Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-10 (2022) 1521–1526

Mathematical Models for Minimizing Total Tardiness on Parallel Additive Manufacturing Machines

Chunlong Yu * **Andrea Matta** ** **Quirico Semeraro** ***
Junjie Lin ****

* *School of Mechanical Engineering, Tongji University, Shanghai 201804, China, (e-mail: chunlong_yu@tongji.edu.cn).*
** *Department of Mechanical Engineering, Politecnico di Milano, Milan 20156, Italy, (e-mail: andrea.matta@polimi.it).*
*** *Department of Mechanical Engineering, Politecnico di Milano, Milan 20156, Italy, (e-mail: quirico.semeraro@polimi.it).*
**** *Sino-German College, University of Shanghai for Science and Technology, Shanghai 200093, China, (e-mail: 593200291@qq.com).*

Abstract: In this research we tackle the scheduling problem in additive manufacturing for unrelated parallel machines. Both the nesting and scheduling aspects are considered. Parts have several alternative build orientations. The goal is to minimize the total tardiness of parts. We propose a mixed-integer linear programming model which considers the nesting subproblem as a 2D bin-packing problem, as well as a model which simplifies the nesting subproblem to a 1D bin-packing problem. The computational efficiency and properties of the proposed models are investigated by numerical experiments. Results show that the total tardiness optimization significantly increases the complexity of the problem, only the simple instances are solved optimally, whereas the makespan variant is able to solve all testing instances. Using the 1D bin-packing simplification allows for solving more instances to optimality, but with a risk of obtaining nesting-infeasibility. We also observed the compromise between the total tardiness and makespan objectives, which originates from the dilemma of "packing more parts to benefit from the common machine setup/recoating time" or "packing less parts to maintain the flexibility for handling distributed duedates".

Keywords: additive manufacturing; production scheduling; bin-packing problem; mathematical programming models; total tardiness.

1. INTRODUCTION

Additive Manufacturing (AM) is versatile at fabricating complex designs using digital representations and thus, AM acts as a key for highly customized orders. This stimulates the on-demand business models such as Factory-as-a-Service (FaaS) (Kang et al., 2018). In FaaS, customers can place production orders via the online market, and the factory has to perform the delivery before a duedate, otherwise a penalty will be paid. Due to the layer-by-layer construction feature, AM process is generally time consuming, production decisions are hence having great impacts on the system performance. To satisfiy the increasing demand, how to efficiently schedule the production tasks becomes a critical problem for the on-time order delivery.

In recent years, the scheduling problem for AM machines has drawn an increasing attention (Oh et al., 2020). Generally, the production scheduling is to allocate the jobs to the machines and decide their processing sequence to optimize certain objectives. Compared to the scheduling problems in conventional manufacturing environment, some specific features of AM introduce new challenges.

An AM machine is generally capable of printing a set of parts, known as a *build*, at a time. We use the term "batch" to represent a general concept in the following context. The nesting problem thus arises during the production, which concerns the placement of parts in the limited build platform. The decisions include part displacement and rotation around the z-axis (vertical axis). A proper nesting leads to a high machine usage rate. There are two types of nesting, 2D and 3D, depending on whether the parts can be put upon each other. In this study, we consider the case of selective laser melting (SLM) technique, and thus, no stacks are permitted because supporting structures are required for heat dissipation and fixing parts on the platform (Zhang et al., 2020). Hence, the nesting problem concerns the packing of parts' projections into the platform, which is usually a rectangular area. To reduce the difficulty of considering complex part geometries, a common way is to represent parts' projections by their minimum bounding boxes, which leads to a 2D bin-packing problem (2D-BPP) (Lodi et al., 2002). However, even for this case, the nesting problem is NP-hard.

⋆ This work is sponsored by Shanghai Pujiang Program (21PJ1413300).

10.1016/j.ifacol.2022.09.606

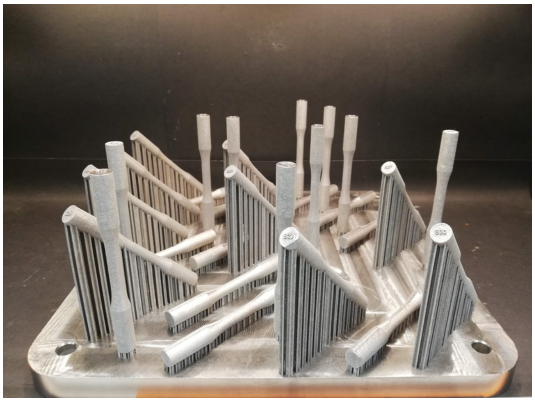


Fig. 1. A batch of parts with different build orientations

The selection of build orientation of parts is also important. Fig. 1 illustrates a batch of specimens for tensile trails, which are printed with different build orientations of 0° (vertical), 45° and 90° (horizontal) deviation from the z-axis. Obviously, vertical orientation accounts for the smallest projection area but the largest height, while the horizontal orientation is the opposite case. Since the batch processing time depends on the maximum height of the parts in the batch, the selection of build orientation can be viewed as a trade-off between the maximum number of nested parts and the batch processing time. Thus, the build orientation selection has an impact on the schedule performance.

In the literature, the scheduling and nesting are generally tackled in steps, i.e., parts are firstly grouped to form batches (nesting), then, the batches are assigned and sequenced on the AM machines (scheduling).This divide-and-conquer methodology inevitably leads to potential gap from the optimal joint decisions. Recently, the number of works tackling the joint nesting-scheduling problems is increasing, such as Li and Zhang (2018), Zhang et al. (2020) and Alicastro et al. (2021). Further, most studies assume that the build orientation is fixed before the nesting and scheduling. This simplifies the problem but also restricts the potential to find better schedules by adopting different orientation strategies, which has been shown by Griffiths et al. (2019) and Che et al. (2021). Lastly, machine productivity and cost are the most common objectives, while under the on-demand manufacturing mode, the duedate performance is more critical.

Several mathematical models have been proposed for the AM scheduling problems. Kucukkoc (2019) proposed three models for different shop environments without considering the 2D-BPP. Altekin and Bukchin (2021) extended one of these models for the cost objective. Although the model of Li and Zhang (2018) tackles the joint problem of bin-packing and scheduling, it adopts the "p-batch" (parallel batch) assumption, i.e., the batch processing time equals to the greatest processing time of the batched parts. This might not fit well for the AM procedure. Che et al. (2021) provided the first model considering alternative build orientations. However, none of these models address the total tardiness objective.

In this study, we investigate the nesting-scheduling problem in a system of parallel AM machines. Machines are unrelated due to different platform capacities (sizes). A set of parts are to be scheduled. Each part comes with a duedate, and a set of alternative build orientations satisfying the quality constraint. When printing, we consider the case that a part is not allowed to stack upon another. We aim at deciding the joint decision of build orientation selection, nesting and scheduling of the parts, to minimize the total tardiness. The contributions of this research are:

- Two mixed-integer linear programming (MILP) models considering different fidelity levels of the underlying bin-packing problem (1D and 2D) are provided. To improve the solvability, the constraints are linearized. To the best of our knowledge, this is the first time that this nesting-scheduling problem is modeled and tackled.
- The performance of the proposed models are tested and compared on instances taken from the literature. By doing so, we provide some insights including: (a) the difference between optimizing the makespan and total tardiness objective (2) the pros and cons of simplifying the bin-packing problem.

The remaining of the paper is organized as follows. Section 2 describes the problem. Section 3 proposes the mathematical models. Section 4 reports the numerical results, and Section 5 concludes the paper.

2. PROBLEM DESCRIPTION

A collection of parts $J = \{1, \ldots, n\}$ are to be printed on a set of parallel machines $I = \{1, \ldots, m\}$. Each part $j \in J$ has a given duedate d_j, a volume v_j and a set of optional build orientations K_j. Each build orientation $k \in K_j$ corresponds to a 3D geometry with four parameters $\{h_{jk}, l_{jk}, w_{jk}, s_{jk}\}$, which are the height, length and width of the rectangle projection, and the volume of the supporting structure, respectively. Each machine $i \in I$ can process a batch of parts simultaneously. The capacity of the build platform of machine i is defined by a cube with length $\mathcal{L}_i$, width $\mathcal{W}_i$ and height $\mathcal{H}_i$. Parts are not allowed to stack on each other, therefore, a set of parts can be placed in a batch only when their projections do not overlap, and their geometries are within the platform boundaries. The batch processing time P_{ib} consists of three components: the machine setup time, the laser scanning time and the recoating time, as given in (5) . The parts must be placed parallel to the length or width of the platform, and can rotate around the z-axis by 90 degrees. Let c_j be the completion time of part j, and C_{ib} be that of the b-th build on machine i, $c_j = C_{ib}$ if part j is in the batch. The tardiness $T_j = \max\{0, c_j - d_j\}$. All parts are available at the beginning. The goal is to decide the build orientation for each part, place and schedule them on the machines to minimize the total tardiness.

3. MATHEMATICAL PROGRAMMING MODELS

This section proposes two MILP models for the nesting-scheduling problem.

3.1 Nesting and Scheduling Model

The MILP model is given as follows. The sets are:

I : Set of machines;
J : Set of parts;
K_j : Set of alternative build orientation of part j;
B_i : Set of available batches on machine i;

The parameters are:

S_i : Setup time of machine i;
V_i : Scanning speed of machine i;
v_j: Volume of part j;
U_i : Recoating speed of machine i;
$\mathcal{L}_i,\mathcal{W}_i,\mathcal{H}_i$: Length, width and height of the build platform on machine i;
$h_{jk}, l_{jk}, w_{jk}, s_{jk}$: Height, length, width and supporting structure volume of the k-th orientation of part j.

The decision variables are:

X_{jib} : equals 1 if part j is assigned to the b-th batch of machine i, 0 otherwise;
Y_{jk} : equals 1 if part j selects the k-th optional build orientation, 0 otherwise;
(x_j, y_j): coordinates of the left-bottom point of part j's projection in a batch;
o_j: equals 1 if part j is placed with its length parallel to the that of the platform, 0 otherwise.

The auxiliary variables are:

Z_{ib} : equals 1 if the b-th batch on machine i is formed, 0 otherwise;
$PL_{jj'}$: equals 1 if part j's right-top point is placed to the left of part j''s left-bottom point in the same batch, 0 otherwise;
$PB_{jj'}$: equals 1 if part j's right-top point is placed below part j''s left-bottom point in the same batch, 0 otherwise;
P_{ib} : processing time of the b-th batch on machine i;
C_{ib} : completion time of the b-th batch on machine i;
H_{ib} : height of the b-th batch on machine i.
c_j : completion time of part j;
T_j : tardiness of part j;
M : a large number.

$PL_{jj'}$,$PB_{jj'}$,$PL_{j'j}$ and $PB_{j'j}$ together describe the position relationship of part j and j'.

The nesting-scheduling problem of parallel AM machines, denoted by $\mathscr{NSPM}$, is formulated as:

$$\mathscr{NSPM}: \quad \min \sum_{j\in J} T_j, \tag{1}$$

s.t.

$$\sum_{j\in J} X_{jib} \le M Z_{ib}, \forall i\in I, b\in B_i, \tag{2}$$

$$\sum_{i\in I}\sum_{b\in B_i} X_{jib} = 1, \forall j\in J, \tag{3}$$

$$\sum_{k\in K_j} Y_{jk} = 1, \forall j\in J, \tag{4}$$

$$P_{ib} = S_i Z_{ib} + V_i \sum_{j\in J}(X_{jib} v_j + X_{jib}\sum_{k\in K_j} Y_{jk} s_{jk}) + U_i H_{ib}, \forall i\in I, b\in B_i, \tag{5}$$

$$x_j + \sum_{k\in K_j} Y_{jk}\cdot l_{jk} \le L_i + M(1-\sum_{b\in B_i} X_{jib}) + M(1-o_j), \forall j\in J, i\in I, \tag{6}$$

$$x_j + \sum_{k\in K_j} Y_{jk}\cdot w_{jk} \le L_i + M(1-\sum_{b\in B_i} X_{jib}) + M o_j, \forall j\in J, i\in I, \tag{7}$$

$$y_j + \sum_{k\in K_j} Y_{jk}\cdot w_{jk} \le W_i + M(1-\sum_{b\in B_i} X_{jib}) + M(1-o_j), \forall j\in J, i\in I, \tag{8}$$

$$y_j + \sum_{k\in K_j} Y_{jk}\cdot l_{jk} \le W_i + M(1-\sum_{b\in B_i} X_{jib}) + M o_j, \forall j\in J, i\in I, \tag{9}$$

$$x_j + \sum_{k\in K_j} Y_{jk}\cdot l_{jk} \le x_{j'} + M(1-PL_{jj'}) + M(1-o_j), \forall j, j'\in J, \tag{10}$$

$$x_j + \sum_{k\in K_j} Y_{jk}\cdot w_{jk} \le x_{j'} + M(1-PL_{jj'}) + M o_j, \forall j, j'\in J, \tag{11}$$

$$y_j + \sum_{k\in K_j} Y_{jk}\cdot w_{jk} \le y_{j'} + M(1-PB_{jj'}) + M(1-o_j), \forall j, j'\in J, \tag{12}$$

$$y_j + \sum_{k\in K_j} Y_{jk}\cdot l_{jk} \le y_{j'} + M(1-PB_{jj'}) + M o_j, \forall j, j'\in J, \tag{13}$$

$$PL_{jj'} + PB_{jj'} + PL_{j'j} + PB_{j'j} \ge X_{jib} + X_{j'ib} - 1, \forall j, j'\in J, j<j', i\in I, b\in B_i, \tag{14}$$

$$H_{ib} \ge \sum_{k\in K_j} Y_{jk} h_{jk} - M(1-X_{jib}), \forall i\in I, b\in B_i, j\in J, \tag{15}$$

$$H_{ib} \le \mathcal{H}_i, \forall i\in I, b\in B_i, \tag{16}$$

$$C_{ib} \ge C_{i(b-1)} + P_{ib}, \forall i\in I, b\in B_i, \tag{17}$$

$$C_{i0} = 0, \forall i\in I, \tag{18}$$

$$c_j \ge C_{ib} - M(1-X_{jib}), \forall j\in J, i\in I, b\in B_i, \tag{19}$$

$$T_j \ge c_j - d_j, \forall j\in J. \tag{20}$$

$$x_j, Z_j \ge 0, \forall j\in J, \tag{21}$$

$$c_j, T_j \ge 0, \forall j\in J, \tag{22}$$

$$P_{ib}, C_{ib}, H_{ib} \ge 0, \forall i\in I, b\in B_i, \tag{23}$$

$$X_{jib}\in\{0,1\}, \forall j\in J, i\in I, b\in B_i, \tag{24}$$

$$Z_{ib}\in\{0,1\}, \forall i\in I, b\in B_i, \tag{25}$$

$$Y_{jk}\in\{0,1\}, \forall j\in J, k\in K_j, \tag{26}$$

$$PL_{jj'}, PB_{jj'}\in\{0,1\}, \forall j, j'\in J. \tag{27}$$

Constraints (2) ensure that a batch cannot be assigned to if not formed. Constraints (3) specify that a part is assigned to only one batch. Constraints (4) ensure that a part selects one build orientation. Constraints (5) calculate the processing time of a batch. Constraints (6) - (9) guarantee that the part is within the build boundaries. Constraints (10) - (13) guarantee that the parts are not overlapped when assigned to the same batch. More specifically, Constraints (10) and (11) impose that when $PL_{jj'} = 1$, i.e., part j is placed to the left of part j', the right edge of part j does not exceed the left edge of part j'. These constraint are relaxed when $PL_{jj'} = 0$. Constraints (12) and (13) impose for the vertical direction with the same logic. Constraints (14) ensure that when part j and j' are allocated on the same batch, i.e., X_{jib} and $X_{j'ib}$

equal 1, there is at least one active positioning relationship. Constraints (17) calculate the batch completion time, while Constraints (19) calculate the part completion time. Constraints (20) calculate the tardiness of a part.

The following constraints are introduced to break the symmetry of batches:

$$Z_{i(b-1)} \geq Z_{ib}, \forall i \in I, b \in B_i \setminus \{1\}. \tag{28}$$

To linearize (5), we introduce an auxiliary variable e_{jib} to represent the support structure volume of part j printed on the b-th batch of machine i, and replace Constraints (5) with the followings:

$$P_{ib} = S_i Z_{ib} + V_i \sum_{j \in J} (X_{jib} v_j + e_{jib}) + U_i H_{ib}, \forall i \in I, b \in B_i, \tag{29}$$

$$e_{jib} \geq \sum_{k \in K_j} Y_{jk} s_{jk} - M(1 - X_{jib}), \forall j \in J, i \in I, k \in K_j, \tag{30}$$

$$e_{jib} \geq 0, \forall j \in J, i \in I, b \in B_i \tag{31}$$

3.2 Nesting and Scheduling Model with Packing Simplification

The integration of the 2D-BPP and the parallel machine scheduling problem greatly increases the difficulty. To tackle this, the common way is to simplify the 2D-BPP into a 1D-BPP, as in Kucukkoc (2019). The basic idea is to replace the two-dimensional machine capacity constraint with a one-dimensional one, i.e., the total projection area of the parts printed on the machine cannot succeed the platform area. Thus, we replace Constraints (7) - (14) by

$$\sum_{j \in J} X_{jib} a_j \leq \mathcal{L}_i \mathcal{W}_i, \forall i \in I, b \in B_i, \tag{32}$$

$$a_j = \sum_{k \in K_j} Y_{jk} l_{jk} w_{jk}, \forall j \in J \tag{33}$$

where $a_j \geq 0, j \in J$ is an auxiliary variable representing the projection area of part j on the build platform. Note that Constraints (32) are non-linear. To linearize them, we introduce the following auxiliary decision variable:

α_{jib} : projection area of part j on the b-th batch of machine i,

and the parameter:

$\bar{a}_j$: maximum projection area of part j, i.e., $\max_{j \in K_j} l_{jk} w_{jk}$.

Constraints (32) are replaced by the followings.

$$\sum_{j \in J} \alpha_{jib} \leq \mathcal{L}_i \mathcal{W}_i, \forall i \in I, b \in B_i, \tag{34}$$

$$\alpha_{jib} \leq \bar{a}_j X_{jib}, \forall j \in J, i \in I, b \in B_i, \tag{35}$$

$$\alpha_{jib} \geq a_j - (1 - X_{jib}) \bar{a}_j, \forall j \in J, i \in I, b \in B_i, \tag{36}$$

$$\alpha_{jib} \leq a_j, \forall j \in J, i \in I, b \in B_i, \tag{37}$$

$$\alpha_{jib} \geq 0, \forall j \in J, i \in I, b \in B_i, \tag{38}$$

Constraints (34) are the capacity constraints. Constraints (35) ensure that α_{jib} is 0 when part j is not assigned to the b-th batch on machine i. Constraints (36) and (37) impose that $\alpha_{jib} = a_j$ when part j is assigend to the b-th batch on machine i.

In summary, the model, denoted by $\mathscr{NSPM} - 1D$, is constructed as below:

$$\mathscr{NSPM} - 1D: \quad \min \sum_{j \in J} T_j, \tag{39}$$

s.t. (2) - (4), (29) -(31), (34 - 38), (33), (28), (15) -(26).

However, one should notice that, the nesting details are not given by $\mathscr{NSPM} - 1D$ and should be generated afterwards by, e.g., some packing heuristics. Also, there is a risk that the given part-to-batch assignment cannot lead to any feasible nesting due to the packing simplification.

4. NUMERICAL RESULTS

In this section, we investigate the model efficiency. Particularly, we are interested in the following questions:

(a) Is it more difficult to optimize the total tardiness than the makespan?
(b) How much can we benefit from simplifying the 2D-BPP into 1D-BPP, and what is the disavantage?

4.1 Testing Instances

The testing instances are adopted from Che et al. (2021) (http://www.computational-logistics.org/orlib/2L-BPM). In these instances, the parts are downloaded from the website "Thingiverse", the part information is collected using a 3D printing application named "Repetier-Host", and the AM machine information is from Kucukkoc (2019). We select five instances as listed in Table 1, in which the number of parts are adjusted for better testing our models. We generate the duedates for the part, e.g., j, from a discrete uniform distribution as:

$$d_j = \text{Unif}[\mu(1 - TF - RDD/2), \mu(1 - TF + RDD/2)], \tag{40}$$

where TF and RDD are the average tardiness factor and the range of duedates, respectively. A large TF value indicates the case of urgent orders. μ is the estimated makespan, which is calculated under the following assumptions: (1) the parts select the build orientation with the largest height; (2) the parts are sequenced in a non-decreasing order of their projection area; (3) the batching is a 1D-BPP; (4) the parts are assigned in sequence to the machine with the earliest completion time.

Table 1. Testing instances

ID	Name	#parts	#machines	#build orientations
1	ht1_1-10	10	2	1 to 3
2	ht1_2-15	15	2	1 to 3
3	ht1_3 -20	20	2	1 to 3
4	ht2_2-25	25	2	1 to 3
5	ht2_1-30	30	2	1 to 3

4.2 Experiment Setup

To investigate question (a) and (b), we perform a full factorial design of experiment with the following factors and levels: Models - $\{\mathcal{M}$: $\mathscr{NSPM}(C_{max})$, $\mathscr{NSPM}$, $\mathscr{NSPM} - 1D$ $\}$, Instance - {1,2,3,4,5}, TF - {0.3, 0.6}, RDD - {0.3, 0.6}.

The performance of $\mathscr{NSPM}(C_{max})$, $\mathscr{NSPM}$, $\mathscr{NSPM} - 1D$ are compared, where $\mathscr{NSPM}(C_{max})$ is identical to

Table 2. Results of $\mathscr{NSPM}(C_{max})$

Instance	Opt.	C_{max}	t (sec.)	OptGap
ht1_1-10	**opt.**	14.5	1.0	0.00
ht1_2-15	**opt.**	14.2	7.0	0.00
ht1_3-20	**opt.**	29.6	59.9	0.00
ht2_2-25	**opt.**	35.0	121.5	0.00
ht2_1-30	**opt.**	46.6	282.3	0.00

Table 3. Results of $\mathscr{NSPM}$

Instance	TF	RDD	Opt.	$\sum T_j$	C_{max}	t	Gap[%]
ht1_1-10	0.3	0.3	**opt.**	0.01	16.91	36.7	0.00
ht1_1-10	0.3	0.6	feas.	0.01	16.89	1800	1.09
ht1_1-10	0.6	0.3	**opt.**	26.05	17.20	72.9	0.00
ht1_1-10	0.6	0.6	**opt.**	26.28	17.20	345	0.00
ht1_2-15	0.3	0.3	**opt.**	0.43	16.70	274	0.09
ht1_2-15	0.3	0.6	feas.	0.01	19.30	1800	2.33
ht1_2-15	0.6	0.3	feas.	32.54	20.75	1800	64.75
ht1_2-15	0.6	0.6	feas.	34.95	18.35	1800	68.89
ht1_3-20	0.3	0.3	feas.	0.02	36.68	1800	7.82
ht1_3-20	0.3	0.6	feas.	0.02	43.56	1800	13.43
ht1_3-20	0.6	0.3	feas.	58.20	39.82	1800	99.98
ht1_3-20	0.6	0.6	feas.	70.17	42.88	1800	92.30
ht2_2-25	0.3	0.3	feas.	28.19	45.29	1800	99.93
ht2_2-25	0.3	0.6	feas.	6.88	42.34	1800	99.75
ht2_2-25	0.6	0.3	feas.	188.24	50.62	1800	99.99
ht2_2-25	0.6	0.6	feas.	179.17	51.19	1800	97.97
ht2_1-30	0.3	0.3	feas.	35.86	59.21	1800	99.93
ht2_1-30	0.3	0.6	feas.	24.55	62.38	1800	99.90
ht2_1-30	0.6	0.3	feas.	154.41	66.19	1800	99.62
ht2_1-30	0.6	0.6	feas.	128.10	62.66	1801	96.48

Table 4. Results of $\mathscr{NSPM}$-1D

Instance	TF	RDD	Opt.	$\sum T_j$	C_{max}	t	Gap[%]
ht1_1-10	0.3	0.3	**opt.**	0.01	16.19	1.3	0.00
ht1_1-10	0.3	0.6	**opt.**	0.01	19.11	1.0	0.00
ht1_1-10	0.6	0.3	**opt.**	23.51*	16.80	93.4	0.00
ht1_1-10	0.6	0.6	**opt.**	23.74*	16.80	217	0.00
ht1_2-15	0.3	0.3	**opt.**	0.43	16.70	135	0.00
ht1_2-15	0.3	0.6	**opt.**	0.01	18.67	8.5	0.00
ht1_2-15	0.6	0.3	feas.	31.68	18.48	1800	72.22
ht1_2-15	0.6	0.6	feas.	31.76	20.41	1800	59.53
ht1_3-20	0.3	0.3	**opt.**	0.01	34.55	10.5	0.00
ht1_3-20	0.3	0.6	**opt.**	0.01	42.91	15.5	0.00
ht1_3-20	0.6	0.3	feas.	44.90	34.97	1800	99.97
ht1_3-20	0.6	0.6	feas.	65.74	40.46	1800	86.10
ht2_2-25	0.3	0.3	feas.	13.40	43.03	1800	99.85
ht2_2-25	0.3	0.6	feas.	1.13	42.82	1800	98.10
ht2_2-25	0.6	0.3	feas.	168.99	47.64	1800	99.99
ht2_2-25	0.6	0.6	feas.	175.34	51.48	1800	96.61
ht2_1-30	0.3	0.3	feas.	24.24	53.09	1800	99.88
ht2_1-30	0.3	0.6	feas.	5.43	54.40	1800	99.45
ht2_1-30	0.6	0.3	feas.	124.64	59.15	1800	99.41
ht2_1-30	0.6	0.6	feas.	109.23	58.76	1800	95.87

* Note that the optimal objective values in this table might be lower than those in Table 3 due to the packing infeasibility.

$\mathscr{NSPM}$ except that the objective is the makespan. The models are solved by Gurobi solver (ver. 9.1.2) in a computer with Intel i7-9750H CPU (2.6GHz), given a maximum CPU time of 1800 seconds. The acceptable optimality gap is set as 0.1%.

4.3 Results Analysis

The results are reported in Tables 2 - 4, respectively. The column "Opt." indicates whether the optimal solution is obtained in the given time (opt.) or not (feas.). The column "t" is the CPU time in seconds. The column "Gap" is the optimality gap of the best feasible solution, given by $(|BestFeas. - BestBound|)/BestFeas.$ The column C_{max} and $\sum T_j$ are the makespan and total tardiness, respectively. Note that the solution of $\mathscr{NSPM}(C_{max})$ is not affected by the TF and RDD values, and thus these two columns are not included in Table 2. In Table 3 and 4, we include the makespan values though the objective function of $\mathscr{NSPM}$ and $\mathscr{NSPM}-1D$ is the total tardiness.

Table 5. $\mathscr{NSPM}(C_{max})$ solution on ht_1_2-15.

NSPM(Cmax)				
Machine	Batch	Completion time	Parts in Batch	Part duedates
1	1	14.14	1,2,3,4,8,9,10, 12,13,14,15	-
2	1	14.18	5,6,7,11	-

Table 6. $\mathscr{NSPM}$ solution on ht_1_2-15 (TF=0.3, RDD=0.3). The tardy part is marked in bold.

Machine	Batch	Completion time	Parts in Batch	Part duedates
1	1	12.71	7,9,12,14	13,14,14,14
1	2	16.42	**13**,15	**16**,17
2	1	10.64	1,2,6,8,11	12,15,11,11,14
2	2	16.7	3,4,5,10	17,17,17,17

As shown, $\mathscr{NSPM}(C_{max})$ solves all instances within 1800 seconds to optimality, whilst $\mathscr{NSPM}$ cannot tackle any instance with #part ≥ 15. The nesting-scheduling problem with total tardiness objective seems more difficult than that with makespan objective.

Table 5 and 6 report the optimal schedules for ht_1_2-15 in terms of the makespan and total tardiness (TF=0.3, RDD=0.3), respectively. As shown, in the case of makespan, the parts are allocated in such a way that the batch completion times are balanced. While in the case of total tardiness, only the parts with similar duedates are batched and processed together. Characterizing by the processing time in (5), the batch is between a p-batch (parallel batch) and an s-batch (serial batch). Packing more parts in the batch brings a benifit of sharing machine setup and recoating time, but also increases the batch processing time (laser scanning), leading to late delivery. Therefore, the "packing as much as possible" rule that works well for the makespan is not necesary true for the total tardiness, which reveals the potential conflict between these two objectives.

We further investigate, when optimizating the total tardiness, how the order urgency (TF) and duedate range (RDD) would influence the makespan values. Figure 2 shows that the makespan value tends to increase with either TF or RDD, with only a few exceptions. This implies a greater conflict between these two objectives when the duedates are tight or widely distributed.

The part placement in the schedule of ht_1_2-15 is illustrated in Figure 3, where the parts are represented by their bounding boxes. The parts are placed as closed to the bottom-left corner as possible, because we added in the objectve function a term of $\eta \sum_{j \in J}(x_j + y_j)$, scaled by a small η factor. Note that other reference points can be

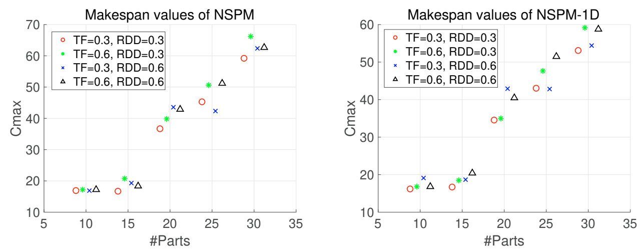
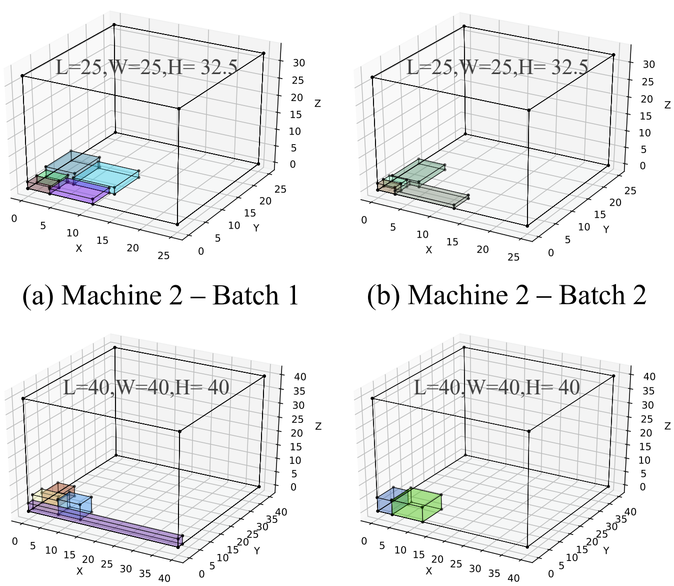


Fig. 2. C_{max} values in intances of different #Parts



(a) Machine 2 – Batch 1 (b) Machine 2 – Batch 2

(c) Machine 1 – Batch 1 (d) Machine 1 – Batch 2

Fig. 3. Part placement for the optimal schedule by NSPM on ht_1_2-15 (TF=0.3, RDD=0.3).

chosen. As observed, since there are enough free area in the platform, the parts tend to select the alternative build orientation with a low height to avoid high recoating time.

The comparison of Table 3 and 4 shows that, the advantage of simplying the bin-packing problem is obvious. $\mathscr{NSPM}-1D$ can solve larger instances and quicker than $\mathscr{NSPM}$ to optimality. However, inconsistent objective values are observed in some cases, e.g., ht1_1-10, TF $=0.6$, RDD $=0.3$. After investigating the optimal solutions, we found that in the schedule provided by $\mathscr{NSPM}-1D$, one of the five batches is actually infeasible, because a part's length (or width) is greater than that of the platform. Therefore, to generate feasible schedules, the batches obtained by $\mathscr{NSPM}-1D$ should undergo a feasibility checking procedure. Then, parts in infeasible batches have to be reassigned to another batch, or at least, selects another build orientation, which inevitably alters the processing times of the batches and one may lose the schedule optimality. This is the main drawback of the MILP models using packing simplification, e.g., those in Kucukkoc (2019) and Altekin and Bukchin (2021).

5. CONCLUSIONS

In this study, we formulate the nesting-scheduling problem in unrelated parallel AM machines where the parts have alternative build orientations. To minimize the total tardiness, two MILP models considering different fidelity levels of the bin-packing problem are proposed.

Under the same settings, optimizing the total tardiness is more difficult than the makespan. Moreover, there exists a compromise between the makespan and total tardiness objectives, which originates from the dilemma of "packing more parts to benefit from the common machine setup/recoating time" or "packing less parts to maintain the flexibility for handling distributed duedates". Further, simplifying the underlying 2D-BPP into 1D-BPP increases the solvability, but also introduces the penalty of creating infeasible part-to-batch assignments.

Future works will consider more technical details of the SLM machine and their impacts in modeling the nesting-scheduling problems, e.g., the preheating and cooling requriement. Also, efficient exact algorithms such as branch and bound, branch and price will be developed to solve the models.

REFERENCES

Alicastro, M., Ferone, D., Festa, P., Fugaro, S., and Pastore, T. (2021). A reinforcement learning iterated local search for makespan minimization in additive manufacturing machine scheduling problems. *Computers & Operations Research*, 105272.

Altekin, F.T. and Bukchin, Y. (2021). A multi-objective optimization approach for exploring the cost and makespan trade-off in additive manufacturing. *European Journal of Operational Research*.

Che, Y., Hu, K., Zhang, Z., and Lim, A. (2021). Machine scheduling with orientation selection and two-dimensional packing for additive manufacturing. *Computers & Operations Research*, 130, 105245.

Griffiths, V., Scanlan, J.P., Eres, M.H., Martinez-Sykora, A., and Chinchapatnam, P. (2019). Cost-driven build orientation and bin packing of parts in selective laser melting (slm). *European Journal of Operational Research*, 273(1), 334–352.

Kang, H.S., Do Noh, S., Son, J.Y., Kim, H., Park, J.H., and Lee, J.Y. (2018). The faas system using additive manufacturing for personalized production. *Rapid Prototyping Journal*.

Kucukkoc, I. (2019). Milp models to minimise makespan in additive manufacturing machine scheduling problems. *Computers & Operations Research*, 105, 58–67.

Li, X. and Zhang, K. (2018). Single batch processing machine scheduling with two-dimensional bin packing constraints. *International Journal of Production Economics*, 196, 113–121.

Lodi, A., Martello, S., and Vigo, D. (2002). Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 123(1-3), 379–396.

Oh, Y., Witherell, P., Lu, Y., and Sprock, T. (2020). Nesting and scheduling problems for additive manufacturing: a taxonomy and review. *Additive Manufacturing*, 101492.

Zhang, J., Yao, X., and Li, Y. (2020). Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing. *International Journal of Production Research*, 58(8), 2263–2282.