Industrial Engineering and Innovation Management (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/ieim.2025.080219
ISSN 2522-6924 Vol. 8 Num. 2

Minimizing maximum tardiness for parallel machine scheduling in additive manufacturing

Wenjing Zhou

Business School, University of Shanghai for Science and Technology, Shanghai, 200093, China
243563560@st.usst.edu.cn

Keywords: Additive manufacturing, Parallel machine scheduling, Heuristic algorithm

Abstract: Additive manufacturing, also known as 3D printing, has unique process characteristics that not only enable the production of integrated and structurally complex parts but also meet customers' customized requirements. Because the processing times different from those in traditional batch scheduling, the scheduling problems in additive manufacturing face some new challenges. It is important to find a feasible solution for the scheduling problems in additive manufacturing in a reasonable time, to let AM fully display its advantages of cost reduction and efficiency improvement. This paper mainly addresses a parallel-machine scheduling problem in additive manufacturing, with the objective of minimizing the maximum tardiness of all parts. We establish a mixed-integer linear programming (MILP) model for this problem and develop a corresponding algorithm (Algorithm P), in which the greedy allocation stage comprehensively considers the impact of batch membership on the objective function. We conduct a large number of numerical experiments, and the results show that Algorithm P has advantages in terms of computational time and the number of opened batches.

1. Introduction

Additive manufacturing (AM) is widely recognized as a disruptive technology capable of fabricating complex, custom-tailored parts with high precision and efficiency[1]. In recent years, the application of AM has attracted increasing attention, as it promotes the transformation of manufacturing structures and production paradigms. Compared with traditional subtractive and forming manufacturing processes, AM utilizes software such as CAD to digitize product structures, which greatly enhances the flexibility and transferability of production design. Through its layer-by-layer material deposition process, AM allows the production of parts with complex internal structures without the need for molds, achieving part integration, high strength, and lightweight design. In recent years, additive manufacturing has gradually become the preferred method for producing urgent parts in industries such as aerospace and medical fields due to its advantages in meeting small-batch customization needs and rapid product development. As its range of applications continues to expand, the uniqueness of additive manufacturing technology and the new demands for delivery time and speed present new challenges to traditional production scheduling.

In recent years, supply chains have been disrupted by the impact of COVID-19 and geopolitical conflicts, which has once again heightened attention on the resilience and efficiency of supply chains[2]. AM has great significance in restructuring supply chain structures and can achieve sustainable competitive advantages with shorter lead times and negligible negative environmental impacts[3]. But AM machines are very expensive, industrial consumables are expensive, and a wide range of materials is available. In practice, AM typically involves processing multiple parts simultaneously within the same batch. Batch scheduling can partially reduce the impact of the high fixed and operating costs associated with AM. However, the characteristics of additive manufacturing significantly increase the complexity of parallel-machine batch scheduling problems. AM operates in a non-preemptive manner, and the processing time for a batch is determined by the setup time, the total volume of the parts, and the maximum height of the tallest part in the batch[4]. Consequently, different combinations of parts during batch formation phase may prolong the overall processing time of the batch. The resulting tardiness accumulates across batch completion times and impacts subsequent batches. Therefore, the way in which parts are grouped into batches is a key factor influencing machine utilization and is one of the primary considerations in scheduling decisions for additive manufacturing. The allocation of parts and the arrangement of batch sequences have the most significant impact on the final delivery time of products. Because the completion times of individual parts are difficult to predict accurately. Evaluating schedules under worst-case assumptions allows potential risks to be identified early. It also provides guidance for resource allocation and production coordination in subsequent stages.

Among commonly used scheduling performance metrics, maximum tardiness provides an intuitive measure of schedule performance under worst-case conditions. When processing times are uncertain, adopting a worst-case evaluation perspective helps prevent cascading tardiness caused by a few severely tardy parts and enhances the stability of the production system. Therefore, developing scheduling strategies that aim to minimize the maximum tardiness of all parts is of considerable practical importance. It is critical to modify and extend the rules in traditional scheduling problems to make them applicable to the scheduling problems in AM. But the uncertainty in processing times makes traditional scheduling rules hard to apply directly. These rules need to be modified and extended to address the unique challenges of additive manufacturing

In this paper, we aim to address a parallel-machine scheduling problem in additive manufacturing, with the objective of minimizing the maximum tardiness of all parts. We establish a mixed-integer linear programming (MILP) model that incorporates the two-dimensional bin packing constraints for this problem. Given that the problem is NP-hard, we also develop a heuristic algorithm (Algorithm P), which is designed based on the characteristics of processing times in AM. Algorithm P thoroughly considers the impact of assigning parts to different available batches on the objective function value. Computational experiments verify the feasibility and performance advantages of the proposed mathematical model and algorithm.

The rest of the paper is organized as follows. Section 2 introduces and summarizes the literature on scheduling problems in additive manufacturing. Section 3 establishes a mixed-integer linear programming (MILP) model and demonstrates its effectiveness through numerical examples. Section 4 develops a heuristic algorithm (Algorithm P) based on the characteristics of additive manufacturing and describes its operational rules in detail. Section 5 presents and compares the results of computational experiments. Section 6 summarizes the content of this paper and discusses potential directions for future research.

2. Literature Review

Research on scheduling optimization in additive manufacturing has built a multi-level framework based on the key features of the technology. Studies have gradually moved from focusing on single-objective optimization to decision-making that considers multiple factors and fits complex application scenarios. Researchers have steadily improved both the efficiency and accuracy of solving these scheduling problems through algorithmic innovations, iterative performance improvements, and the integration of multiple methodological approaches.

Li et al. (2017) first defined the problem to minimize the average production cost per volume of material, and proposed two heuristic rules for this problem[5]. Kucukkoc (2019) extended the machine environment for scheduling problems in additive manufacturing, and established mixed-integer linear programming models with the objective of minimizing the makespan[4]. Dvorak et al. (2018) developed a model based on constraint satisfaction and compatibility graph to solve the integrated optimization problem of nesting and multi-machine scheduling in additive manufacturing, and adopted a local search algorithm incorporating multi-fidelity nesting heuristics for the solution[6]. Che et al. (2021) for the first time simultaneously considered orientation selection and two-dimensional bin packing constraints, and established a mathematical model and a simulated annealing algorithm with the objective of minimizing the makespan[7]. Cakici et al. (2025) established constraint programming models for scheduling problems in different AM machine environments, with the objective function of minimizing the makespan[8].

With the rising demands for personalization and requirements for rapid delivery, researchers have begun to focus on the integrated scheduling problem of production and transportation based on additive manufacturing. Dwivedi et al. (2023) developed a mixed-integer linear programming and variable neighborhood search method for the problem that production and transportation are carried out synchronously[9]. Zehetner and Gansterer (2022) explored the AM scheduling problem with multi-site, combining mixed-integer programming with genetic algorithms to minimize the costs[10]. Kucukkoc (2024) established a mixed-integer linear programming model which considered the costs and carbon emissions caused by production and transportation and verified its performance through numerical experiments[11].

Additionally, numerous scholars have engaged in in-depth discussions regarding delivery timeliness issues in additive manufacturing production scenarios. Chergui et al. (2018) established a mathematical model for parallel machine scheduling with the objective of minimizing total tardiness, designed a heuristic algorithm and demonstrated its effectiveness through numerical experiments[12]. Considering that existing methods have difficulty in solving customer-order-oriented scheduling problems with complex constraints such as batch processing and multiple materials effectively, Zipfel et al. (2024) designed a metaheuristic to minimize the total weighted tardiness of customer orders[13].

Some researchers have extended the packing constraint from one dimension to two dimensions in order to construct scheduling optimization models that better reflect practical scenarios. Aloui and Hadj-Hamou (2021) proposed two models to estimate production times for two different technologies with the objective of minimizing total tardiness, and established a mixed-integer linear programming model as well as heuristic approaches to address scheduling problems that incorporate two-dimensional packing constraints[14]. Nascimento et al. (2021) developed a constraint programming model aimed at reducing operating and tardy-deliveries expenses. Their research investigated the nesting and scheduling strategie for irregularly shaped parts[15]. Building on this foundation, Nascimento et al. (2024) proposed two logic-based Benders decomposition algorithms with the objective of minimizing total tardiness[16].

For the objective of minimizing the maximum tardiness, Kapadia et al. (2019) addressed the scheduling problem for parallel machines with randomly arriving parts. Aiming to minimize the maximum tardiness, they employed an iterative optimization simulation (IOS) model to compare the effects of genetic algorithms on optimizing part orientation and rotation under two different scheduling strategies on delivery performance[17].

Although existing literature has explored scheduling problems based on additive manufacturing from multiple perspectives, research focusing on the objective of minimizing the maximum tardiness is still limited. Therefore, this paper employs minimizing the maximum tardiness of all parts as the objective function to address the parallel machine scheduling problem in additive manufacturing. We incorporate two-dimensional bin packing constraints into the problem formulation, establish a mixed-integer linear programming model, and propose a heuristic algorithm for this problem.

3. Problem Description and Mathematical formulation

3.1. Problem description

This problem can be described as follows. There is a set of parts $J=\{1, 2, ..., n\}$ that need to be grouped in batches $B=\{1, 2, ..., n\}$ on parallel identical AM machines $M=\{1, 2, ..., m\}$. Each part has a due date d_j, a height h_j, a volume v_j, and a length l_j and width w_j of its minimum rectangular bounding box. Multiple AM machines have identical parameters and can operate in parallel. The relevant parameters of the machines are as follows: L, W and H correspond to the length, width and height of the machine respectively. SET is the setup time. VT is the time required for forming per unit volume of material. HT is the time required for powder-layering[4]. To ensure that any part can be built on the machine, we assumed that $l_j \leq L$, $w_j \leq W$, $h_j \leq H$.

3.2. Decision variables

u_{mbj} is a binary variable that equals 1 if part j is assigned to batch b of machine m, and 0 otherwise;

z_{mb} is a binary variable that equals 1 if batch b in machine m is opened, and 0 otherwise;

$left_{mbji}$ is a binary variable that equals 1 if part j is located left to part i in batch b of machine m, and 0 otherwise;

$below_{mbji}$ is a binary variable that equals 1 if part j is located below to part i in batch b of machine m, and 0 otherwise;

(x_j, y_j) denotes the coordinates of the lower-left corner of part j;

h_{mb} is the height of batch b in machine m;

PT_{mb} is the processing time of batch b in machine m;

C_{mb} is the completion time of batch b in machine m;

T_j is the tardiness of part j.

3.3. Mixed-integer linear programming model

This problem can be modeled as follows.

$$\min T_{max} \tag{1}$$

$$\sum_{m \in M} \sum_{b \in B} u_{mbj} = 1, \forall j \in J \tag{2}$$

$$\sum_{j \in J} u_{mbj} \leq M \cdot z_{mb}, \forall b \in B, \forall m \in M \tag{3}$$

$$x_j + w_j \leq W \cdot u_{mbj} + M(1 - u_{mbj}), \forall m \in M, \forall b \in B, \forall j \in J \tag{4}$$

$$y_j+l_j \le L \cdot u_{mbj}+M(1-u_{mbj}), \forall m \in M, \forall b \in B, \forall j \in J \quad (5)$$

$$left_{mbji}+left_{mbij}+below_{mbji}+below_{mbij} \ge u_{mbj}+u_{mbi}-1, \forall m \in M, \forall b \in B, \forall j,i \in J, j \ne i \quad (6)$$

$$x_j+w_j \le x_i+M(1-left_{mbji}), \forall m \in M, \forall b \in B, \forall j,i \in J, j \ne i \quad (7)$$

$$y_j+l_j \le y_i+M(1-below_{mbji}), \forall m \in M, \forall b \in B, \forall j,i \in J, j \ne i \quad (8)$$

$$z_{m(b+1)} \le z_{mb}, \forall m \in M, \forall b \in B \quad (9)$$

$$h_{mb} \ge h_j \cdot u_{mbj}, \forall m \in M, \forall b \in B, \forall j \in J \quad (10)$$

$$PT_{mb}=SET \cdot z_{mb}+VT \cdot \sum_{j \in J} v_j \cdot u_{mbj}+HT \cdot h_{mb}, \forall m \in M, \forall b \in B \quad (11)$$

$$C_{mb} \ge C_{m(b-1)}+PT_{mb}, \forall m \in M, \forall b \in B \quad (12)$$

$$T_j \ge C_{mb}-d_j \cdot u_{mbj}-M \cdot (1-u_{mbj}), \forall m \in M, \forall b \in B, \forall j \in J \quad (13)$$

$$T_{max} \ge T_j, \forall j \in J \quad (14)$$

$$u_{mbj}, z_{mb}, left_{mbji}, below_{mbji} \in \{0,1\}, x_j, y_j, C_{mb}, T_j \ge 0, \forall m \in M, \forall b \in B, \forall j,i \in J, j \ne i \quad (15)$$

The objective function (1) aims to minimize the maximum tardiness of all parts. Constraint (2) guarantees that each part j can be allocated to only one batch b on machine m. Constraint (3) represents that part j can be allocated to batch b only if the batch is opened. Constraints (4) and (5) are used to limit the physical dimensions of the parts, ensuring that the length and width of each part do not exceed the length and width of the machine to which it is assigned. Constraints (6) to (8) guarantee that parts assigned to the same batch must be placed on the machine without overlap, where $left_{mbji}$ and $below_{mbji}$ denote the positional relationship between part j and part i. Constraint (9) is used to ensure the opening sequence between batches in the machine m, , meaning that a new batch can only be opened on machine m when the preceding batch has been opened. Constraint (10) determines the height of batch b on machine m, whose value is the maximum height of all parts in the batch. Constraint (11) defines the processing time of batch b. Constraint (12) defines the completion time of each batch. Constraints (13) and (14) provide the definitions for calculating the tardiness of part j and the objective function, where the tardiness value of part j is the difference between the completion time of its affiliated batch and its due date. Constraint (15) is the definition of variables, and M is a sufficiently large constant.

4. Heuristic Method

In order to solve large-scale instances with higher efficiency, we propose a heuristic algorithm (Algorithm P) tailored to improve the solution efficiency by incorporating the processing time characteristics of additive manufacturing. Algorithm P incorporates the relevant research findings from xx and consists of two phases: sorting and allocation. It aims to obtain feasible solutions within a reasonable timeframe. The sorting phase of Algorithm P establishes clear part priority criteria, providing a decision foundation for the subsequent allocation phase. This achieves logical integration and efficient coordination between the two critical phases. It also provides a crucial procedural basis for the scheduling performance of Algorithm P.

The allocation phase is the core component of Algorithm P. It produces the final feasible scheduling plan by using the part priority sequence output by the sorting phase and taking into account the processing characteristics of additive manufacturing technology. The design quality of the rules directly determines the solution quality and efficiency of Algorithm P. Since we consider the two-dimensional packing constraints in this problem, Algorithm P incorporates auxiliary algorithms to determine both the batches for parts and their specific placement locations. The detailed rules of Algorithm P are as follows.

(1) Phase 1: Sorting. Since the objective is to minimize the maximum tardiness of all parts, we integrate the Earliest Due Date (EDD) rule into Algorithm P, which ensures that parts are sorted in ascending order of their due dates. As the number of parts increases, some parts are likely to have identical due dates. To address this practical challenge, Algorithm P refines the sorting rules based on the problem's specific characteristics. If due dates are the same, parts are sorted in ascending order of their estimated build times. We use the batch processing time calculation formula (Constraints (4)) to estimate the workload of each part. Prioritizing parts with smaller workloads enables faster capacity release, which helps reduce the risk of severe part tardiness in the scheduling scheme. If estimated processing times are also identical, parts are sorted in ascending order of their heights. When the heights of different parts are equal, Algorithm P sorts these parts by volume in ascending order. Such a multi-level sorting rule can effectively resolve the situation where multiple parts have identical feature parameters. The pseudocode of the sorting phase is shown in Algorithm 4.1.

Algorithm 4.1 Algorithm P: Sorting

for $j = 1$ to $|J|$ **do**
    Calculate the estimated build time PT_j for part j
end for
Sort the parts in ascending order of their due dates (d_j)
if $\forall\ i, j \in J\ (d_i = d_j)$ **then**
    Sort the parts in ascending order of their estimated processing times (PT_j)
    if $\forall\ i, j \in J\ (PT_i = PT_j)$ **then**
        Sort the parts in ascending order of their heights (h_j)
        if $\forall\ i, j \in J\ (h_i = h_j)$ **then**
            Sort the parts in ascending order of their volumes (v_j)
        end if
    end if
end if
Add the sequence of parts obtained according to the above sorting rules to the list of unassigned parts $List_{unassigned}$.

(2) Phase 2: Allocation. The allocation phase of Algorithm P takes maximizing the utilization of existing batches as its core principle and does not open new batches blindly. This phase evaluates each part sequentially based on the result of the sorting phase. For each candidate batch, it evaluates placing the part into the batch and calculates the maximum tardiness. The allocation phase selects the batch that produces the smallest maximum tardiness to place the part. The particular rules are described as follows.

Under the height constraint, Algorithm P uses Algorithm 4.3 to search available areas in all existing batches for each part. From all available areas that meet the physical constraints of the part, the one with the smallest area is selected as the candidate area for the part. Each existing batch generates at most one such candidate area. Algorithm P then calculates the current maximum tardiness after the part is placed into each candidate area.

If none of the existing batches on a machine can place the part, Algorithm P will create a new batch on this machine. This operation is only performed when the total batch number constraint is satisfied. Algorithm P then places the part into the new batch and calculates the current maximum tardiness.

Compare the solutions and the batch that produces the smallest current maximum tardiness is selected as the final placement for the part. If multiple candidate areas lead to the same value, the one with the smallest size is chosen. After determining the final placement, Algorithm P calls Algorithm 4.4 to remove the occupied available area. Based on the placement position of the part, new available areas are generated by cutting along the edges of the part.

Repeat the three steps until all parts have been allocates. The pseudocode of the allocation stage is shown in Algorithm 4.2.

Algorithm 4.2 Algorithm P: Allocation

for $j = 1$ to $|List_{unassigned}|$ **do**
    Initialization: List of exist batches $\leftarrow \emptyset$.
    for $m = 1$ to $|M|$ **do**
        for $b = 1$ to $|B|$ **do**
            Call : Algorithm 4.3 find the candidate area $A_{candidate}$ for part j.
            if $A_{candidate}$ exists **then**
                Call : Algorithm 4.4
                Calculate T_{max}
                List of exist batches $\leftarrow A_{candidate}$
            end if
        end for
    end for
    Initialization: List of new batches $\leftarrow \emptyset$.
    Calculate the total number of batches on all current machines num_b.
    if current total number of batches $num_b <$ total number of parts n **then**
        for the machine where the candidate area $A_{candidate}$ was not found **do**
            Create a new batch A_{new}
            if part j can be placed in A_{new} **then**
                Calculate T_{max}
                List of new batches $\leftarrow A_{new}$
            end if
        end for
    end if
    Combine List of exist batches and List of new batches as List of candidate batches.
    Select the batch from List of candidate batches that produces the smallest T_{max}
    if there exist multiple identical smallest T_{max} values **then**
        Choose the one with the smallest size
    end if
    Update T_{max}
end for

Algorithm 4.3 Candidate area search rule

Initialization: available area $\leftarrow$ the size of machine $(W * L)$
function CANDIDATE AREA SEARCH RULE(part j)
    From all available areas in existing batches, select area a where area width $\geq$ width of part j and area length $\geq$ length of part j.
    if multiple available areas a are found **then**
        Select the area with the smallest size as the candidate area $A_{candidate}$.
    end if
    Record the position information of $A_{candidate}$.
end function

Algorithm 4.4 2D placement rule

```
function 2D PLACEMENT RULE(part j)
    Get the position information of A_candidate.
    Place part j into A_candidate.
    Remove the A_candidate and update the list of available areas
end function
```

5. Computational Experiments

Machine parameters and the length, width, and height of the parts were selected from Che et al. (2021)[7]. According to Chergui et al. (2018)[12], part due dates were randomly generated from 8 hours to 72 hours for small-scale instances and from 8 hours to 160 hours for large-scale instances.

5.1. Model feasibility verification

Assume that 10 parts are to be processed on two additive manufacturing machines with identical parameters. The detailed parameters of the machines and parts are presented in Table 1 and Table 2, respectively. The computation time of the MILP models are limited to 3600 seconds.

Table 1: Machine parameters

Length (cm)	Width (cm)	Height (cm)	SET (h)	VT (cm^3 /h)	HT (cm/h)
25	25	32.5	2	0.030864	0.7

Table 2: Part parameters

Part ID	Due date	Volume	Height	Length	Width
1	10.0	60.0	6.0	5.0	4.0
2	9.0	80.0	6.0	5.0	5.5
3	17.0	50.0	4.0	8.0	3.0
4	8.0	75.0	5.0	8.0	8.0
5	29.0	15.0	1.0	6.0	3.0
6	49.0	70.0	7.0	3.0	5.0
7	28.0	10.5	3.8	2.5	2.0
8	64.0	50.0	4.0	11.0	3.0
9	13.0	50.0	4.0	8.0	3.0
10	51.0	60.0	3.0	7.0	7.0

Table 3: Computational result

Machine ID	Parts	Completion time	Processing time	Total volume	Batch height
1	4、9	9.36	9.36	125.0	5.0
1	5、10	15.77	6.41	75.0	3.0
1	8	22.12	6.34	50.0	4.0
1	7	27.10	4.98	10.5	3.8
2	1、2	10.52	10.52	140.0	6.0
2	3	16.86	6.34	50.0	4.0
2	6	25.92	9.06	70.0	7.0

Table 4: Tardiness of each part

Part ID	Completion time	Due date	Tardiness
1	10.52	10.00	0.52
2	10.52	9.00	1.52
3	16.86	17.00	0.00
4	9.36	8.00	1.36
5	15.77	29.00	0.00
6	25.92	49.00	0.00
7	27.10	28.00	0.00
8	22.12	64.00	0.00
9	9.36	13.00	0.00
10	15.77	51.00	0.00

Table 3 presents the part allocation scheme in detail. Table 4 illustrates in detail the tardiness of each part. As shown in Table 4, a total of 7 batches were initiated on the two machines for the scheduling of 10 parts. The Gurobi solver obtained the optimal solution of the problem within 2.1500s, with the optimal objective value of 1.52.

The experimental results clearly demonstrate that the batch height constraints, processing sequence constraints, and completion time constraints in the proposed model all exhibit effectiveness. These constraints ensure the stability and orderliness of the production process through their synergistic effect, fully verifying the feasibility of the model.

5.2. Comparison of results between the MILP Model and Algorithm P

Each instance is a combination of a given number of parts and machines. For example, P5M2 means scheduling 5 parts in 2 identical AM machines. Table 5 shows the comparison between the MILP model and Algorithm P in objective values and computation times on small-scale instances.

We can find that with the increasing of the number of parts and machines, the computational efficiency of the Gurobi solver significantly slows down. In contrast, Algorithm P achieves shorter computation times and a more stable growth trend, indicating a clear advantage in computational efficiency when solving instances of the same scale.

Table 5: Computational results of the MILP model and Algorithm P on small-scale instances

	Gurobi			Algorithm P		
	Objective value	Batches	Time (s)	Objective value	Batches	Time (s)
P5M2	0.00	1	0.0500	0.00	1	0.0070
P10M2	9.09	5	0.7900	9.09	2	0.0091
P10M3	0.00	6	1.0600	0.00	3	0.0101
P15M3	0.07	15	8.2500	5.29	3	0.0203
P15M4	0.00	9	3.8700	0.00	4	0.0242
P20M4	0.00	14	49.9600	1.63	6	0.0394
P25M5	13.18	19	160.1100	13.18	4	0.0580

It is not difficult to find that when the solutions of two methods are the same, the number of batches opened by Algorithm P is smaller than the Gurobi solver. For example, the number of batches generated by the Gurobi solver is 15 for instance P15M3. It means that each part is individually assigned to a batch. But for the same instance, Algorithm P produces only 3 batches, mainly because it tends to group parts that satisfy the two-dimensional placement constraints into the same batch whenever possible, thereby reducing idle space and improving overall resource utilization.

Table 6: Computational results of Algorithm P on large-scale instances

Instances	Objective value	Batches	Time (s)
P50M2	30.45	5	0.1257
P50M3	0.75	6	0.1486
P100M3	31.08	11	0.3904
P100M5	3.79	12	0.5467
P150M5	22.12	15	1.2621
P150M8	9.16	17	1.4492
P200M8	15.58	20	2.6132
P200M10	18.97	22	3.1994

Algorithm P is adopted to solve large-scale instances and the computational results are presented in Table 6. By comparing the computational results reported in Table 5 and Table 6, a clear performance difference between Algorithm P and the MILP formulation can be observed. Under a computation time of 0.7900s, MILP is only capable of effectively solving problem instances of P10M2. In contrast, Algorithm P can successfully handle instances with sizes reaching up to P100M5 in 0.5467s. It can be observed that Algorithm P is able to solve instances that are significantly larger and more complex than those handled by the MILP model, while requiring substantially less computational time. The results indicate that Algorithm P outperforms the MILP model in both efficiency and scalability.

In addition, as the problem scale increases, the computation time of the proposed algorithm grows at a relatively moderate rate. Even for larger instances, it is still able to produce feasible solutions within an acceptable time frame. These results clearly demonstrate the efficiency and practicality of the proposed approach, and provide solid experimental evidence supporting its applicability in real-world production scheduling scenarios.

6. Conclusion

In this paper, we examine a parallel-machine scheduling problem with two-dimensional packing constraints in additive manufacturing, and the objective of minimizing the maximum tardiness. To address this problem, a mixed-integer linear programming model is developed. Based on the process characteristics of additive manufacturing and the type of objective function, a heuristic algorithm (Algorithm P) is designed. Numerical experiments are conducted to evaluate and verify the effectiveness of the proposed model and algorithm, as well as the advantages of the algorithm in solution efficiency.

Future research may extend this work in several directions, such as formulating multi-objective optimization models and incorporating more complex and realistic production environments.

References

[1] Inayathullah S, Buddala R. Review of machine learning applications in additive manufacturing[J]. Results in Engineering, 2025, 25: 103676.

[2] Ghobakhloo M, Iranmanesh M, Foroughi B, et al. Industry 4.0 digital transformation and opportunities for supply chain resilience: a comprehensive review and a strategic roadmap[J]. Production planning & control, 2025, 36(1): 61-91.

[3] Singh S, Mohanty R P, Mangla S K, et al. Critical success factors of additive manufacturing for higher sustainable competitive advantage in supply chains[J]. Journal of Cleaner Production, 2023, 425: 138908.

[4] Kucukkoc I. MILP models to minimise makespan in additive manufacturing machine scheduling problems[J]. Computers & Operations Research, 2019, 105: 58-67.

[5] Li Q, Kucukkoc I, Zhang D Z. Production planning in additive manufacturing and 3D printing[J]. Computers & Operations Research, 2017, 83: 157-172.
[6] Dvorak F, Micali M, Mathieug M. Planning and scheduling in additive manufacturing[J]. Inteligencia Artificial, 2018, 21(62): 40-52.
[7] Che Y, Hu K, Zhang Z, et al. Machine scheduling with orientation selection and two-dimensional packing for additive manufacturing[J]. Computers & Operations Research, 2021, 130: 105245.
[8] Cakici E, Kucukkoc I, Akdemir M. Advanced constraint programming formulations for additive manufacturing machine scheduling problems[J]. Journal of the Operational Research Society, 2025, 76(3): 590-605.
[9] Dwivedi G, Chakraborty S, Agarwal Y K, et al. Simultaneous production and transportation problem: A case of additive manufacturing[J]. Transportation Science, 2023, 57(3): 741-755.
[10] Zehetner D, Gansterer M. The collaborative batching problem in multi-site additive manufacturing[J]. International Journal of Production Economics, 2022, 248: 108432.
[11] Kucukkoc I. Scheduling of distributed additive manufacturing machines considering carbon emissions[J]. An International Journal of Optimization and Control: Theories & Applications (IJOCTA), 2024, 14(1): 20-31.
[12] Chergui A, Hadj-Hamou K, Vignat F. Production scheduling and nesting in additive manufacturing[J]. Computers & Industrial Engineering, 2018, 126: 292-301.
[13] Zipfel B, Neufeld J, Buscher U. An iterated local search for customer order scheduling in additive manufacturing[J]. International Journal of Production Research, 2024, 62(3): 605-625.
[14] Aloui A, Hadj-Hamou K. A heuristic approach for a scheduling problem in additive manufacturing under technological constraints[J]. Computers & Industrial Engineering, 2021, 154: 107115.
[15] Nascimento P, Silva C, Mueller S, et al. Nesting and scheduling for additive manufacturing: an approach considering order due dates[C]//Congress of APDIO, the Portuguese Operational Research Society. Cham: Springer International Publishing, 2021: 117-128.
[16] Nascimento P J, Silva C, Antunes C H, et al. Optimal decomposition approach for solving large nesting and scheduling problems of additive manufacturing systems[J]. European Journal of Operational Research, 2024, 317(1): 92-110.
[17] Kapadia M S, Starly B, Thomas A, et al. Impact of scheduling policies on the performance of an additive manufacturing production system[J]. Procedia Manufacturing, 2019, 39: 447-456.