



# Customer Order Scheduling in an Additive Manufacturing Environment

Benedikt Zipfel, Janis S. Neufeld, Udo Buscher

## ► To cite this version:

Benedikt Zipfel, Janis S. Neufeld, Udo Buscher. Customer Order Scheduling in an Additive Manufacturing Environment. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2021, Nantes, France. pp.101-109, 10.1007/978-3-030-85910-7\_11 . hal-03806477

**HAL Id: hal-03806477**

**<https://inria.hal.science/hal-03806477>**

Submitted on 7 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# Customer Order Scheduling in an Additive Manufacturing Environment

Benedikt Zipfel, Janis S. Neufeld, and Udo Buscher

Chair of Business Management, esp. Industrial Management,  
TU Dresden, 01069 Dresden, Germany  
`benedikt.zipfel@tu-dresden.de`  
`janis.sebastian.neufeld@tu-dresden.de`  
`udo.buscher@tu-dresden.de`

**Abstract.** This paper investigates the customer order scheduling problem on unrelated parallel additive manufacturing machines. The discussed problem comprises the splitting of orders into jobs, the allocation of those jobs to builds and finally the sequencing of builds on 3D printers. A mixed-integer programming model is presented that integrates practical requirements, such as printing profiles and different materials, and minimises total weighted tardiness. Using the GUROBI solver computational results are then given for a comprehensive test bed. It is shown, that medium sized problems can be solved using the proposed model, and that the consideration of printing profiles has a relevant impact on the scheduling task in additive manufacturing.

**Keywords:** Additive manufacturing · Customer order scheduling · Unrelated parallel machines.

## 1 Introduction and Related Work

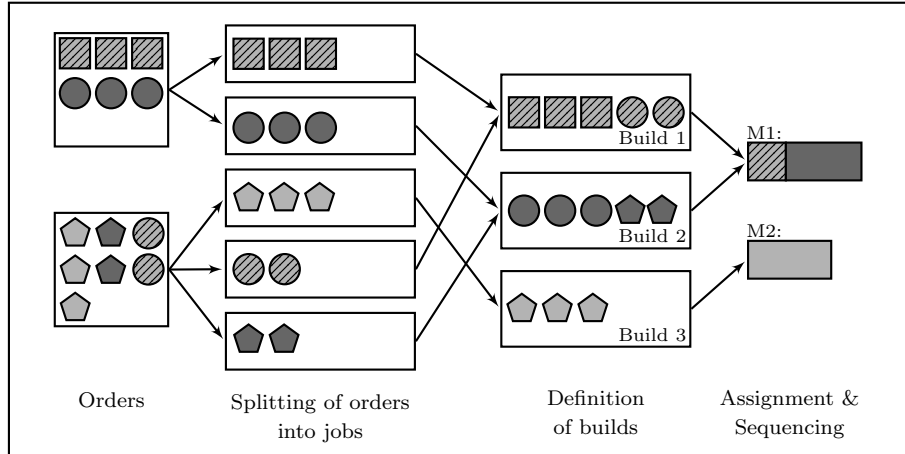
Technologies of additive manufacturing (AM) have become the foundation of new on-demand services, such as Factory-as-a-Service and Production-as-a-Service [4]. In the presence of growing amounts of heterogeneous products, that have to be produced in AM facilities, efficient planning and operations management for AM are essential. Due to the customer orientation of the newly emerged on-demand services, we propose a model for customer order scheduling in AM, where multiple types of 3D printers are available. Jobs can be manufactured with different printing profiles, which have an effect on the speed and quality of the production process. This extension to existing scheduling problems for AM is also derived from the customer oriented perspective of additive manufacturing offering a wide palette of materials and various quality levels to their customers. At the same time, it leads to incompatibilities since not all products can be manufactured together. Hence, the main contribution of this paper is the integration of customer orientation into the scheduling problem for AM taking into account new variability regarding materials and quality. To the best of our knowledge, this problem configuration has not been addressed before. Due to their leading

role in industrial applications of metallic AM, we focus on the technologies selective laser melting (SLM) and direct metal laser sintering (DMLS)[3]. The considered problem combines several aspects of scheduling. Firstly, it comprises the issue of scheduling customer orders, where optimisation objectives shift from a job focus to an order focus. While [1] introduce the problem for a single machine configuration with different job families, [5] recently study order scheduling with the objective to minimise the total weighted completion times of orders for a parallel machine configuration. Secondly, as 3D printers are able to process several products at the same time, the problem can be classified as a batch processing machine scheduling problem (for relevant studies see, e.g. [6, 8]). Finally, a growing amount of recent publications can be found focusing on the problems of nesting and scheduling in AM. In [2] three models are presented dealing with the problem of scheduling for AM in single, identical and non-identical parallel machines environments. [7] introduce a model formulation with two objectives of minimising makespan as well as total tardiness and introduce predefined material types. With our model we will expand the perspective of scheduling for AM by combining the stated aspects addressed by previous research, extending the scope by printing profiles and proposing an integrated model formulation.

The remainder of this paper is organized as follows. In Section 2 we define the problem and present a mixed-integer programming (MIP) model. Section 3 describes a computational study with 60 randomly generated instances, while Section 4 concludes with prospects of future research.

## 2 Problem Formulation

Before describing the problem, let us give a brief explanation of the examined context. An additive manufacturing company is offering an on-demand 3D print-



**Fig. 1.** Schematic description of the optimisation problem

ing service to its customers. To process the ordered products, the manufacturer's facility is equipped with several different SLM/DMLS printers. Each product must be made from a specific material  $w \in W$  and can be produced in different levels of quality. According to the requested quality a printing profile  $d \in D$  must be applied to available 3D printers. Each printer  $k \in M$  has a restricted production height, production area and is capable of processing an individual subset of all possible materials. Let a customer order  $i \in I$  consist of a subset of products in the manufacturers portfolio with defined quantities, materials, and the desired quality levels.

In this context, the considered problem is illustrated in Figure 1 by an example with two orders. Herein, the shapes indicate different types of products, while the fill colors of the products symbolize the materials. The existence of patterns on the products implies different quality levels, and hence, the need of different printing profiles. An order  $i$  can be split into several jobs  $j \in J_i$ , each of which consists of a defined quantity of products with identical product type, quality level, and material requirement. The described splitting is represented in the second stage of Figure 1. The objective is to assign all resulting jobs to builds with respect to material and quality similarity, while the splitting of jobs into different builds is allowed. Furthermore, the builds have to be allocated and sequenced on the available printers according to the machine specifications with the objective of minimising the total weighted tardiness of all orders,  $\sum_i w_i T_i$ . These sub problems are illustrated with the third and fourth stage of Figure 1. Due to the focus on SLM/DMLS technology, we allow products only to be placed on the build plate of a printer, i.e., stacking of products within the three-dimensional build space is prohibited. To ensure a valid nesting of products into builds, projections including tolerances are used for the production area of a product type [2]. The notations used for the model are as follows:

---

**Sets**


---

|       |                              |          |  |
|-------|------------------------------|----------|--|
| $B$   | Set of builds                | $S^J$    | Set of incompatible jobs                                   |
| $D$   | Set of profiles              | $S^M$    | Set of incompatible combinations of jobs and machines      |
| $D_j$ | Set of profiles of a job $j$ | $S^{MW}$ | Set of incompatible combinations of machines and materials |
| $I$   | Set of orders                | $S^W$    | Set of incompatible combinations of jobs and materials     |
| $J$   | Set of jobs                  | $S^D$    | Set of incompatible combinations of jobs and profiles      |
| $J_i$ | Set of jobs of an order $i$  | $W$      | Set of materials   |
| $M$   | Set of machines              |          |  |

---

**Parameters**


---

|       |  |           |   |
|-------|--|-----------|---|
| $a_j$ | Production area of the product type of job $j$ | $H_k$     | Maximum production height of machine $k$                            |
| $h_j$ | Height of the product type of job $j$          | $HT_{kd}$ | Processing time for one height unit by machine $k$ with profile $d$ |

|       |   |                 |  |
|-------|---|-----------------|--|
| $v_j$ | Volume of the product type of job $j$                 | $RT_k$          | Removal time of processed build on machine $k$   |
| $d_i$ | Due date of order $i$                                 | $ST_k$          | Setup time on machine $k$ without changing materials   |
| $q_j$ | Amount of products of job $j$                         | $ST_k^w$        | Setup time on machine $k$ with changing materials  |
| $w_i$ | Penalty costs for delay of one time unit of order $i$ | $VT_{kd}$       | Processing time to build one volume unit by machine $k$ with profile $d$   |
| $A_k$ | Maximum production area of machine $k$                | $\gamma_{bvcw}$ | Binary matrix to indicate, if setup between build $b$ with material $v$ and build $c$ with material $w$ is necessary |

$G$  Big Number

---

### Variables

---

|            |   |            |  |
|------------|---|------------|--|
| $p_{bk}$   | Processing time of build $b$ on machine $k$   | $CB_{bk}$  | Completion time of build $b$ on machine $k$                    |
| $x_{jbk}$  | Amount of items of job $j$ in build $b$ on machine $k$                                      | $CI_i$     | Completion time of order $i$                                   |
| $y_{bkwd}$ | 1, if build $b$ is scheduled on machine $k$ with material $w$ and profile $d$ , 0 otherwise | $HB_{bkd}$ | Production height of build $b$ on machine $k$ with profile $d$ |
| $z_{jbk}$  | 1, if items of job $j$ are processed in build $b$ on machine $k$ , 0 otherwise              | $T_i$      | Tardiness of part $i$  |

---

The studied problem can be classified as an unrelated parallel machine scheduling and nesting problem with batching and incompatible job families. The objective is to minimise the total weighted tardiness of all orders. Inspired by existing models from [2] and [7] we formulate the described problem as a MIP model.

$$\sum_{i \in I} w_i \cdot T_i \rightarrow \min \quad (1)$$

s.t.:

$$T_i \geq CI_i - d_i \quad \forall i \in I \quad (2)$$

$$CI_i \geq CB_{bk} - G \cdot (1 - z_{jbk}) \quad \forall i \in I, j \in J_i, b \in B, k \in M \quad (3)$$

$$\sum_{b \in B} \sum_{k \in M} x_{jbk} = q_j \quad \forall j \in J \quad (4)$$

$$x_{jbk} \leq q_j \cdot z_{jbk} \quad \forall j \in J, b \in B, k \in M \quad (5)$$

$$x_{jbk} \geq z_{jbk} \quad \forall j \in J, b \in B, k \in M \quad (6)$$

$$\sum_{j \in J} (a_j \cdot x_{jbk}) \leq A_k \quad \forall b \in B, k \in M \quad (7)$$

$$z_{jbk} \cdot h_j \leq HB_{bkd} \quad \forall j \in J, d \in D_j \quad (8)$$

$$b \in B, k \in M$$

$$HB_{bkd} \leq H_k \quad \forall d \in D, b \in B, \quad (9)$$

$$k \in M$$

$$\sum_{j \in J} z_{jb+1,k} \leq G \cdot \sum_{j \in J} z_{jbk} \quad \forall b \in B \mid b < |B|, k \in M \quad (10)$$

$$z_{jbk} + z_{lbk} \leq 1 \quad \forall (j, l) \in S^J \mid j < l, \quad (11)$$

$$b \in B, k \in M$$

$$z_{jbk} = 0 \quad \forall (j, k) \in S^M, b \in B \quad (12)$$

$$\sum_{w \in W} \sum_{d \in D} y_{bkwd} \leq 1 \quad \forall b \in B, k \in M \quad (13)$$

$$z_{jbk} \leq y_{bkwd} \quad \forall b \in B, d \in D, j \in J, \quad (14)$$

$$k \in M, w \in W,$$

$$(w, j) \notin S^W, (d, j) \notin S^D$$

$$y_{bkwd} \leq 1 \quad \forall b \in B, d \in D, k \in M, \quad (15)$$

$$w \in W, (w, k) \notin S^{MW}$$

$$CB_{bk} \geq \sum_{w \in W} \sum_{d \in D} y_{bkwd} \cdot ST_k^w + p_{bk} \quad \forall b \in B, k \in M \quad (16)$$

$$CB_{bk} \geq CB_{b-1,k} + p_{bk} + ST_k^w \cdot \gamma_{b-1,vbw} \quad \forall b \in B \mid b > 0, k \in M, \quad (17)$$

$$+ ST_k \cdot (1 - \gamma_{b-1,vbw})$$

$$- G \cdot (2 - y_{b-1,kvd} - y_{bkwe})$$

$$w, v \in W, d, e \in D$$

$$p_{bk} = \sum_{d \in D} (HB_{bkd} \cdot HT_{kd})$$

$$+ \sum_{j \in J} \sum_{d \in D_j} (x_{jbk} \cdot v_j \cdot VT_{kd}) + RT_k \quad \forall b \in B, k \in M \quad (18)$$

$$x_{jbk} \in \mathbb{N}_0, y_{bkwd} \in \{0, 1\}, z_{jbk} \in \{0, 1\} \quad \forall j \in J, b \in B, k \in M, \quad (19)$$

$$w \in W, d \in D$$

On the basis of a customer oriented scheduling, we consider (1) as the objective function minimising the total weighted tardiness of all orders. (2) computes the tardiness of an order by subtracting the due date from the completion time of the order. By introducing (3) the completion time of an order  $i$  is at least equal to the completion time of a build  $b$  on machine  $k$  if the specific build processes jobs of the respective order. Constraints (4) ensure that the requested order quantity of each job is produced within all scheduled builds. (5) and (6) connect the binary variable  $z_{jbk}$  with its integer equivalent  $x_{jbk}$ . Furthermore, we adopt the machine restrictions introduced by [2] and adjust them for our purpose in (7), (8) and (9). (7) ensures that the builds fit into the production

area of the available printers, while (8) and (9) guarantee that the height of the build does not exceed the maximum production height of the machine. Note that we are not considering a constraint to check for a feasible production volume since the production area of a product is considered as a rectangular projection of the product onto the printing platform as in [2]. With (10) we verify that the builds are filled incrementally one after another. To avoid incompatible job combinations in builds we use (11). The equation allows only one of the jobs  $j$  and  $l$  to be in a build  $b$  on machine  $k$  if the tuple  $(j, l)$  is in the set  $S^J$ . Constraints (12) apply the same logic to incompatible combinations of jobs and machines. With equation (13) we specify that a build  $b$  on machine  $k$  can only be processed using one material  $w$  and only one printing profile  $d$ . Constraints (14) define the relation between decision variables  $z_{jbk}$  and  $y_{bkwd}$  using sets of incompatibility. We denote with (15) that a build  $b$  on machine  $k$  can only be manufactured if the machine is capable of processing the required material  $w$ , i.e., the tuple  $(w, k)$  is not in the incompatibility set. Inequalities (16) guarantee that the completion time of a build on a certain machine is greater than or equal to the sum of the setup time for a new material and its processing time. For a batch which has at least one predecessor (17) ensures that the completion time is greater than or equal to the sum of the completion time of the predecessor, its own processing time and a required setup time. The latter depends on whether there is a material change between two consecutive builds or not. This is controlled by a binary matrix  $\gamma_{bvcw}$  indicating if a setup with material change is necessary. In addition, we define the processing time of a build as the sum of the time needed to print the height of the build, the printing time for the volume of a build and the time to remove the build from the machine in (18). Finally, (19) defines the ranges of the decision variables.

### 3 Computational Study

The proposed model has been implemented with GUROBI using the provided PYTHON API. The study has been conducted on an Intel(R) Xenon(R) CPU E5-2630 v2 with 2.6 GHz clock speed and 384 GB RAM. For solving the model the maximum number of threads was set to four and a time limit of 1800 seconds was chosen. Table 2 describes the structure of the test instances. Six configurations are defined to study the impact of variations concerning materials and profiles as well as the impact of growing problem size. For the first three configurations *config1* to *config3* we set the number of available materials and printing profiles to three and two, respectively. For *config4* to *config6* these values are increased to five materials and three profiles. For both categories, we choose different numbers of investigated orders, machines and product types. Based on the data set P62 from [2], which provides information for 46 different product types and three unrelated machines, for each test instance the respective number of product types and machines is randomly picked. For each order the number of different products is set between one and 60 percent of the total number of product types, while the quantities for each product within an order are randomly



chosen between one and five. The resulting average number of jobs and products for each configuration that have to be assigned to builds are also displayed in Table 2. As can be seen from this summary the defined structure yields in a test design reaching from small to reasonable problem sizes, i.e., on average from 15.9 products (*config1*) up to 76.0 products (*config6*) and a maximum of five parallel machines. These instances can represent practical scheduling tasks of small on-demand 3D printing service providers.

|                        | <i>config1</i> | <i>config2</i> | <i>config3</i> | <i>config4</i> | <i>config5</i> | <i>config6</i> |
|------------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Total instances        | 10             | 10             | 10             | 10             | 10             | 10             |
| Total orders $ I $     | 3              | 5              | 7              | 3              | 5              | 7              |
| Total machines $ M $   | 2              | 3              | 5              | 2              | 3              | 5              |
| Total product types    | 5              | 7              | 10             | 5              | 7              | 10             |
| Total profiles $ D $   | 2              | 2              | 2              | 3              | 3              | 3              |
| Total materials $ W $  | 3              | 3              | 3              | 5              | 5              | 5              |
| Average total jobs     | 5.7            | 15.4           | 22.4           | 6.6            | 14.8           | 25.2           |
| Average total products | 15.9           | 45.2           | 66.8           | 18.3           | 42.2           | 76.0           |

**Table 2.** Design of problem instances

A major factor influencing the computational performance of the proposed model is the upper bound of possible builds  $|B|$ . As noted by [2], the value must be high enough to ensure that better feasible solutions will not be cut off. In contrast, a large value for  $|B|$  results in a much longer computation time. To overcome this issue, instead of setting  $|B| = \sum_{j \in J} q_j$ , we compute the minimum number of builds to fit in all ordered products by solving a simple bin packing problem in preprocessing. We then use this result as a basis for our tests by iteratively solving the problem instance and increasing  $|B|$  until no better solution can be found. This method gradually increases the slackness of the model allowing more builds to be empty.

|   | $\varnothing B $ | # opt. sol. | $\varnothing$ MIPGap (%) | $\varnothing$ Runtime (s) |
|---|------------------|-------------|--------------------------|---------------------------|
| <i>config1</i>                            | 3.6              | 10 / 10     | 0.0                      | 3.6                       |
| <i>config2</i>                            | 7.6              | 5 / 10      | 38.6                     | 904.1                     |
| <i>config2<sub>opt</sub></i> <sup>a</sup> | 6.0              | 5           | 0.0                      | 8.2                       |
| <i>config3</i>                            | 10.9             | 2 / 10      | 35.3                     | 1446.0                    |
| <i>config3<sub>opt</sub></i> <sup>a</sup> | 9.0              | 2           | 0.0                      | 30.0                      |
| <i>config4</i>                            | 4.7              | 10 / 10     | 0.0                      | 0.5                       |
| <i>config5</i>                            | 8.6              | 4 / 10      | 44.4                     | 1190.1                    |
| <i>config5<sub>opt</sub></i> <sup>a</sup> | 7.0              | 4           | 0.0                      | 275.3                     |
| <i>config6</i>                            | 9.1              | 2 / 10      | 53.5                     | 1569.9                    |
| <i>config6<sub>opt</sub></i> <sup>a</sup> | 7.5              | 2           | 0.0                      | 649.2                     |

<sup>a</sup> Average values only for instances from those configurations that have been solved to optimality.

**Table 3.** Results of the computational study

The results of the computational study are illustrated in Table 3. The first column of the table shows the average number of builds on each available machine for the optimal (or best found) solution, while the second one denotes the number of instances solved to optimality. The third and fourth column outline the average values of each configuration for MIP gap and runtime of the solver, respectively. All instances of *config1* and *config4* can be solved to optimality, while the optimum solution can be found only for about half of the instances with intermediate problem size (*config2* and *config5*). In each of *config3* and *config6* only two instances can be solved to optimality. Instances of reasonable size with up to 27 jobs and 88 products could be solved to optimality. However, quite high MIP gaps exist for instances, where no optimal solution was found within the time limit. This indicates that the solution is highly dependent on the specific characteristics of the considered test problem. With these results, the presented model can already provide helpful decision support for small 3D printing manufacturers. However, for larger real-world problems a heuristic approach would be necessary to ensure high-quality solutions.

Furthermore, the results show a growing computational effort with the increase of variability in material and printing profile options. The gaps to the known lower bounds as well as the runtime of the solver are reasonable lower in *config2* and *config3* (low number of printing profiles and materials) in comparison with *config5* and *config6* (higher number of printing profiles and materials). This shows that the integration of the practical requirements of printing profiles and materials is relevant for the scheduling in AM.

## 4 Conclusion

This paper extended the research on scheduling for AM by combining the premises of order scheduling with the existing model formulations with the aim of minimising total weighted tardiness of customer orders. The problem has been considered in an unrelated parallel machine environment, formulated as MIP model and explored in an computational study with various characteristics. The numerical results in this paper demonstrate the complexity and difficulty of customer order scheduling in an AM environment and the relevance of considering printing profiles and materials in the scheduling. For small to medium problem instances the MIP model can be used by companies to generate optimal schedules. The findings for larger problems demonstrate the need of heuristic algorithms to solve such problem instances efficiently. Apart from this, the proposed model can be adjusted to integrate product geometries by extending the nesting of products from a one dimensional packing problem to a more precise procedure evaluating two or even three dimensions of the products.

## References

1. Gupta, J.N., Ho, J.C., van der Veen, J.A.: Single machine hierarchical scheduling with customer orders and multiple job classes. *Annals of Operations Research* **70**, 127–143 (1997). <https://doi.org/10.1023/A:1018913902852>

2. Kucukkoc, I.: MILP models to minimise makespan in additive manufacturing machine scheduling problems. *Computers & Operations Research* **105**, 58–67 (2019). <https://doi.org/10.1016/j.cor.2019.01.006>
3. Li, Q., Kucukkoc, I., Zhang, D.Z.: Production planning in additive manufacturing and 3D printing. *Computers & Operations Research* **83**, 157–172 (2017). <https://doi.org/10.1016/j.cor.2017.01.013>
4. Oh, Y., Witherell, P., Lu, Y., Sprock, T.: Nesting and Scheduling Problems for Additive Manufacturing: A Taxonomy and Review. *Additive Manufacturing* **36**, 101492 (2020). <https://doi.org/10.1016/j.addma.2020.101492>
5. Shi, Z., Wang, L., Liu, P., Shi, L.: Minimizing Completion Time for Order Scheduling: Formulation and Heuristic Algorithm. *IEEE Transactions on Automation Science and Engineering* **14**(4), 1558–1569 (2017). <https://doi.org/10.1109/TASE.2015.2456131>
6. Shi, Z., Huang, Z., Shi, L.: Customer order scheduling on batch processing machines with incompatible job families. *International Journal of Production Research* **56**(1-2), 795–808 (2018). <https://doi.org/10.1080/00207543.2017.1401247>
7. Tavakkoli-Moghaddam, R., Shirazian, S., Vahedi-Nouri, B.: A Bi-objective Scheduling Model for Additive Manufacturing with Multiple Materials and Sequence-Dependent Setup Time. In: Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D. (eds.) *Advances in Production Management Systems. Towards Smart and Digital Manufacturing*. pp. 451–459. IFIP Advances in Information and Communication Technology, Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-57997-5\\_52](https://doi.org/10.1007/978-3-030-57997-5_52)
8. Zhang, C., Shi, Z., Huang, Z., Wu, Y., Shi, L.: Flow shop scheduling with a batch processor and limited buffer. *International Journal of Production Research* **55**(11), 3217–3233 (2017). <https://doi.org/10.1080/00207543.2016.1268730>