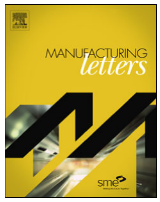
Available online at www.sciencedirect.com

Manufacturing Letters

Manufacturing Letters 35 (2023) 321–332



51st SME North American Manufacturing Research Conference (NAMRC 51, 2023)

Integration of discrete-event dynamics and machining dynamics for machine tool: modeling, analysis and algorithms☆

Mason Ma[a], Alisa Ren[a], Christopher Tyler[b], Jaydeep Karandikar[b], Michael Gomez[c], Tony Shi[a,*], Tony Schmitz[b,d]

[a]*Department of Industrial and Systems Engineering, University of Tennessee Knoxville, Knoxville, TN 37996, USA*
[b]*Manufacturing Science Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA*
[c]*MSC Industrial Supply Company, Knoxville, TN 37932, USA*
[d]*Department of Mechanical, Aerospace and Biomedical Engineering, University of Tennessee Knoxville, Knoxville, TN 37996, USA*

Abstract

Machining dynamics research lays a solid foundation for machining operations by providing stable combinations of spindle speed and depth of cut. Furthermore, machine learning has been applied to predict tool life as a function of cutting speed. However, the existing research does not consider the discrete-event dynamics in machine shop, i.e., the machine tool needs to process a series of parts in queue under various practical production requirements. This paper addresses the integration of discrete-event dynamics and machining dynamics to achieve cost savings in machining. A learning-based cost function is first proposed for the studied integrated optimization problem of machine tool. The proposed cost function utilizes the predicted tool life under different stable cutting speeds for further optimizing speed selection of machine tool to deal with the discrete-event dynamics in machine shop. Then, according to the practical production requirements, effective mathematical optimization models are developed for the related integrated optimization problems with the consideration of cost, makespan and due date, respectively. Numerical results show the effectiveness of our proposed methods and also the potential to be used in practice.

Keywords: Machine tool; machining dynamics; discrete-event dynamics; mathematical optimization; operational excellence

1. Introduction

The daily operations of machine tools aim to produce products to fulfill orders with various production requirements at the minimum cost. This involves both the machining dynamics that govern the material cutting by the machine tool, and the discrete-event dynamics that govern the production of a series of parts queued at the machine tool. Studies in the two fields have independently made important contributions to the operational excellence of machine tools in production environments.

The research of machining dynamics has laid a solid foundation for the operational excellence of machine tool by providing a set of stable cutting parameters, including spindle speed. The related studies focus on the physical constraints and modeling of machining processes [1, 2], the stable and chatter machining parameters identification [3, 4], tool wear and tool life prediction [5, 6], cutting force modeling and measurement [7, 8] and etc. The machining of a single part is typically considered in these research studies. However, few research efforts have considered the practical production requirements for hundreds or

☆ This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (`http://energy.gov/downloads/doe-public-access-plan`).

* Corresponding author. Tel.: +1-865-974-7654.
E-mail address: tony.shi@utk.edu (Tony Shi).

Nomenclature

$\mathcal{J}$ $\mathcal{J} = \{1, \ldots, n\}$ is the set of n orders, indexed by j, k or l
$\mathcal{I}$ $\mathcal{I} = \{1, \ldots, m\}$ is the index set of m candidate speeds, indexed by i
l_j the quantity of parts for order j
v_j volume to be removed for each part in order j, mm^3
d_j the due date for order j, min
a radial depth of cut, mm
b axial depth of cut, mm
f_t feed per tooth, mm
N_t number of teeth
Ω spindle speed, rpm
V cutting speed, mm/min
D diameter of the tool edge, mm
t machining time for one part, min
T tool life, min
MRR material removal rate, mm^3/min
r_m machine tool operating cost, \$/min
t_{ch} time required for changing a tool edge, min
C_{te} cost per tool edge, \$
C_p total cost per part, \$
C_o total cost per order, \$
Ω_{ij} i-th spindle speed for processing order j, rpm
T_{ij} tool life when using speed i to process order j, min
p_{ij} total machining time of order j with speed i, min
q_{ij} number of tool edges needed for order j by speed i
h_{ij} total processing time of order j with speed i, min
$\mathbf{\Omega}$ spindle speed assignment matrix, $\mathbf{\Omega} = \{\Omega_{ij}\}_{m\times n}$
$\mathbf{T}$ tool life time matrix, $\mathbf{T} = \{T_{ij}\}_{m\times n}$
$\mathbf{P}$ total machining time matrix, $\mathbf{P} = \{p_{ij}\}_{m\times n}$
$\mathbf{Q}$ matrix for number of tool edges, $\mathbf{Q} = \{q_{ij}\}_{m\times n}$
$\mathbf{H}$ total processing time matrix, $\mathbf{H} = \{h_{ij}\}_{m\times n}$
x_{ij} binary variable, $x_{ij} = 1$ if speed i is chosen for processing order j; otherwise $x_{ij} = 0$
$f_{cost}(x)$ cost of all orders as a function of x, \$
c a given constant of the upper bound for the makespan, min
M M is a sufficiently large number
y_{jk} binary variable, $y_{jk} = 1$ if order j is processed before order k; otherwise $y_{jk} = 0$
C_j completion time of order j, min
z_{jk} binary variable, $z_{jk} = 1$ if order j is assigned on position k; otherwise $z_{jk} = 0$
C_k completion time of the order at position k, min

even thousands of parts from multiple accepted orders.

In the meantime, the discrete-event dynamics research for manufacturing typically considers the production scheduling of a series of parts that are in the queue of the machine tool. The operational excellence is usually achieved at the machine shop level by considering various production requirements, such as order scheduling on parallel machines [9, 10, 11], production scheduling on batch machines [12, 13], buffers for machines [14, 15], due date requirement [16] and etc. However, these studies do not consider the physical constraints and practical operating of the machine tools, such as the machining stability and tool life, which can significantly impact the accuracy of processing times. Most existing studies assume that the processing times are fixed and known. As such, the solutions from these studies are often impractical.

Therefore, the integration of machining dynamics and discrete-event dynamics is critical and promising. In the literature, a few studies have considered the production scheduling of machine tool, such as machine tool feed rate scheduling for processing parts with various geometry [17, 18], computer numerical control (CNC) machine scheduling with controllable processing times constrained by tool life [19, 20], tool replacement scheduling to balance cost of machining conditions, product quality and tool life [21, 22]. However, to our knowledge, no research to date addresses how to integrate machining dynamics, including the machining stability, various cutting speeds and tool life, into discrete-event dynamics for the operational excellence of machine tool. We seek to fill this gap.

In this paper, we address the problem of integration of discrete-event dynamics and machining dynamics, denoted by IDM, to increase the profit of machine shop. In industrial practice, cutting speed, feed, depth of cut along with cutting geometry and tool material system (substrate, coating, and type of coating method) make a difference in the cutting time, tool life, makespan and eventually the cost. As the first attempt for such integration, the scope of this paper is to select an appropriate cutting speed for each order to reduce the total cost. We analyze the problem under two scenarios of the practical production constraints with increasing complexities: 1) The integrated optimization of cost and makespan; 2) The integrated optimization of cost and due date. While only optimizing cost is the foundation, the above two scenarios increase the fidelity of our studied problem to practical production. We show that we can take advantage of high speed machining and proper production schedule to reduce the overall cost.

We organize the rest of the paper as follows. In Section 2, we present the detailed problem settings for both the discrete-event dynamics and the machining dynamics. A learning-based cost function for machine tool is given in Section 3. Different production scenarios are studied in Sections 4 and 5. The mathematical optimization models, analysis and algorithms are provided. In Section 6, we present numerical results for the proposed methods. Section 7 concludes this paper.

2. Problem Settings

In this section, we describe our problem settings of IDM in terms of orders and parts, stable spindle speed, processing time and tool life, and machine shop production and the cost.

2.1. Discrete-Event Dynamics: Order and Part Setting

We specify the production for n accumulated orders, each consisting of a number of parts. For example, a machine shop may receive more than 10 orders from the master production

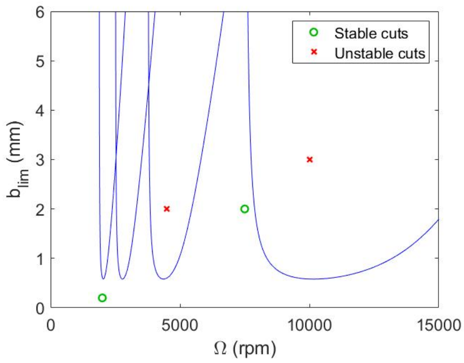


Fig. 1. Stability lobe diagram (SLD) to determine the stable and unstable cutting (chatter) by choosing parameters spindle speed (Ω) and axial depth of cut (b).

plan for daily operations. We address how to schedule the production of the machine tool by identifying the best speeds for each of these orders and scheduling their sequences in the queue of the machine tool to save total production cost.

Mathematically, we define the set of n orders as $\mathcal{J} = \{1, 2, \ldots, n\}$. Each order consists of a specific quantity of parts, denoted by l_j, $j \in \mathcal{J}$. In this paper, we assume all the parts of an order are the same type, but parts for different orders can be different. The different part types can be resulted by different situations, such as material type, part geometry, and etc. For the single type of part in an order j, let the volume to be removed for each part be v_j, such that the total volume to be removed (i.e., workload of order j) can be calculated by $v_j l_j$. All the orders are released at time zero. A due date d_j is associated for order j before which all parts of the order must be produced.

The challenges for such discrete-event dynamics research are that the processing times of orders are various due to different cutting speeds for different part types. To obtain the operational excellence for machine tool, the critical issue is to obtain the accurate estimation of processing time for orders.

2.2. Machining Dynamics: Spindle Speed and Tool Life Setting

Machining dynamics provides the foundation for accurate processing time estimation via the machining stability and tool life estimation under a wide variety of spindle speeds. Without loss of generality, we adopt the milling settings throughout this paper. The stability lobe diagram (SLD) enables to select the best stable spindle speeds at increased axial depth of cut, as shown in Fig. 1. The machining process can either be stable, that are the green circles below the stability boundary (blue curve), and chatter (red cross). In practice, for a given tool-material combination, the SLD can be obtained by conducting impact test. For more details, see the SLD and its governing relationship in [23]. In this paper, we assume the stable spindle speeds are selected from SLD generated by impact test.

Given a stable spindle speed Ω by SLD and a part with the volume to be removed as v, the machining time then can be calculated as

$$t = \frac{v}{MRR} = \frac{v}{abf_t N_t \Omega}, \tag{1}$$

where MRR is the material removal rate defined as the average volume of material removed per unit time with a chosen stable spindle speed Ω from the SLD, a and b the radial and axial depth of cut, and f_t is the feed per tooth. In this paper, we calculate t by considering various Ω with fixed a, b and f_t as the same setting in [6].

Notably, the tool life is decided directly given by the machining parameters, like the spindle speed Ω. In general, this is given by the Taylor's tool life equation [24]:

$$VT^N = C, \tag{2}$$

where the cutting speed V is the peripheral velocity by $V = \pi D\Omega$, D is the diameter of the tool, and T is the tool life. N and C are constants. Whilst higher spindle speed can increase the machining efficiency of jobs with less machining time t, it leads to reduction of tool life. Thus, there exists a balance of tool life and the machining time which will eventually impact the cost. To obtain the tool life model is usually by machine learning methods. In this paper, we use a state-of-the-art machine learning model in [6] to generate the tool life as input for our method, as discussed in Section 3.

2.3. Production and Cost Settings

For production setting, we specify there is a single machine tool. There are sufficient tool edges of the same type for processing all the n orders. In practice, it is often infeasible to complete a fractional number of a part for a tool edge as it requires to change the tool edge in the middle of the cut. As a result, when the rest of tool life is not long enough for machining a single part, the tool edge needs to be changed with a new one. The orders are produced sequentially in the queue of the machine tool. No preemption and interruption of orders are allowed. For all parts in one order, the chosen cutting speed is the same. For different orders, the cutting speeds can be different.

The objective is to minimize the overall cost introduced by machine tool operating cost for machining parts plus the tool edges. In [6], a cost model per part was provided as

$$C_p = r_m t + \frac{(r_m t_{ch} + C_{te})t}{T}, \tag{3}$$

where we denote r_m, t_{ch}, C_{te} and C_p as the machine tool operating cost in \$ per unit time, the time required for changing a tool edge, the cost in \$ per tool edge and the total cost per part, respectively. This model, however, considers the cost for a single part, rather than the overall cost of orders for hundreds or even thousands of parts made by different materials.

In this study, we extend the above model to represent the overall cost for producing multiple parts in multiple orders as

$$C_o = \sum_{j \in \mathcal{J}} [r_m p_j + (r_m t_{ch} + C_{te}) q_j], \tag{4}$$

where the first term $r_m p_j$ calculates the machine tool operating cost for processing order j and p_j refers to the total machining time with a chosen speed. The second term $(r_m t_{ch} + C_{te}) q_j$ calculates the cost induced by changing tool edges, and q_j is the number of tool edges needed for processing order j.

Eq. (4) is the generalized form of Eq. (3) to include all orders in the practical machine shop production. In next section, we will present a new cost function through the calculation of p_j and q_j under the context of multiple orders in machine shop.

3. Learning-based Cost Function for Machine Tool

The determination of cost of the orders depends on the machining time and number of tool edges used, which is directly decided by the prediction of tool life. In this section, we present a new cost function based on the tool life prediction in the sense of expectation with a state-of-the-art machine learning model.

3.1. A Machine Learning Model for Tool Life Prediction

Karandikar et al. (2021) [6] developed a physics-guided logistic regression model to predict tool worn probability with cutting speed V and cutting time T as inputs. Through introducing into machine learning Taylor's power law-based tool life description (see Eq. (2)), the model has a powerful performance in prediction accuracy. The logistic decision boundary of the model, that is, when the probability of tool worn is 0.5, provides an accurate estimation for tool life in the sense of expectation as

$$\log(T) = -\frac{\theta_1}{\theta_2} \log(V) - \frac{\theta_0}{\theta_2}, \tag{5}$$

where the coefficients $\theta_0 = -89.57$, $\theta_1 = 13.57$ and $\theta_2 = 5.26$ are estimated through experimental data for a single-insert endmill (Kennametal KICR073SD30333C) with $D = 19.05$ mm diameter and a square uncoated carbide insert (Kennametal 107,888,126 C9 JC). Equivalently, this model can be transformed into the below Taylor's tool life model after using logarithmic transformation,

$$T = \left(\frac{C}{V}\right)^{1/N} = \left(\frac{C}{\pi D \Omega}\right)^{1/N}, \tag{6}$$

where $N = 0.388$ and $C = 735.2$ m/min using the above values of θ_0, θ_1 and θ_2. The logistic model in the logarithmic and original space of V and T is shown in the two panels of Fig. 2. The gray scale indicates the probability of tool worn with a cutting speed V (horizontal axis) and cutting time T (vertical axis). We will use Eq. (6) throughout this paper for tool life prediction.

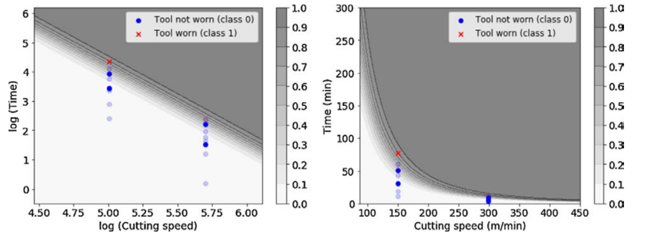


Fig. 2. Probability of tool worn as a function of cutting speed (logarithmic/original scale in left/right panel) from the trained logistic model; the train data points are shown in light blue, while the test data points in dark blue (tool not worn) and red cross (tool worn). The gray scale indicates the probability of tool worn with a cutting speed (horizontal axis) and cutting time (vertical axis). Good extrapolation outside of the train data space of cutting speed can be obtained, as shown by the dark blue dots and red cross. This extrapolation capability enables the accurate prediction of tool life for wide range of cutting speeds, which will be used in the experiments. Figures are reproduced from [6].

3.2. Calculation of Learning-based Cost Function

Based on the learning-based tool life model in [6], a cost function is developed for multiple orders. As the parts for different orders can be made by different materials, the SLD of the same tool edge for different orders (parts) are different. As a result, the candidate stable spindle speeds for orders are also different. Thus, we define the matrix of spindle speed as

$$\mathbf{\Omega} = \begin{bmatrix} \Omega_{11} & \Omega_{12} & \cdots & \Omega_{1n} \\ \Omega_{21} & \Omega_{22} & \cdots & \Omega_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \Omega_{m1} & \Omega_{m2} & \cdots & \Omega_{mn} \end{bmatrix}, \tag{7}$$

where the entry Ω_{ij} refers to the spindle speed for processing order j with spindle speed i, $i \in \mathcal{I}$, and $\mathcal{I} = \{1, 2, \ldots, m\}$ is defined as the index set of available stable spindle speed.

For a given spindle speed Ω_{ij} to process order j, the tool life T_{ij} is calculated by Eq. (6). Also, the total machining time p_{ij} for processing order j with spindle speed Ω_{ij} calculated as

$$p_{ij} = \frac{v_j l_j}{a b f_t N_t \Omega_{ij}}, \tag{8}$$

where $v_j l_j$ indicates total volume of order j by multiplying its part quantity l_j with the part volume v_j. When arranged T_{ij} and p_{ij} in matrix form, we can obtain the below matrices for tool life and machining time:

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1n} \\ T_{21} & T_{22} & \cdots & T_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ T_{m1} & T_{m2} & \cdots & T_{mn} \end{bmatrix} \text{ and } \mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix}, \tag{9}$$

Using $\mathbf{T}$, we can now compute the number of tool edges needed for each order with different spindle speeds as matrix

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} \end{bmatrix}, \tag{10}$$

where each entry is calculated by

$$q_{ij} = \left\lceil \frac{l_j}{\lfloor T_{ij}/t_{ij} \rfloor} \right\rceil, \quad \text{and} \quad t_{ij} = \frac{v_j}{abf_tN_t\Omega_{ij}}. \tag{11}$$

The term t_{ij} is the machining time for a single part of order j with spindle speed Ω_{ij}. Thus, with floor function, $\lfloor T_{ij}/t_{ij} \rfloor$ refers to the number of parts of order j that can be finished by a single tool edge. Finally, the number of tool edges for processing l_j parts can be obtained by the ceiling function.

We can further obtain the summation of total machining time and tool edge change time for each order under all its candidate spindle speeds. This summation, denoted by h_{ij}, represents the total processing time occupied by order j if with speed i:

$$h_{ij} = p_{ij} + t_{ch}q_{ij}. \tag{12}$$

The associated total time matrix is denoted by

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mn} \end{bmatrix}. \tag{13}$$

Finally, with matrices **P**, **Q** and **H**, we can obtain the cost function for processing all orders as

$$f_{cost}(x) = \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}} (r_m h_{ij} + C_{te}q_{ij})x_{ij}. \tag{14}$$

Note here we introduce a binary variable x_{ij} defined as

$$x_{ij} := \begin{cases} 1, & \text{if speed } i \text{ is chosen for processing order } j, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Eq. (14) integrates both the machining dynamics, that is, term $r_m h_{ij} + C_{te}q_{ij}$ based on the cost function for parts of orders, and the discrete-event dynamics, that is, discrete variable x_{ij} in Eq. (15) for choosing different speeds for different orders to achieve the reduction of production cost.

In next section, we will optimize the speed selection through controlling x_{ij} under the discrete-event dynamics environment of machine shop such that $f_{cost}(x)$ can be minimized for processing all orders. Before that, we show an illustrative example.

3.3. Illustration of Cost Saving by the Proposed Integration

As an illustrative example of our proposed integration, we show two scenarios in Fig. 3 for processing an order: with faster speed in (a) obtained by the expected cost method in [6], and slower speed in (b) by solving our developed M1 model in Section 4. The light grey period is for tool machining, the light green period is for changing the tool edge. The red dashed sections refer to the waste time that is not enough for machining a single part before its tool life comes to end, while the one in the last tool edge is wasted because all parts are done.

By choosing slower speed in (b), we process one more part per tool edge in (b) in comparison with (a), namely from 9 part per tool edge to 10. The last three parts on the 4th tool edge of (a) are thus moved back to be processed with the 3rd tool edge, saving the last tool edge and reducing the waste time for each tool edge in (a). Notably, the production cost is reduced from \$244.18 to \$223.14. The makespan is increased from 80.82 min to 81.09 min as a sacrifice of the reduced cost. Such trade-off between cost and makespan is meaningful in practical production. Note Figure 3 only gives the case when smaller speed will lead to cost reduction, it is more often and realistic to reduce cost by increasing the speed for high speed machining, and this exactly shows our motivation for this study.

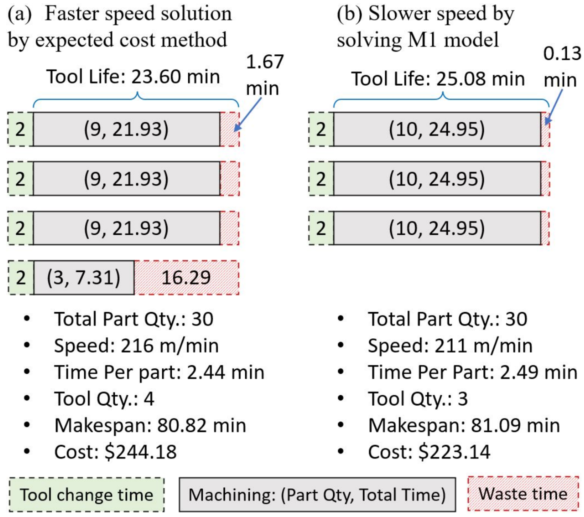


Fig. 3. Illustrative example for choosing different speeds to process multiple parts of an order by the expected cost method in [6] and solving our developed model M1 in Section 4. By choosing slower speed, we process one more part per tool edge in (b) in comparison with (a). The last three parts on the 4th tool edge of (a) are thus moved back to the 3rd tool edge. This leads to saving the last tool edge and reducing the waste time for each tool edge in (a), and importantly, reducing the cost from \$244.18 to \$223.14. See details in Section 6.

4. Integrated Optimization of Cost and Makespan

We denote our problem when only cost is considered as IDM-C, which is studied in Sections 4.1 - 4.2 . In section 4.3, we study the IDM problem when integrated optimization of cost and makespan is considered, denoted by IDM-CM. Integer optimization (IO) models are presented for both problems.

4.1. Model with Only Cost Consideration

When only cost is under consideration, the below IO model M1 aims to find the optimal spindle speed for each order, such that the cost function can be minimized.

$$\min \quad f_{cost}(x) \tag{M1}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}} x_{ij} = 1, \qquad \forall j \in \mathcal{J}, \tag{16}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}. \tag{17}$$

The objective is to minimize the overall cost $f_{cost}(x)$ of producing n orders. Constraints (16) ensure that each order must be processed by only one spindle speed. Constraints (17) provide the binary variable restriction on x_{ij}.

We show the constraint matrix of formulation M1 is totally unimodular. Thus, the IO model can be solved as a linear programming (LP) model. To do this, we arrange the variables in the sequence of $x_{11}, x_{21}, \ldots, x_{m1}; x_{12}, \ldots x_{m2}; \ldots; x_{1n}, \ldots, x_{mn}$ as columns of the constraint matrix. The constraint matrix $\mathbf{A} \in \mathbb{Z}^{n \times mn}$ for constraints (16) is as below:

$$\mathbf{A} = \begin{bmatrix} 1^{th} & \ldots & m^{th} & (m+1)^{th} & \ldots & 2m^{th} & \ldots & ((n-1)m+1)^{th} & \ldots & nm^{th} \\ 1 & \ldots & 1 & 0 & \ldots & 0 & \ldots & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1 & \ldots & 1 & \ldots & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 0 & \ldots & 1 & \ldots & 1 \end{bmatrix}. \tag{18}$$

The above 0, 1 matrix $\mathbf{A}$ is totally unimodular because it satisfies the following two conditions: (1) there are at most two nonzero elements in each column of $\mathbf{A}$, which is number 1 and there is only one entry for each column to be 1; (2) $\mathbf{A}$ admits an equitable row-bicoloring. In particular, the rows of $\mathbf{A}$ can be partitioned into two sets, say "red set" as row 1 and "blue set" as rows 2 to n. It's easy to check that the sum of the red rows minus the sum of the blue rows is a vector with entries ± 1.

As a result, the polyhedron for formulation M1 is integral, which has integral extreme points. Thus, it always admits an integral solution and is equivalent to solving the LP relaxation:

$$\begin{aligned} \min \quad & f_{cost}(x) && \text{(M1-LP)} \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} x_{ij} = 1, \quad \forall j \in \mathcal{J}, && (19) \\ & x_{ij} \in [0,1], \quad \forall i \in \mathcal{I}, j \in \mathcal{J}, && (20) \end{aligned}$$

where we relax the binary variable constraints $x_{ij} \in \{0,1\}$ into the interval $x_{ij} \in [0,1]$. Next, we develop an exact algorithm.

4.2. Decomposition-based Greedy Algorithm

Formulation M1 naturally adopts a decomposition for both the objective and the constraints for each order. While every order must be assigned with a spindle speed, there are no additional constraints that couples the assignment of spindle speed for different orders. Therefore, the assignment of spindle speed for each order is independent. That is, for each order j, we can select the optimal speed from m candidates that minimize its cost $r_m h_{ij} + C_{te} q_{ij}$ in the objective. In general, we denote the described procedure as Decomposition-based Greedy (DG) and summarize as the below Algorithm 1.

For Algorithm 1, we have the below lemma to show the optimality of its solution.

Lemma 4.1. *Given n orders and m candidate speeds, Algorithm 1 solves problem IDM-C optimally in* $O(18mn + n)$ *time.*

Proof. To prove DG algorithm is optimal, we do it by contradiction. Suppose the solution of DG is not optimal, then there must exist at least one order, denoted by $\hat{j}$, that selects a spindle speed with smaller cost $r_m h_{ij} + C_{te} q_{ij}$ than $\hat{j}$. This contradicts with steps 7-9 of DG, which selects the spindle speed with the smallest cost. Thus, Algorithm 1 is an exact algorithm.

Algorithm 1: The DG algorithm

Input : Data matrices $\mathbf{\Omega}$; order set $\mathcal{J}$ and associated l_j and v_j; the cost parameters r_m, t_{ch} and C_{te}.

1 Initialize $f_{cost}(x) = 0$; $x_{ij} = 0, \forall i \in \mathcal{I}, j \in \mathcal{J}$;
2 Calculate matrices $\mathbf{T}$, $\mathbf{Q}$, $\mathbf{P}$ and $\mathbf{H}$ with Eqs. (6) - (12);
3 **for** $j \in \mathcal{J}$ **do** `/* Decompose as n subproblems */`
4   **for** $i \in \mathcal{I}$ **do** `/* Find best speed for j */`
5   Compute $Obj_{ij} = r_m h_{ij} + C_{te} q_{ij}$;
6   **end**
7   Find the best speed i^* of job j by $i^* := \arg\min_{i \in \mathcal{I}} Obj_{ij}$;
8   Update $x_{i^* j} = 1$;
9   $f_{cost}(x) \leftarrow f_{cost}(x) + Obj_{i^* j}$
10 **end**
11 **Output** Optimal cost $f_{cost}(x)$, speed $x_{ij}, \forall i \in \mathcal{I}, j \in \mathcal{J}$

Next, we consider the solution time of Algorithm 1. For preparing $\mathbf{T}$, $\mathbf{Q}$, $\mathbf{P}$ and $\mathbf{H}$, it takes 2 operations to calculate T_{ij}, 6 operations to calculate q_{ij}, 3 operations to calculate p_{ij} and 2 operations to calculate h_{ij}. Together with the coupled loops, this takes $14mn$ times. For each j, it takes $3m$ operations to compute Obj_{ij} for all $i \in \mathcal{I}$, m operations to find the speed i^* with minimum Obj_{ij}, and 1 operation to update the overall cost $f_{cost}(x)$, respectively. To sum up, we can get the total time complexity of Algorithm 1 is $O(18mn + n)$, which completes the proof. □

4.3. Integration with Makespan

Makespan constraint refers to the hard constraint that all the orders must be completed within a given time period. For example, this can describe the fixed time length for the worker shift. To model problem IDM-CM, we need an additional parameter:

c : a given constant of the upper bound for the makespan.

With this additional parameter, we can model problem IDM-CM by modifying formulation M1 with makespan constraint as below IO model, denoted by M2:

$$\begin{aligned} \min \quad & f_{cost}(x) && \text{(M2)} \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} x_{ij} = 1, \quad \forall j \in \mathcal{J}, && (21) \\ & \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}} h_{ij} x_{ij} \leq c, && (22) \\ & x_{ij} \in \{0,1\}, \quad \forall i \in \mathcal{I}, j \in \mathcal{J}. && (23) \end{aligned}$$

Constraint (22) ensures the makespan of producing all the orders not exceeding the given time length c. The makespan constraint adds additional complexity for the problem. We remark the complexity of problem IDM-CM is still open.

5. Integrated Optimization of Cost and Due Date

We consider problem IDM when the integrated optimization of cost and due date is to be achieved, denoted by IDM-CD. Due to the due date constraints, the sequences of order processing need to be considered to determine the cost. We propose three categories of model based on the choices of decision variables which determine the order sequences. That are, completion time variables (5.1), linear order variables (5.2) and positional variables (5.3) of orders, respectively. In the first category, we propose a mixed-integer optimization (MIO) model that can be efficiently solved by off-the-shelf optimization solvers. The latter two categories of model involve quadratic constraints. Considering that off-the-shelf optimization solvers cannot efficiently deal with this kind of nonlinear problem, we adopt linearization strategies for the proposed formulations.

5.1. Modeling with Completion Time Variables

In the first model, we use completion time variables to indicate the order sequence for problem IDM-CD. The needed additional variables are defined as follows:

y_{jk} : $y_{jk} = 1$ if order j is processed before order k, $\forall j, k \in \mathcal{J}$ and $j \neq k$. Otherwise $y_{jk} = 0$;

C_j : completion time of order j, $\forall j \in \mathcal{J}$.

The problem IDM-CD can be formulated as below.

$$
\begin{aligned}
\min\quad & f_{cost}(x) && && \text{(M3-1)}\\
\text{s.t.}\quad & \sum_{i\in\mathcal{I}} x_{ij} = 1, && \forall j \in \mathcal{J}, && (24)\\
& C_j \geq \sum_{i\in\mathcal{I}} h_{ij}x_{ij}, && \forall j \in \mathcal{J}, && (25)\\
& C_j + \sum_{i\in\mathcal{I}} h_{ik}x_{ik} \leq C_k + M(1-y_{jk}), && \forall j < k \in \mathcal{J}, && (26)\\
& C_k + \sum_{i\in\mathcal{I}} h_{ij}x_{ij} \leq C_j + My_{jk}, && \forall j < k \in \mathcal{J}, && (27)\\
& C_j \in [0, d_j], && \forall j \in \mathcal{J}, && (28)\\
& x_{ij}, y_{jk} \in \{0,1\}, && \forall i \in \mathcal{I}, j,k \in \mathcal{J}. && (29)
\end{aligned}
$$

Constraints (25) ensure that the completion time of each order is no later than its processing time. Constraints (26) and (27) are disjunctive constraints which ensure that either order j is processed before order k, i.e. $y_{jk} = 1$ in constraints (26), or order k is before order j, that is $y_{jk} = 0$ in constraints (27). M is a sufficiently large number. Constraints (28) enforce the completion time is non-negative and no larger than the due date. In this formulation, the M is a natural upper bound for C_j for arbitrary $j \in \mathcal{J}$. Thus, we can set $M = \sum_{j\in\mathcal{J}} \max_{i\in\mathcal{I}}\{h_{ij}\}$. The total number of variables for formulation M3-1 is $n^2 + mn + n$, which includes $n^2 + mn$ binary variables and $n^2 + 3n$ constraints.

Remark 1. *If $d_j = c$ for $j \in \mathcal{J}$, problem IDM-CD is reduced to problem IDM-CM.*

To check the above remark is valid, it's obvious to see that when $d_j = c$ for $j \in \mathcal{J}$, then the makespan of any feasible solutions to problem IDM-CD also satisfy the makespan limit c. Thus, problem IDM-CM is a special case of problem IDM-CD. It is more general for us to study problem IDM-CD.

5.2. Modeling with Linear Ordering Variables

5.2.1. Proposed Integer Optimization Model

This formulation is based on the linear order variables y_{jk} and y_{kj} that either of them should be equal to 1 for arbitrary $j < k \in \mathcal{J}$. The following formulation M3-2 is considered.

$$
\begin{aligned}
\min\quad & f_{cost}(x) && && \text{(M3-2)}\\
\text{s.t.}\quad & \sum_{i\in\mathcal{I}} x_{ij} = 1, && \forall j \in \mathcal{J}, && (30)\\
& y_{jk} + y_{kj} = 1, && \forall j,k \in \mathcal{J}, j < k, && (31)\\
& y_{jk} + y_{kl} + y_{lj} \leq 2, && \forall j,k,l \in \mathcal{J}, j \neq k \neq l, && (32)\\
& \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} h_{ij}x_{ij}y_{jk} + \sum_{i\in\mathcal{I}} h_{ik}x_{ik} \leq d_k, && \forall k \in \mathcal{J}, && (33)\\
& x_{ij}, y_{jk} \in \{0,1\}, && \forall i \in \mathcal{I}, j,k \in \mathcal{J}. && (34)
\end{aligned}
$$

Constraints (31) are a set of conflict constraints, which ensure that either order j is processed before order k or vice versa. Note here $j < k$ because of symmetry. Constraints (32) indicate the transitivity constraints that ensure a linear order of three arbitrary orders. Constraints (33) indicate the starting time of order k plus its processing time must be no larger than its due date. The total number of variables of formulation M3-2 with m speeds and n orders is $n^2 + mn$, which are all binary variables and involve $n^3 - 2.5n^2 + 3.5n$ constraints.

Note that M3-2 is a quadratically constrained IO formulation due to constraints (33). Considering that the current optimization solvers like CPLEX cannot efficiently deal with this kind of nonlinear problem, we adopt linearization strategies for this formulation in the following section.

5.2.2. Reformulation with Linearization Technique

To enhance the solvability of formulation M3-2, we first apply the linearization technique proposed in [25] to derive an MIO model, which is equivalent to M3-2. For arbitrary $j \in \mathcal{J}$, $k \in \mathcal{J}$, let $\alpha_{jk} = (\sum_{i\in\mathcal{I}} h_{ij}x_{ij})y_{jk}$ and substitute the cross-product $\sum_{i\in\mathcal{I}} h_{ij}x_{ij}y_{jk}$ in M3-2 using the following inequalities:

$$
\begin{cases}
\sum_{i\in\mathcal{I}} h_{ij}x_{ij} - M(1-y_{jk}) \leq \alpha_{jk} \leq \sum_{i\in\mathcal{I}} h_{ij}x_{ij} + M(1-y_{jk}),\\
-My_{jk} \leq \alpha_{jk} \leq My_{jk},
\end{cases}
\tag{35}
$$

where M is a sufficiently large number. For arbitrary $j, k \in \mathcal{J}$, if $y_{jk} = 1$, then $\alpha_{jk} = \sum_{i\in\mathcal{I}} h_{ij}x_{ij}$, and the inequalities become $\sum_{i\in\mathcal{I}} h_{ij}x_{ij} \leq \alpha_{jk} \leq \sum_{i\in\mathcal{I}} h_{ij}x_{ij}$ and $-M \leq \alpha_{jk} \leq M$. Clearly, the equivalence holds. Otherwise, if $y_{jk} = 0$, the inequalities are

$\sum_{i\in\mathcal{I}} h_{ij}x_{ij} - M \le \alpha_{jk} \le \sum_{i\in\mathcal{I}} h_{ij}x_{ij} + M$ and $0 \le \alpha_{jk} \le 0$, which indicates $\alpha_{jk} = 0$. Thus, we get an equivalent model to M3-2.

We further reduce the above inequality constraints (35) as

$$\begin{cases} \sum_{i\in\mathcal{I}} h_{ij}x_{ij} - M(1-y_{jk}) \le \alpha_{jk}, \\ 0 \le \alpha_{jk} \le My_{jk}. \end{cases} \tag{36}$$

Thus, we obtain a new linearized formulation M3-2-L as below:

$$\begin{aligned}
\min \quad & f_{cost}(x) && \text{(M3-2-L)}\\
\text{s.t.} \quad & \text{Constraints (30) – (32)}, && (37)\\
& \sum_{j\in\mathcal{J}} \alpha_{jk} + \sum_{i\in\mathcal{I}} h_{ik}x_{ik} \le d_k, \quad \forall k \in \mathcal{J}, && (38)\\
& \sum_{i\in\mathcal{I}} h_{ij}x_{ij} - M(1-y_{jk}) \le \alpha_{jk} \le My_{jk}, \quad \forall j,k \in \mathcal{J}, && (39)\\
& \alpha_{jk} \ge 0, \quad \forall j,k \in \mathcal{J}, && (40)\\
& x_{ij}, y_{jk} \in \{0,1\}, \quad \forall i \in \mathcal{I}, j,k \in \mathcal{J}. && (41)
\end{aligned}$$

This formulation is obtained by eliminating the right-hand-side of the first inequality and the left-hand-side of the second inequality in (35). The value of M is set as the maximum total processing time of the orders, that is $M = \max_{i,j}\{h_{ij}\}$. The total number of variables is $2n^2 + mn$, which involves $n^2 + mn$ binary variables and $n^3 + 0.5n^2 + 3.5n$ constraints. For formulations M3-2-L and M3-2, we have the below Lemma 5.1.

Lemma 5.1. *For arbitrary optimal solution to M3-2-L, there exists an optimal solution to M3-2 with the same objective value.*

Proof. For arbitrary optimal solution (x^*, y^*, α^*) to M3-2-L, we can show (x^*, y^*) as an optimal solution to M3-2 with the same objective value.

We first show (x^*, y^*) is a feasible solution to M3-2. For arbitrary $j,k \in \mathcal{J}$, if $y^*_{jk} = 0$, then $\alpha^*_{jk} = 0$ based on constraints (39) and (40). Then we have $\sum_{i\in\mathcal{I}} h_{ik}x^*_{ik} \le d_k$ in constraints (38). This indicates (x^*, y^*) satisfies constraints (33) in formulation M3-2 when $y^*_{jk} = 0$. If $y^*_{jk} = 1$, then by substituting into constraints (39) first and then (38), we have

$$\sum_{i\in\mathcal{I}} h_{ij}x^*_{ij} \le \alpha^*_{jk} \le M, \tag{42}$$

$$\sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} h_{ij}x^*_{ij} + \sum_{i\in\mathcal{I}} h_{ik}x^*_{ik} \le \sum_{j\in\mathcal{J}} \alpha^*_{jk} + \sum_{i\in\mathcal{I}} h_{ik}x^*_{ik} \le d_k. \tag{43}$$

Inequalities (43) indicate (x^*, y^*) satisfies constraints (33) when $y^*_{jk} = 1$. To sum up, (x^*, y^*) is a feasible solution to M3-2.

Next, we prove the objective value of (x^*, y^*) is less than or equal to that of any feasible solutions to M3-2. Let $\mathcal{S}_2$ be the solution space of formulation M3-2, i.e.,

$$\mathcal{S}_2 := \{(x,y) \ : \ \text{Constraints (30) - (34)}\}. \tag{44}$$

As M3-2-L is a relaxation of M3-2, the objective value of the optimal solution (x^*, y^*, α^*) will be less than or equal to that of any feasible solutions to M3-2. That is

$$f_{cost}(x^*) = \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} (r_m h_{ij} + C_{te}q_{ij})x^*_{ij} \le f_{cost}(x), \ \forall (x,y) \in \mathcal{S}_2. \tag{45}$$

Note $f_{cost}(x^*)$ is also the objective value of feasible solution (x^*, y^*) to M3-2. This indicates the objective value of (x^*, y^*) is no larger than any feasible solutions to M3-2.

Therefore, (x^*, y^*) is an optimal solution to M3-2 with the same objective value $f_{cost}(x^*)$, which completes the proof. □

5.3. Modeling with Positional Variables

5.3.1. Proposed Integer Optimization Model

This model is based on the positional variables that describe the relationship between arbitrary order with the position it is assigned in the schedule. The additional decision variables are defined as follows.

z_{jk} : $z_{jk} = 1$ if order j is assigned on position k, $\forall j,k \in \mathcal{J}$. Otherwise $z_{jk} = 0$;

C_k : completion time of the order at position k, $\forall k \in \mathcal{J}$.

A new IO model can be formulated as below.

$$\begin{aligned}
\min \quad & f_{cost}(x) && \text{(M3-3)}\\
\text{s.t.} \quad & \sum_{i\in\mathcal{I}} x_{ij} = 1, \quad \forall j \in \mathcal{J}, && (46)\\
& \sum_{k\in\mathcal{J}} z_{jk} = 1, \quad \forall j \in \mathcal{J}, && (47)\\
& \sum_{j\in\mathcal{J}} z_{jk} = 1, \quad \forall k \in \mathcal{J}, && (48)\\
& 0 \le C_k \le \sum_{j\in\mathcal{J}} d_j z_{jk}, \quad \forall k \in \mathcal{J}, && (49)\\
& C_1 \ge \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} h_{ij}x_{ij}z_{j1}, && (50)\\
& C_k \ge C_{k-1} + \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} h_{ij}x_{ij}z_{jk}, \quad \forall k \in \mathcal{J}/\{1\}, && (51)\\
& x_{ij}, z_{jk} \in \{0,1\}, \quad \forall i \in \mathcal{I}, j,k \in \mathcal{J}. && (52)
\end{aligned}$$

Constraints (47) and (48) ensure that a particular order can only be assigned to exactly one position and each position can be assigned with exactly one order. Constraints (49) ensure the completion time of order at position k is non-negative and no later than its due time. Constraints (50) and (51) describe the completion time of the order at position k. The total number of variables in this formulation is $n^2 + mn + n$, which involves $n^2 + mn$ binary variables and $6n$ constraints. Note that M3-3 is a quadratically constrained IO formulation. We will also apply linearization technique to reformulate M3-3.

5.3.2. Reformulation with Linearization Technique

For arbitrary $j,\ k \in \mathcal{J}$, define a new variable $\beta_{jk} = (\sum_{i\in\mathcal{I}} h_{ij}x_{ij})z_{jk}$. An equivalent MIO model can be obtained by substituting $\sum_{i\in\mathcal{I}} h_{ij}x_{ij}z_{jk}$ in M3-3 and with the below additional constraints

$$\begin{cases}\sum_{i\in\mathcal{I}} h_{ij}x_{ij} - M(1-z_{jk}) \le \beta_{jk} \le \sum_{i\in\mathcal{I}} h_{ij}x_{ij} + M(1-z_{jk}),\\ -Mz_{jk} \le \beta_{jk} \le Mz_{jk}.\end{cases} \tag{53}$$

Note the equivalence validation of the above inequalities is the same with (35). For concise presentation we skip it.

We further reduce the above inequality constraints (53) as

$$\begin{cases}\sum_{i\in\mathcal{I}} h_{ij}x_{ij} - M(1-z_{jk}) \le \beta_{jk},\\ 0 \le \beta_{jk} \le Mz_{jk}.\end{cases} \tag{54}$$

Thus, a new linearized formulation M3-3-L can be obtained as:

$$\begin{aligned}
\min \quad & f_{cost}(x) && \text{(M3-3-L)}\\
\text{s.t.} \quad & \text{Constraints (46)} - (49), && (55)\\
& C_1 \ge \sum_{j\in\mathcal{J}} \beta_{j1}, && (56)\\
& C_k \ge C_{k-1} + \sum_{j\in\mathcal{J}} \beta_{jk}, \quad \forall k \in \mathcal{J}/\{1\}, && (57)\\
& \sum_{i\in\mathcal{I}} h_{ij}x_{ij} - M(1-z_{jk}) \le \beta_{jk} \le Mz_{jk}, \quad \forall j,k \in \mathcal{J}, && (58)\\
& \beta_{jk} \ge 0, \quad \forall j,k \in \mathcal{J}, && (59)\\
& x_{ij}, z_{jk} \in \{0,1\}, \quad \forall i \in \mathcal{I}, j,k \in \mathcal{J}. && (60)
\end{aligned}$$

The value of M is set as the maximum total processing time among all the orders, that is $M = \max_{i,j}\{h_{ij}\}$. The total number of variables is $2n^2 + mn + n$, including $n^2 + mn$ binary variables and $2n^2 + 6n$ constraints. For formulations M3-3 and M3-3-L, we have the below Lemma 5.2.

Lemma 5.2. *For arbitrary optimal solution to M3-3-L, there exists an optimal solution to M3-3 with the same objective value.*

Proof. The proof idea is similar to Lemma 5.1. For arbitrary optimal solution (x^*, z^*, C^*, β^*) to M3-3-L, we will construct (x^*, z^*, C^*) as an optimal solution to M3-3 with the same objective value.

We first show (x^*, z^*, C^*) is a feasible solution to M3-3. It is simple to check (x^*, z^*, C^*) is feasible when $z^*_{jk} = 0$ by substituting $z^*_{jk} = 0$ into (58) and then (56) and (57). Otherwise, when substituting $z^*_{jk} = 1$ we have

$$\sum_{i\in\mathcal{I}} h_{ij}x^*_{ij} \le \beta^*_{jk} \le M, \quad \forall j,k \in \mathcal{J}, \tag{61}$$

$$C^*_1 \ge \sum_{j\in\mathcal{J}} \beta^*_{j1} \ge \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} h_{ij}x^*_{ij}, \tag{62}$$

$$C^*_k \ge C^*_{k-1} + \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}} h_{ij}x^*_{ij}, \ \forall k \in \mathcal{J}/\{1\} \tag{63}$$

Inequalities (62) and (63) indicate (x^*, z^*, C^*) is a feasible solution when $z^*_{jk} = 1$.

Next, similar to Lemma 5.1, we define the solution space of formulation M3-3 as $\mathcal{S}_3 := \{(x, z, C) : \text{Constraints (46) - (52)}\}$, and we can get that $f_{cost}(x^*) \le f_{cost}(x), \forall (x, z, C) \in \mathcal{S}_3$ as M3-3-L is a relaxation of M3-3. Thus, we conclude (x^*, z^*, C^*) is an optimal solution to M3-3. □

Based on Lemmas 5.1 and 5.2, we will directly solve models M3-2-L and M3-3-L using MIO solver instead of M3-2 and M3-3, respectively. The comparison of the three MIO formulations is discussed in next section.

6. Computational Experiments

Comprehensive computational experiments are conducted for different problem settings. For problem IDM-C, we compare the performance of methods by solving models M1 and M1-LP with modern optimization solver `CPLEX`, DG algorithm and the expected cost method (ECM henceforth) from [6]. For problem IDM-CM, we compare methods by solving M2 using `CPLEX` with ECM. The three MIO models for problem IDM-CD, i.e., M3-1, M3-2-L and M3-3-L, are solved by `CPLEX` and compared in terms of their computational efficiency.

All the methods are coded in Python. The optimization solver `CPLEX 20.1.0` is used for solving all the IO and MIO models with the Python interface `docplex`. All the settings for `CPLEX` remain default except for time limit as 600 seconds. The experiments are deployed on a mobile workstation with Intel(R) Xeon(R) W-10885M CPU @ 2.40GHz, 128 GB memory, 64 bit Windows 10 Pro operating system for workstations.

6.1. Data Generation

Machine tool. The settings for machine tool including the tool geometry, cutting parameters (except for spindle speed set) and tool life model are the same with that in [6]. In particular, the tool is a single-insert endmill with diameter $D = 19.05$ mm and $N_t = 1$. The radial depth of cut is $a = 4.7$ mm, the axial depth of cut is $b = 3$ mm and the feed per tooth is $f_t = 0.06$ mm, respectively. The constants in the Taylor tool life equation are $N = 0.388$ and $C = 735.2$ m/min.

Order and part. The number of orders n is from a set $\{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$, with one single part type for each order. The quantity l_j of the part for one order is an integer number randomly sampled from uniform distribution $U[10, 200]$. The volume v_j to be removed per part is an integer number randomly sampled from uniform distribution $U[5000, 15000]$ with unit mm^3. The makespan limitation c under each case of n is decided by considering two values: the makespan of the solution by solving problem IDM-C which only considers the cost, and the minimum makespan when adopting the minimum h_{ij} for each order j. We set c as the integer number by rounding the average of these two values.

Due date. We specify the due date by generating the data without regarding for feasibility and then discarding instances that have no feasible schedule. Specifically, we first obtain the

Table 1. Numerical results of solving problem IDM-C by the expected cost method (ECM) [6], solving models M1 and M1-LP, and the exact algorithm DG.

Order #	ECM		M1			M1-LP			DG		
	Obj: Cost	Time (s)	Obj: Cost	CRR (%)	Time (s)	Obj: Cost	CRR (%)	Time (s)	Obj: Cost	CRR (%)	Time (s)
2	2082.55	2.81	**1977.23**	5.06	0.01	**1977.23**	5.06	0.01	**1977.23**	5.06	**0.000**
4	3301.91	5.58	**3174.74**	3.85	0.01	**3174.74**	3.85	0.02	**3174.74**	3.85	**0.000**
6	5418.85	8.28	**5184.44**	4.33	0.01	**5184.44**	4.33	0.03	**5184.44**	4.33	**0.000**
8	6670.49	11.01	**6406.96**	3.95	0.01	**6406.96**	3.95	0.03	**6406.96**	3.95	**0.003**
10	9688.88	13.87	**9314.33**	3.87	0.01	**9314.33**	3.87	0.03	**9314.33**	3.87	**0.003**
12	11872.06	16.66	**11363.36**	4.28	0.01	**11363.36**	4.28	0.05	**11363.36**	4.28	**0.003**
14	13597.67	19.27	**13061.32**	3.94	0.02	**13061.32**	3.94	0.05	**13061.32**	3.94	**0.003**
16	15417.88	21.77	**14768.81**	4.21	0.02	**14768.81**	4.21	0.06	**14768.81**	4.21	**0.005**
18	16849.63	24.57	**16127.64**	4.28	0.02	**16127.64**	4.28	0.06	**16127.64**	4.28	**0.003**
20	20654.23	27.29	**19883.29**	3.73	0.02	**19883.29**	3.73	0.08	**19883.29**	3.73	**0.003**

* All values in all columns are the average results of 10 independent runs for each order number.
* All the 10 independent instances for all the order numbers are solved to optimality with Gap 0.00% by M1, M1-LP and DG (shown by bold numbers).

expected cutting speed for each order using the ECM in [6]. Henceforth, the makespan $\bar{C}_{max}$ and the mean value of total processing time $\bar{h}$ among the n orders can be obtained. The initial due date d_j is generated by randomly sampling from uniform distribution $U[0.4\bar{C}_{max}, 0.5\bar{C}_{max}]$. A schedule is then generated using earliest due date rule based on d_j, and the completion time C_j for each order is thus obtained. We then check the feasibility of d_j, i.e., d_j is accepted if $d_j \geq C_j$; otherwise $d_j = C_j + r$, where r is a randomly sampled integer number from uniform distribution $U[0, \bar{h}]$. By this way of setting d_j, we ensure at least the expected speeds are feasible solution to problem IDM-CD.

Production and cost. The candidate cutting speed V is evenly spaced in $[220, 420]$ m/min with the interval 2 m/min. Thus, in total $m = 100$ speeds are used for one order. Note here without loss of generality, we use the same set of cutting speed for all the n orders. The spindle speed matrix $\mathbf{\Omega}$ is thus constructed, with the equivalent minimum spindle speed 3676.02 rpm and the maximum spindle speed 6984.44 rpm. Lastly, the coefficients of the cost model are set as $r_m = 2$ \$/min, $t_{ch} = 2$ min and $C_{te} = \$12.5$ per tool edge.

For each order number, 10 independent data instances using different random seeds are generated with the above data generation procedure. Therefore, in total 100 instances are used. All the reported results are the average values obtained by the 10 independent instances for each order number.

6.2. Results for Problem IDM-C

Table 1 presents the results for solving problem IDM-C by the ECM in [6], solving models M1 and M1-LP, and the DG algorithm 1. The columns show the objectives (minimum cost in \$ by each method) and the solution time in second. In the column "CRR", we calculate the cost reduction ratio (CRR) as

$$CRR = \frac{Obj_{ECM} - Obj}{Obj_{ECM}} \times 100\%, \qquad (64)$$

where Obj_{ECM} and Obj, respectively, refer to the objective value from ECM and other methods under comparison.

As shown in Table 1, solving M1 and M1-LP by `CPLEX`, and the DG algorithm all find the optimal solution for all the 100 instances within 0.1 second, shown as the bold numbers. Remarkably, the DG algorithm preforms as the fastest among the three, within 0.005 second for all the 100 instances. In comparison with the ECM, formulations M1 and M1-LP and DG algorithm show that, by selecting different cutting speeds, the overall production cost can be reduced by 3% - 5% on average, as shown by the highlighted columns in gray in Table 1. Figure 4 shows results of CRR in Table 1. Note the CRR for formulations M1, M1-LP and DG are the same and thus are overlapped.

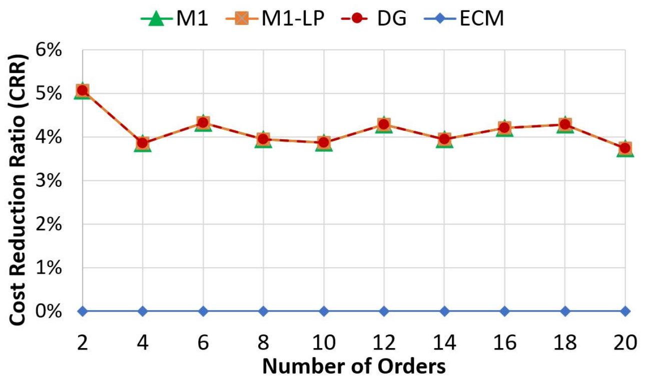


Fig. 4. The comparison of cost reduction ratio (CRR) for problem IDM-C. M1, M1-LP and DG are overlapped as they all obtain the optimal solutions.

Essentially, the cause of 3-5% CRR comes from the rationale of our integration idea. That is, the ECM from machining dynamics view considers only one part to choose the best speed. Thus, when multiple parts of an order are considered from the discrete event dynamics view, this speed might not be optimal since it might lead to larger waste at the rear of the tool life, then larger tool edge quantities and higher cost. Instead, our methods choose the best speed with a balance of the number of tool edges and the tool life, thus could avoid such waste of tool life and save the cost (see illustrative example in Figure 3).

6.3. Results for Problem IDM-CM

Table 2 shows the results for solving problem IDM-CM. In the column "Makespan Limit: c", we show the average makespan limit within which all parts must be finished. Formulation M2 is solved to optimality (with the gap 0.00) for all the 10 instances under each order number within 0.11 second,

Table 2. Numerical results of solving problem IDM-CM by expected cost method (ECM) [6] and solving model M2.

Order #	Makespan Limit: c	ECM			M2					
		Obj: Cost	Time (s)	Makespan	Obj: Cost	Num	Gap (%)	LP Relaxation	Time (s)	Makespan
2	633	2082.55	2.81	763.15	**2052.42**	**10**	0.00	2034.58	0.00	624.96
4	1027	3301.91	5.58	1209.71	**3245.44**	**10**	0.00	3233.46	0.05	1021.47
6	1678	5418.85	8.28	2000.68	**5281.53**	**10**	0.00	5271.44	0.08	1672.02
8	2070	6670.49	11.01	2449.62	**6532.52**	**10**	0.00	6526.56	0.08	2067.51
10	3007	9688.88	13.87	3580.07	**9512.87**	**10**	0.00	9502.75	0.11	3003.93
12	3669	11872.06	16.66	4362.28	**11571.73**	**10**	0.00	11566.20	0.08	3667.11
14	4198	13597.67	19.27	5013.83	**13364.26**	**10**	0.00	13354.87	0.05	4195.26
16	4775	15417.88	21.77	5687.06	**15047.90**	**10**	0.00	15042.45	0.08	4772.70
18	5202	16849.63	24.57	6221.69	**16426.20**	**10**	0.00	16421.71	0.05	5199.98
20	6413	20654.23	27.29	7656.49	**20286.36**	**10**	0.00	20281.15	0.08	6411.31

* All values in all columns except for “Num” are the average results of 10 independent runs for each order number.
* Column “Num” indicates the number of instances solved to optimality among the 10 instances.

Table 3. Numerical results of problem IDM-CD by solving models M3-1, M3-2-L and M3-3-L.

Order #	M3-1				M3-2-L				M3-3-L			
	Obj: Cost	Num	Gap (%)	Time (s)	Obj: Cost	Num	Gap (%)	Time (s)	Obj: Cost	Num	Gap (%)	Time (s)
2	**1920.68**	**10**	0.00	0.02	**1920.68**	**10**	0.00	0.02	**1920.68**	**10**	0.00	0.03
4	**4463.55**	**10**	0.00	0.08	**4463.55**	**10**	0.00	0.05	**4463.55**	**10**	0.00	0.07
6	**5412.14**	**10**	0.00	0.25	**5412.14**	**10**	0.00	0.07	**5412.14**	**10**	0.00	0.14
8	**7293.43**	**10**	0.00	0.27	**7293.43**	**10**	0.00	0.10	**7293.43**	**10**	0.00	1.83
10	**11129.40**	**10**	0.00	1.02	**11129.40**	**10**	0.00	0.39	**11129.40**	**10**	0.01	25.43
12	**12700.74**	**10**	0.01	14.17	**12700.74**	**10**	0.00	2.66	12700.74	5	3.14	508.27
14	12615.67	8	0.44	163.11	**12615.67**	**10**	0.01	10.80	12615.67	1	6.05	601.27
16	13232.52	4	2.52	485.71	**13232.52**	**10**	0.01	45.09	13232.52	1	6.06	602.53
18	18001.38	0	6.47	600.30	**18001.38**	**10**	0.01	184.39	18001.38	0	8.06	602.33
20	21394.91	0	8.76	600.60	**21394.91**	**8**	0.76	380.35	21394.91	0	8.76	602.19

* All values in all columns except for “Num” are the average results of 10 independent runs for each order number.
* Column “Num” indicates the number of instances solved to optimality among the 10 instances.

shown as the bold numbers. The LP relaxation results show M2 is tight, which can explain for its good computational efficiency.

Notably, due to the makespan limit, the minimum cost obtained in solving problem IDM-C in Table 1 cannot be obtained in solving problem IDM-CM. Nevertheless, all these optimal solutions satisfy the makespan limit. While the ECM does not support to set makespan limit, the results here show that our integration provides the capability to set many practical production requirements. Furthermore, for all the 10 order numbers, both the cost and makespan of ECM solutions are larger than those of the M2 solutions. In summary, by using our method, it is practical for the machine shop to achieve solutions that lead to both smaller cost and shorter makespan.

6.4. Results for Problem IDM-CD

In general, the problem with due date consideration (IDM-CD) is harder to solve. The proposed three formulations (M3-1, M3-2-L and M3-3-L) can perform differently in terms of their solution quality (measured by Gap, the smaller the better; the Number of instances solved to optimality, the larger the better) and time efficiency (measured by Time, the smaller the better).

Table 3 presents such numerical results. Obviously, the solution quality of M3-2-L is the best among the three. This can be seen from the number of instances solved to optimality. Among all the 100 instances, M3-2-L solves 98 to optimality except for 2 instances at $n = 20$ orders, while that of M3-1 and M3-3-L are 72 and 66, respectively. The same conclusions can be drawn from the gap and the solution time. For example, at $n = 18$, M3-2-L takes 184.39 seconds to solve all 10 instances to optimality, while M3-1 takes 600.30 seconds yet still has gap 6.47%, and M3-3-L takes 602.33 seconds with gap 8.06%. Note that time above 600 seconds means the solving procedure is terminated due to time limit of `CPLEX`. It is interesting to see although with different gaps for the three models, the objectives are the same. In our experience, it can often be the case that `CPLEX` has found the optimal solution in 5 minutes, though it takes much longer time to certify its optimality. Thus, formulations with fast increasing lower bounds can certify the optimality in shorter time.

In summary, these models consider the cost, makespan and due date, and could have large potentials to be used for industrial practice. Among them, model M3-2-L is the best one that gives the best solution within the smallest time. This validates our idea of integrating discrete event dynamics with machining dynamics for the operational excellence of machine tool.

7. Conclusion, Discussion and Future Work

We have identified and studied a significant problem of the integration of discrete-event dynamics and machining dynamics for the operational excellence of machine tool for the first time. The machining stability, various cutting speeds and tool life were incorporated into the production of a series of parts from multiple orders in the queue of machine tool. Based on a state-of-the-art logistic regression model for tool life prediction, a new cost function was developed for orders with cutting speed choices as the variables. Then the cost was minimized via choosing different stable cutting speeds.

We have demonstrated a set of models and algorithms by considering various practical production scenarios, including cost, makespan and due date. When considering the integrated optimization of cost and makespan, we developed integer optimization models and exact algorithm. Then, we considered the integrated optimization of cost and due date. Three categories of mixed-integer optimization models were presented based on the choices of decision variables. The linearization technique was applied to obtain the models. The results showed while satisfying the makespan and due date requirements, our methods achieved on average 3% - 5% reduction of the cost, which has the potential for cost saving in practice.

Future work includes the deeper integration from both the discrete-event dynamics and the machining dynamics. Besides cutting speed, other machining parameters can be as decision variables, including axial depth of cut and feed rate. Thus, we may have nonlinear and nonconvex constraints from the stability lobe diagram. Also, the stochastic tool life prediction can be considered. In the meantime, the complexities of the problems for makespan and due date constraints are one future direction. The practical extensions of the problems are to consider different candidate speeds for orders, various part types of an order, and multiple machine tools in the machine shop.

Acknowledgements

This material is based upon work supported in part by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), Advanced Manufacturing Office (AMO), which is operated by UT-Battelle, LLC, for the U.S. Department of Energy under Contract DE-AC05-00OR22725. This work is supported in part by the University of Tennessee Knoxville under the Graduate Advancement Training and Education (GATE) program of Science Alliance, and the Southeastern Advanced Machine Tools Networks (SEAMTN).

References

[1] C. T. Tyler and T. L. Schmitz, "Analytical process damping stability prediction," *Journal of Manufacturing Processes*, vol. 15, no. 1, pp. 69–76, 2013.

[2] C. T. Tyler, J. R. Troutman, and T. L. Schmitz, "A coupled dynamics, multiple degree of freedom process damping model, part 1: Turning," *Precision Engineering*, vol. 46, pp. 65–72, 2016.

[3] Y. Altintas, G. Stepan, E. Budak, T. Schmitz, and Z. M. Kilic, "Chatter stability of machining operations," *Journal of Manufacturing Science and Engineering*, vol. 142, no. 11, p. 110801, 2020.

[4] J. Karandikar, A. Honeycutt, T. Schmitz, and S. Smith, "Stability boundary and optimal operating parameter identification in milling using bayesian learning," *Journal of Manufacturing Processes*, vol. 56, pp. 1252–1262, 2020.

[5] J. Karandikar, T. McLeay, S. Turner, and T. Schmitz, "Tool wear monitoring using naive bayes classifiers," *The International Journal of Advanced Manufacturing Technology*, vol. 77, no. 9, pp. 1613–1626, 2015.

[6] J. Karandikar, T. Schmitz, and S. Smith, "Physics-guided logistic classification for tool life modeling and process parameter optimization in machining," *Journal of Manufacturing Systems*, vol. 59, pp. 522–534, 2021.

[7] T. No, M. Gomez, R. Copenhaver, J. U. Perez, C. Tyler, and T. L. Schmitz, "Force and stability modeling for non-standard edge geometry endmills," *Journal of Manufacturing Science and Engineering*, vol. 141, no. 12, p. 121002, 2019.

[8] M. F. Gomez and T. L. Schmitz, "Displacement-based dynamometer for milling force measurement," *Procedia Manufacturing*, vol. 34, pp. 867–875, 2019.

[9] L. Wang, Z. Shi, and L. Shi, "A novel quadratic formulation for customer order scheduling problem," in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 576–580, IEEE, 2013.

[10] Z. Shi, L. Wang, P. Liu, and L. Shi, "Minimizing completion time for order scheduling: Formulation and heuristic algorithm," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1558–1569, 2015.

[11] Z. Shi, H. Ma, M. Ren, T. Wu, and J. Y. Andrew, "A learning-based two-stage optimization method for customer order scheduling," *Computers & Operations Research*, vol. 136, p. 105488, 2021.

[12] Z. Shi, Z. Huang, and L. Shi, "Customer order scheduling on batch processing machines with incompatible job families," *International Journal of Production Research*, vol. 56, no. 1-2, pp. 795–808, 2018.

[13] M. Qin, R. Wang, Z. Shi, L. Liu, and L. Shi, "A genetic programming-based scheduling approach for hybrid flow shop with a batch processor and waiting time constraint," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 94–105, 2019.

[14] C. Zhang, Z. Shi, Z. Huang, Y. Wu, and L. Shi, "Flow shop scheduling with a batch processor and limited buffer," *International Journal of Production Research*, vol. 55, no. 11, pp. 3217–3233, 2017.

[15] H. Ma, H. K. Lee, Z. Shi, and J. Li, "Workforce allocation in motorcycle transmission assembly lines: A case study on modeling, analysis, and improvement," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4164–4171, 2020.

[16] J. M. Framinan and P. Perez-Gonzalez, "Order scheduling with tardiness objective: Improved approximate solutions," *European Journal of Operational Research*, vol. 266, no. 3, pp. 840–850, 2018.

[17] W. Ferry and Y. Altintas, "Virtual five-axis flank milling of jet engine impellers—part ii: feed rate optimization of five-axis flank milling," *Journal of Manufacturing Science and Engineering*, vol. 130, no. 1, p. 011013, 2008.

[18] X. Beudaert, P.-Y. Pechard, and C. Tournier, "5-axis tool path smoothing based on drive constraints," *International Journal of Machine Tools and Manufacture*, vol. 51, no. 12, pp. 958–965, 2011.

[19] R. K. Kayan and M. S. Akturk, "A new bounding mechanism for the cnc machine scheduling problems with controllable processing times," *European Journal of Operational Research*, vol. 167, no. 3, pp. 624–643, 2005.

[20] S. Gurel and M. S. Akturk, "Considering manufacturing cost and scheduling performance on a cnc turning machine," *European Journal of Operational Research*, vol. 177, no. 1, pp. 325–343, 2007.

[21] Z. Vagnorius, M. Rausand, and K. Sørby, "Determining optimal replacement time for metal cutting tools," *European Journal of Operational Research*, vol. 206, no. 2, pp. 407–416, 2010.

[22] A. Zaretalab, H. S. Haghighi, S. Mansour, and M. S. Sajadieh, "A mathematical model for the joint optimization of machining conditions and tool replacement policy with stochastic tool life in the milling process," *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 5, pp. 2319–2339, 2018.

[23] T. L. Schmitz and K. S. Smith, "Machining dynamics: Frequency response to improved productivity, 2nd edition," *Springer*, p. 303, 2018.

[24] F. W. Taylor, "On the art of cutting metals," *Transactions of the ASME*, vol. 28, pp. 31–248, 1906.

[25] F. Glover and E. Woolsey, "Converting the 0-1 polynomial programming problem to a 0-1 linear program," *Operations Research*, vol. 22, no. 1, pp. 180–182, 1974.