simultaneously more than one job as a batch. The basic principle in batch processing is that the jobs in each batch will be processed with common starting and ending times. The main reasons for concentrating on the scheduling of BPMs are due to the very high processing time requirement of batch operation when we compare with other processes and batching decision may affect the performance of the entire manufacturing/service industry (Table 2).

**Table 1** BPM problem parameters and their attributes in scheduling of BPM

| *BPM problem parameters* | *Attribute* |
|---|---|
| Number of BPM | Single BPM |
| | Multiple and homogeneous type BPM |
| | Multiple and heterogeneous type BPM |
| Capacity restriction of BPM | Bounded |
| | Un-bounded |
| Family of jobs | Single family of jobs |
| | Multiple and compatible job-families |
| | Multiple and in-compatible job-families |
| Size of the jobs | Identical job size |
| | Non-identical job size |
| Dimension/volume of the jobs | Identical job dimension/volume |
| | Non-identical job dimension/volume |
| Splitting of jobs between the batches | Allowed |
| | Not allowed |
| Batch processing time | Dependent on the jobs in the batch |
| | Independent of the jobs in the batch |
| Set-up time | Included in the processing time |
| | Not-included in the processing time |
| Scheduling objective | Completion time-based |
| | Due-date based |
| | Cost-based |
| Number of scheduling objectives | Single |
| | Multiple |
| Date availability | Deterministic and known |
| | Stochastic |
| | Fuzzy |
| Scheduling problem nature | Static |
| | Dynamic considering only future arrival of jobs |
| | Dynamic considering future arrival of jobs and real-time events |

**Table 2** Scheduling of BPM in various industry (in alphabetical order) with a sample latest reference

| *Batch operations/processor* | *Industry* | *Sample latest reference(s)* |
|---|---|---|
| Furnace for bending phase | Automotive safety glass manufacturing facility | Mora et al. (2020) |
| Hardening of synthetic parts using oven | Aircraft industry | Van der Zee et al. (2001) |
| Furnaces used to heat the aluminium Ingots | Aluminium manufacturing | Jia et al. (2016) |
| Hardening and soaking/ heat-treatment furnace | Automobile gear manufacturing | Ravindra and Mathirajan (2014) |
| Batch distillation process | Chemical industry | Tang and Yan (2009) |
| Dyeing machine | Clothing industry | Zhang et al. (2017) |
| Environmental stress screening (ESS) chambers | Electronics manufacturing industry | Damodaran, et al. (2009) and Alipour et al. (2020) |
| Thermal chamber | Electronics manufacturing industry | Damodaran and Wiechman (2015) |
| Cutting machine | Furniture manufacturing | Ogun and Alabas-Uslu (2018) |
| Dry Kiln | Furniture manufacturing industry | Yaghubian et al. (2001) |
| Annealing Kiln | Glass container industry | Fachini et al. (2017) |
| Tissue processors | Hospital histopathology laboratory | Leeftink et al. (2018) |
| Washer – washing of reusable medical devices | Hospital sterilisation services | Ozturk et al. (2010) |
| IP machine used in watch and clock industry | Ion plating (IP) industry | Chan et al. (2007) |
| Multi-head hole-punching machine | Iron and steel industry | Oulamara (2007) |
| Heat-treatment furnace (HTF) | Metalworking industry | Dupont and Dhaenens-Flipo (2002) |
| Bake-out/box-oven | Multi-layer-ceramic capacitor | Koh et al. (2004) |
| Pottery Kiln | Pottery manufacturing | Jia et al. (2020) |
| Diffusion furnace/machine | Semiconductor manufacturing | Rani and Mathirajan (2020) |
| E-beam writer | Semiconductor manufacturing | Hung (1998) |
| Etching tank | Semiconductor manufacturing | Sung and Kim (2002) |
| Baking machine in wafer probe | Semiconductor manufacturing | Huang and Lin (1998) |
| Pallet in assembly stage | Semiconductor manufacturing | Cheng et al. (1996) |

**Table 2** Scheduling of BPM in various industry (in alphabetical order) with a sample latest reference (continued)

| *Batch operations/processor* | *Industry* | *Sample latest reference(s)* |
|---|---|---|
| Burn-in oven | Semiconductor manufacturing | Li et al. (2019) |
| Cleaning multiple subtracts of sensors/bath | Sensor manufacturing facility | Rojas-Santiago et al. (2017) and Maya et al. (2014) |
| Hole punching | Sheet metal industry | Boudhar (2003) |
| Carousel | Shoe manufacturing factory | Fanti et al. (1996) |
| Heat treatment furnace (HTF) | Steel casting industry | Mathirajan and Sivakumar (2006b) |
| Soaking bit furnace | Steel Ingot production | Li et al. (2011) |
| HTF | Steel manufacturing | Zheng and Li (2009) |
| Annealing furnace | Steel production | Ozturk (2020) |
| Aging test operation | Thin film transistor liquid crystal displays (TFT-LCD) manufacturing | Chung et al. (2009) |
| Tyre curing/mould | Tyre manufacturing | Bellanger and Oulamara (2009) |
| Sterilisation machine | Professional waste disposal services | Tsai and Chou (2016) |

This study particularly addresses the scheduling of the burn-in process, that is scheduling of burn-in oven (BO) (a BPM), of the final testing stage of the back-end manufacturing operation of semiconductor manufacturing (SM). The purpose of the burn-in process is to bring out latent defects due to infant mortality of chips. To achieve this, the chips of each lot (job) are loaded on specific burn-in boards and exposed to a high temperature and voltages for a long period. Chips are stressed electrically and thermally, that is, they are placed in an oven at temperatures up to 150°C and voltage, which may be as high as 1.5 times the normal operating voltage, which are then applied, at high temperature for a period of time which may be as short as a few hours or as long as 48 hours.

The analysis of the literature indicated that dispatching rule-based greedy heuristic algorithm (DR-GHA) provides efficient solution as quick as possible in scheduling (Sarin et al., 2011). Furthermore, it is observed that, DR-GHA is widely used in industries such as SM industry (Varadarajan and Sarin, 2006; Hildebrandt et al., 2010; Chen and Wang, 2012). The reasons for its utilisation are mostly based on the fact that DR-GHAs perform efficiently in a wide range of environments and particularly these algorithms are relatively easy to understand, easy to implement, require only minimal computational time and can cope with dynamic changes (Nguyen et al., 2013).

By and large all the DR-GHAs consider job-priority-index, computed based on dispatching rule and the job-priority-data, as a criterion to construct a batch for scheduling in BO (Lee et al., 1992; Mathirajan et al., 2010; Li et al., 2019). The quality of the DR-GHA for scheduling a BPM is expected to vary when the job-priority-data is changed. Thus, this study proposes an application of a back-propagation artificial neural network (ANN), which is an extension of Parsa et al. (2019), to achieve more accurate parameter estimations in calculating the job-priority-data for the jobs and to enhance the performance of the DR-GHA for scheduling a BPM.

The overall arrangement of the paper is as follows. Problem description with assumptions is discussed in Section 2. In Section 3, the closely related literature on scheduling of BO with due-date based scheduling objectives is presented. The proposed multiple variants of

a DR-GHA

b machine learning (ML) approach, implemented based on back-propagation neural network, are discussed in Section 4.

Results of the computational experiments to evaluate the performance of the multiple variants of ML approach are discussed in Section 5. Finally, the conclusion of the study along with the implications, limitations and the directions for future research is presented in Section 6.

## 2 Problem description and assumptions

IC chips (called as product) arrive at the burn-in area in lots. Each lot consists of several IC chips of the same product type. Each lot or product is referred to as a job. In the burn-in operation, IC chips of each job are loaded onto boards (often product-specific). As each job has different lot sizes, the number of boards required to place the entire lot (job) is *different*. The boards are then placed into a BO. Typically, an oven's capacity is measured in terms of number of boards placed in an oven. Each IC chips of a lot has a pre-specified minimum burn-in time, depending on its type and/or the customer's requirements. With these basic briefs on burn-in operation, in this study, there are 'N' jobs (lots) that need to be scheduled in a BO. Accordingly, each of the jobs has different job size of $S_j$ (that is, number of boards required for each job), processing requirement of $P_j$, due-date of $D_j$, weight (job-priority-data) of $W_j$, and non-zero release time of $R_j$ (at which it becomes available for processing, and consider all future arrival of jobs until the decision making time epoch for scheduling a batch to capture the dynamic nature of scheduling), and the release time(s) and due-date(s) are non-agreeable [that is, job-processing need not necessarily follow the first-in-first-out, based on the release time (that is, if $R_i \leq R_j$ not-implied $D_i \leq D_j$)]. The release time of a batch for scheduling is given by the longest ready time of all jobs in the batch. As IC chips may stay in the BO for a period longer than their minimum required burn-in time, the processing time of each batch equals the longest minimum exposure time among all the products (jobs) in the batch.

In the competitive global environment, customers' voices have forced production managers to consider on-time delivery as an important management performance. So, for the research problem characteristics defined here, the scheduling objective is to minimise total weighted tardiness (TWT). TWT is defined as the sum of product of size and tardiness of job 'j' (that is, $TWT = \sum_{j=1}^{N} S_j * T_j$, where $T_j$ and $S_j$ are tardiness and size of job 'j', respectively). And $T_j$ is defined as how much late the job 'j' is completed as compared to the due-date (that is, $T_j = \max(0, C_j - D_j)$, where $C_j$ and $D_j$ are completion time and due-date of job 'j', respectively). The reason for considering the scheduling objective of minimising TWT is that it is a measure that incurs a penalty for each job that finishes processing after its committed due-date. As this penalty increases with the magnitude of the tardiness, schedules that minimise the weighted sum of the penalties provide good on-time delivery performance (Perez et al., 2005).

The research problem described here can be concisely represented using the three-field notation of Graham et al. (1979) as "1/p-batch, dynamic job-arrivals, non-identical job sizes, non-identical processing time, due-dates, release time, non-agreeable release time(s) and due-date(s)/TWT" and we make the following assumptions:

- Data required for dynamic scheduling of a BO problem defined in this study are assumed to be deterministic and known a priori. This assumption is a valid one as in practice, particularly in SM, estimates of the required parameters' values for this problem can be obtained from the existing shop-floor computerised information system.
- Each job requires one operation, and all jobs are independent.
- The BO has a capacity 'B', measured in terms of number of boards it can hold. The number of boards required for any job (that is $S_j$) must be less than or equal to 'B'. That is, splitting of lot for processing is not permitted.
- Once processing of a batch is initiated, it cannot be interrupted, and other jobs cannot be introduced into the BO until processing of the batch is complete.

## 3 A closely related literature review

Scheduling of BPM was very first time studied by Ikura and Gimple (1986). Though there are many studies addressing on scheduling BPM across various industries (Table 2), most of the studies are related to SM and particularly related to scheduling of:

a diffusion furnace

b BO only (Mathirajan and Sivakumar, 2006a; Monch et al., 2011).

Since, this study is pertaining to scheduling of BO problem; the existing studies on scheduling of BO are reviewed and classified based on the type of BPM problem parameter considered. Accordingly, all the existing studies on scheduling of BO problems can be grouped into completion time-based scheduling objective and due-date based scheduling objective. As this study is concentrating on due-date based scheduling objective, all the existing studies considering due-date based scheduling objective are reviewed and discussed in this section. However, for the existing studies considering completion time-based scheduling objective, we refer to Parsa et al. (2016), Jia et al. (2017), Beldar and Costa (2018), Chung and Sun (2018) and Alizadeh and Kashan (2019). Furthermore, all the existing studies are grouped into single objective and multi-objectives. As this study is related to single due-date based scheduling objective, all the existing studies considering single due-date based scheduling objective which are reviewed and discussed in this section. Whereas the existing studies considering multi objectives we refer to Tsai and Chou (2016), Zhang et al. (2017), Feng et al. (2020) and Jia et al. (2020).

The problem of scheduling burn-in operation in SM was first introduced by Lee et al. (1992) in which jobs are of the same size, same processing time and different arrival time. The batches were constrained by the number of jobs that could be included, as the jobs are assumed to have the same size. Number of problems considering single and parallel BO(s) were studied with single and multi-objectives such as minimising maximum tardiness, number of tardy jobs, maximum lateness, and makespan. They proposed dynamic programming (DP)-based algorithms. Based on the computational analyses, they showed that some of the problems studied may be solved in polynomial time, but others are NP-hard.

Dynamic scheduling of a BO problem with jobs having identical job sizes, non-identical processing time and agreeable release time(s) and due date(s)to minimise number of tardy jobs is addressed by Hochbaum and Landy (1994). Due to computational intractability in solving large size problems they proposed simple greedy heuristic algorithms (GHAs).

Wang and Uzsoy (2002) extend the problem configuration considered in Hochbaum and Landy (1994) by including non-agreeable release time(s) and due date(s) to minimise maximum lateness (Lmax). They combine a DP-based algorithm, with a random key encoding scheme to develop a genetic algorithm (GA) for the problem configuration considered in their study. Ventura and Kim (2000, 2003) extend the problem configuration considered in Wang and Uzsoy (2002) by introducing parallel identical BPs and non-identical job-size with the scheduling objective of minimising the earliness-tardiness for the static and dynamic situation along with additional constraint on number of boards required to fill the lot (job). Further, Ventura and Kim (2003) assumed that all job processing times are same.

Mönch et al. (2006) consider the scheduling of the single BO with jobs having an unrestrictive late common due-date using a combination of dominance properties and a GA with the objective of minimising the earliness-tardiness of the jobs. They also include a constraint on the maximum allowable tardiness. Subsequently, Mönch and Unbehaun (2007) extend this research by considering the case of parallel identical BOs and proposed DP.

Raghavan and Venkataramana (2006) considered a static version of scheduling of multiple and identical BOs problem with jobs having identical job sizes, non-identical processing time and agreeable release time(s) and due-date(s) to minimise TWT. They proposed mixed integer linear programming (MILP) model. Due to computational intractability of the proposed MILP model, they proposed ant colony algorithm (ACO) for efficiently addressing large sized real-life static scheduling problems.

Li and Chen (2014), Cabo et al. (2015), Li et al. (2015) and Parsa et al. (2017a, 2017b) considered the same job characteristics of the problem studied by Raghavan and Venkataramana (2006) for a single BO problem to minimise number of tardy jobs, minimise the maximum lateness, minimise earliness and tardiness, and minimise the total tardiness, respectively and proposed GHA, and GA respectively. Moreover, Li and Chen (2014) considered non-agreeable release time(s) and due date(s) as additional constraint and Li et al. (2015), and Parsa et al. (2017a, 2017b) considered non-identical job size.

Chou and Wang (2012) and Xu and Bean (2016) considered dynamic scheduling of multiple and non-identical BOs with non-identical job size, non-identical processing time, and non-agreeable release time(s) and due-date(s) to minimise TWT. Both the studies proposed MILP model. Due to computational difficulty in getting optimal solution they proposed simulated annealing (SA) and GA respectively. Hulett et al. (2017) considered the problem studied by Chou and Wang (2012) with static situation and agreeable release time(s) and due date(s). This study also proposed MILP model. Due to computational difficulty in getting optimal solution they proposed particle swarm optimisation (PSO).

Dynamic scheduling of a BO problem with jobs having non-identical job sizes, non-identical processing time and non-agreeable release time(s) and due date(s) to minimise TWT is addressed by Chou and Wang (2008), Mathirajan et al. (2010) and Wang (2011). They proposed MILP model and demonstrated its computational intractability in solving real-life sized problem on scheduling a BO. First two studies proposed meta heuristics: GA, SA respectively and other two studies proposed GHA, for efficiently addressing large sized real-life problems on scheduling a BO.

Condotta et al. (2010) and Zhou et al. (2018) studied the same problem addressed in Chou and Wang (2008) with scheduling objective of Lmax and proposed simple GHA and meta heuristic: PSO, respectively. However, Condotta et al. (2010) assumed an identical job size, identical processing time.

Li et al. (2019) considered static scheduling of single BO with non-identical job size, multiple job-families and non-identical processing time to minimise the maximum lateness. They proposed MILP model and due to its computational difficulty in getting optimal solution for large sized problem they proposed simple GHA. Very recently Keshavarz (2021) developed a lower bound method, based on column generation approach, for a static scheduling of single BO with non-identical processing time and non-identical job-size to minimise the total earliness and tardiness. Further, this researcher empirically proved that the proposed lower bound method could enhance the lower bound around 41% in average comparing with the best known lower bounding method in the literature.

A summary on the review of closely related existing studies on scheduling of BO with due-date based objective, discussed in this section, is presented in Table 3. From Table 3, one can observe that a scant treatment has been given in the literature on scheduling of a BO with non-agreeability of release time(s) and due-date(s), non-identical job-sizes, and non-identical processing times to minimise the customer-based scheduling objective of TWT.

**Table 3** A summary on closely related scheduling of BO with due-date based single scheduling objective

| No. | Researchers | Year | *Burn-in oven characteristics* | | | *Job characteristics* | | | | | | | | | *Nature of scheduling* | | *Additional constraints* | | *DD-based objective* | *Solution methodology* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | *SBO* | *MIBO* | *MNIBO* | *Number of family* | | *Job size (JS)* | | *Processing time (PT)* | | *Priority/ weight* | *Release time (RT)* | *Due-date (DD)* | *Static* | *Dynamic* | *Board* | *Non-agreeable RT & DD* | | *Math. model* | *Greedy heuristic* | *Meta heuristic* |
| | | | | | | *SF* | *MF* | *IJS* | *NIJS* | *IPT* | *NIPT* | | | | | | | | | | | |
| 1 | Hochbaum and Landy | 1994 | x | | | x | | x | | | x | x | x | x | | x | | | NT | | x | |
| 2 | Ventura and Kim | 2000 | | x | | x | | | x | | x | | x | x | x | | x | | E-T | MILP | x | |
| 3 | Wang and Uzsoy | 2002 | x | | | x | | x | | | x | | x | x | | x | | x | Lmax | DP | x | GA |
| 4 | Ventura and Kim | 2003 | | x | | x | | | x | x | | | x | x | | x | x | x | E-T | MILP | x | |
| 5 | Mönch et al. | 2006 | x | | | x | | x | | | x | | x | x | x | | | | E-T | | x | GA |
| 6 | Raghavan and Venkataramana | 2006 | | x | | | x | x | | | x | x | x | x | x | | | | TWT | MILP | | ACO |
| 7 | Mönch and Unbehaun | 2007 | | x | | x | | x | | | x | | x | x | x | | | | E-T | DP | x | GA |
| 8 | Chou and Wang | 2008 | x | | | x | | | x | | x | x | x | x | | x | | x | TWT | MILP, DP | x | GA |
| 9 | Mathirajan et al. | 2010 | x | | | x | | | x | | x | x | x | x | | x | | x | TWT | MILP | x | SA |
| 10 | Condotta et al. | 2010 | x | | | x | | x | | x | | | x | x | | x | | x | Lmax | | x | |
| 11 | Wang | 2011 | x | | | x | | | x | | x | x | x | x | | x | | x | TWT | | x | |
| 12 | Chou and Wang | 2012 | | | | x | | | x | | x | x | x | x | | x | | x | TWT | MILP | x | SA |
| 13 | Li and Chen | 2014 | x | | | | x | x | | | x | x | x | x | x | | | x | NT | | x | |
| 14 | Cabo et al. | 2015 | x | | | x | | x | | | x | | x | x | x | | | | Lmax | DP | x | |
| 15 | Li et al. | 2015 | x | | | x | | | x | | x | | x | x | x | | | | E-T | MILP | x | GA |
| 16 | Xu and Bean | 2016 | | | x | x | | | x | | x | x | x | x | | x | | x | TWT | NLP | | GA |
| 17 | Parsa et al. | 2017a | x | | | x | | | x | | x | | x | x | x | | | | TT | DP | x | |
| 18 | Parsa et al. | 2017b | x | | | x | | | x | | x | | x | x | x | | | | E/L | DP | | GA |
| 19 | Hulett et al. | 2017 | | | x | x | | | x | | x | x | x | x | x | | | | TWT | MILP | | PSO |
| 20 | Zhou et al. | 2018 | x | | | x | | | x | | x | | x | x | | x | | x | Lmax | | | PSO |
| 21 | Li et al. | 2019 | x | | | | x | | x | | x | | x | x | x | | | | Lmax | MILP | x | |
| 22 | Keshavarz | 2021 | x | | | x | | | x | | x | | | x | x | | | | E-T | MILP | x | Memetic algorithm |

Though, the analysis of the literature indicated that the application of ML approach for scheduling is not a new to the scheduling (Aytug et al., 1994; Priore et al., 2014), ML, and neural networks are not developed enough to solve the scheduling problem (Melnik and Nasonov, 2019) in general. Particularly there is a very scant treatment has been given towards ML approaches for scheduling BPM. Furthermore, though there are two studies proposed ML approaches for scheduling a BO with completion time-based scheduling objective (Shao et al., 2008; Parsa et al., 2019), there has been no research for scheduling a BO with the problem configuration, close to the reality, defined in this study that has proposed a ML approach. Furthermore, it is observed from the literature that ML approaches can be used to capture complex processing environments in a way such that scheduling policies, in particular dispatching rules, can be derived (Benda et al., 2019). Thus, this study proposes a ML approach, particularly a back-propagation ANN to achieve more accurate parameter estimations in calculating the job-priority-data of the jobs and integrated the same with a few proposed dispatching rules based GHA for scheduling a BO problem defined in this study.

## 4 Proposed approaches

The research problem, on scheduling a BO, defined in this study is empirically shown to be NP hard by Mathirajan et al. (2010). Due to the computational intractability in getting optimal solution for real-life large sized instances, a hybrid neural network (HNN) algorithm is proposed in this study to find an efficient solution. The proposed HNN is an integration of the ANN approach and a simple DR-GHA. So, before presenting the proposed HNN algorithm, we first give a quick review of different dispatching rules considered for:

a developing GHA,

b applying a back-propagation ANN to achieve more accurate parameter estimations in calculating the job-priority-data of the jobs and in turn for scheduling the research problem considered in the study.

*Dispatching rules considered for scheduling a BO:* the analysis of the literature indicated that simple GHA, based on dispatching rules, are developed to obtain efficient solution as quick as possible in scheduling (Sarin et al., 2011). It is also noticed that dispatching rules are widely used in manufacturing industry, especially industry like SM (Hildebrandt et al., 2010). The reasons for the popularity could be due to fact that they perform reasonably well in a wide range of environments, relatively easy to understand and need only minimal computational time. In addition to that, they are easy to implement and can cope with dynamic changes (Nguyen et al., 2013). Accordingly, keeping the importance of various job characteristics associated with burn-in operations as well as the better performing dispatching rules as claimed in the literature, the following dispatching rules are used for developing multiple variants of greedy algorithms:

- *Highest job size (HJS):* it is a function of the job characteristic 'size ($S_j$)' of a job. As per HJS rule jobs available in-front of BO will be sorted, at the decision-making time epoch, based on size of the job (highest to lowest), and based on that the job-index is developed and assigned as follows:

$$\text{Job-Index}(j) = S_j$$

- *Longest processing time (LPT):* it is a function of the job-characteristic 'processing time ($P_j$)' of a job. Based on LPT rule jobs available in-front of BO will be sorted, at the decision-making time epoch, based on processing-time of the job (longest to least), and based on that the Job-Index is developed and assigned as follows:

$$\text{Job-Index}(j) = P_j$$

- *Earliest release time (ERT):* it is a function of the job-characteristic 'release time ($R_j$), at which job becomes available for processing'. According ERT rule, jobs available in-front of BO will be sorted, at the decision-making time epoch, based on release-time of the job (earliest to late), and based on that the Job-Index is developed and assigned as follows:

$$\text{Job-Index}(j) = R_j$$

- *Earliest due-date (EDD):* it is a function of the job characteristic 'due-date ($D_j$)'. As per EDD rule, jobs available in-front of BO will be sorted, at the decision-making time epoch, based on due-date of the job (earliest to late), and based on that the job-index is developed and assigned as follows:

$$\text{Job-Index}(j) = D_j$$

- *Flow due-date (FDD):* FDD is proposed by Jayamohan and Rajendran (2000). According to FDD rule jobs available in-front of BO will be sorted, at the decision-making time epoch, based on FDD of the job (least to longest), and based on that the job-index is developed and assigned as follows:

$$\text{Job-Index}(j) = R_j + \sum_{q=1}^{m} P_{jq}$$

where

$P_{jq}$ process time required for job 'j' for the operation 'q'

$m$ current operation.

*Note:* It is to be noted that this study concerns about only one operation: burn-in operation, the total processing time of job 'j' till the operation 'm' (here $m = 1$) is equal to the processing time of job 'j' ($P_j$) for burn-in operation. Accordingly, this study modified the job-index of job 'j' using FDD is as follows.

$$\text{Job-Index}(j) = R_j + P_j$$

- *Operational due-date (ODD):* Rose (2003) probably is the first one applied ODD as dispatching rule in scheduling SM problem. Based on ODD rule jobs available in-front of BO will be sorted, at the decision-making time epoch, based on ODD of the job (least to longest), and based on that the Job-Index is developed and assigned as follows:

$$\text{Job-Index}(j) = R_j + c * \sum_{q=1}^{m} P_{jq}$$

where

$c$ Due-date allowance factor (and it is assumed to be equal to 3).

*Note:* a note given for FDD is applicable to ODD also. So, the job-index becomes as follows.

$$\text{Job-Index}(j) = R_j + c * P_j$$

- *Latest start time (LST):* this rule is adopted from Mathirajan et al. (2010). LST is defined as the difference between the due-date of the job and the processing time of the job. This is fundamentally the latest time at which the job has to be started to process so that it can be finished before its due-date. Accordingly, the job-index is developed and assigned as follows:

$$\text{Job-Index}(j) = D_j - P_j$$

- *Composite index (CI):* this rule is adopted from Mathirajan et al. (2010). We introduce a CI that attempts to obtain the jobs that have early latest *start time*, a short *processing time* and a large *job size* first. With this, the job-index is developed and assigned as follows:

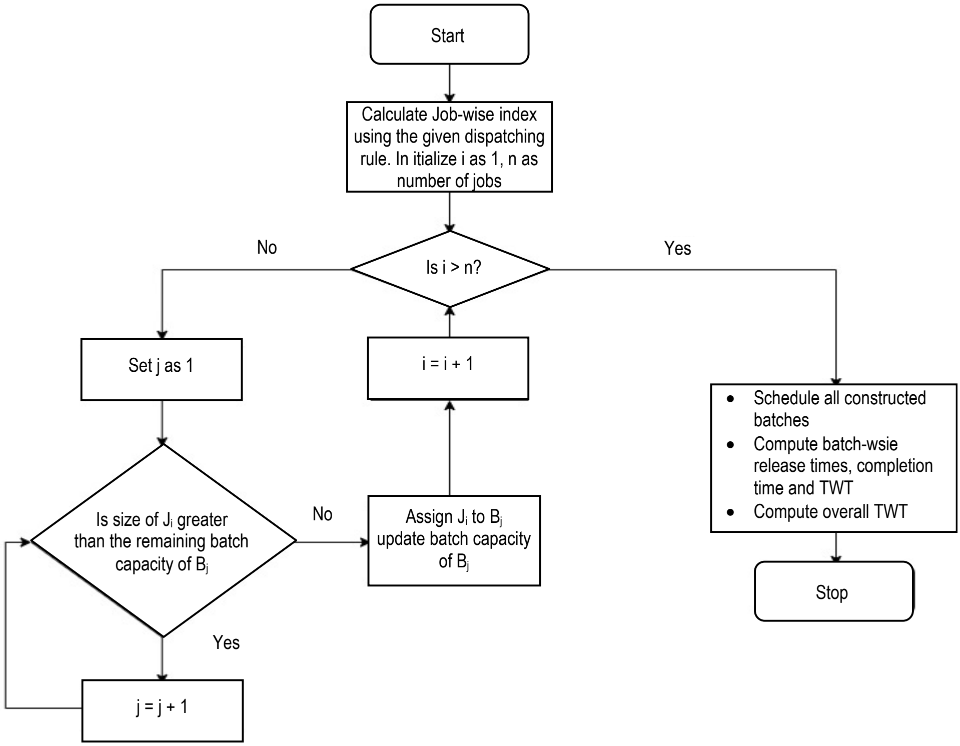
$$\text{Job-Index}(j) = \left[\left(D_j - P_j\right) \times P_j\right] / S_j$$

The first four dispatching rules (HJS, LPT, ERT, and EDD) considered in this study are simple (as we considered a single criterion to sort the available jobs) and static in nature (as they are not dependent on time). The dispatching rules: FDD, ODD, LST and CI are composite (as we considered multiple characteristics of a job as criterion to sort the available jobs) and static in nature.

## *4.1 Proposed DR-GHA*

A simple DR-GHA is developed by following five steps to schedule a batch for processing in BO. In the first step, by applying dispatching rule, every job is assigned an index, called as job-index. In the second step, if there is a job-priority-data for each job is given then job-index is computed appropriately using both dispatching rule and the job-priority-data and is called as job-priority-index. If job-priority-data is not given, then both job-index and job-priority-index are one and the same (or it is assumed that the job-priority-data for each job is given as same and constant). In the third step, the jobs are sequenced or sorted based on the computed job-priority-index. In the fourth step, a set of jobs are selected, to form batches parallel, from the top of the sorted-listed-jobs until the batch capacity constraint is satisfied. Finally, the constructed batches are scheduled in the BO. This five-step process is repeated until all the jobs are scheduled for the given planning period. The working mechanism of this simple DR-GHA is given in Figure 1. Considering each of the 8 different dispatching rules defined in this section we have coded these eight variants of DR-GHA in Python for scheduling a single BO problem defined in this study.

**Figure 1** Working mechanism of the proposed DR-GHA




## *4.2 Proposed HNN algorithm*

By and large, the quality of the dispatching rule based scheduling algorithm is expected to vary when the job-priority-data (nothing but a weight for penalising tardiness) is changed (Park et al., 2000). Due to that, this study integrates an application of ANN to enhance the performance of DR-GHA for scheduling a BO by learning and generating efficient job-priority-data. This integrated approach is called as HNN algorithm (Parsa et al., 2019).

That is, in the HNN, we applied a back-propagation ANN to achieve more accurate parameter estimations in calculating the job-priority-data for scheduling a BO by utilising the proposed DR-GHA. To get the benefit of HNN for scheduling a BO problem defined in this study, the ANN is integrated with the proposed DR-GHA, like the HNN proposed in Parsa et al. (2019). So, the proposed HNN is an extension to the existing HNN for scheduling a BO problem with six main differences (please refer to Table 4) between Parsa et al. (2019) approach and that of ours in this paper. However, all the important parameters of HNN: maximum number of epochs, learning rate, initial weights, and reinforcement factor are assumed to have the same values set in Parsa et al. (2019), as this study is also related to scheduling a BO. With these additional features on the existing HNN, the working mechanism of the HNN is as follows:

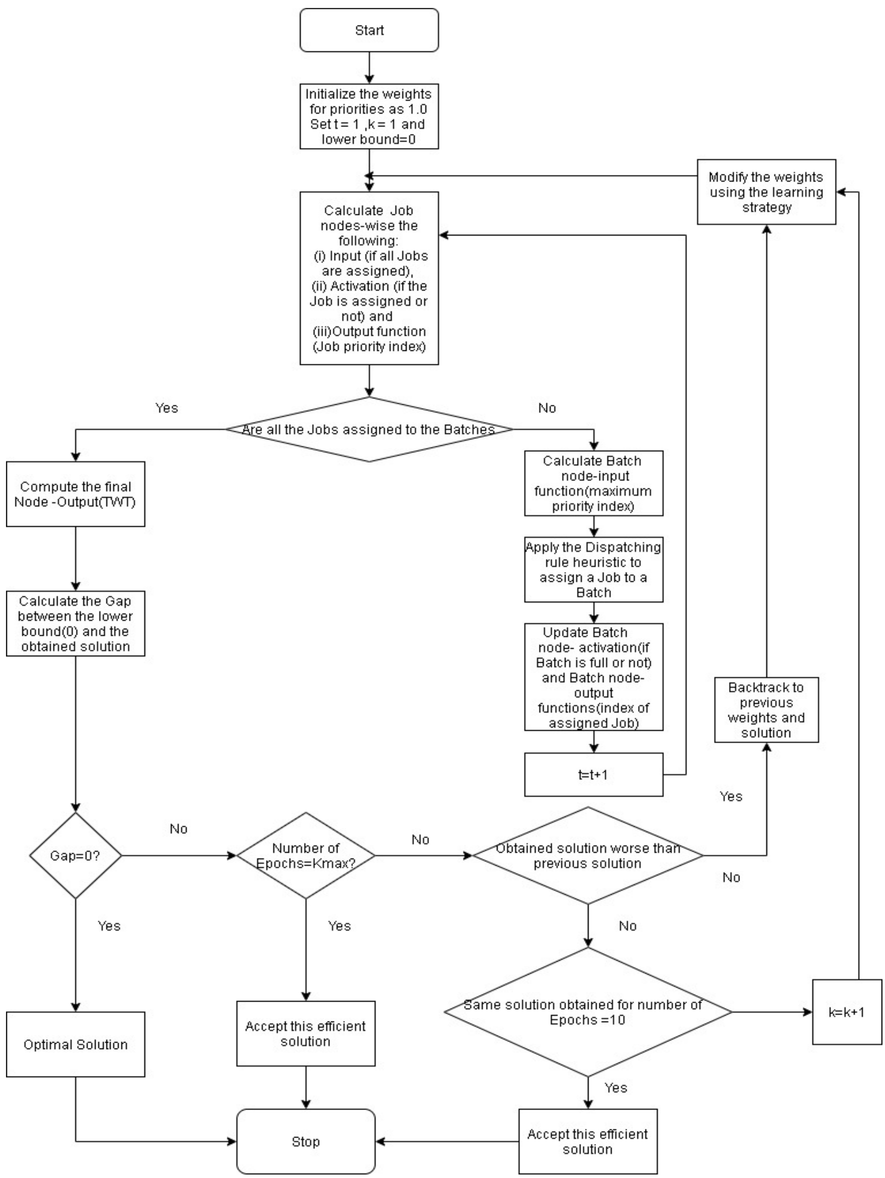
**Table 4** Difference between the proposed HNN and the existing HNN

| *No.* | *Current study* | *Study by Parsa et al. (2019)* |
|---|---|---|
| 1 | Introduced additional job-characteristics (close to real-life) such as different release time, different due-date, and non-agreeable release time(s) and due-date(s) requirement to describe the research problem on scheduling BO. | Job characteristics: processing time and non-identical job-sizes are only considered for scheduling BO. |
| 2 | Considered dynamic scheduling (that is, in this study jobs are having different release time and considering all the future arrivals of jobs until the decision-making epoch of the scheduling of BO) | Considered static scheduling (that is, jobs are having release time equal to zero or constant) |
| 3 | Considered customer perspective objective, that is minimisation of TWT. | Considered organisation perspective objective, that is minimisation of total completion time. |
| 4 | Applied a back-propagation ANN on each of the variants of GHAs (proposed for scheduling the research problem considered in this study) to achieve more accurate parameter estimations in calculating the job priority-data. With this there are eight variants of HNN developed for scheduling a BO. Furthermore, the performance evaluation of each of the proposed eight variants of:<br>a DR-GHA<br>b HNN is carried out in comparison with<br>1 optimal solution (on 160 small instances)<br>2 estimated optimal solution (on 640 large scale instances) obtained using the procedure given in Rardin and Uzsoy (2001). | Applied a back-propagation ANN on only one GHAs FFLPT (first-fit longest processing time) to achieve more accurate parameter estimations in calculating the job priority-data. And developed HNN algorithm for scheduling a BO. Furthermore, the performance evaluation of the proposed HNN algorithm is carried out in comparison with a set of existing GHAs and random algorithm, in comparison with the lower bound obtained using the procedure given in Uzsoy (1994). |
| 5 | Number of epochs and minimum TWT (= Zero), as there is a possibility of getting TWT as zero, are given as stopping conditions of the ANN. | Number of epochs and minimum total completion time (= lower bound obtained, using the procedure in Uzsoy (1994)) are given as stopping conditions of the ANN. |

In our study, initially, the job-priority-data (that is weight $W_j$) for all job are constant and equal to 1. Now, in the proposed HNN and at epoch = 1, the job-priority-index is computed based on the dispatching-rule-criterion w.r.t. DR-GHA and the job-priority-data. The computed job-priority-index is utilised in each of the iterations of the proposed DR-GHA for scheduling a BO. During the next epoch of the HNN, the weight $W_j$ are modified for all jobs based on ANN and recalculate all job-priority-index and re-apply the same DR-GHA. The modified weights will generate new schedule in the next epoch. If an improvement on TWT happens in an epoch, the ANN tends to reinforce the current weights and stores the best TWT known at this point. If no improvement on the TWT, the ANN continues without reinforcement.

Furthermore, if there is no improvement on TWT for a particular number of epochs (= 50), the ANN assumes that the last acceptance of the solution is not appropriate, so the ANN backtrack to the last best set of values of job-priority-data of the jobs (that is weights). Subsequently a certain number of epochs are implemented; the best TWT is stored, and reported. That is, in the proposed HNN we monitor improvement and reinforce the values of the job-priority-data of the jobs (that is weights) that lead to improvements and backtrack to the last set of best values of the job-priority-data of the jobs if no improvement achieved for a predefined number of epochs. The idea of reinforcement is a classical feature of neural network learning. By reinforcing the weights of the previous epochs applying a reinforcement factor, the relative weights are essentially retained for a few epochs and the probability of finding better weight combinations will be increased in the new few epochs.

**Figure 2** Working mechanism of the proposed HNN



It is clear from the above brief on the proposed HNN, after a number of epochs, it is expected that neural network learns a proper set of estimated values of the job-priority-data of each job that would generate efficient solution in comparison with applying only the DR-GHA without integrating it with the neural network. With these, the working mechanism of the HNN, explained briefly here, is represented in Figure 2. For more technical details on the working mechanism of the HNN the readers are requested to refer the study by Parsa et al. (2019) as this study exactly followed the technical aspects presented in that study. Furthermore, in the proposed HNN, we use eight different DR-GHA for embedding into the neural network. And in this process, we have eight different versions of HNN. Python is used to code all these eight variants of HNN for dynamic scheduling of a BO.

## 5 Computational experiments

To understand the efficiency of the proposed 16 approaches: eight proposed variants of DR-GHA and eight proposed variants of HNN, for dynamic scheduling of a BO, a series of computational experiments is carried out. To perform appropriate computational experiments, we need to have experimental design, benchmark solution (BS) procedure, and performance measure(s). The required details of these are presented in this section.

### 5.1 Experimental design

The aim of the experimental design is to generate the required suitable data for the problem defined under study. Accordingly, the two experimental designs presented in Mathirajan et al. (2010) are used for generating suitable data, as the problem configuration defined in the current study is exactly matching with the problem configuration defined in Mathirajan et al. (2010). Accordingly, the summary on the experimental designs considered in this study are given in Table 5 and Table 6.

**Table 5** A summary of experimental design for generating small scale problem instances

| *No.* | *Parameters* | *Levels* | *Number of levels* |
|---|---|---|---|
| 1 | Number of jobs (N) | 10, 12 | 2 |
| 2 | Release time of jobs ($R_j$) | Uniform distributions [1, 20], [1, 30] | 2 |
| 3 | Processing time of jobs ($P_j$) | Uniform distributions [1, 10], [1, 15] | 2 |
| 4 | Due-date of jobs ($D_j$) | $R_j + P_j$ + uniform distributions [1, 30], [1, 45] | 2 |
| 5 | Size of jobs ($S_j$) | Uniform distributions [4, 10], [4, 14] | 2 |
| Number of problem configurations | | | $2 \times 2 \times 2 \times 2 \times 2 = 32$ |
| Problem instances per configuration | | | 5 |
| Total problem instances | | | 160 |

*Source:* Mathirajan et al. (2010)

The experimental design summary given in Table 5 is used particularly for generating small scale problem instances, which are possible to solve and obtain optimal solution using exact approach in a reasonable computational time. Particularly, the experimental design summary given in Table 5 is used for two cases. In the *first case*, it is used to understand the closeness of the solution of each of the proposed 16 approaches: eight variants of DR-GHA and eight variants of HNN in comparison to optimal solution (obtained from the MILP model proposed in Mathirajan et al., 2010). In the *second case*, the quality of the estimated optimal solution (which could be used as BS for large sized real-life problem of burn-in operation and can be obtained for each of the problem instances using the procedure given in Rardin and Uzsoy, 2001) will be analysed in comparison to the optimal solution obtained on small scale instances. The experimental design summary given in Table 6 is used for generating large scale problem instances, which are possible to solve by each of the proposed 16 approaches within a reasonable computational time. These large sale instances are used to understand the performances of each of the proposed 16 approaches in comparison to estimated optimal solution.

**Table 6** A summary of experimental design for generating large scale problem instances

| *No.* | *Parameters* | *Levels* | *Number of Levels* |
|---|---|---|---|
| 1 | Number of jobs (N) | 25, 50, 75, 100 | 4 |
| 2 | Release time of jobs ($R_j$) | Uniform distributions [1, 20], [1, 30] | 2 |
| 3 | Processing time of jobs ($P_j$) | Uniform distributions [1, 10], [1, 15] | 2 |
| 4 | Due-date of jobs ($D_j$) | $R_j + P_j$ + uniform distributions [1, 30], [1, 45] | 2 |
| 5 | Size of jobs ($S_j$) | Uniform distributions [4, 10], [4, 14] | 2 |
| Number of problem configurations | | | 4 × 2 × 2 × 2 × 2 = 64 |
| Problem instances per configuration | | | 10 |
| Total problem instances | | | 640 |

*Source:* Mathirajan et al. (2010)

## 5.2 BS procedure

For small-scale problem, optimal solution obtained from the MILP model proposed in Mathirajan et al. (2010) is considered as a BS for performance evaluation of the proposed 16 approaches: eight variants of DR-GHA and eight variants of HNN. Whereas for large-scale problem, estimated optimal value is considered as BS.

## 5.3 Performance measures

Since the efficiency of each of the proposed approaches may vary over a range of problem instances, the efficiency is analysed both empirically and statistically. For empirical analysis, the performance measures: deviation/proximity, average relative percentage deviation (ARPD), and IRANK are used. For *statistical analysis*, this study conducts descriptive statistics and Kruskal-Wallis test. The details of these measures are given below.

- *Deviation/proximity:* for each problem instance and for each variant of DR-GHA and for each variant of HNN, we compute the 'deviation' as per equation (1).

$$D_{ij} = \left(FS_{ij} - BS_i\right) \tag{1}$$

where

- $i$ Problem instances and $i \in [1, 160]$ for small scale instances and $i \in [1, 640]$ for large scale instances.
- $j$ Proposed eight variants of DR-GHA and eight variants of HNN and $j \in [1, 16]$.
- $FS_{ij}$ Feasible solution obtained for $i^{th}$ problem instance using $j^{th}$ proposed approach.
- $BS_i$ Equal to optimal value of '$i^{th}$' problem instance, for small-scale problem.
- $BS_i$ Equal to estimated optimal value of '$i^{th}$' problem instance, for large-scale problem.
- $D_{ij}$ Deviation (or proximity) between the solution obtained from $j^{th}$ variant of the proposed algorithm and the optimal (benchmark) solution for the $i^{th}$ instance.

- *ARPD:* for each problem instance and for each variant of DR-GHA and for each variant of HNN, we first compute the relative percentage deviation as per equation (2):

$$RPD_{ij} = \left(D_{ij} / BS_i\right) * 100 \tag{2}$$

where

- $RPD_{ij}$ relative percentage deviation of '$j^{th}$' variant of the proposed algorithm for '$i^{th}$' problem instance

After computing RPD score, next we compute the average of RPD (ARPD) score for each of the proposed 16 approaches: eight variants of DR-GHA and eight variants of HNN over the number of problem instances planned in each problem configurations as well as the total number of problem instances in the experimental design as per the equation (3).

$$ARPD_j = \sum_{i=1}^{N} RPD_{ij} / N \tag{3}$$

where

- $ARPD_j$ average relative percentage deviation of '$j^{th}$' proposed variant.
- $N$ = 5 when ARPD is computed problem configuration wise for small scale problems.
- $N$ = 10 when ARPD is computed problem configuration wise for large scale problems.
- $N$ = 640 when ARPD is computed considering entire large scale problem instances.

- *IRANK:* for each of the proposed approaches, this study computes IRANK for triangulating the performance analyses using performance measures: deviation (for the small-scale instances) and ARPD (for large scale instances), as per the equation (4).

$$IRANK_j = \left(\sum_{r=1}^{Maxrank} \{a(r, j) * r\}\right) \Big/ \sum_{r=1}^{Maxrank} a(r, j) \tag{4}$$

where

$r$ rank and $r \in [1, 16]$

$j$ proposed approach $j \in [1, 16]$

$a(r, j)$ number of times the 'j$^{th}$' proposed approach in rank 'r'

*Maxrank* Maximum rank possible (*Maxrank* = 16 = number of proposed approaches).

$IRANK_j$ integrated rank of 'j$^{th}$' proposed approach.

For *statistical analysis*, this study first computes descriptive statistics (mean, median, standard deviation and 95% confidence interval), for each of the proposed 16 approaches; eight variants of DR-GHA and eight variants of HNN. As the normality test is failed over the obtained RPD scores, Kruskal-Wallis tests (a non-parametric test) on the medians are conducted to compare the proposed approaches (Beldar and Costa, 2018).

## *5.4 Experimental results and analysis*

For evaluating the performance of the proposed 16 approaches: eight variants of DR-GHA and eight variants of HNN for scheduling a BO problem, defined in this study, the following empirical and statistical performance analyses are carried out:

### *5.4.1 Empirical performance analyses*

In the empirical performance analyses, the proposed 16 approaches are evaluated in comparison with:

a optimal solution on 160 small scale instances

b estimated optimal solution on 640 large sale problem instances. The details of these empirical analyses are as follows:

*Performance analysis in comparison with optimal solution on small scale instances:* the 160 small scale problem instances generated and used by Mathirajan et al. (2010), including the optimal solution obtained, are used in this study for performance analysis of the proposed 16 approaches in comparison with optimal solution. Accordingly, each of the 160 problem instances are applied on each of the proposed 16 approaches and obtained feasible solution with TWT. Some of the small-scale instances, out of 160 instances, when applied with the MILP model proposed in Mathirajan et al. (2010) resulted the optimal solution as zero (that is zero TWT). Due to this, finding RPD scores on TWT in comparison with the optimal solution becomes 'infinite'. To avoid this, for understating the performance of the proposed 16 approaches w.r.t. optimal solution, we used the performance measures 'deviation/proximity', and IRANK. Accordingly, using equation (1) the 'deviation/proximity' score is obtained, for each of the proposed approaches and for each of the 160 problem instances, considering bench-mark solution as an optimal solution. Considering this score, we computed and presented (Table 7) for each of the proposed approaches:

1 number of times the optimal solution yielded

2 minimum, mean, maximum, and variance on the deviation score w.r.t. optimal solution over the 160 small scale instances.

Table 7, clearly indicates that each of the HNN variants are performing efficiently in comparison with its respective DR-GHA, which is not integrated with ANN. From this, particularly based on small scale problem instances, we infer that the back-propagation ANN is providing more accurate parameter estimations in calculating the job-priority-data for the jobs and in turn used in DR-GHA for obtaining efficient solution.

**Table 7** Performance of the proposed 16 approaches based on 'deviation/proximity' w.r.t. optimal solution

| *No.* | *Proposed approaches* | *Variants* | *No. of times yielded optimal solution* | *Deviation from optimal solution over 160 instances* | | | |
|---|---|---|---|---|---|---|---|
| | | | | *Minimum* | *Mean* | *Maximum* | *Variance* |
| 1 | GHA | HJS | 0 | 8 | 883.53 | 2,493 | 1,983.81 |
| 2 | | LPT | 0 | 30 | 887.73 | 2,808 | 2,521.37 |
| 3 | | ERT | 7 | 0 | 349.95 | 1,776 | 743.98 |
| 4 | | EDD | 18 | 0 | 353.44 | 2,105 | 1,034.17 |
| 5 | | FDD | 8 | 0 | 332.09 | 1,730 | 592.33 |
| 6 | | ODD | 5 | 0 | 383.1 | 1,509 | 530.72 |
| 7 | | LST | 16 | 0 | 376.68 | 2,456 | 1,109.56 |
| 8 | | CI | 2 | 0 | 937.84 | 3,087 | 2,324.7 |
| 9 | HNN | ANN-HJS | 114 | 0 | 65.52 | 883 | 77.46 |
| 10 | | ANN-LPT | 73 | 0 | 76.34 | 883 | 82.82 |
| 11 | | ANN-ERT | 133 | 0 | 58.55 | 888 | 76.72 |
| 12 | | ANN-EDD | 132 | 0 | 61.08 | 875 | 74.02 |
| 13 | | ANN-FDD | 134 | 0 | 57.56 | 875 | 74.58 |
| 14 | | ANN-ODD | 122 | 0 | 59.1 | 875 | 75.04 |
| 15 | | ANN-LST | 120 | 0 | 60.01 | 875 | 77.56 |
| 16 | | ANN-CI | 113 | 0 | 61.275 | 875 | 76.17 |

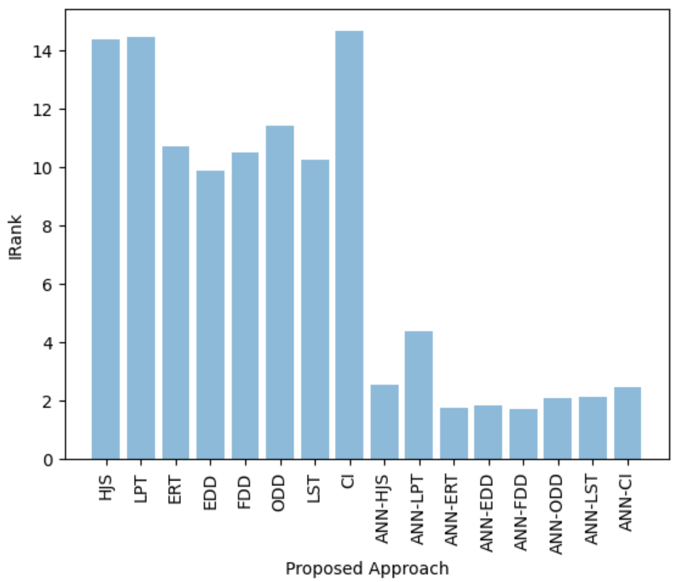
In addition, Table 7 clearly indicates, by and large, the proposed HNN variants: ANN-FDD and ANN-ODD are outperforming. The rule ODD seeks to minimise the deviation of completion times of a job from its operation due-date, and hence good performance for this rule is observed. Basically, the rule FDD defines a milestone for every operation of a job, and hence attempts to ensure the timely completion of operations. Further it is important to note that when the allowance factor, (i.e., c) is equal to 1, both FDD and ODD are same.

**Table 8** Ranking matrix – 160 small scale instances

| *Rank* | *Variant of DR-GHA based on the dispatching criterion* | | | | | | | | *Variant of HNN based on the integration of* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *HJS* | *LPT* | *ERT* | *EDD* | *FDD* | *ODD* | *LST* | *CI* | *ANN-HJS* | *ANN-LPT* | *ANN-ERT* | *ANN-EDD* | *ANN-FDD* | *ANN-ODD* | *ANN-LST* | *ANN-CI* |
| 1 | 0 | 0 | 7 | 18 | 8 | 5 | 16 | 2 | 114 | 73 | 133 | 132 | 134 | 122 | 120 | 113 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 4 | 2 | 3 | 3 | 4 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 | 4 | 3 | 5 | 3 |
| 4 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 6 | 5 | 5 | 5 | 3 | 4 | 3 | 4 |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 4 | 5 | 4 | 0 | 4 | 7 | 7 | 7 |
| 6 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 7 | 11 | 6 | 3 | 5 | 7 | 7 | 8 |
| 7 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 12 | 15 | 2 | 1 | 4 | 9 | 8 | 10 |
| 8 | 0 | 0 | 2 | 5 | 5 | 1 | 1 | 0 | 12 | 35 | 5 | 12 | 3 | 5 | 7 | 9 |
| 9 | 2 | 1 | 31 | 33 | 20 | 16 | 21 | 0 | 1 | 10 | 1 | 0 | 1 | 0 | 0 | 1 |
| 10 | 2 | 5 | 27 | 27 | 26 | 22 | 30 | 3 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 |
| 11 | 11 | 4 | 28 | 15 | 38 | 35 | 19 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 3 | 4 | 21 | 24 | 37 | 23 | 40 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 16 | 20 | 27 | 18 | 17 | 26 | 14 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 33 | 36 | 13 | 10 | 5 | 24 | 9 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 54 | 42 | 3 | 3 | 1 | 6 | 7 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 39 | 48 | 0 | 4 | 1 | 1 | 1 | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Integrated rank | 14.37 | 14.46 | 10.72 | 9.86 | 10.52 | 11.41 | 10.26 | 14.68 | 2.54 | 4.36 | 1.76 | 1.81 | 1.71 | 2.08 | 2.14 | 2.44 |
| | 14 | 15 | 12 | 9 | 11 | 13 | 10 | 16 | 7 | 8 | 2 | 3 | 1 | 4 | 5 | 6 |

To triangulate the inferences obtained based on the performance measure: Deviation/Proximity, we consider the performance measure: IRANK. To obtain IRANK for each of the proposed 16 approaches, first we computed ranking matrix considering the TWT scores obtained for each of the 160 instances using each of the proposed 16 approaches. The computed ranking matrix (16 approaches × 16 ranks) considering the 160 small scale instances is presented in Table 8. The value $a_{ij}$ in the ranking matrix (Table 8) represents number of times the $i^{th}$ proposed approach resulted $j^{th}$ ranking solution out of $j$ = 1, 2,…16 approaches. Using the ranking matrix, the IRANK score for each of the proposed 16 approaches is computed using the equation (4) and the same is given in Figure 3. Table 8 and Figure 3 clearly endorse the same inferences obtained based on the performance measure: deviation/proximity.

**Figure 3** Performance of the proposed approaches, on small instances, based on IRANK performance measure (see online version for colours)




In general, the findings obtained considering small-scale instances, on the performances of the proposed approaches, cannot be extrapolated for large scale instances for any conclusion and generalisation. So, this study carried out the performance analyses of the proposed approaches considering 640 large scale instances and the details of the same is discussed as follows:

*Performance analysis in comparison with estimated optimal solution:* writing a Python code and using the experiential design given in Table 6, 640 problem instances are generated. For each of the 640 problem instances, each of the proposed approaches are applied and obtained feasible solution with TWT. For each of the 640 problem instances the estimated optimal solution (TWT) is computed using the:

a 16 TWT values available (obtained from each of the 16 proposed approaches) for each of the 640 problem instances

b procedure discussed in Rardin and Uzsoy (2001).

**Table 9** (a) Performance of the proposed approaches w.r.t. estimated optimal solution and $N = 25$

| *Problem configuration* | *Variant of DR-GHA based on dispatching criterion* | | | | | | | | *Variant of HNN based on the integration of* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *HJS* | *LPT* | *ERT* | *EDD* | *FDD* | *ODD* | *LST* | *CI* | *ANN-HJS* | *ANN-LPT* | *ANN-ERT* | *ANN-EDD* | *ANN-FDD* | *ANN-ODD* | *ANN-LST* | *ANN-CI* |
| 25_20_10_30_6 | 232.99 | 269.2 | 132.03 | 183.2 | 100.2 | 100.8 | 184.3 | 178.1 | 58.13 | 17.73 | 84.22 | 29.86 | 10.76 | 24.7 | 73.62 | 63.04 |
| 25_20_10_30_10 | 156.52 | 172.9 | 78.96 | 89.44 | 54.26 | 49.8 | 94.05 | 82.4 | 24.49 | 11.47 | 18.73 | 12.59 | 9.58 | 6.95 | 26.89 | 31.19 |
| 25_20_10_45_6 | 725.51 | 685.5 | 377.39 | 311.6 | 296 | 260.5 | 382.9 | 245.2 | 207.5 | 118.9 | 326.3 | 9.57 | 71.78 | 93.6 | 204 | 178.9 |
| 25_20_10_45_10 | 227.76 | 284 | 128.48 | 156.7 | 112.4 | 106.8 | 155.1 | 126.3 | 34.72 | 19.52 | 40.57 | 29.74 | 15.85 | 22.2 | 41.01 | 38.83 |
| 25_20_15_30_6 | 134.62 | 141.2 | 112.2 | 99.18 | 62.78 | 36.96 | 100.7 | 103.3 | 27.79 | 21.43 | 20.14 | 18.59 | 15.71 | 10.6 | 38.24 | 39.57 |
| 25_20_15_30_10 | 118.27 | 132.3 | 77.75 | 65.09 | 52.67 | 38.42 | 93.13 | 79.21 | 23.55 | 16.29 | 6.51 | 16.67 | 9.54 | 11.9 | 21.33 | 24.43 |
| 25_20_15_45_6 | 275.1 | 276.8 | 181.45 | 154.9 | 82.42 | 50.15 | 201.9 | 189.6 | 58.88 | 41.95 | 75.26 | 35.29 | 12.83 | 5.04 | 58.54 | 62.74 |
| 25_20_15_45_10 | 137.99 | 133.3 | 91.17 | 87.77 | 58.8 | 47.64 | 90.78 | 98.42 | 28.55 | 11.37 | 7.21 | 18.26 | 7.28 | 11.8 | 22.08 | 25.11 |
| 25_30_10_30_6 | 431.01 | 518.7 | 194.59 | 289.6 | 164.6 | 237.4 | 274.6 | 242 | 127 | 25.7 | 119.1 | 44.96 | 31.18 | 57 | 133 | 118.5 |
| 25_30_10_30_10 | 235.15 | 295.4 | 116.83 | 164.9 | 105.6 | 107 | 143 | 127.3 | 45.4 | 9.79 | 32.96 | 30.59 | 8.16 | 26 | 50.67 | 44.43 |
| 25_30_10_45_6 | 809.19 | 956.4 | 334.32 | 294.9 | 373.5 | 325.9 | 308.8 | 329.7 | 257.5 | 94.14 | 235 | 31.77 | 97.32 | 90.9 | 198.5 | 247 |
| 25_30_10_45_10 | 504.65 | 511 | 142.07 | 250.5 | 132.1 | 193.2 | 232.6 | 250.3 | 96.5 | 21.79 | 117.4 | 11.31 | 25.22 | 29.5 | 91.87 | 104.8 |
| 25_30_15_30_6 | 150.02 | 158.4 | 69.91 | 97.73 | 64.27 | 59.86 | 104.8 | 96.4 | 35.96 | 4.56 | 20.76 | 17.42 | 10.19 | 22.2 | 38.69 | 34.58 |
| 25_30_15_30_10 | 105.96 | 132.8 | 62.27 | 76.76 | 65.27 | 51.01 | 82.88 | 66.74 | 19.99 | 6.77 | 8.51 | 13.18 | 9.23 | 9.57 | 22.12 | 21.41 |
| 25_30_15_45_6 | 309.87 | 344.7 | 123.82 | 130.1 | 132.6 | 108.6 | 157.2 | 170.7 | 62.65 | 12.92 | 29.64 | 26.28 | 15.61 | 29.1 | 82.1 | 76.1 |
| 25_30_15_45_10 | 214.82 | 219 | 94.39 | 123.3 | 90.66 | 80.24 | 141.8 | 100.3 | 48.52 | 3.94 | 37.22 | 34.33 | 19.22 | 18.8 | 46.93 | 42.21 |
| *ARPD over 160 instances* | 298.09 | 327 | 144.85 | 161 | 121.7 | 115.9 | 171.8 | 155.4 | 72.32 | 27.39 | 73.72 | 23.78 | 23.09 | 29.4 | 71.85 | 72.05 |

**Table 9** (b) Performance of the proposed approaches w.r.t. estimated optimal solution and $N = 50$

| | *Variant of DR-GHA based on dispatching criterion* | | | | | | | | *Variant of HNN based on the integration of* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *HJS* | *LPT* | *ERT* | *EDD* | *FDD* | *ODD* | *LST* | *CI* | *ANN-HJS* | *ANN-LPT* | *ANN-ERT* | *ANN-EDD* | *ANN-FDD* | *ANN-ODD* | *ANN-LST* | *ANN-CI* |
| 50_20_10_30_6 | 115.26 | 120.2 | 95.46 | 89.79 | 67.76 | 39.15 | 105.6 | 96.23 | 51.24 | 48.74 | 81.93 | 42.71 | 25.64 | 5.59 | 58.49 | 57.06 |
| 50_20_10_30_10 | 93.19 | 113 | 86.24 | 72.25 | 63.32 | 28.16 | 95.7 | 74.4 | 37.42 | 40.03 | 61.4 | 33.03 | 23.29 | 4.36 | 46.99 | 46.51 |
| 50_20_10_45_6 | 139.7 | 138.7 | 106.76 | 113.4 | 73.66 | 29.51 | 118.9 | 116.5 | 64.65 | 55.07 | 100.2 | 52.64 | 31.37 | 5.68 | 64.62 | 68.18 |
| 50_20_10_45_10 | 85.48 | 102.4 | 69.97 | 68.1 | 52.3 | 24.31 | 80.67 | 73.39 | 26.62 | 28.99 | 51.71 | 27.98 | 14.5 | 1.33 | 36.82 | 38.41 |
| 50_20_15_30_6 | 104.74 | 102 | 93.12 | 76.2 | 52.34 | 20.17 | 98.52 | 90.96 | 51.77 | 46.98 | 83.74 | 37.82 | 18.8 | 2.25 | 52.4 | 50.75 |
| 50_20_15_30_10 | 70.56 | 92.43 | 58.99 | 56.31 | 28.9 | 10.97 | 74.79 | 57.87 | 29.23 | 27.99 | 29.08 | 21.97 | 11.34 | 0.66 | 35.06 | 33.25 |
| 50_20_15_45_6 | 106.03 | 110.8 | 97.03 | 85.82 | 56.55 | 22.03 | 110.8 | 96.76 | 48.32 | 54.09 | 86.2 | 42.85 | 23.37 | 3.53 | 55.74 | 59.01 |
| 50_20_15_45_10 | 86.21 | 99.49 | 69.22 | 65.72 | 37.58 | 15.58 | 85.7 | 69.93 | 34.74 | 34.37 | 37.18 | 25.38 | 15.69 | 1.45 | 38.77 | 40.58 |
| 50_30_10_30_6 | 138.17 | 144.6 | 67.66 | 84.74 | 52.75 | 37.06 | 91.52 | 87.07 | 56.05 | 31.55 | 96.41 | 25.84 | 15.92 | 1.75 | 63.38 | 63.77 |
| 50_30_10_30_10 | 85.23 | 103.2 | 56.57 | 48.42 | 40.4 | 27.87 | 59.17 | 51.44 | 27.32 | 14.77 | 45.18 | 14.65 | 10.07 | 1.54 | 41.17 | 40.35 |
| 50_30_10_45_6 | 120.48 | 134.1 | 67.3 | 66.33 | 58.36 | 26.15 | 78.58 | 64.38 | 49.58 | 27.28 | 86.67 | 17.74 | 13.15 | 1.65 | 47.5 | 50.21 |
| 50_30_10_45_10 | 96.77 | 124.7 | 54.15 | 62.84 | 45.46 | 26.22 | 70.2 | 50.19 | 27.35 | 24.92 | 54.66 | 17.11 | 12.21 | 1.24 | 39.95 | 40.32 |
| 50_30_15_30_6 | 110.81 | 119.8 | 88.14 | 93.62 | 60.7 | 31.18 | 106.7 | 94.38 | 47.2 | 46.61 | 75.71 | 32.62 | 26.67 | 4.24 | 57.96 | 59.29 |
| 50_30_15_30_10 | 68.78 | 80.45 | 53.35 | 43.12 | 36.97 | 16.12 | 57.59 | 52.62 | 26.06 | 19.28 | 24.22 | 16.49 | 11.23 | 2.11 | 27.9 | 28.66 |
| 50_30_15_45_6 | 120.06 | 121.9 | 74.52 | 82.01 | 53.37 | 25.54 | 86.48 | 75.18 | 48.3 | 37.24 | 84.68 | 31.95 | 17.28 | 2.28 | 54.92 | 56.91 |
| 50_30_15_45_10 | 88.11 | 97.11 | 65.63 | 50.52 | 45.82 | 20.07 | 55.77 | 55.08 | 32.27 | 20.5 | 27.46 | 20.65 | 16.1 | 2.32 | 34.32 | 38.56 |
| *ARPD over 160 instances* | 101.85 | 112.8 | 75.257 | 72.45 | 51.64 | 25.01 | 86.04 | 75.4 | 41.13 | 34.9 | 64.15 | 28.84 | 17.91 | 2.62 | 47.25 | 48.24 |

**Table 9** (c) Performance of the proposed approaches w.r.t. estimated optimal solution and $N = 75$

| *Problem configuration* | *Variant of DR-GHA based on dispatching criterion* | | | | | | | | *Variant of HNN based on the integration of* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *HJS* | *LPT* | *ERT* | *EDD* | *FDD* | *ODD* | *LST* | *CI* | *ANN-HJS* | *ANN-LPT* | *ANN-ERT* | *ANN-EDD* | *ANN-FDD* | *ANN-ODD* | *ANN-LST* | *ANN-CI* |
| 75_20_10_30_6 | 101.41 | 99.86 | 81.63 | 69.95 | 56.9 | 21.34 | 90.14 | 80.06 | 59.57 | 53.72 | 76 | 46.32 | 30.82 | 4.55 | 58.77 | 63.83 |
| 75_20_10_30_10 | 76.59 | 90.36 | 66.26 | 58.81 | 45.51 | 18.28 | 69.6 | 62.41 | 39.97 | 45.57 | 69.61 | 36.11 | 26.72 | 4.16 | 49.15 | 46.13 |
| 75_20_10_45_6 | 120.37 | 111.7 | 101.24 | 97.74 | 74.98 | 37.65 | 106.5 | 101.3 | 71.86 | 70.66 | 87.61 | 63.33 | 46.74 | 14 | 72.15 | 78.1 |
| 75_20_10_45_10 | 80.72 | 91.15 | 69.88 | 66.2 | 49.91 | 23.64 | 77.38 | 67.4 | 43.38 | 47.52 | 73.12 | 43.34 | 28.57 | 6.38 | 45.7 | 47.28 |
| 75_20_15_30_6 | 90.74 | 95.94 | 90.87 | 71.57 | 52.49 | 17.68 | 93.19 | 88.95 | 60.35 | 65.23 | 79.93 | 47.61 | 31.32 | 3.34 | 64.48 | 63.64 |
| 75_20_15_30_10 | 73.77 | 83.75 | 71.17 | 59.26 | 41.57 | 15.92 | 75.38 | 65.96 | 37.45 | 48.57 | 72.21 | 35.9 | 23.86 | 3.23 | 45.2 | 44.68 |
| 75_20_15_45_6 | 101.8 | 102.1 | 101.99 | 86.68 | 51.32 | 15.69 | 111.1 | 100.8 | 63.25 | 68.52 | 85.28 | 58.57 | 28.93 | 2.49 | 72.13 | 72.28 |
| 75_20_15_45_10 | 84.49 | 97.97 | 77.85 | 75.56 | 47.14 | 20.3 | 85.84 | 78.47 | 46.58 | 57.06 | 82.79 | 52.54 | 28.87 | 4.41 | 50.53 | 54.46 |
| 75_30_10_30_6 | 88.48 | 89.53 | 65.82 | 69.45 | 51.6 | 23.38 | 77.79 | 63.94 | 44.26 | 39.28 | 60.79 | 36.35 | 26.14 | 5.12 | 52.51 | 54.64 |
| 75_30_10_30_10 | 69.69 | 80.81 | 47.37 | 55.79 | 38.13 | 25.05 | 57.83 | 49.66 | 30.55 | 30.44 | 61.17 | 26.9 | 19.38 | 4.56 | 35.61 | 38.01 |
| 75_30_10_45_6 | 104.83 | 104.4 | 67.37 | 66.08 | 52.16 | 20.57 | 75.33 | 74.44 | 54.1 | 41.11 | 75.53 | 37.19 | 24.76 | 4.21 | 59.6 | 58.2 |
| 75_30_10_45_10 | 83.18 | 98.52 | 60.82 | 63.48 | 43.5 | 22.14 | 68.99 | 58.64 | 43.22 | 40.3 | 71.26 | 33.91 | 24.48 | 4.53 | 43.91 | 41.03 |
| 75_30_15_30_6 | 100.32 | 91.72 | 75.1 | 66.32 | 54.11 | 23.45 | 86.51 | 77.15 | 61.2 | 52.98 | 76.12 | 42.24 | 29.95 | 5.22 | 59.41 | 58.69 |
| 75_30_15_30_10 | 74.15 | 85.03 | 68.42 | 55.34 | 47.11 | 21.78 | 72.62 | 64.8 | 37.54 | 45.32 | 70.4 | 33.84 | 28.46 | 5.31 | 43.65 | 43.89 |
| 75_30_15_45_6 | 95.62 | 94.86 | 78 | 71.48 | 55.32 | 23.99 | 90.77 | 77.04 | 55.19 | 51.92 | 71.42 | 47.65 | 30.96 | 5.29 | 57.41 | 59.57 |
| 75_30_15_45_10 | 79.9 | 102.4 | 73.13 | 63.93 | 52.22 | 26.23 | 78.5 | 70.57 | 42.79 | 50.13 | 79.48 | 40.46 | 29.92 | 6.36 | 50.12 | 53.27 |
| *ARPD over 160 instances* | 89.129 | 95 | 74.808 | 68.6 | 50.87 | 22.32 | 82.34 | 73.85 | 49.45 | 50.52 | 74.55 | 42.64 | 28.74 | 5.19 | 53.77 | 54.86 |

**Table 9** (d) Performance of the proposed approaches w.r.t. estimated optimal solution and $N = 100$

| *Problem configuration* | *Variant of DR-GHA based on dispatching criterion* | | | | | | | | *Variant of HNN based on the integration of* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *HJS* | *LPT* | *ERT* | *EDD* | *FDD* | *ODD* | *LST* | *CI* | *ANN-HJS* | *ANN-LPT* | *ANN-ERT* | *ANN-EDD* | *ANN-FDD* | *ANN-ODD* | *ANN-LST* | *ANN-CI* |
| 100_20_10_30_6 | 127.09 | 121.4 | 115.09 | 107.9 | 86.27 | 51.94 | 121.4 | 113.7 | 87.06 | 90.46 | 102.9 | 82.63 | 65.82 | 30.7 | 94.72 | 95.63 |
| 100_20_10_30_10 | 66.64 | 79.09 | 57.93 | 54.35 | 38.63 | 8.84 | 59.49 | 55.63 | 40.69 | 43.72 | 64.66 | 37.07 | 23.43 | 1.14 | 41.44 | 42.63 |
| 100_20_10_45_6 | 82.19 | 80.15 | 72.87 | 75.78 | 54.6 | 17.53 | 80.22 | 75.33 | 52.47 | 51.02 | 63.11 | 54.06 | 32.09 | 4.05 | 57.44 | 58.77 |
| 100_20_10_45_10 | 69.38 | 80.15 | 61.3 | 54.66 | 38.39 | 11.01 | 63.19 | 59.53 | 42.12 | 42.41 | 64.23 | 39.91 | 23.8 | 1.61 | 45.4 | 45.13 |
| 100_20_15_30_6 | 81.69 | 79.06 | 76.97 | 64.95 | 41.04 | 10.53 | 80.54 | 75.61 | 53.5 | 57.45 | 65.53 | 47.61 | 27.04 | 1.65 | 58.22 | 58.82 |
| 100_20_15_30_10 | 63.29 | 75.05 | 62.88 | 44.37 | 34.42 | 7.42 | 61.11 | 58.9 | 41.51 | 45.42 | 66.58 | 31.68 | 21.05 | 0.89 | 43.63 | 44.07 |
| 100_20_15_45_6 | 82.93 | 84.03 | 86.34 | 76.58 | 46.26 | 11.91 | 93.21 | 88.35 | 57.95 | 62.72 | 70.77 | 56.85 | 29.8 | 1.7 | 63.77 | 63.7 |
| 100_20_15_45_10 | 68.41 | 81.21 | 73.07 | 54.92 | 39.48 | 10.39 | 71.88 | 61.53 | 46.16 | 52.65 | 71.95 | 40.59 | 22.39 | 1.46 | 48.44 | 49.4 |
| 100_30_10_30_6 | 74.81 | 74.41 | 59.56 | 55.16 | 43.59 | 18.08 | 63.48 | 60.2 | 41.23 | 39.52 | 55.59 | 38.96 | 26.16 | 4.52 | 50.44 | 49.59 |
| 100_30_10_30_10 | 54.42 | 65.93 | 45.16 | 41.27 | 32.85 | 12.58 | 49.13 | 45.73 | 29.42 | 30.52 | 50.27 | 26.77 | 19.82 | 2.87 | 31.12 | 30.31 |
| 100_30_10_45_6 | 81.82 | 86.67 | 67.72 | 66.04 | 51.69 | 25.56 | 68.74 | 63.33 | 48.74 | 46.21 | 61.48 | 47.62 | 32.72 | 9.46 | 58.1 | 56.94 |
| 100_30_10_45_10 | 58.21 | 62.16 | 46.33 | 46.36 | 34.47 | 12.45 | 52.91 | 42.26 | 27.64 | 30.5 | 48.82 | 32.36 | 20.42 | 2.9 | 32.39 | 31.7 |
| 100_30_15_30_6 | 77.36 | 78.3 | 69.54 | 59.58 | 45.69 | 14.26 | 76.69 | 71.99 | 48.52 | 50.4 | 60.53 | 40.44 | 28.76 | 2.85 | 55.31 | 55.08 |
| 100_30_15_30_10 | 65.39 | 75.73 | 56.46 | 44.61 | 36.76 | 11.72 | 55.12 | 50.93 | 40.86 | 43.76 | 64.77 | 31.27 | 21.96 | 1.96 | 40.6 | 41.64 |
| 100_30_15_45_6 | 75.09 | 75.29 | 75.03 | 61.25 | 45.48 | 17.65 | 70.99 | 63.38 | 44.53 | 50.35 | 62.58 | 44.29 | 29.93 | 4.12 | 53.15 | 55.02 |
| 100_30_15_45_10 | 63.22 | 77.62 | 57.14 | 56.82 | 36.84 | 10.29 | 66.3 | 57.08 | 39.87 | 40.55 | 62.85 | 40.36 | 24.29 | 1.46 | 43.04 | 39.91 |
| *ARPD over160 instances* | 74.496 | 79.76 | 67.712 | 60.29 | 44.15 | 15.76 | 70.9 | 65.22 | 46.39 | 48.6 | 64.79 | 43.28 | 28.09 | 4.59 | 51.08 | 51.15 |

**Table 10** Ranking matrix – 640 large scale instances

| Rank | *Variant of DR-GHA based on the dispatching criterion* | | | | | | | | *Variant of HNN based on the integration of* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *HJS* | *LPT* | *ERT* | *EDD* | *FDD* | *ODD* | *LST* | *CI* | *ANN-HJS* | *ANN-LPT* | *ANN-ERT* | *ANN-EDD* | *ANN-FDD* | *ANN-ODD* | *ANN-LST* | *ANN-CI* |
| 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 6 | 40 | 21 | 30 | 42 | 501 | 1 | 2 |
| 2 | 0 | 0 | 0 | 0 | 0 | 324 | 1 | 0 | 10 | 36 | 23 | 43 | 161 | 33 | 1 | 6 |
| 3 | 0 | 0 | 1 | 2 | 4 | 71 | 1 | 2 | 18 | 42 | 21 | 80 | 347 | 37 | 10 | 6 |
| 4 | 0 | 0 | 0 | 9 | 84 | 30 | 0 | 1 | 81 | 82 | 26 | 194 | 55 | 26 | 32 | 27 |
| 5 | 0 | 0 | 3 | 10 | 68 | 28 | 1 | 7 | 115 | 117 | 31 | 125 | 21 | 17 | 49 | 49 |
| 6 | 1 | 0 | 7 | 18 | 67 | 33 | 0 | 6 | 125 | 100 | 17 | 63 | 9 | 13 | 91 | 95 |
| 7 | 0 | 0 | 8 | 20 | 68 | 17 | 2 | 8 | 102 | 85 | 19 | 49 | 4 | 10 | 129 | 115 |
| 8 | 3 | 0 | 15 | 13 | 68 | 24 | 4 | 21 | 71 | 66 | 28 | 33 | 1 | 3 | 138 | 139 |
| 9 | 2 | 0 | 26 | 29 | 81 | 41 | 7 | 34 | 65 | 51 | 50 | 17 | 0 | 0 | 114 | 114 |
| 10 | 13 | 7 | 73 | 69 | 85 | 23 | 25 | 55 | 31 | 17 | 78 | 4 | 0 | 0 | 51 | 54 |
| 11 | 28 | 6 | 131 | 130 | 55 | 15 | 71 | 109 | 8 | 1 | 77 | 1 | 0 | 0 | 14 | 24 |
| 12 | 49 | 30 | 100 | 125 | 27 | 12 | 91 | 131 | 7 | 3 | 56 | 1 | 0 | 0 | 5 | 6 |
| 13 | 63 | 21 | 106 | 102 | 18 | 11 | 134 | 127 | 1 | 0 | 74 | 0 | 0 | 0 | 4 | 2 |
| 14 | 100 | 58 | 96 | 71 | 11 | 7 | 138 | 83 | 0 | 0 | 82 | 0 | 0 | 0 | 1 | 1 |
| 15 | 208 | 168 | 49 | 27 | 3 | 4 | 106 | 38 | 0 | 0 | 37 | 0 | 0 | 0 | 0 | 0 |
| 16 | 173 | 350 | 25 | 13 | 1 | 0 | 59 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Integrated ranks | 14.35 | 15.16 | 12.03 | 11.34 | 7.92 | 4.44 | 13.2 | 11.88 | 6.39 | 5.55 | 9.78 | 4.62 | 2.84 | 1.62 | 7.46 | 7.52 |
| | 15 | 16 | 13 | 11 | 9 | 3 | 14 | 12 | 6 | 5 | 10 | 4 | 2 | 1 | 7 | 8 |

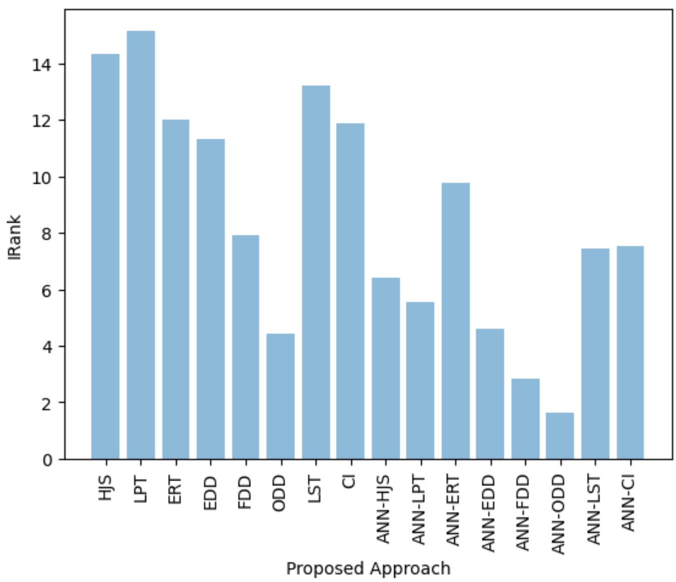
Now for each of the proposed 16 approaches, considering the

1 feasible TWT scores

2 estimated optimal TWT score obtained for each of the 640 problem instances, the RPD scores are obtained using equation (2).

Using the computed RPD scores of each of the proposed 16 approaches, the ARPD score on TWT is computed using equation (3) over ten problem instances as per the problem configuration defined in Table 6. Proposed approach wise and for each value of $N$ (that is $N$ = 25, $N$ = 50, $N$ = 75, and $N$ =100) the computed ARPD scores is presented in Tables 9(a) to 9(d). Furthermore, the proposed approach wise the ARPD scores over 640 instances, representing irrespective of the problem configurations, are computed and represented in Figure 4.

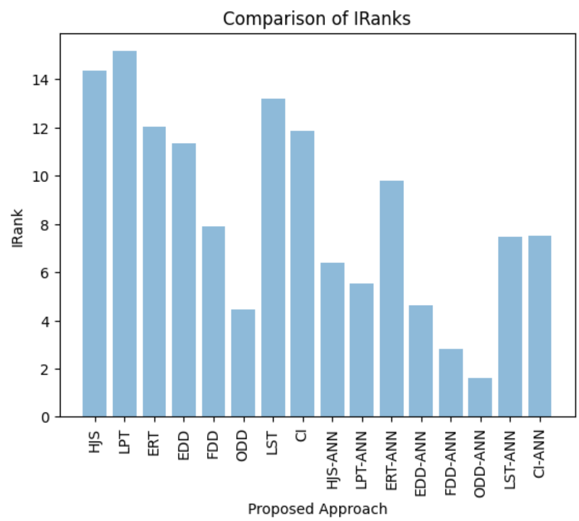
**Figure 4** Performance of the proposed approaches based on ARPD score over 640 large instances (see online version for colours)




From Tables 9(a) to 9(d) and Figure 4, it is observed that each of the HNN variants is performing efficiently than the respective DR-GHA. This clearly shows, similar to the small-scale instances, the power of back-propagation ANN for achieving more accurate parameter estimations in calculating the job-priority-data for each of jobs and in turn providing efficient scheduling of BO using HNN. Further, from Tables 9(a) to 9(d), it is observed that when the problem size increases the HNN variant: ANN-ODD is outperforming. This is even true from Figure 4. In addition, from the results, as a second-best choice, one can choose the HNN variant: ANN-FDD. Furthermore, for triangulating the findings observed from the ARPD analysis presented in Figure 4, the performance measure: IRANK is used. To compute IRANK, first, a [16 × 16] ranking matrix is developed and presented in Table 10 in connection with 640 problem instances. Using the ranking matrix (Table 10) related to 640 instances, the IRANK is computed using equation (4) and the same is presented in Figure 5. From Figure 5, the proposed

HNN variant: ANN-ODD is outperforming and followed with ANN-FDD. This observation endorses with the findings observed from the ARPD analysis presented in Tables 9(a) to 9(d) and Figure 4. The better performing approaches identified based on the empirical analyses considering 160 small scale instances (in comparison with optimal solution) and 640 problem instances (in comparison with estimated optimal solution) are further verified by conducting statistical analysis and the same is discussed in the next section.

**Figure 5** Performance of the proposed approaches, on large instances, based on IRANK performance measure (see online version for colours)




### *5.4.2 Statistical analyses on the performance of the proposed 16 approaches*

To understand statistically the performance of the proposed 16 approaches this study conducts statistical analyses: descriptive statistics and Kruskal-Wallis test (a non-parametric test), considering 160 small scale instances and 640 large scale instances, as follows:

#### *5.4.2.1 Statistical analyses considering 160 small scale instances*

For each of the 16 proposed approaches the 160 TWT values obtained (considering 160 small scale instances) are used to compute the descriptive statistics: mean, median, standard deviation and 95% confidence interval. The computed descriptive scores are presented in Table 11 to statistically analyse the performance of the proposed approaches. It is observed from Table 11 that:

a each of the eight variants of the proposed HNN is performing better than each of its DR-GHA which is not integrated with ANN

b within the proposed eight variants of HNN, the variant HNN: ANN-ODD is outperforming in comparison with the other variants.

**Table 11** Descriptive statistics of proposed variants-small scale problem instances

| *Proposed variants based on* | *Descriptive statistics* | | | |
|---|---|---|---|---|
| | *Mean* | *Median* | *Standard deviation* | *95% confidence interval* |
| HJS | 894.80625 | 697 | 630.18886 | (797.16, 992.45) |
| LPT | 949.0125 | 707.5 | 705.99577 | (839.62, 1,058.41) |
| ERT | 416.225 | 280 | 419.8741 | (351.17, 481.28) |
| EDD | 414.7125 | 247 | 478.93781 | (340.5, 488.92) |
| FDD | 393.36875 | 269 | 378.00043 | (334.8, 451.94) |
| ODD | 445.2125 | 352.5 | 363.28208 | (388.92, 501.5) |
| LST | 437.9625 | 264.5 | 502.54655 | (360.09, 515.83) |
| CI | 999.11875 | 945.5 | 675.60935 | (894.43, 1,103.8) |
| ANN-HJS | 126.8 | 56 | 203.64795 | (95.25, 158.35) |
| ANN-LPT | 137.61875 | 69.5 | 205.68725 | (105.75, 169.49) |
| ANN-ERT | 119.83125 | 55 | 199.48731 | (88.92, 150.74) |
| ANN-EDD | 122.35625 | 55.5 | 199.01772 | (91.52, 153.19) |
| ANN-FDD | 118.8375 | 55.5 | 198.01448 | (88.16, 149.52) |
| ANN-ODD | 120.36875 | 53.5 | 199.18597 | (89.51, 151.23) |
| ANN-LST | 121.2875 | 54.5 | 200.6667 | (90.19, 152.38) |
| ANN-CI | 122.55 | 55.5 | 198.44258 | (91.8, 153.3) |

**Table 12** Kruskal-Wallis test on deviation values w.r.t optimal solution – small scale instances

| *Test statistics*[a,b] | | | |
|---|---|---|---|
| | | *Deviation* | |
| Kruskal-Wallis H | | 1,417.005 | |
| df | | 15 | |
| Asymp. sig. | | 01.000 | |
| *Ranks* | | | |
| | *Variants of algorithm* | *N* | *Mean rank* |
| Deviation | HJS | 160 | 2,113.33 |
| | LPT | 160 | 2,127.93 |
| | ERT | 160 | 1,604.13 |
| | EDD | 160 | 1,494.72 |
| | FDD | 160 | 1,580.29 |
| | ODD | 160 | 1,716.67 |
| | LST | 160 | 1,558.68 |
| | CI | 160 | 2,160.94 |
| | ANN-HJS | 160 | 787.85 |
| | ANN-LPT | 160 | 862.71 |
| | ANN-ERT | 160 | 735.02 |
| | ANN-EDD | 160 | 755.93 |
| | ANN-FDD | 160 | 730.69 |
| | ANN-ODD | 160 | 740.74 |
| | ANN-LST | 160 | 750.57 |
| | ANN-CI | 160 | 767.80 |
| | Total | 2,560 | |

Notes: [a]Kruskal Wallis test. [b]Grouping variable: variants of algorithm.

In addition to the descriptive statistical analysis on understanding the best performing proposed approach(s), this study is further checking that, whether the distribution of deviation/proximity score (that is, loss of optimality score) across the proposed approaches is same or not. To study this, Kruskal-Wallis (a non-parametric) test is conducted and the results are presented in Table 12. It is observed from Table 12 that:

a there is a statistically significant difference in the distribution of deviation scores across the proposed approaches, as the P-Values is 0.000, which is less than 0.05, and

b the HNN variants: ANN-ODD and ANN-FDD are relatively top variants among the proposed approaches in optimally scheduling the research problem defined in this study.

#### *5.4.2.2 Statistical analyses considering 640 large scale instances*

The statistical analyses carried out to understand the performance of the proposed approaches considering small scale instances is exactly conducted for the 640 large scale problem instances with a change in input to Kruskal-Wallis test. That is, in the case of large-scale problem instances, we studied whether the RPD score (that is, loss of optimality score) across the proposed approaches is same or not. Accordingly, the results obtained based on descriptive analysis and Kruskal-Wall test analyses are presented in Table 13 and Table 14, respectively. The inferences from these analyses exactly matched with the inferences obtained from the small-scale instances.

**Table 13** Descriptive statistics of proposed variants – large scale problem instances

| *Proposed variants based on* | *Descriptive statistics* | | | |
|---|---|---|---|---|
| | *Mean* | *Median* | *Standard deviation* | *95% confidence interval* |
| HJS | 62,849.89 | 47,328 | 56,440.97 | (58,477.16, 67,222.62) |
| LPT | 65,642.52 | 48,122 | 59,584.46 | (61,026.25, 70,258.79) |
| ERT | 58,993.8 | 41,927.5 | 55,562.8 | (54,689.1, 63,298.49) |
| EDD | 56,601.45 | 41,182 | 52,129.12 | (52,562.78, 60,640.12) |
| FDD | 50,794.72 | 36,857.5 | 47,438.48 | (47,119.46, 54,469.99) |
| ODD | 41,438.7 | 30,163 | 38,354.73 | (38,467.19, 44,410.21) |
| LST | 60,772.73 | 43,689 | 56,094.91 | (56,426.81, 65,118.65) |
| CI | 58,324.8 | 42,279.5 | 54,311.22 | (54,117.07, 62,532.53) |
| ANN-HJS | 50,478.29 | 35,544 | 48,817.55 | (46,696.18, 54,260.4) |
| ANN-LPT | 57,781.32 | 38,287.5 | 56,887.12 | (53,374.03, 62,188.62) |
| ANN-ERT | 50,916.82 | 33,945.5 | 50,358.85 | (47,015.3, 54,818.34) |
| ANN-EDD | 48,686.15 | 33,524.5 | 47,667.82 | (44,993.12, 52,379.19) |
| ANN-FDD | 43,964.13 | 30,162.5 | 42,974.44 | (40,634.71, 47,293.54) |
| ANN-ODD | 36,630.08 | 26,021.5 | 35,466.71 | (33,882.32, 39,377.84) |
| ANN-LST | 51,953.95 | 36,222.5 | 49,790.06 | (48,096.5, 55,811.41) |
| ANN-CI | 52,046.45 | 36,682.5 | 49,712.03 | (48,195.04, 55,897.86) |

**Table 14** Kruskal-Wallis test on RPD values of proposed variants – large scale instances

| | *Test statistics*[a,b] | | |
|---|---|---|---|
| | *RPD* | | |
| Kruskal-Wallis H | 5,307.226 | | |
| df | 15 | | |
| Asymp. sig. | 0.000 | | |
| | *Ranks* | | |
| | *Variants of algorithm* | *N* | *Mean rank* |
| RPD | HJS | 640 | 8,051.78 |
| | LPT | 640 | 8,425.83 |
| | ERT | 640 | 6,962.64 |
| | EDD | 640 | 6,659.83 |
| | FDD | 640 | 5,160.02 |
| | ODD | 640 | 3,041.75 |
| | LST | 640 | 7,458.11 |
| | CI | 640 | 6,944.60 |
| | ANN-HJS | 640 | 4,318.35 |
| | ANN-LPT | 640 | 5,652.18 |
| | ANN-ERT | 640 | 3,590.46 |
| | ANN-EDD | 640 | 3,166.66 |
| | ANN-FDD | 640 | 2,051.19 |
| | ANN-ODD | 640 | 963.68 |
| | ANN-LST | 640 | 4,739.77 |
| | ANN-CI | 640 | 4,741.16 |
| | Total | 10,240 | |

Notes: [a]Kruskal Wallis test.
[b]Grouping variable: variants of algorithm.

Overall, the inferences obtained from the statistical analyses on the performances of the proposed approaches, considering both cases of small-scale instances and large-scale instances, exactly endorses the observations obtained from the empirical analyses on the performance of the proposed approaches.

## 6 Conclusions

We have considered a new dynamic scheduling of BP problem with jobs characteristics: different release time, different due-dates, different processing time, non-agreeable release time(s) and due-date(s) with the scheduling objective of minimising TWT. We proposed:

a eight variants of DR-GHA

b eight variants of HNN algorithms, applying an ANN for dynamic scheduling of a BO problem defined in this study.

The main purpose of integrating ANN with DR-GHA is to:

a obtain accurate parameter estimation for the job-priority-data of each of the jobs by applying an iterative learning strategy

b utilising the estimated value of the job-priority-data of each of the jobs with DR-GHA for providing near optimal/estimated optimal schedule for burn-in operation.

From the series of computational analyses, it is observed that:

a the performance of each of the proposed eight variants of the HNN algorithms is far better than each of respective DR-GHA, which are not integrated with ANN, in comparison with both optimal solution (on small scale instances) and estimated optimal solution (on large scale instances)

b each of the proposed eight variants of HNN algorithms could find the optimal or near optimal solution for small size problem instances

c when the problem size increases, the proposed HNN variant: HNN-ODD is outperforming relatively with other HNN variants.

In addition, the HNN variant: HNN-FDD is consistently performing well as a second choice, out of the eight variants of HNN proposed in this study.

Both empirically and statistically, with the series of computational analyses, this study proved that more accurate parameter estimation of the job-priority-data of each of the jobs through a back-propagation ANN and utilising it in any DR-GHA leads to high-quality schedules with respect to TWT.

Only selected dispatching rules are considered, in this study, for proposing DR-GHA and to integrate with ANN for scheduling a BO problem defined in this study. However, there are many dispatching rules in literature and in practice. So, one could consider all the possible dispatching rules, widely used in SM, or extensively claimed as good dispatching rules in the literature, for the scheduling objective of TWT and develop a series of computational experiments to arrive at a set of best performing HNN variants for scheduling:

a single BO

b multiple and non-identical BO(s) could be very important immediate future research topics.

Another important future work could be to develop an efficient lower bound procedure, as the estimated optimal solution has its own demerits, for developing a BS for performance evaluation.

## References

Alipour, P., Nguyen, C. and 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*, Vol. 35, No. 1, pp.80–99.

Alizadeh, N. and Kashan, A.H. (2019) 'Enhanced grouping league championship and optics inspired optimization algorithms for scheduling a batch processing machine with job conflicts and non-identical job sizes', *Applied Soft Computing Journal*, Vol. 83, No. 1, pp.1–16.

Aytug, H., Bhattacharya, S., Koehler, G.J. and Snowdon, J.L. (1994) 'A review of machine learning in scheduling', *IEEE Transactions on Engineering Management*, Vol. 41, No. 2, pp.165–171.

Bellanger, A. and Oulamara, A. (2009) 'Scheduling hybrid flowshop with parallel batching machines and compatibilities', *Computers and Operations Research*, Vol. 36, No. 6, pp.1982–1992.

Beldar, P and Costa, A. (2018) 'Single machine batch processing problem with release dates to minimize total completion time', *International Journal of Industrial Engineering Computations*, Vol. 9, No. 3, pp.331–348.

Benda, F., Braune, R., Doerner, K.F. and Hartl, R.F. (2019) 'A machine learning approach for flow shop scheduling problems with alternative resources, sequence-dependent setup times and blocking', *OR Spectrum*, Vol. 41, No. 4, pp.871–893.

Boudhar, M. (2003) 'Scheduling a batch processing machine with bipartite compatibility graphs', *Mathematical Methods of Operations Research*, Vol. 57, No. 3, pp.513–527.

Cabo, M., Possani, E., Potts, C.N. and Song, X. (2015) 'Split-merge: using exponential neighborhood search for scheduling a batching machine', *Computers and Operations Research*, Vol. 63, No. 1, pp.125–135.

Chan, F.T.S., Au, K.C., Chan, L.Y., Lau, T.L. and Choy, K.L. (2007) 'A genetic algorithm approach to machine flexibility problems in an ion plating cell', *The International Journal of Advanced Manufacturing Technology*, Vol. 31, Nos. 11–12, pp.1127–1134.

Chen, T. and Wang, Y-C. (2012) 'Enhancing scheduling performance for a wafer fabrication factory: the bi-objective slack-diversifying nonlinear fluctuation-smoothing rule', *Computational Intelligence and Neuroscience*, Vol. 2012, pp.1–12, Article ID: 404806, https://doi.org/ 10.1155/2012/404806 (accessed 6 April 2020).

Cheng, T.C.E., Gordon, V.S. and Kovalyov, M.Y. (1996) 'Single machine scheduling with batch deliveries', *European Journal of Operational Research*, Vol. 94, No. 2, pp.277–283.

Chou, F. and Wang, H.M. (2012) 'Minimizing total weighted tardiness on parallel batch-processing machine scheduling problems with varying machine capacities', *Applied Mechanics and Materials*, Vol. 110, No. 116, pp.3906–3913.

Chou, F-D. and Wang, H-M. (2008) 'Scheduling for a single semiconductor batch-processing machine to minimize total weighted tardiness', *Journal of the Chinese Institute of Industrial Engineers*, Vol. 25, No. 2, pp.136–147.

Chung, T.P. and Sun, H. (2018) 'Scheduling batch processing machine problem with non-identical job-sizes via artificial immune system', *Journal of Industrial and Production Engineering*, Vol. 35, No. 3, pp.129–134.

Chung, S.H., Tai, Y.T. and 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*, Vol. 47, No. 18, pp.5109–5128.

Condotta, A., Knust, S. and Shakhlevich, N.V. (2010) 'Parallel batch scheduling of equal-length jobs with release and due-dates', *Journal of Scheduling*, Vol. 13, No. 5, pp.463–477.

Damodaran, P. and Wiechman, N. (2015) 'A column generation approach for scheduling a batch processing machine with makespan objective', *International Journal of Industrial and Systems Engineering*, Vol. 21, No. 3, pp.334–355.

Damodaran, P., Hirani, N.S. and Velez-Gallego, M.C. (2009) 'Scheduling identical parallel batch processing machines to minimize makespan using genetic algorithms', *European Journal of Industrial Engineering*, Vol. 3, No. 2, pp.187–206.

Dupont, L. and Dhaenens-Flipo, C. (2002) 'Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure', *Computers and Operations Research*, Vol. 29, No. 7, pp.807–819.

Fachini, R.F., Esposto, K.F. and Camargo, V.C.B. (2017) 'Glass container production planning with warm-ups and furnace extraction variation losses', *The International Journal of Advanced Manufacturing Technology*, Vol. 90, Nos. 1–4, pp.527–543.

Fanti, M.P., Maione, B., Piscitelli, G. and Turchiano, B. (1996) 'Heuristic scheduling of jobs on a multi-product batch-processing machine', *International Journal of Production Research*, Vol. 34, No. 8, pp.2163–2186.

Feng, Q., Shang, W-P., Jiao, C-W. and Li, W-J. (2020) 'Two-agent scheduling on a bounded parallel-batching machine with makespan and maximum lateness objectives', *Journal of the Operations Research Society of China*, Vol. 8, No. 1, pp.189–196.

Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinnooy, K.A.H.G. (1979) 'Optimization and approximation in deterministic sequencing and scheduling: a survey', *Annals of Discrete Mathematics*, Vol. 5, No. 1, pp.287–326.

Hildebrandt, T., Heger, J. and Scholz-Reiter, B. (2010) 'Towards improved dispatching rules for complex shop floor scenarios – a genetic programming approach', in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO '10*, pp.257–264.

Hochbaum, D.S. and Landy, D. (1994) 'Scheduling with batching: minimizing the weighted number of tardy jobs', *Operations Research Letters*, Vol. 16, No. 2, pp.79–86.

Huang, S-C. and Lin, J.T. (1998) 'An interactive scheduler for a wafer probe centre in semiconductor manufacturing', *International Journal of Production Research*, Vol. 36, No. 7, pp.1883–1900.

Hulett, M., Damodaran, P. and Amouie, M. (2017) 'Scheduling non-identical parallel batch processing machines to minimize total weighted tardiness using particle swarm optimization', *Computers and Industrial Engineering*, Vol. 113, No. 1, pp.425–436.

Hung, Y-F. (1998) 'Scheduling of mask shop e-beam writers', *IEEE Transactions on Semiconductor Manufacturing*, Vol. 11, No. 1, pp.165–172.

Ikura, Y. and Gimple, M. (1986) 'Efficient scheduling algorithms for a single batch processing machine', *Operations Research Letters*, Vol. 5, No. 2, pp.61–65.

Jayamohan, M.S. and Rajendran, C. (2000) 'New dispatching rules for shop scheduling: a step forward', *International Journal of Production Research*, Vol. 38, No. 3, pp.563–586.

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

Jia, Z., Wang, C., Wen, T.T., Leung, J.Y-T. and Li, K. (2017) 'Effective heuristics for makespan minimization in parallel batch machines with non-identical capacities and job release times', *Journal of Industrial and Management Optimization*, Vol. 13, No. 2, pp.977–993.

Jia, Z., Wang, C. and Leung, J.Y-T. (2016) 'An ACO algorithm for makespan minimization in parallel batch machines with non-identical job sizes and incompatible job families', *Applied Soft Computing*, Vol. 38, No. 10, pp.395–404.

Keshavarz, T. (2021) 'A lower bounding method for earliness and tardiness minimization on a single batch processing machine', *Journal of Industrial and Systems Engineering*, Vol. 13, No. 2, pp.9–26.

Koh, S.G., Koo, P.H., Ha, J.W. and Lee, W.S. (2004) 'Scheduling parallel batch processing machines with arbitrary job sizes and incompatible job families', *International Journal of Production Research*, Vol. 42, No. 19, pp.4091–4107.

Lee, C-Y., Uzsoy, R. and Martin-Vega, L.A. (1992) 'Efficient algorithms for scheduling semiconductor burn-in operations', *Operations Research*, Vol. 40, No. 4, pp.764–775.

Leeftink, A.G., Boucherie, R.J., Hans, E.W., Verdaasdonk, M.A.M., Vliegen, I.M.H. and van Diest, P.J. (2018) 'Batch scheduling in the histopathology laboratory', *Flexible Service Manufacturing Journal*, Vol. 30, Nos. 1–2, pp.171–197.

Li, S.S. and Chen, R.X. (2014) 'Single-machine parallel-batching scheduling with family jobs to minimize weighted number of tardy jobs', *Computers and Industrial Engineering*, Vol. 73, No. 1, pp.5–10.

Li, S., Ng, C.T., Cheng, T.C.E. and Yuan, J. (2011) 'Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan', *European Journal of Operational Research*, Vol. 210, No. 3, pp.482–488.

Li, X.L., Li, Y.P. and 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*, Vol. 106, No. 1, pp.91–101, Elsevier Ltd.

Li, Z., Chen, H., Xu, R. and Li, X. (2015) 'Earliness-tardiness minimization on scheduling a batch processing machine with non-identical job sizes', *Computers and Industrial Engineering*, Vol. 87, No. 1, pp.590–599, Elsevier Ltd.

Mathirajan, M. and Sivakumar, A.I. (2006a) 'A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor', *International Journal of Advanced Manufacturing Technology*, Vol. 29, Nos. 9–10, pp.990–1001.

Mathirajan, M. and Sivakumar, A.I. (2006b) 'Minimizing total weighted tardiness on heterogeneous batch processing machines with incompatible job families', *International Journal of Advanced Manufacturing Technology*, Vol. 28, Nos. 9–10, pp.1038–1047.

Mathirajan, M., Bhargav, V. and 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*, Vol. 48, Nos. 9–12, pp.1133–1148.

Maya, J., Muthusamy, S., Velez-Gallego, M.C. and Rojas-Santiago, M. (2014) 'Minimizing makespan in a no-wait flow shop with two batch processing machines: a grasp algorithm', *International Journal of Industrial and Systems Engineering*, Vol. 17, No. 2, pp.152–169.

Melnik, M. and Nasonov, D. (2019) 'Workflow scheduling using neural networks and reinforcement learning', *Procedia Computer Science*, Vol. 156, pp.29–36.

Mönch, L. and Unbehaun, R. (2007) 'Decomposition heuristics for minimizing earliness-tardiness on parallel burn-in ovens with a common due-date', *Computers and Operations Research*, Vol. 34, No. 11, pp.3380–3396.

Mönch, L., Unbehaun, R. and 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*, Vol. 28, No. 2, pp.177–198.

Monch, L., Fowler, J., Dauzère-Pérès, S., Mason, S. and Rose, O. (2011) 'A survey of problems, solution techniques and future challenges in scheduling semiconductor manufacturing operations', *Journal of Scheduling*, Vol. 14, No. 6, pp.583–599.

Mora, J.F., Nait-Abdallah, R., Lozano, A.J., Montoya, C. and Otero-Caicedo, R. (2020) *Batch Assignment of Parallel Machines in an Automotive Safety Glass Manufacturing Facility*, Ing. Univ., Vol. 24 [online] https://revistas.javeriana.edu.co/index.php/iyu/article/view/23228 (accessed 12 January 2021).

Nguyen, S., Zhang, M., Johnston, M. and 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*, Vol. 17, No. 5, pp.621–639.

Ogun, B. and Alabas-Uslu, C. (2018) 'Mathematical models for a batch scheduling problem to minikize earliness and tardiness', *Journal of Industrial Engineering and Management*, Vol. 11, No. 3, pp.390–405.

Oulamara, A. (2007) 'Makespan minimization in a no-wait flow shop problem with two batching machines', *Computers and Operations Research*, Vol. 34, No. 4, pp.1033–1050.

Ozturk, O. (2020) 'A truncated column generation algorithm for the parallel batch scheduling problem to minimize total flow time', *European Journal of Operational Research*, Vol. 286, No. 2, pp.432–443.

Ozturk, O., Espinouse, M.L, Mascolo, D. and Gouin, A. (2010) 'Optimizing the makespan of washing operations of medical devices in hospital sterilization services', *IEEE Workshop on Health Care Management*, pp.1–6.

Park, Y., Kim, S. and Lee, Y.H. (2000) 'Scheduling jobs on parallel machines applying neural network and heuristic rules', *Computers and Industrial Engineering*, Vol. 38, No. 1, pp.189–202.

Parsa, N.R., Karimi, B. and Husseini, S.M.M. (2017a) 'Exact and heuristic algorithms for the just-in-time scheduling problem in a batch processing system', *Computers and Operations Research*, Vol. 80, No. 1, pp.173–183.

Parsa, N.R., Karimi, B., Mohammad, S. and Husseini, M. (2017b) 'An improved memetic algorithm to minimize the earliness – tardiness on a single batch processing machine', *Journal of Industrial and Systems Engineering*, Vol. 10, pp.57–72, Special Issue on Production and Inventory.

Parsa, N.R., Karimi, B. and Husseini, S.M.M. (2016) 'Minimizing total flow time on a batch processing time using a hybrid max-min ant system', *Computers and Industrial Engineering*, Vol. 99, No. 1, pp.372–381.

Parsa, N.R., Karimi, B., Husseini, M. and Taha, K. (2019) 'A hybrid neural network approach to minimize total completion time on a single batch processing machine', *International Transactions in Operational Research*, Vol. 28, No. 5, pp.2867–2899, https://doi.org/10.1111/itor.12665 (accessed 6 April 2020).

Perez, I.C., Fowler, J.W. and Carlyle, W.M. (2005) 'Minimizing total weighted tardiness on a single batch process machine with incompatible job families', *Computers and Operations Research*, Vol. 32, No. 2, pp.327–341.

Priore, P., Gomez, A., Pino, R. and Rosillo, R. (2014) 'Dynamic scheduling of manufacturing systems using machine learning: an updated review', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 28, No. 1, pp.83–97.

Raghavan, N.R.S. and Venkataramana, M. (2006) 'Scheduling parallel batch processors with incompatible job families using ant colony optimization', *IEEE International Conference on Automation Science and Engineering*, pp.507–512.

Rardin, R.L. and Uzsoy, R. (2001) 'Experimental evaluation of heuristic optimization algorithms: a tutorial', *Journal of Heuristic*, Vol. 7, No. 3, pp.261–304.

Ravindra, G. and Mathirajan, M. (2014) 'Minimizing total weighted tardiness on heterogeneous batch processors with incompatible job families', *The International Journal of Advanced Manufacturing Technology*, Vol. 70, Nos. 9–12, pp.1563–1578.

Rojas-Santiago, M., Muthusamy, S., Damodaran, P. and Velez-Gallego, M.C. (2017) 'Scheduling batch processing machines in a no-wait flow shop ACO to minimize the makespan', *International Journal of Industrial and Systems Engineering*, Vol. 27, No. 3, pp.390–411.

Rose, O. (2003*) Comparison of Due-Date Oriented Dispatching Rules in Semiconductor Manufacturing* [online] https://www.researchgate.net/publication/2859216_Comparison_of_Due-date_Oriented_Dispatch_Rules_in_Semiconductor_Manufacturing (accessed 6 April 2020).

Sarin, S.C., Varadarajan, A. and Wang, L. (2011) 'A survey of dispatching rules for operational control in wafer fabrication', *Production Planning and Control*, Vol. 22, No. 1, pp.4–24.

Shao, H., Chen, H.P., Xu, R. and Zhang, W.G. (2008) 'Minimizing makespan for a single batch processing machine with non-identical job sizes using neural nets approach', *Fourth International Conference on Natural Computation*, pp.511–514, DOI: 10.1109/ICNC.2008.368 (accessed 6 April 2020).

Sung, C.S. and Kim, Y.H. (2002) 'Minimizing makespan in a two-machine flowshop with dynamic arrivals allowed', *Computers and Operations Research*, Vol. 29, No. 3, pp.275–294.

Tang, L. and Yan, P. (2009) 'Particle swarm optimization algorithm for a batching problem in the process industry', *Industrial and Engineering Chemistry Research*, Vol. 48, No. 20, pp.9186–9194.

Tsai, Y-C. and Chou, F-D. (2016) 'A bi-criterion simulated annealing method for a single sbatch-processing machine scheduling problem with dynamic job release times', in *International Conference on Artificial Intelligence and Computer Science*, pp.103–107.

Varadarajan, A. and Sarin, S.C. (2006) 'A survey of dispatching rules for operations control in wafer fabrication', *IFAC Proceedings*, Vol. 39, No. 3, pp.715–726.

Ventura, J.A. and Kim, D. (2000) 'Parallel machine scheduling about an unrestricted due-date and additional resource constraints', *IIE Transactions (Institute of Industrial Engineers)*, Vol. 32, No. 2, pp.147–153.

Ventura, J.A. and Kim, D. (2003) 'Parallel machine scheduling with earliness – tardiness penalties and additional resource constraints', *Computers and Operation Research*, Vol. 30, No. 13, pp.1945–1958.

Rani, M.V. and Mathirajan, M. (2020) 'Meta-heuristics for dynamic real time scheduling of diffusion furnace in semiconductor manufacturing industry', *International Journal of Industrial and Systems Engineering*, Vol. 34, No. 3, pp.365–395.

Wang, C.S. and Uzsoy, R. (2002) 'A genetic algorithm to minimize maximum lateness on a batch processing machine', *Computers and Operation Research*, Vol. 29, No. 12, pp.1621–1640.

Wang, H. (2011) 'Solving single batch-processing machine problems using an iterated heuristic', *International Journal of Production Research*, Vol. 49, No. 14, pp.4245–4261.

Xu, S. and Bean, J.C. (2016) 'Scheduling parallel-machine batch operations to maximize on-time delivery performance', *Journal of Scheduling*, Vol. 19, No. 5, pp.583–600.

Yaghubian, A.R., Hodgson, T.J. and Joines, J.A. (2001) 'Dry-or-buy decision support for dry kiln scheduling in furniture production', *IIE Transactions*, Vol. 33, No. 2, pp.131–136.

Van der Zee, D.J., van Harten, A. and Schuur, P. (2001) 'On-line scheduling of multi-server batch operations', *IIE Transactions (Institute of Industrial Engineers)*, Vol. 33, No. 7, pp.569–586.

Zhang, R., Chang, P-C., Song, S. and Wu, C. (2017) 'A multi-objective artificial bee colony algorithm for parallel batch-processing machine scheduling in fabric dyeing processes', *Knowledge-based Systems*, Vol. 116, No. 7, pp.114–129.

Zheng, R. and Li, H. (2009) 'Minimizing weighted number of tardy jobs on unbounded single batching machine with family jobs', in *IEEE International Conference on Industrial Engineering and Engineering Management*, pp.1718–1722.

Zhou, H., Pang, J., Chen, P. and Chou, F. (2018) 'A modified particle swarm optimization algorithm for a batch-processing machine scheduling problem with arbitrary release times and non-identical job sizes', *Computers and Industrial Engineering*, Vol. 123, No. 1, pp.67–81.