IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 29, 2020, accepted April 13, 2020, date of publication April 16, 2020, date of current version May 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988274

Learning-Based Metaheuristic for Scheduling Unrelated Parallel Machines With Uncertain Setup Times

CHEN-YANG CHENG[1], **POURYA POURHEJAZY**[1], **KUO-CHING YING**[1], **SHU-FEN LI**[2], **AND CHIEH-WEN CHANG**[1,3]
[1]Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 10608, Taiwan
[2]Department of Industrial Engineering and Management, National Chin-Yi University of Technology, Taichung 41170, Taiwan
[3]Wistron Corporation, Taipei 22181, Taiwan

Corresponding author: Kuo-Ching Ying (kcying@ntut.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, R. O. C., under Grant MOST108-2221-E-027-025.

ABSTRACT Setup time consists of all the activities that need to be completed before the production process takes place. The extant scheduling predominantly relies on simplistic methods, like the average value obtained from historical data, to estimate setup times. However, such methods are incapable of representing the real industry situation, especially when the setup time is subject to significant uncertainties. In this situation, the estimation error increases proportionally to the problem size. This study proposes a Random-Forest-based metaheuristic to minimize the makespan in an Unrelated Parallel Machines Scheduling Problem (UPMSP) with uncertain machine-dependent and job sequence-dependent setup times (MDJSDSTs). Taking the forging industry as an example, the numerical experiments show that the error percentage for the setup time estimation substantially decreases when the proposed approach is applied. This improvement is particularly significant when large-scale problems are sought. Overall, this study highlights the role of advanced analytics in bridging the gap between scheduling theory and practice.

INDEX TERMS Scheduling, unrelated parallel machines, setup times, random-forest, metaheuristic.

I. INTRODUCTION

In a saturated market place, the strategic intention has been directed towards customer satisfaction, where product on-time delivery is a principal attribute. In this situation, production scheduling is of critical importance for large businesses to ensure timely response to the demand surges. The production research literature shows a growing flow of research papers expanding scheduling problems, inspired by various industry-specific needs and situations. The existing body of scheduling literature predominantly focused on different machine characteristics, job constraints, and pursuing various objectives [1]. The vast majority of these scheduling extensions ignored uncertain setup times, exposing the modeling outcomes to different levels of imprecision [2]–[4]. This may be due to the complexities involved in implementing stochastic or simulation-based optimization approaches. Setup procedure includes all the inevitable preparation activities from machine adjustment, and mold installation/removal, to cleaning fixtures and the other involved tools. The idle time caused by setups, among the other non-value-adding activities (see [5]), should be estimated carefully to obtain dependable scheduling solutions.

The setup time estimation in practice is predominantly based on the operators' experience, or, at best, simple average-based methods. Such approaches are not effective in the production systems with general-purpose machinery, and the possible errors can be extensive when estimating the delivery due date for large-size orders. Sequence-dependent setup time is a prime example when the uncertainties prevail [6] because the estimation depends not only on the current job, but also the job immediately preceding it, and the characteristics of the respective machine [2]. This situation is particularly prevalent in the Unrelated Parallel Machines Scheduling Problem (UPMSP). UPMSPs with machine-dependent and job sequence-dependent setup

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwu Li.

times (MDJSDSTs) represent the real production situation in various industries, from dicing operations in semiconductor wafer manufacturing, and drilling operations in printed circuit board fabrication, to warp making, weaving, dyeing, and cloth cutting in textile manufacturing [7]. Despite the wide industry applications of this family of scheduling problems, research in this field is relatively under-developed [8]. From the existing literature, [9] included uncertainties in job arrival time and due date but dismissed the setup time uncertainties. Reference [10] studied optimization under uncertainty in UPMSPs with MDJSDSTs, addressing the issue from a solution algorithm standpoint. No studies considered setup time estimation complexities, and the inherent uncertainties, while the errors caused due to this shortcoming may jeopardize the consistency of the scheduling outcomes. Overall, one-dimensional statistical methods may not be effective to address the setup time uncertainties, particularly when implemented to solve complex and large-scale scheduling situations.

Various integrations have been proposed to improve the analytical aspects of optimization in the supply chain and manufacturing contexts [11]. Advanced analytics can help address the uncertainties involved in the mathematical model parameters, particularly time value estimations [12]. Inspired by this idea, this study sought to integrate a learning-based method into scheduling optimization problems to help narrow the gap between research and practice. Given the relative significance and the complexities involved in the estimation of MDJSDSTs in UPMSPs, this scheduling extension is considered as a baseline to provide insights and informs other scheduling situations. The main contribution of this work is, therefore, to propose a Random-Forest-based Hybrid Artificial Bee Colony (RF-HABC) algorithm to minimize the makespan of UPMSPs with MDJSDSTs.

The remainder of this manuscript is organized into four sections. Section 2 consists of a literature review to support the stated research gap. The applied methodological tools, the research process, algorithm and analysis methods are briefly explained in section 3. Section 4 provides exhaustive numerical results and experimental analysis. Finally, concluding remarks, and directions for future research works are provided in section 5.

II. LITERATURE REVIEW

Setup operations are case-specific. That is, production needs, and characteristics determine the sort of preparation activities required before the value-adding operations take place. Setup times can particularly be substantial when general-purpose machines are used. Behavior-dependent characteristics of setup times have been the focus of many scheduling studies [13]. Overall, sequence-dependent setups are subject to more complexities, because they depend on the characteristics of the consecutive jobs, and machines [2], [6].

Scheduling with stochastic time parameters was the first time introduced by the seminal works of [14], and [15]. Given the significant impact of the time parameters uncertainty on the overall performance of the system, i.e. completion time fluctuations [16], various methods are used to address this issue. More particularly, studies addressed the issues pertinent to the setup time uncertainty considering processing and setup times within certain intervals [17]–[19], known as lower- and upper-bounds [20], [21], and using fuzzy time variables [22], [23].

Zhu *et al.* [24] addressed the scheduling of cluster tools, which is a special case of UPMSP [25], using Petri Nets to model the close-down process for the wafer lots switch and maintenance. Although Petri nets are capable of addressing setup time and time-window [26], and widely used in the scheduling context [27]–[29], they are simplistic and limited to effectively model the real-world systems [30]. Parallel machines scheduling problems considered release and delivery time constraints, but in a deterministic environment [31]. Other studies addressed this shortcoming, for example through considering the processing time as a random variable following a negative exponential distribution for optimizing UPMSP [32]; this study did not take into account setup times. Besides, such one-dimensional statistical methods alleviate uncertainties but cannot be effective when limited data is available [33], or significant uncertainties should be confronted [34].

More recent studies, therefore, addressed time uncertainties applying robust deviation methods [35], [36]. Reference [35] considered distribution functions, along with a robust deviation method to minimize the maximum regret value. Reference [36] applied a similar approach to address the setup time uncertainties in parallel machine scheduling with order dependence. Estimating the accuracy was not subject to study in the mentioned papers, and the authors merely focused on optimization results analysis. In other words, it is unknown if their proposed method improved the accuracy of setup estimates.

Reference [12] suggested that advanced analytics, and more particularly learning-based methods can be applied to address the uncertainties involved in time parameter estimations. There are several studies where decision trees are incorporated into scheduling problems, addressing various computational aspects. Reference [37] adopted a decision tree to select proper dispatching rules in reentrant hybrid flowshops. They applied [38] Iterative Dichotomiser 3 –ID3 algorithm to alleviate computational difficulties for the simulation analysis. Reference [39] incorporated the C4.5 decision tree (see [40]) into a metaheuristic approach to explore assignment rules while considering utilization rate, processing and expiration times. They showed that the learning-based technique significantly reduces the actual delay time. Applying a similar approach, [41] dealt with a multi-target scheduling problem, simultaneously minimizing the makespan and maximum lateness, and minimizing total delays. Reference [42] applied decision trees for due date assignment in a dynamic job shop scheduling problem. Their study claims the lowest tardiness norm, reducing the errors for total work content estimation.

Surprisingly, limited attention has been given to advanced analytics applications to overcome the limitations of extant scheduling. Learning-based tools, and more particularly RF, have not been applied for the estimation of time parameters, and more particularly setup times in the scheduling context. To bridge this gap, our study proposes an RF-HABC algorithm to solve UPMSPs with uncertain MDJSDSTs, comparing its error rates with that of simple average and decision tree methods. Following the notation system introduced by [43], the problem is hereafter denoted by $R_m|S_{ijk}|C_{\max}$. The first term of this notation, R_m, indicates a UPM production system, where the job sequence on each machine can be different; S_{ijk} specifies MDJSDSTs, to be obtained through RF analysis. The last term, $C_{\max}$, specifies makespan as the objective function.

III. RESEARCH METHOD

This section first describes the RF method to deal with MDJSDST estimation. The mixed-integer programming (MIP) formulation of the UPMSP with uncertain MDJSDSTs is then detailed. The section continues with a brief explanation of the solution algorithm phase, HABC, and its computational steps. We finish up with elaborating on the developed two-phase methodology.

A. RANDOM FOREST

Proposed by [44], RF consists of a group of tree predictors that make individual decisions, and are formed by a set of branches and sub-branches, each of which representing various decision-making scenarios. Employing Bootstrap Aggregating (Bagging), random subspace method, and the classification model of the decision tree, RF uses training data, and input features, to train the individual trees. In this procedure, Bagging uses the bootstrap replicates of the learning data to help generate several versions of the same predictor, and form the aggregated predictor [45]. The random subspace method develops independent tree-classifiers such that the combination attains higher accuracy [46]. The last major element, the decision tree, is a supervised feature extraction method, consisting of training and prediction phases. Through classifying large sets of data, decision trees extract, and apply, generic rules to identify unknown samples [47].

Decision trees are tree-like flowcharts, mimicking the decision-making processes in the human brain [48]. A decision tree consists of a root node, and outgoing branches comprising internal, and leaf nodes [49]. Nodes are subsets of examples, and the branches represent respective conditions and certain attributes. Categorizing decision trees into classification or regression trees, they have been widely applied in classification and predictive modeling [50]. Evaluating chi-squared automatic interaction detection (CHAID; [51]), C4.5 programs for machine learning [52], and classification and regression trees (CART: [53]) are some seminal examples of decision tree applications.

Similar to decision trees, RF expands the population by generating new trees, from the existing ones, using the Classification and Regression Trees (CART) method [44], [53]. For this purpose, different models and predictive variables are used to ensure that there is no correlation between the developed individuals in the forest. The pruning method is often applied to avoid over-fitting in the presence of a large population size. As a final step, the forest should be screened considering the predicted value. Given a large number of tree predictors, the model identifies the most popular class based on the group votes, and draw respective decisions. RF is capable of effectively processing the data, and filling in the missing values while maintaining high accuracy.

RF has been successfully applied in various fields, where accurate predictive performance in the presence of noise in data is essential [54]. Despite the merits of RF compared to the existing learning tools [55], particularly its proven accuracy in time estimation [56], RF applications in the scheduling context are quite limited. From the existing studies, [57] applied RF to predict the gaps in machines, where several heuristics behavior were compared in job-shop scheduling. In the latter work, [58] integrated RF into a metaheuristic, Particle Swarm Optimization, to generate divergent schedules at the initialization phase. For a thorough review of the learning-based classification approaches, we refer the readers to the exhaustive review of [59].

B. MATHEMATICAL FORMULATION

MIP is applied to schedule the forging process, simultaneously determining the sequence of the jobs and the associated time variables. UPMSPs are characterized by parallel machines that are either different in features or the tasks they perform. We adapted UPMSP with MDJSDSTs, developed by [60], to model the problem at hand. In our formulation, the dummy variable, C_0, is considered to account for the completion time of the first job in the sequence. Besides, L is a very large positive integer. The remainder of indices, parameters, and the decision variables are introduced in the following.

1) INDICES

k	Machine tag, $k \in M = \{1, 2, \ldots, m\}$
i, j	Job index, $i, j \in N = \{1, 2, \ldots, n\}$

2) PARAMETERS

m	Number of machines
n	Number of jobs
$P_{i,k}$	Processing time of job i on the machine k

3) DECISION VARIABLES

$X_{i,j}^{k}$	Binary variable; $= 1$ if the job j is processed immediately after job i on the machine k; $= 0$, otherwise.
C_j	Completion time of the job j

$ST_{i,j}^{k}$ Adjusted setup time for processing of the job j, succeeding job i and processed on the machine k

The MIP formulation of the problem is now presented.

$$\text{Minimize } C_{\max} = \max\{C_j\}_{1\leq j\leq n} \tag{1}$$

$$\text{Subject to } \sum_{k=1}^{m}\sum_{i=1}^{n} X_{i,j}^{k} = 1, \quad \forall j \in N, j \neq i \tag{2}$$

$$\sum_{j=1}^{n} X_{0,i}^{k} = 1, \forall k \in M \tag{3}$$

$$\sum_{\substack{i=1 \\ i\neq q}}^{n} X_{i,q}^{h} - \sum_{\substack{j=1 \\ q\neq j}}^{n} X_{q,j}^{h} = 1, \quad \forall q \in N, \forall h \in M \tag{4}$$

$$C_0 = 0 \tag{5}$$

$$C_j - \left[C_i + \sum_{k=1}^{m} X_{i,j}^{k} \begin{pmatrix} ST_{i,j}^{k} + P_j^{k} + \\ L \times (\sum_{k=1}^{m} X_{i,j}^{k} - 1) \end{pmatrix} \right] \geq 0, \tag{6}$$

$$\forall i \in \{0,\ldots,n\}, \forall j \in \{1,\ldots,n\}$$

$$X_{i,j}^{k} \in \{0,1\}, C_j, ST_{i,j}^{k} \in Z^{+}, \forall i \in \{0,\ldots,n\}, \forall j \in \{1,\ldots,n\}, \forall k \in \{1,\ldots,m\} \tag{7}$$

Equation (1) calculates the makespan, to be minimized in the optimization process. Constraint (2) ensures that each work is assigned to one and only one machine. According to equation (3), each machine can only be assigned one job at a time. The sequence of jobs will be determined using equation (4). Equation (5) assigns a zero value to the completion time of task 0. Constraint (6) helps calculate the completion time of other tasks, ensuring that each of them is associated with only one completion time value and that the completion time value is non-negative. As a final constraint, (7) specifies that the decision variable $X_{i,j}^{k}$ is binary, and the time variables only accept positive integer values.

C. HYBRID ARTIFICIAL BEE COLONY

Developed by [61], Artificial Bee Colony (ABC) is a population-based metaheuristic algorithm, inspired by the collective behavior of bee colonies in search of food sources. ABC performs the search in solution space based on two major features; the division of bees, and the information sharing among them. In this approach, the bee colony is divided into three categories, simulating the bees' honey collecting mechanism: collecting, observing and scouting bees work together to find the best nectar source. Scout bees randomly search for promising new sources. ABC's exploration procedure mimics the scout bees carry out. Given a particular nectar source, collecting bees transfer food and the associated information (i.e. proximity, richness and the ease of extraction) to the onlooker bees in the hive. This procedure forms the exploitation capability of the algorithm. ABCs have been widely applied in combinatorial optimization, and more particularly scheduling problems [62], [63].

ABCs have been widely applied to solve various scheduling problems; single machine scheduling problems [63], economic lot scheduling problems [64], permutation flow shop scheduling problems [65], hybrid flow shop scheduling problems ([66], UPMSP with MDJSDSTs [67], job-shop scheduling problems with no-wait constraint [68], and flexible job shop scheduling problems [69] are the most recent examples. The hybrid artificial bee colony (HABC) is different from its extant versions in that the initial population generation, adaptive function value, proximity procedures are enhanced and a local search, the well-known Simulated Annealing algorithms, is integrated to ensure that ABC, which is a global search algorithm, can effectively avoid getting trapped in local optima.

HABC operates based on six major parameters. The number of food sources, SN, specifies the number of onlookers. *limit* determines the threshold after which the food source will no longer be explored. T_0 and β are considered to adjust the initial temperature, and the temperature change ratio, respectively. The computational iterations will be continued until the maximum time limit $T_{\max}$ is reached. On this basis, the current temperature T is defined, and gets updated at each iteration $T \leftarrow \beta T$, where $0 < \beta < 1$. This phase of RF-HABC results in a (near-) optimal solution, π_{best}. The pseudocode of the HABC algorithm is provided in Figure 1.

The major elements of HABC are now described.

1) INITIALIZATION

The jobs assigned to each machine should be first sorted randomly. Figure 2 is a generic example of a solution structure, where the work sequence in different machines, π_i, can be separated by placing a zero value between two consecutive sequences. In the initialization stage, one should also set the numbers of total and the observing bees equal to the number of solutions.

2) FITNESS VALUE CALCULATION

The fitness function value of a solution, π, determines its quality with respect to the defined performance indicator. $fit(\pi)$ can be obtained using equation (8); that is, the smaller the makespan, the better the solution is. The algorithm searches for new neighborhoods considering equation (9), where $\pi_{\min}^{i}$ and $\pi_{\max}^{i}$ are the lower- and upper-bounds of the solutions, respectively.

$$fit(\pi) = \frac{1}{C_{\max}(\pi)} \tag{8}$$

$$\pi_{\pi}^{i} = \pi_{\min}^{i} + rand_{[0,1]}\left(\pi_{\max}^{i} - \pi_{\min}^{i}\right) \tag{9}$$

3) NEIGHBORHOOD SOLUTION

The adjacent solutions to a current solution, $\pi^{current}$, can be obtained by applying the destruction and construction steps

```
HABC algorithm
Input: $SN, limit, \beta, T_0, T_{\max}$ ; Output: $C_{max}$
Begin
{
    Initialize each solution $\pi^i$ in the first population by random, $i = 1, 2, ..., SN$;
    $T = T_0$; Let $\pi_{best}$ be the best solution;
    Calculate the fitness of each solution $fit(\pi^i)$, $i = 1, 2, ..., SN$;
    Do {
       For $i = 1, 2, ..., SN$ {//Place the employed bees on their food sources;
          Obtain a new solution $\pi^i_{new}$ by $IG(\pi^i)$;
          Apply local search to $\pi^i_{new}$;
           If $\Delta = C_{\max}\left(\pi^i_{new}\right) - C_{\max}\left(\pi^i\right) \le 0$, Let $\left\{\pi^i = \pi^i_{new};\right\}$
            Else {Generate r~$U\left(0,1\right)$;
               If $r < Exp(-\Delta / T)$, Let $\left\{\pi^i = \pi^i_{new};\right\}$
               Else {Discard $\pi^i_{new}$}
            }
      }
      //Place the onlooker bees on the food sources depending on their nectar amounts
      For $i = 1, 2, ..., SN$; {
           selects a food source $\pi^j$ depending on the probability value $prob_j = fit(\pi^j) / \sum_{s=1}^{SN} fit(\pi^s)$;
          Obtain a new solution $\pi^j_{new}$ by $IG(\pi^j)$;
          Apply local search to $\pi^j_{new}$
           If $\Delta = C_{\max}\left(\pi^i_{new}\right) - C_{\max}\left(\pi^i\right) \le 0$, Let $\left\{\pi^i = \pi^i_{new};\right\}$
            Else {Generate r~$U\left(0,1\right)$;
               If $r < Exp(-\Delta / T)$, Let $\left\{\pi^i = \pi^i_{new};\right\}$
               Else {Discard $\pi^i_{new}$;}
            }
      }
      For $i = 1, 2, ..., SN$; // Send the scouts to search new food sources;
      {
          If (food source $\pi^i$ is not changed in the successive limit times)
             Regenerated $\pi^i$ randomly;
      }
      For $i = 1, 2, ..., SN${ // Memorize the best food source found so far
          If $C_{max}(\pi^j_{new}) < C_{max}(\pi_{best})$, Let $\left\{\pi_{best} = \pi^i_{new}\right\}$
      }
      If ( $G \bmod limit = 0$ ){ $T = \beta \times T$ ;}
    } while the computational time is less than $T_{\max}$
       Output the best food source found so far ( $\pi_{best}$ );
}
End
```

FIGURE 1. Pseudocode of the hybrid artificial bee colony (HABC).

0	15 2 5 4 10 11	0	8 18 9 7 6 14	0	1 3 13 16 12 17
Jobs sequence of machines 1		Jobs sequence of machines 2		Jobs sequence of machines 3	

FIGURE 2. **Initial group sorting diagram.**

```
Procedure Local_Search ($\pi$)

Begin
        $L = n + m - 1$ (the number of elements in the solution list);
        $\pi' = \pi$;
        $i = 1$;
        while ($i \le L - 1$)
        {
            $j = i + 1$;
            while ($j \le L$)
            {
                If exchanging the $i^{th}$ and $j^{th}$ elements of $\pi'$ may improve $\pi'$
                {
                    $\pi_{temp}$ = solution obtained by exchanging the $i^{th}$ and $j^{th}$ elements of $\pi'$;
                    If ($C_{\max}(\pi_{temp})$ is less than $C_{\max}(\pi')$)
                    {
                        $\pi' = \pi_{temp}$;
                        $i = 1$;
                        $j = L$;
                    }
                    $j = j + 1$;
                }
            }
            $i = i + 1$;
        }
        return $\pi'$;
End
```

FIGURE 3. **Local search procedure.**

of the Iterative Greedy (IG) algorithm. In this approach, $\pi^{current}$, $\pi^{removed}$, $\pi^{remainder}$, and π^{new} represent a current sequence, list of d removed jobs, the $n - d$ remaining jobs, and the updated, new, sequence, respectively. Destruction step consists of randomly selecting n jobs from the job lists assigned to the machine with the largest completion time. Remove the selected jobs from $\pi^{current}$, and add them to $\pi^{removed}$ keeping the same job order. $\pi^{remainder}$ represents the partial sequence of $\pi^{current}$ after removing the randomly selected jobs. The construction step iteratively adds the elements of $\pi^{removed}$ into every possible position of $\pi^{remainder}$ until a whole sequence, π^{new}, is obtained. The resulted sequence is considered a neighborhood solution of $\pi^{current}$.

4) LOCAL SEARCH

Given a new neighborhood solution, π^{new}, local search approach is employed iteratively to improve the solution with respect to the fitness function value, makespan. It is done by relocating the jobs within the respective sequence. If a better solution is obtained, the local search will be applied another

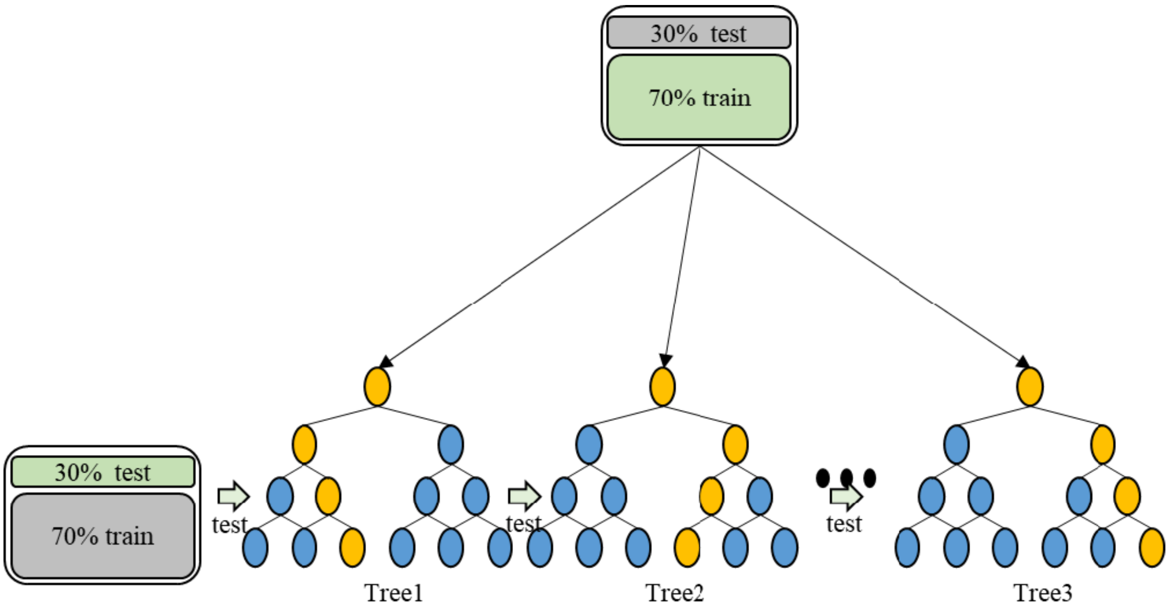


FIGURE 4. **A generic representation of the random forest.**

time until the changes become non-effective. The pseudocode of this procedure is provided in Figure 3.

5) EMPLOYED, ONLOOKER AND SCOUT SEARCH

A solution, π, corresponds to an employed bee, while the new neighborhood solution, π^{new}, is a food source for the employed bees; it replaces π if appears to be a better alternative. Onlooker bee evaluates the fitness values delivered by the employed bees. On this basis, a solution, π, is selected as the incumbent with a probability $prob = makespan(\pi)/\sum_{s=1}^{SN} makespan(\pi^{s})$; the higher a solutions fitness value is, the more likely it will be its selection as incumbent to the *SN* solutions. Once a solution is selected, a corresponding neighborhood solution will be obtained through the neighborhood and local search procedures, until better alternatives are found. A solution will be abandoned if it can no longer be improved. In this situation, the solutions corresponding employed bee become a scout bee.

D. THE PROPOSED RF-HABC

The proposed solution approach, RF-HABC, consists of two major phases. It first employs a learning-based method to estimate MDJSDSTs. The HABC algorithm then solves the $R_m|S_{ijk}|C_{\max}$ problem.

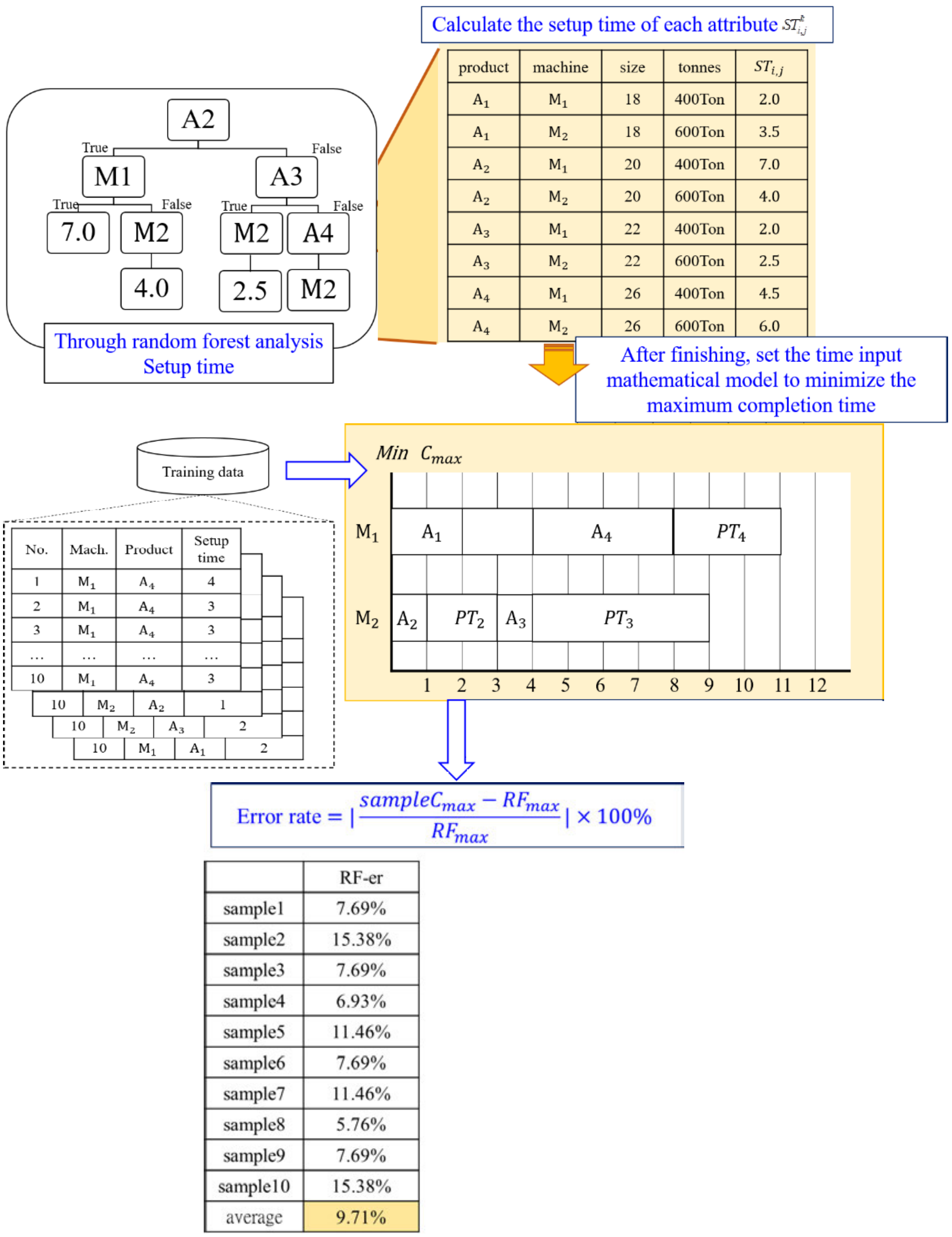
a: PHASE I. RF ANALYSIS FOR SETUP TIME ESTIMATION

RF operates based on the regression or classification trees [70], [71]. This study applies the former approach, RF analysis based on the regression tree that comprises training and prediction steps. Given the parts' information (i.e. number, type, and size) in each machine, and the immediately preceding station, as well as the respective specifications of machines (i.e. number and tonnage), Phase I extends to estimate a setup time value for each of the attribute combinations. A generic example of this procedure is provided in Figure 4.

To estimate the MDJSDSTs, the Bootstrap method is first applied to select N data, and form a sample D, and use them as training material. Considering different attributes Y_i represents the target variable for each of the inputs in the sample $D = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)\}$. Given a set of selected attributes as growth candidates, we form the main node branch. We then continue branching and selecting attributes and repeat until none of the attributes meet the branching conditions. This repetitive extraction of training data from RFs makes the results less vulnerable to noise and outliers. Next, new sample data, D' are generated through predictions; they are then assigned to the K decision trees. We then obtain the average value from the K decision trees such that the predicted value of the new sample data D' is defined. Finally, the correlation between the attributes is calculated. The resulted correlation data and the MDJSDST, $ST_{i,j}^{k}$, form the matrix of machine-job-sequence.

Given the required rule attributes and the obtained MDJSDSTs, the setup time information should be sorted following the mathematical model and the required input format by the solution algorithm. As suggested by [72], the classification method of the Bootstrap sampling is only applied to the observation data, while the RF is applied to both observations and variables. This approach enhances the model's capability in searching for correlations between variables, and improving accuracy.

As a part of numerical experiments, this study applies an alternative approach, the decision tree method, to estimate MDJSDSTs considering case-specific production conditions. A decision tree is employed to extract the knowledge of setup


A2
True
False
M1
A3
True
False
True
False
7.0
M2
M2
A4
4.0
2.5
M2
Through random forest analysis
Setup time
Calculate the setup time of each attribute $ST^{t}_{i,j}$
After finishing, set the time input mathematical model to minimize the maximum completion time
Training data
Min C_{max}
M$_1$
M$_2$
A$_1$
A$_4$
PT_4
A$_2$
PT_2
A$_3$
PT_3
1
2
3
4
5
6
7
8
9
10
11
12

product	machine	size	tonnes	$ST_{i,j}$
A_1	M_1	18	400Ton	2.0
A_1	M_2	18	600Ton	3.5
A_2	M_1	20	400Ton	7.0
A_2	M_2	20	600Ton	4.0
A_3	M_1	22	400Ton	2.0
A_3	M_2	22	600Ton	2.5
A_4	M_1	26	400Ton	4.5
A_4	M_2	26	600Ton	6.0

No.	Mach.	Product	Setup time
1	M_1	A_4	4
2	M_1	A_4	3
3	M_1	A_4	3
…	…	…	…
10	M_1	A_4	3
10	M_2	A_2	1
10	M_2	A_3	2
10	M_1	A_1	2

$$\text{Error rate} = \left|\frac{sampleC_{max} - RF_{max}}{RF_{max}}\right| \times 100\%$$

	RF-er
sample1	7.69%
sample2	15.38%
sample3	7.69%
sample4	6.93%
sample5	11.46%
sample6	7.69%
sample7	11.46%
sample8	5.76%
sample9	7.69%
sample10	15.38%
average	9.71%

FIGURE 5. Graphical representation of the methodology.

time estimations based on various attributes, and generate a set of IF-THEN rules. The generated rules are mutually exclusive and collectively exhaustive, so estimations can be made based on the set of examples [73].

An illustrative example is now provided to explain the computational steps. Let assume a small forging workshop, where two machines are designated to process a total of four products. Product type, product size, and machine tonnage attributes are considered to determine the setup times in a production system with general-purpose machinery. Figure 5 is a graphical illustration of the computational steps.

This procedure consists of test and forecast steps, considering machine-dependent and job sequence-dependent

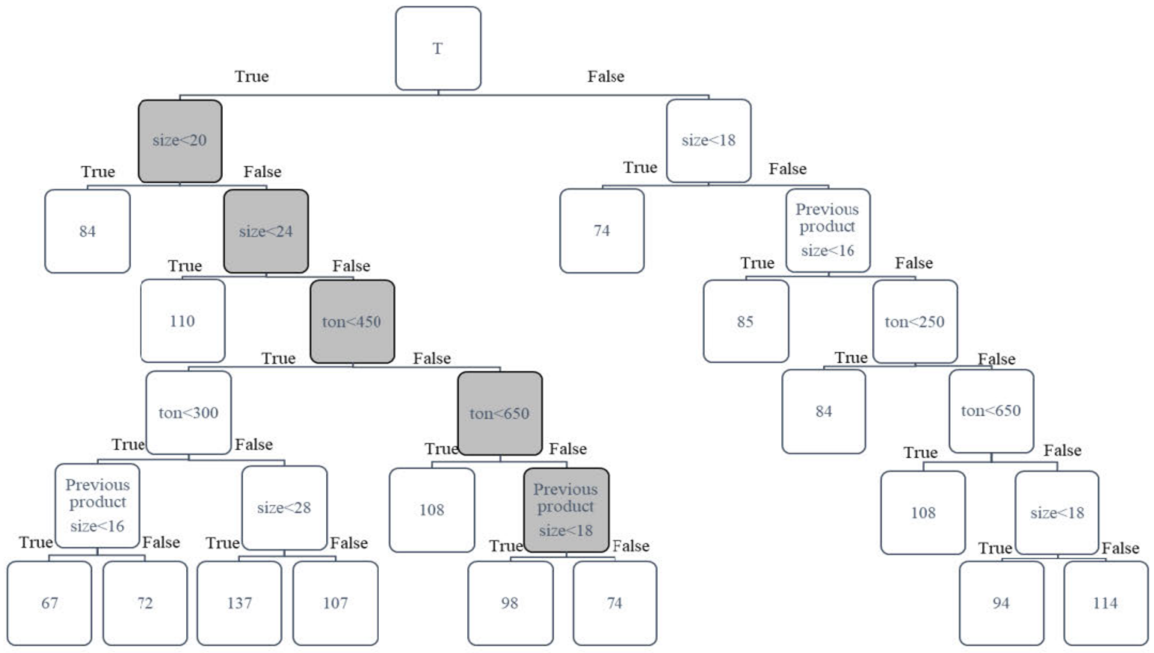


FIGURE 6. Decision tree for an exemplary product.

setup situations, each of which representing the possible preparation time before the machine begins a given operation. Given the defined attributes, the sample data will be used as training input to the RF procedure. The output of this procedure is summarized in forms of a matrix, to be used as input to the solution algorithm, which results in the initial job sequence to the $R_m|S_{ijk}|C_{\max}$ problem, applying the RF method ($RF_{\max}$). Given the defined rules, the next step considers ten independent samples from the historic data to resolve the $R_m|S_{ijk}|C_{\max}$ problem, and obtain the respective objective function values. In the illustrative example, the completion time of the operations, O_1 and O_2, on M_1 and M_2, when considering the first sample, is as follows: $M_1A_1 + P_{11} + M_1A_4 + P_{14} = 2 + 2 + 4 + 3 = 11$ and $M_2A_2 + P_{22} + M_2A_3 + P_{23} = 1 + 2 + 1 + 5 = 9$; therefore, the makespan of the sample is equal to $C_{\max}(O_1) = 11$.

Given the objective function value for each of the samples, equation (10) is used to calculate the respective error rate $RF_{er(i)}$. On this basis, the resulted average error, 9.71 in the illustrative example determines the measure on the accuracy of the results when applying the RF method.

$$Error = \frac{C_{\max}(i) - RF_{\max}}{RF_{\max}} \times 100\% \qquad (10)$$

b: PHASE II. SOLUTION APPROACH

Given the obtained setup times in Phase I, the HABC algorithm can be applied to find the (near-)optimal solution to the $R_m|S_{ijk}|C_{\max}$ problem. The mathematical formulation and optimization procedures were detailed in sections 3.2 and 3.3.

IV. NUMERICAL ANALYSIS

This section presents the computational results and analysis of the proposed framework. The numerical analysis begins with a description of the test data. RF analysis is then presented. Finally, verification and evaluation steps are elaborated to show the superiority of the proposed approach. The numerical experiments are conducted on a personal computer with an Intel(R) Core (TM) i7-6700HQ (2.6GHz) processor, 24GB of RAM, and a Windows 10 operating system. The solution algorithm was coded in Python programming language.

A. CASE DESCRIPTION

Taking the forging industry as an example, the test data is partially obtained from a small-parts forging company. Forging refers to the procedure of deforming metals, applying comprehensive force on the work-in-progress material using heavy equipment. Given the processing temperature, forging can be categorized into hot satin, warm satin, and cold satin groups. Cold forging consists of molding the metal at room temperature, and pressing the plate, or bar, after repeated extrusion. In cold satin, it is relatively easy to obtain high dimensional accuracy and surface smoothness, when compared to hot satin. However, the plasticity of the forged material can be rather low.

Different pressure norms may be required considering the dimension of the product. The force generated by a compression molding machine is directly proportionate to its tonnage, that is, the heavier compression machines are often characterized by larger pressure upper limit [74]. Given that machinery with the same tonnage may also perform differently,

TABLE 1. Computational results of the estimation accuracy for [50-100] setup time interval.

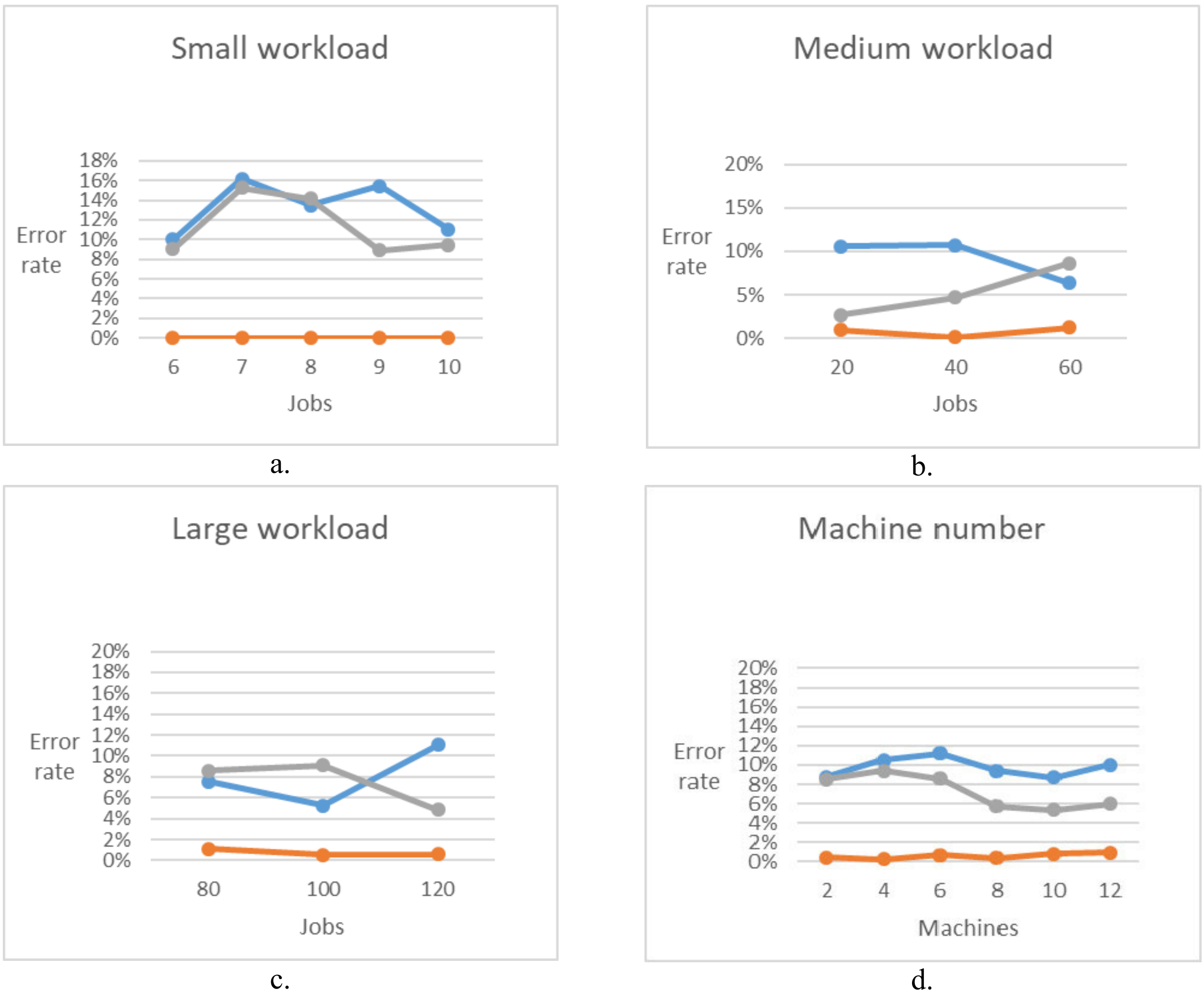
		AV-HABC				*DT-HABC*				*RF-HABC*			
m	*n*	*Min*	*Avg*	*Max*	*StD*	*Min*	*Avg*	*Max*	*StD*	*Min*	*Avg*	*Max*	*StD*
2	6	0.43	4.21	10.18	4.04	3.45	6.75	10.02	2.49	0.00	3.76	7.25	2.95
	7	0.18	5.42	17.10	6.72	0.17	5.10	13.85	5.42	0.00	4.53	9.27	4.14
	8	1.57	6.34	14.18	4.71	2.57	7.08	10.13	3.0%	0.00	5.42	9.55	4.03
	9	3.80	6.06	10.83	2.83	2.03	7.10	12.92	4.61	0.00	2.27	6.32	2.4%
	10	2.59	6.09	9.26	2.39	5.94	7.76	12.77	2.84	0.00	3.17	8.94	3.38
	20	1.67	2.74	4.58	1.14	6.06	6.86	7.73	0.67	0.23	0.83	2.24	0.88
	40	4.48	5.71	7.64	1.28	7.09	7.97	8.52	0.58	0.04	0.53	1.08	0.40
	60	6.66	7.84	10.08	1.39	5.94	7.32	8.65	1.07	0.49	0.98	1.62	0.52
	80	10.70	11.84	13.32	0.98	7.22	7.53	7.95	0.32	0.42	0.79	1.43	0.40
	100	10.07	11.07	12.73	1.04	5.21	6.25	7.86	0.98	0.67	0.97	1.17	0.21
	120	4.23	5.41	6.31	0.75	9.59	9.98	10.53	0.40	0.16	0.87	1.45	0.47
4	6	0.73	5.65	11.50	3.97	2.41	7.43	12.38	3.53	0.00	2.78	5.37	1.95
	7	6.42	8.38	13.44	2.96	7.54	11.47	18.50	4.45	0.00	3.24	6.65	3.04
	8	0.73	7.79	15.48	6.43	6.91	9.23	12.38	2.77	0.00	6.18	15.12	6.34
	9	1.58	3.77	6.50	2.19	4.56	11.05	17.51	5.56	0.00	2.16	3.46	1.34
	10	2.62	8.78	14.25	4.63	6.34	9.19	13.07	3.21	0.00	4.88	8.47	3.16
	20	2.21	4.29	6.73	1.70	4.63	6.89	9.25	1.65	0.16	0.92	2.73	1.03
	40	1.25	3.09	5.58	1.64	9.90	10.62	11.82	0.73	0.08	0.48	1.59	0.63
	60	13.02	13.71	14.70	0.64	1.47	3.57	5.04	1.50	1.20	1.59	2.37	0.45
	80	10.15	10.77	11.85	0.66	1.87	4.03	5.19	1.37	0.04	0.54	1.53	0.58
	100	10.82	11.93	12.96	0.83	4.29	5.13	6.08	0.81	0.03	0.61	1.45	0.71
	120	4.68	6.02	6.73	0.83	10.56	10.66	10.76	0.09	0.17	0.70	1.04	0.33
6	8	8.13	11.47	13.33	2.03	4.76	11.61	17.98	5.30	0.00	9.08	16.67	6.27
	9	6.14	9.54	13.33	2.55	6.47	11.95	17.98	4.77	0.00	9.38	21.43	7.94
	10	5.05	7.29	10.24	2.15	6.48	9.13	13.56	3.04	0.00	4.88	7.84	3.33
	20	1.98	3.90	5.53	1.71	10.63	11.16	11.96	0.61	0.21	1.48	2.62	0.95
	40	5.07	7.16	10.11	2.54	12.65	13.62	14.66	0.71	0.25	0.54	0.85	0.25
	60	11.64	12.31	13.17	0.64	1.08	3.67	5.58	1.94	0.23	1.04	2.16	0.71
	80	8.35	9.33	11.29	1.18	4.29	6.54	7.87	1.47	0.61	1.21	1.77	0.46
	100	12.17	12.69	13.25	0.47	1.90	2.58	3.63	0.77	0.00	0.58	2.15	0.90
	120	4.14	5.78	7.07	1.28	10.48	11.18	11.75	0.61	0.04	0.42	0.69	0.26
8	20	4.56	6.57	8.72	1.69	13.41	15.27	18.95	2.23	0.28	1.33	3.55	1.31
	40	5.41	7.12	8.56	1.31	10.73	11.62	12.81	0.93	0.00	0.82	1.63	0.70
	60	6.33	8.83	11.65	1.96	5.29	8.35	10.13	1.90	0.22	1.08	2.35	0.91
	80	5.00	6.87	8.43	1.25	8.04	10.60	12.62	1.78	0.85	1.73	3.49	1.07
	100	6.61	8.09	9.67	1.46	5.75	7.40	8.95	1.14	0.12	0.39	0.84	0.32
	120	6.47	8.05	9.44	1.09	10.96	11.31	11.64	0.24	0.23	0.44	0.99	0.31
10	20	0.00	2.45	4.64	2.01	9.32	14.30	19.68	4.22	0.00	0.94	1.58	0.60
	40	4.73	6.05	8.53	1.55	11.74	12.76	14.26	1.01	0.00	1.98	3.80	1.48
	60	7.14	9.18	11.22	1.60	7.17	9.69	12.08	1.77	1.13	2.54	4.10	1.32
	80	6.44	7.09	8.29	0.78	8.73	12.32	14.18	2.23	0.85	1.82	4.42	1.47
	100	7.61	9.01	9.80	0.96	1.24	3.15	4.37	1.17	0.22	0.70	1.27	0.38
	120	4.89	6.37	7.66	1.18	9.40	10.29	11.10	0.63	0.14	0.87	1.77	0.62
12	20	4.83	7.16	9.22	1.63	14.89	18.06	21.31	2.93	0.41	2.22	4.69	1.57
	40	6.61	9.16	12.89	2.76	7.33	9.49	13.50	2.35	0.00	1.16	3.59	1.43
	60	4.84	7.84	9.64	1.85	7.30	10.79	13.86	2.34	1.69	2.47	3.64	0.82
	80	6.43	7.80	8.60	0.90	8.23	11.41	13.89	2.17	0.50	1.36	2.33	0.74
	100	5.86	6.56	7.15	0.46	6.57	7.95	9.80	1.45	0.09	0.86	1.68	0.64
	120	4.65	6.82	8.26	1.42	10.20	11.29	12.06	0.70	0.17	0.45	0.69	0.25

TABLE 2. Computational results of the estimation accuracy for [50-150] setup time interval.

		AV-HABC				*DT-HABC*				*RF-HABC*			
m	*n*	*Min*	*Avg*	*Max*	*StD*	*Min*	*Avg*	*Max*	*StD*	*Min*	*Avg*	*Max*	*StD*
2	6	2.70	4.81	10.57	3.28	4.81	8.42	13.89	3.65	0.00	2.96	11.25	4.70
	7	6.06	7.72	12.05	2.48	5.01	9.59	14.75	4.79	0.15	1.88	5.63	2.26
	8	2.40	4.51	7.42	2.29	0.44	5.08	11.70	4.46	0.12	0.83	1.74	0.73
	9	9.70	12.63	15.24	2.61	0.36	1.41	4.76	1.87	0.00	1.14	3.82	1.58
	10	8.73	11.85	17.01	3.49	9.88	11.08	12.92	1.12	0.01	0.85	2.95	1.22
	20	14.22	14.22	14.22	0.00	9.44	9.44	9.44	0.00	0.99	0.99	0.99	0.00
	40	15.07	16.25	18.00	1.07	8.53	9.63	10.38	0.82	0.61	1.29	1.84	0.51
	60	14.07	14.54	14.99	0.40	5.91	6.81	8.08	0.94	0.08	0.42	1.09	0.42
	80	12.98	14.76	15.42	1.01	6.18	7.35	11.29	2.2	0.10	0.43	1.14	0.41
	100	3.25	5.15	9.25	2.37	7.50	11.49	16.47	4.39	0.29	0.48	0.59	0.15
	120	1.48	3.41	7.59	2.42	5.40	9.64	16.82	4.47	0.28	2.00	5.30	2.05
4	6	0.61	7.01	13.60	6.04	0.28	4.91	11.47	5.11	0.52	2.99	11.97	5.02
	7	7.00	9.44	11.80	1.70	10.72	13.17	15.94	2.00	0.00	0.22	0.61	0.31
	8	20.95	21.61	23.81	1.23	7.31	8.61	9.22	0.83	0.00	0.30	1.48	0.66
	9	11.30	13.44	17.28	2.41	10.20	11.63	14.29	1.56	0.34	2.42	7.50	3.02
	10	17.90	19.79	21.75	1.41	6.86	8.15	9.28	0.96	0.14	0.44	0.73	0.23
	20	12.98	13.94	15.39	0.91	10.20	11.24	12.24	0.87	0.10	0.46	0.75	0.31
	40	15.39	16.43	17.77	0.88	11.28	11.89	12.14	0.35	0.08	0.25	0.57	0.19
	60	17.52	19.60	22.58	2.56	0.80	2.77	5.31	2.23	0.29	2.23	7.31	2.89
	80	4.09	6.92	12.61	3.29	9.49	11.63	12.30	1.20	0.00	1.14	5.70	2.55
	100	5.52	8.41	13.73	3.14	10.82	11.29	12.44	0.65	0.30	0.89	1.68	0.61
	120	18.31	19.42	20.85	0.99	10.42	13.08	18.46	3.23	0.41	1.20	2.29	0.72
6	8	17.84	18.64	19.85	0.93	10.41	11.94	13.39	1.12	0.00	0.85	3.76	1.63
	9	15.08	16.81	18.18	1.23	8.53	11.46	12.42	1.65	0.08	0.25	0.46	0.15
	10	13.79	15.04	16.75	1.12	7.13	8.14	10.11	1.22	0.35	1.31	4.78	1.94
	20	9.49	9.78	10.13	0.29	5.44	9.12	12.50	2.95	0.04	0.82	1.77	0.66
	40	18.03	18.96	20.35	1.07	12.83	16.76	18.51	2.31	0.13	1.74	5.30	2.05
	60	15.34	16.44	16.82	0.62	10.11	12.63	15.38	2.35	0.30	1.83	4.78	1.84
	80	15.17	17.02	19.44	1.84	8.10	10.98	13.74	2.51	0.62	1.38	2.36	0.83
	100	13.21	17.00	18.43	2.16	16.32	18.44	25.00	3.76	0.00	2.73	8.17	3.19
	120	12.72	13.71	15.02	1.10	8.63	10.61	12.36	1.56	0.19	1.02	1.85	0.66
8	20	14.83	17.55	19.44	2.15	10.79	12.91	16.37	2.12	0.13	1.74	3.56	1.45
	40	15.68	17.52	19.57	1.68	11.22	13.07	14.85	1.63	0.29	2.02	4.42	1.73
	60	16.63	17.51	18.30	0.66	10.92	11.36	12.09	0.44	0.31	2.25	5.25	1.83
	80	13.28	15.33	17.91	1.68	19.03	20.44	24.59	2.33	0.32	0.79	2.15	0.78
	100	12.09	14.58	21.47	3.88	14.29	15.65	17.69	1.37	0.00	0.03	0.16	0.07
	120	8.12	15.23	17.53	3.99	10.56	11.58	12.93	0.95	0.31	2.11	5.25	1.85
10	20	4.99	9.66	17.30	4.58	7.83	11.92	14.81	2.56	0.00	0.51	1.66	0.69
	40	14.64	16.24	17.62	1.31	13.88	15.30	16.97	1.18	0.04	1.57	6.56	2.80
	60	0.23	1.14	3.57	1.39	12.22	14.01	15.60	1.30	0.23	1.14	3.57	1.39
	80	17.51	19.27	20.76	1.27	9.67	9.94	10.37	0.31	0.16	1.92	3.49	1.54
	100	17.68	18.37	19.49	0.69	9.33	10.66	11.35	0.86	0.31	1.18	2.38	0.77
	120	14.64	16.24	17.62	1.31	13.88	15.30	16.97	1.18	0.04	1.57	6.56	2.80
12	20	0.23	1.14	3.57	1.39	12.22	14.01	15.60	1.30	0.23	1.14	3.57	1.39
	40	13.77	14.23	14.56	0.30	6.83	7.14	7.53	0.31	0.07	0.41	0.93	0.34
	60	17.09	17.27	17.69	0.24	9.11	9.43	9.84	0.33	0.05	0.42	0.93	0.32
	80	15.62	16.89	18.05	0.94	7.92	8.97	9.93	0.76	0.05	0.28	0.56	0.19
	100	15.98	16.40	16.90	0.34	9.93	10.41	10.82	0.40	0.20	0.54	0.87	0.27
	120	16.12	18.14	19.68	1.47	9.40	11.19	12.45	1.15	0.30	0.51	1.04	0.3%

TABLE 3. Computational results of the estimation accuracy for [50-200] setup time interval.

		AV-HABC				*DT-HABC*				*RF-HABC*			
m	*n*	*Min*	*Avg*	*Max*	*StD*	*Min*	*Avg*	*Max*	*StD*	*Min*	*Avg*	*Max*	*StD*
2	6	11.01	3.74	1.16	5.29	10.05	3.60	0.88	4.06	7.53	2.62	11.01	3.74
	7	18.10	6.85	0.36	7.51	19.13	9.62	0.59	1.23	2.41	0.74	18.10	6.85
	8	21.06	3.99	0.24	2.27	3.97	1.84	0.38	2.17	4.48	1.63	21.06	3.99
	9	29.52	9.24	4.26	8.03	13.27	3.28	0.76	2.68	5.10	1.94	29.52	9.24
	10	24.68	3.89	2.75	5.28	7.08	1.88	0.15	1.82	6.62	2.73	24.68	3.89
	20	31.11	5.91	0.03	2.34	7.23	2.80	0.70	2.08	4.53	1.53	31.11	5.91
	40	28.24	2.12	7.12	14.35	18.90	4.68	9.24	13.42	15.53	2.50	28.24	2.12
	60	21.32	2.58	1.68	3.41	6.48	1.85	0.40	1.00	2.09	0.72	21.32	2.58
	80	30.83	5.25	11.78	8.02	8.85	1.04	0.74	6.73	20.24	8.00	30.83	5.25
	100	37.65	5.16	2.98	9.99	16.73	6.39	0.34	3.19	8.36	3.19	37.65	5.16
	120	20.30	4.20	4.55	6.24	8.25	1.71	0.00	2.05	6.65	2.77	20.30	4.20
4	6	27.47	5.11	7.00	14.35	24.12	9.01	0.39	2.54	8.74	3.49	27.47	5.11
	7	34.71	5.40	1.78	5.24	12.45	4.53	0.34	3.19	8.36	3.19	34.71	5.40
	8	9.63	2.21	3.28	9.08	13.52	4.41	0.38	2.17	3.84	1.30	9.63	2.21
	9	6.55	2.51	1.48	3.47	5.14	1.72	0.51	1.31	2.37	0.96	6.55	2.51
	10	6.22	1.27	0.81	4.17	8.30	2.73	0.03	2.86	6.04	2.62	6.22	1.27
	20	14.18	6.34	1.79	7.01	9.21	3.16	0.72	6.20	15.78	6.15	14.18	6.34
	40	21.23	7.13	11.78	9.99	15.93	5.08	0.00	8.38	16.23	6.14	21.23	7.13
	60	5.64	1.82	0.96	4.29	8.55	3.07	0.08	1.06	1.93	0.79	5.64	1.82
	80	25.86	2.44	0.47	3.65	7.03	2.63	0.11	0.32	0.87	0.32	25.86	2.44
	100	26.85	3.03	0.76	1.68	2.88	0.90	0.04	0.77	2.30	0.90	26.85	3.03
	120	32.93	4.60	0.55	4.58	8.62	3.78	0.13	2.44	6.66	2.97	32.93	4.60
6	8	30.32	2.09	0.01	5.24	8.11	3.23	0.73	2.99	7.00	2.36	30.32	2.09
	9	35.03	4.28	7.00	4.47	10.79	4.64	0.56	2.09	4.63	1.64	35.03	4.28
	10	32.05	1.59	1.12	4.91	8.22	3.08	0.48	1.73	2.66	1.03	32.05	1.59
	20	15.77	2.37	0.61	2.01	3.69	1.29	1.62	1.95	2.31	0.28	15.77	2.37
	40	5.74	1.71	0.53	2.98	6.07	2.33	0.47	1.03	1.57	0.45	5.74	1.71
	60	17.37	5.28	3.38	5.04	6.30	1.06	0.66	3.00	4.57	1.66	17.37	5.28
	80	19.52	3.10	1.62	2.08	2.82	0.48	0.89	1.57	2.04	0.53	19.52	3.10
	100	7.56	1.89	0.72	3.81	7.15	2.68	0.49	2.31	6.37	2.39	7.56	1.89
	120	18.15	6.32	0.99	2.52	4.19	1.31	1.31	2.50	4.37	1.31	18.15	6.32
8	20	11.79	0.98	0.81	1.65	2.42	0.68	0.05	0.44	0.67	0.25	11.79	0.98
	40	9.59	1.61	1.04	2.79	5.46	1.88	0.46	1.22	2.28	0.78	9.59	1.61
	60	11.10	3.70	3.07	7.01	9.93	2.73	0.66	2.75	4.36	1.68	11.10	3.70
	80	10.10	2.91	2.23	3.99	5.68	1.48	1.56	3.18	6.69	2.03	10.10	2.91
	100	15.23	2.50	0.28	3.78	5.93	2.21	1.90	3.65	7.40	2.16	15.23	2.50
	120	6.61	2.12	1.15	4.92	8.40	2.65	0.32	1.37	3.83	1.42	6.61	2.12
10	20	9.93	0.40	0.62	2.83	4.25	1.39	1.03	2.80	4.11	1.33	9.93	0.40
	40	8.56	3.17	1.71	2.65	3.72	0.75	0.11	2.26	4.24	1.84	8.56	3.17
	60	7.11	1.81	0.17	2.38	5.15	2.25	0.17	1.31	3.13	1.13	7.11	1.81
	80	15.23	2.20	0.11	2.23	5.51	2.21	0.75	2.18	4.00	1.31	15.23	2.20
	100	12.30	2.35	0.19	2.78	6.16	2.56	3.17	5.84	8.67	2.48	12.30	2.35
	120	12.92	3.04	0.10	1.76	3.74	1.43	0.87	1.67	2.48	0.61	12.92	3.04
12	20	24.30	1.01	1.55	2.12	2.92	0.57	0.03	0.29	0.85	0.35	24.30	1.01
	40	24.09	1.05	0.96	1.89	2.60	0.77	0.03	0.36	0.81	0.30	24.09	1.05
	60	27.53	0.37	0.27	2.26	5.56	2.22	0.19	0.76	1.62	0.57	27.53	0.37
	80	28.19	1.96	1.38	1.93	2.74	0.61	0.07	1.89	3.46	1.29	28.19	1.96
	100	29.37	1.85	0.39	2.04	5.61	2.09	1.08	1.67	2.47	0.60	29.37	1.85
	120	27.77	1.38	0.39	1.34	2.54	0.78	0.64	1.06	2.33	0.72	27.77	1.38



Blue-line: DT-HABC, Green-line: AV-HABC, Red-line: RF-HABC

FIGURE 7. Error rate changes in (a) small-workload, (b) medium-workload, (c) large-workload, and (d) number of machines, within [50,100] interval.

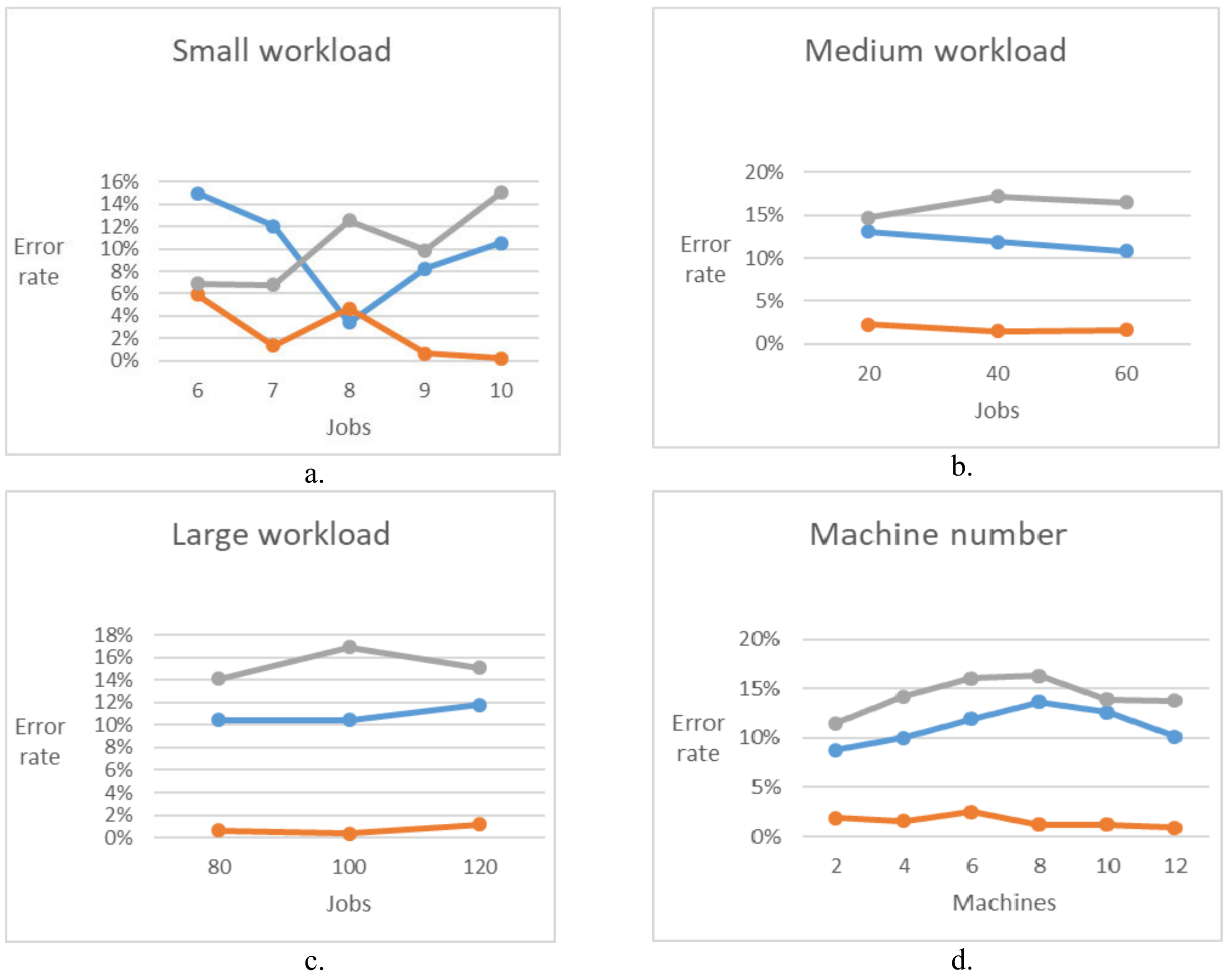
the processing time of a product differs from one machine to another. Likewise, setup time may vary depending on the machine, as well as the new and preceding products.

Case factory data is obtained to estimate MDJSDSTs using the RF method. Small-, medium- and large-scale test problems are considered to analyze the problem size effect on estimation accuracy. For this purpose, and given the wide range of products and machinery in the case company, ten random samples from the top 20 percent of the products are considered to generate the test problems. Due to the complications involved in collecting the rest of the required data, random data has been used.

Overall, small-, medium-, and large-scale problem instances are considered to conduct numerical tests. Small-size instances are configured by $n = \{6, 7, 8, 9, 10\}$ jobs and $m = \{2, 4, 6\}$ machines. Considering ten instances for each of the $m \times n$ combinations, a total of $10 \times \{6, 7, 8, 9, 10\} \times \{2, 4, 6\} \times \{[50, 100], [50, 150], [50, 200]\} = 450$ test examples have resulted. The medium- and large-size problems comprise $n = \{20, 40, 60, 80, 100, 120\}$ jobs and $m = \{2, 4, 6, 8, 10, 12\}$ machines, resulting in a total of $10 \times 6 \times 6 \times 3 = 1080$ examples.

B. RANDOM FOREST ANALYSIS FOR ESTIMATING MDJSDSTs

Given the obtained case company data, RF is now applied to analyze the correlation among the attributes, and estimate the setup times. Figure 6 is a visual representation of the decision tree for an exemplary product from the case company, where the product type is downwardly expanded to obtain the respective MDJSDSTs values. For example, if the frame dimension is smaller than 20, and the outer diameter of the product is also smaller than 20, the estimated setup time is approximately 84 seconds; otherwise, two distinct situations may result. If the outer diameter is less than 24, a setup time



Blue-line: DT-HABC, Green-line: AV-HABC, Red-line: RF-HABC

FIGURE 8. Error rate changes in (a) small-workload, (b) medium-workload, (c) large-workload, and (d) number of machines, within [50,150] interval.

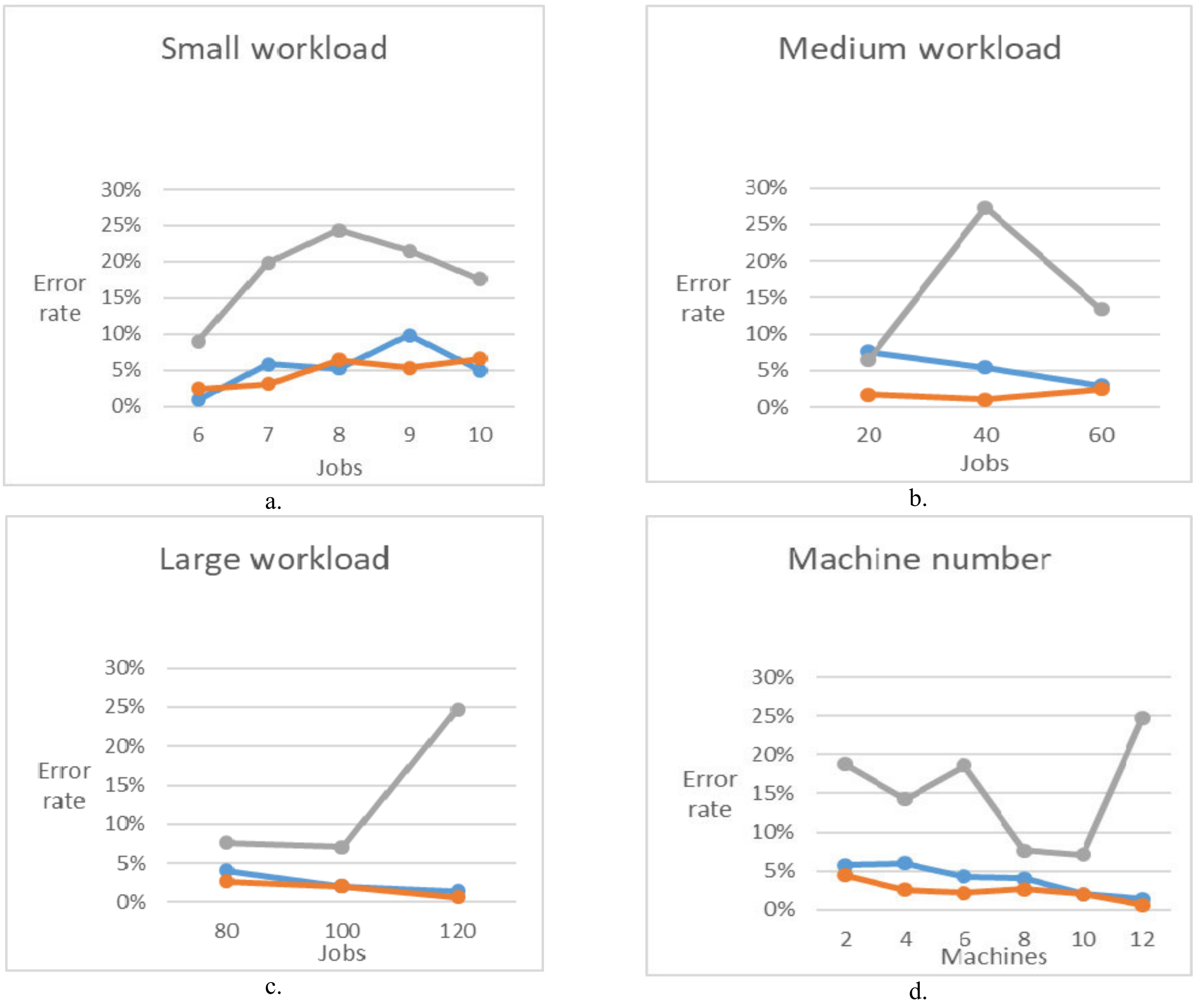
of 110 seconds can be expected, or else, the non-conformity is less than 450, and the expected setup time will be determined based on the machine's tonnage, and the previous work on the machine.

C. EXPERIMENTAL RESULTS

After estimating the MDJSDSTs using the RF approach in phase I, the HABC algorithm in phase II of the RF-HABC is applied to find the (near-)optimal solution to the $R_m|S_{ijk}|C_{\max}$ problem. The simple average (AV), and decision trees (DT) methods are considered to replace the proposed RF learning method in phase I of the algorithm, named AV-HABC and DT-HABC, to analyze the experimental results. To compare the accuracy of the mentioned methods, small, medium and large instances including 6-10, 20-60, and 80-120 jobs, respectively, to be loaded on two to twelve machines, are considered. Besides, three different distribution patterns for setup time interval data, [50, 100], [50, 150] and [50, 200], which are all in seconds, are taken into consideration to explore different data distribution conditions and their respective impact on the estimation methods accuracy. These intervals are separately analyzed.

The error ratios for [50, 100], [50, 150] and [50, 200] setup time intervals are separately presented in Tables 1-3. Given ten runs for each of the test problems, the minimum (Min), average (Avg), maximum (Max) error ratios, as well as the standard deviation (StD) of the test instances are summarized in these tables. As expected, the average error of all three approaches increases when more jobs and machines are included in the problem. However, RF-HABC demonstrates a relatively stable error ratio over the test instances. In other words, problem size does not significantly influence the solution quality when the RF method is applied.

Figures 7-9 visually compare error trend lines amongst RF-HABC, DT-HABC and AV-HABC methods, considering [50, 100], [50, 150] and [50, 200] time intervals, respectively. Each of the figures consists of four separate diagrams to detect the difference in accuracy changes in the small,



Blue-line: DT-HABC, Green-line: AV-HABC, Red-line: RF-HABC

FIGURE 9. Error rate changes in (a) small-workload, (b) medium-workload, (c) large-workload, and (d) number of machines, within [50,200] interval.

TABLE 4. Percentage error under different setup time intervals.

Intervals	AV-HABC	DT-HABC	RF-HABC
[50,100]	7.28	9.77	0.60
[50,150]	14.28	11.18	1.54
[50,200]	15.17	3.95	2.43
Average (%)	12.24	8.30	1.52

medium, and large workloads, as well as the defined range of machinery, respectively.

In general, the trend line of RF-HABC's error ratio is placed below that of DT-HABC and AV-HABC methods. Evidently, RF-HABC demonstrates a relatively stable performance across various problem situations. To be more specific, RF-HABC meaningfully outperforms the other two methods when [50, 100] and [50, 150] time intervals are the case. The difference in the accuracy of RF-HABC and DT-HABC methods, however, tends to become less significant, when a loose data interval, [50, 200], is considered. In other words, DT-HABC demonstrates an accuracy comparable to that of

RF-HABC in some of the large test instances, when the setup times are distributed in a low-density pattern. This general conclusion can be drawn that RF-based estimation methods are strongly preferred when small, and medium time intervals are prevalent.

Table 4 provides a summary of the average errors for each of the applied methods. Overall, RF-HABC's error ratio is about 1.52 percent, which is significantly -about 87 percent- more accurate than the cases where DT-HABC and AV-HABC methods are applied, with 8.30 and 12.24 percentage of the solution, respectively. This difference may become more significant when industry-scale problems are solved.

V. CONCLUSION

To effectively model the scheduling problems, different aspects need to be taken into consideration among which, time parameters uncertainties play a significant role. Setup time includes a variety of preparation activities that need to be executed before the value-adding production processes, and are particularly important when general-purpose machinery dominate. The extant scheduling problems rely on simplistic methods for the estimation of setup times, although the inherent errors caused by inaccurate approaches may result in significant delays in the production plans.

This study proposed an RF-HABC metaheuristic algorithm to solve the $R_m|S_{ijk}|C_{\max}$ problem, considering setup time uncertainties in a machine-dependent and job sequence-dependent scheduling environment. Through exhaustive numerical analysis, it is shown that RF-HABC outperforms the simple averaging, and decision tree approaches concerning accuracy. With an average error rate of 1.52 percent, the RF-based solution algorithm accurately estimates the uncertain MDJSDSTs in UPMSP, where DT-HABC and AV-HABC recorded 12.24 and 8.30 percent of error, respectively. This difference in accuracy was more meaningful when the problem size increases, and the setup time intervals got larger.

Given the abundance of factors affecting setup times, we feel that a deeper analysis of this important scheduling aspect can contribute to the scheduling literature, and help narrow the gap between scheduling theory and practice. In this study, RF and decision tree methods were successfully applied to estimate MDJSDSTs. Future studies can apply other advanced analytics and machine learning methods to further improve the data exploration process in other industrial situations. Last but not least, our study is limited in that it only estimates the setup time uncertainty. Future research can address processing time, and more generally, waiting time, uncertainties through applying simulation-based optimization approaches.

REFERENCES

[1] I. Ribas, R. Leisten, and J. M. Framiñan, "Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective," *Comput. Oper. Res.*, vol. 37, no. 8, pp. 1439–1454, Aug. 2010.

[2] G. M. Kopanos, J. M. Laínez, and L. Puigjaner, "An efficient mixed-integer linear programming scheduling framework for addressing sequence-dependent setup issues in batch plants," *Ind. Eng. Chem. Res.*, vol. 48, no. 13, pp. 6346–6357, Jul. 2009.

[3] A. Allahverdi, "The third comprehensive survey on scheduling problems with setup times/costs," *Eur. J. Oper. Res.*, vol. 246, no. 2, pp. 345–378, Oct. 2015.

[4] A. Allahverdi and H. M. Soroush, "The significance of reducing setup times/setup costs," *Eur. J. Oper. Res.*, vol. 187, no. 3, pp. 978–984, Jun. 2008.

[5] H. H. Miyata and M. S. Nagano, "The blocking flow shop scheduling problem: A comprehensive and conceptual review," *Expert Syst. Appl.*, vol. 137, pp. 130–156, Dec. 2019.

[6] X. Zhu and W. E. Wilhelm, "Scheduling and lot sizing with sequence-dependent setup: A literature review," *IIE Trans.*, vol. 38, no. 11, pp. 987–1007, Nov. 2006.

[7] C.-L. Chen and C.-L. Chen, "Hybrid metaheuristics for unrelated parallel machine scheduling with sequence-dependent setup times," *Int. J. Adv. Manuf. Technol.*, vol. 43, nos. 1–2, pp. 161–169, Jul. 2009.

[8] L. Fanjul-Peyro, R. Ruiz, and F. Perea, "Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times," *Comput. Oper. Res.*, vol. 101, pp. 173–182, Jan. 2019.

[9] Z. Zhang, L. Zheng, N. Li, W. Wang, S. Zhong, and K. Hu, "Minimizing mean weighted tardiness in unrelated parallel machine scheduling with reinforcement learning," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1315–1324, Jul. 2012.

[10] H. G. Santos, T. A. M. Toffolo, C. L. T. F. Silva, and G. V. Berghe, "Analysis of stochastic local search methods for the unrelated parallel machine scheduling problem," *Int. Trans. Oper. Res.*, vol. 26, no. 2, pp. 707–724, Mar. 2019.

[11] P. Pourhejazy and O. Kwon, "The new generation of operations research methods in supply chain optimization: A review," *Sustainability*, vol. 8, no. 10, p. 1033, Oct. 2016, doi: 10.3390/su8101033.

[12] A. Öztürk, S. Kayalıgil, and N. E. Özdemirel, "Manufacturing lead time estimation using data mining," *Eur. J. Oper. Res.*, vol. 173, no. 2, pp. 683–700, Sep. 2006.

[13] S. C. Kim and P. M. Bobrowski, "Scheduling jobs with uncertain setup times and sequence dependency," *Omega*, vol. 25, no. 4, pp. 437–447, Aug. 1997.

[14] L. van der Heyden, "Scheduling jobs with exponential processing and arrival times on identical processors so as to minimize the expected makespan," *Math. Oper. Res.*, vol. 6, no. 2, pp. 305–312, May 1981.

[15] R. R. Weber, "Scheduling jobs by stochastic processing requirements on parallel machines to minimize makespan or flowtime," *J. Appl. Probab.*, vol. 19, no. 1, pp. 167–182, Mar. 1982.

[16] Y. Qiao, N. Wu, F. Yang, M. Zhou, and Q. Zhu, "Wafer sojourn time fluctuation analysis of time-constrained dual-arm cluster tools with wafer revisiting and activity time variation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 4, pp. 622–636, Apr. 2018.

[17] C.-C. Lu, S.-W. Lin, and K.-C. Ying, "Robust scheduling on a single machine to minimize total flow time," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1682–1691, Jul. 2012.

[18] C.-C. Lu, S.-W. Lin, and K.-C. Ying, "Minimizing worst-case regret of makespan on a single machine with uncertain processing and setup times," *Appl. Soft Comput.*, vol. 23, pp. 144–151, Oct. 2014.

[19] A. Aydilek, H. Aydilek, and A. Allahverdi, "Increasing the profitability and competitiveness in a production environment with random and bounded setup times," *Int. J. Prod. Res.*, vol. 51, no. 1, pp. 106–117, Jan. 2013.

[20] H. Aydilek and A. Allahverdi, "A polynomial time heuristic for the two-machine flowshop scheduling problem with setup times and random processing times," *Appl. Math. Model.*, vol. 37, nos. 12–13, pp. 7164–7173, Jul. 2013.

[21] A. Allahverdi, "Three-machine flowshop scheduling problem to minimize makespan with bounded setup and processing times," *J. Chin. Inst. Ind. Eng.*, vol. 25, no. 1, pp. 52–61, Jan. 2008.

[22] S. A. Torabi, N. Sahebjamnia, S. A. Mansouri, and M. A. Bajestani, "A particle swarm optimization for a fuzzy multi-objective unrelated parallel machines scheduling problem," *Appl. Soft Comput.*, vol. 13, no. 12, pp. 4750–4762, Dec. 2013.

[23] M. Naderi-Beni, E. Ghobadian, S. Ebrahimnejad, and R. Tavakkoli-Moghaddam, "Fuzzy bi-objective formulation for a parallel machine scheduling problem with machine eligibility restrictions and sequence-dependent setup times," *Int. J. Prod. Res.*, vol. 52, no. 19, pp. 5799–5822, Oct. 2014.

[24] Q. Zhu, M. Zhou, Y. Qiao, and N. Wu, "Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 3, pp. 389–400, Mar. 2018.

[25] R. Unbehaun and O. Rose, "The use of slow down factors for the analysis and development of scheduling algorithms for parallel cluster tools," in *Proc. Winter Simulation Conf.*, Dec. 2006, pp. 1840–1847.

[26] H.-J. Kim, J.-H. Lee, and T.-E. Lee, "Noncyclic scheduling of cluster tools with a branch and bound algorithm," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 690–700, Apr. 2015.

[27] L. Bai, N. Wu, Z. Li, and M. Zhou, "Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 10, pp. 1456–1467, Oct. 2016.

[28] N. Wu, F. Chu, C. Chu, and M. Zhou, "Petri net modeling and cycle-time analysis of dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 1, pp. 196–207, Jan. 2013.

[29] N. Qi Wu and M. Zhou, "Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 446–454, Apr. 2012.

[30] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 1981.

[31] L. Hidri, A. M. Al-Samhan, and M. M. Mabkhot, "Bounding strategies for the parallel processors scheduling problem with no-idle time constraint, release date, and delivery time," *IEEE Access*, vol. 7, pp. 170392–170405, 2019.

[32] M. Nouri and M. Ghodsi, "Scheduling tasks with exponential duration on unrelated parallel machines," *Discrete Appl. Math.*, vol. 160, nos. 16–17, pp. 2462–2473, Nov. 2012.

[33] X. Xu, W. Cui, J. Lin, and Y. Qian, "Robust makespan minimisation in identical parallel machine scheduling problem with interval data," *Int. J. Prod. Res.*, vol. 51, no. 12, pp. 3532–3548, Jun. 2013.

[34] J. Yang and G. Yu, "On the robust single machine scheduling problem," *J. Combinat. Optim.*, vol. 6, no. 1, pp. 17–33, 2002.

[35] X. Xu, J. Lin, and W. Cui, "Hedge against total flow time uncertainty of the uniform parallel machine scheduling problem with interval data," *Int. J. Prod. Res.*, vol. 52, no. 19, pp. 5611–5625, Oct. 2014.

[36] H. Hu, K. K. H. Ng, and Y. Qin, "Robust parallel machine scheduling problem with uncertainties and sequence-dependent setup time," *Sci. Program.*, vol. 2016, pp. 1–13, Nov. 2016.

[37] H.-S. Choi, J.-S. Kim, and D.-H. Lee, "Real-time scheduling for reentrant hybrid flow shops: A decision tree based mechanism and its application to a TFT-LCD line," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3514–3521, Apr. 2011.

[38] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[39] C.-Y. Lee, S. Piramuthu, and Y.-K. Tsai, "Job shop scheduling with a genetic algorithm and machine learning," *Int. J. Prod. Res.*, vol. 35, no. 4, pp. 1171–1191, Apr. 1997.

[40] J. R. Quinlan, "Decision trees and decision-making," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 339–346, Mar. 1990.

[41] C. D. Geiger, R. Uzsoy, and H. Aytuğ, "Rapid modeling and discovery of priority dispatching rules: An autonomous learning approach," *J. Scheduling*, vol. 9, no. 1, pp. 7–34, Feb. 2006.

[42] D. Y. Sha and C.-H. Liu, "Using data mining for due date assignment in a dynamic job shop environment," *Int. J. Adv. Manuf. Technol.*, vol. 25, nos. 11–12, pp. 1164–1174, Jun. 2005.

[43] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," in *Annals of Discrete Mathematics*, vol. 5. Amsterdam, The Netherlands: Elsevier, 1979, pp. 287–326.

[44] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[45] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[46] T. K. Ho, "Nearest neighbors in random subspaces," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR), Struct. Syntactic Pattern Recognit. (SSPR)*, 1998, pp. 640–648.

[47] A. Pal, J. S. Thorp, T. Khan, and S. S. Young, "Classification trees for complex synchrophasor data," *Electr. Power Compon. Syst.*, vol. 41, no. 14, pp. 1381–1396, Oct. 2013.

[48] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[49] R. Su, X. Chen, S. Cao, and X. Zhang, "Random forest-based recognition of isolated sign language subwords using data from accelerometers and surface electromyographic sensors," *Sensors*, vol. 16, no. 1, p. 100, 2016.

[50] X. J. Chen, Z. G. Zhang, and Y. Tong, "An improved ID3 decision tree algorithm," *Adv. Mater. Res.*, vols. 962–965, pp. 2842–2847, Jun. 2014.

[51] M. van Diepen and P. H. Franses, "Evaluating chi-squared automatic interaction detection," *Inf. Syst.*, vol. 31, no. 8, pp. 814–831, Dec. 2006.

[52] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2014.

[53] L. Breiman, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.

[54] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty, "Optimal number of features as a function of sample size for various classification rules," *Bioinformatics*, vol. 21, no. 8, pp. 1509–1515, Apr. 2005.

[55] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.

[56] Y. Arakawa, K. Yasumoto, K. Pattamasiriwat, and T. Mizumoto, "Improving recognition accuracy for activities of daily living by adding time and area related features," in *Proc. 10th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU)*, Oct. 2017, pp. 1–6.

[57] P. Abreu, C. Soares, and J. M. S. Valente, "Selection of heuristics for the job-shop scheduling problem based on the prediction of gaps in machines," in *Proc. Int. Conf. Learn. Intell. Optim.*, 2009, pp. 134–147.

[58] K. Bala and A. Kumar, "A hybrid approach for load balancing: Using random forest and PSO approach (RFPSO)," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, pp. 1554–1559, 2017.

[59] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, 2009.

[60] M. Helal, G. Rabadi, and A. Al-Salem, "A tabu search algorithm to minimize the makespan for the unrelated parallel machines scheduling problem with setup times," *Int. J. Oper. Res.*, vol. 3, no. 3, pp. 182–192, 2006.

[61] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes Univ., Kayseri, Turkey, Tech. Rep. TECHNICAL REPORT-TR06, 2005.

[62] H. Fei, Q. Li, and D. Sun, "A survey of recent research on optimization models and algorithms for operations management from the process view," *Sci. Program.*, vol. 2017, pp. 1–19, Jul. 2017.

[63] G. Tian, Y. Ren, Y. Feng, M. Zhou, H. Zhang, and J. Tan, "Modeling and planning for dual-objective selective disassembly using AND/OR graph and discrete artificial bee colony," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2456–2468, Dec. 2018.

[64] A. Yurtkuran and E. Emel, "A discrete artificial bee colony algorithm for single machine scheduling problems," *Int. J. Prod. Res.*, vol. 54, no. 22, pp. 6860–6878, Nov. 2016.

[65] O. Bulut and M. F. Tasgetiren, "An artificial bee colony algorithm for the economic lot scheduling problem," *Int. J. Prod. Res.*, vol. 52, no. 4, pp. 1150–1170, Feb. 2014.

[66] G. Deng, H. Yang, and S. Zhang, "An enhanced discrete artificial bee colony algorithm to minimize the total flow time in permutation flow shop scheduling with limited buffers," *Math. Problems Eng.*, vol. 2016, pp. 1–11, Jul. 2016.

[67] S.-W. Lin, K.-C. Ying, and C.-Y. Huang, "Multiprocessor task scheduling in multistage hybrid flowshops: A hybrid artificial bee colony algorithm with bi-directional planning," *Comput. Oper. Res.*, vol. 40, no. 5, pp. 1186–1195, May 2013.

[68] S.-W. Lin and K.-C. Ying, "ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times," *Comput. Oper. Res.*, vol. 51, pp. 172–181, Nov. 2014.

[69] S. Sundar, P. N. Suganthan, C. T. Jin, C. T. Xiang, and C. C. Soon, "A hybrid artificial bee colony algorithm for the job-shop scheduling problem with no-wait constraint," *Soft Comput.*, vol. 21, no. 5, pp. 1193–1202, Mar. 2017.

[70] G. Gong, R. Chiong, Q. Deng, and X. Gong, "A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility," *Int. J. Prod. Res.*, vol. 6, pp. 1–15, Aug. 2019.
[71] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
[72] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, Jan. 2005.
[73] U. M. Fayyad and K. B. Irani, "What should be minimized in a decision tree?" in *Proc. AAAI*, vol. 90, 1990, pp. 749–754.
[74] T. Altan, G. Ngaile, and G. Shen, *Cold and Hot Forging: Fundamentals and Applications*, vol. 1. Cleveland, OH, USA: ASM International, 2004.

CHEN-YANG CHENG received the Ph.D. degree in industrial and manufacturing engineering from Penn State University. He is currently a Professor with the Department of Industrial Engineering and Management, National Taipei University of Technology. His research interests include computer integrated manufacturing, human–computer interaction, distributed systems and control, and intelligent systems.

POURYA POURHEJAZY received the master's degree in industrial engineering from University Technology Malaysia, in 2014, and the Ph.D. degree in logistics engineering from INHA University, South Korea, in 2017. He is currently an Adjunct Assistant Professor and a Postdoctoral Fellow with the National Taipei University of Technology, Taiwan. His research interests include optimization and decision analysis in supply chain, production, and transportation contexts.

KUO-CHING YING is currently a Distinguished Professor of the Department of Industrial Engineering and Management, National Taipei University of Technology. He has published over 100 academic research articles in refereed international journals, such as *Applied Intelligence*, *Applied Soft Computing*, *Computers and Operations Research*, *Computers and Industrial Engineering*, the *European Journal of Industrial Engineering*, the *European Journal of Operational Research*, IEEE Access, the *International Journal of Production Economics*, the *International Journal of Advanced Manufacturing Technology*, the *International Journal of Innovational Computing, Information and Control*, the *International Journal of Production Research*, the *Journal of the Operational Research Society*, the *OMEGA-The International Journal of Management Sciences*, *Production Planning & Control*, *Transportation Research Part E: Logistics and Transport Review*, and among others. His research interests include operations scheduling and combinatorial optimization. He is the Senior Editor of ten international journals.

SHU-FEN LI received the Ph.D. degree in industrial engineering and enterprise information from Tunhai University, Taiwan, in 2017. She is currently an Assistant Professor with the Department of Industrial Engineering and Management, National Chin-Yi University of Technology. Her research interests include operations scheduling, computational intelligence, data mining, and lean production.

CHIEH-WEN CHANG received the master's degree in industrial engineering and management from the National Taipei University of Technology, Taiwan, in 2019. She is currently working as an Industrial Engineer with Wistron Corporation. Her research interest is in data mining on production scheduling.