Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

AUTOMATING THE PRODUCTION PLANNING OF A 3D PRINTING FACTORY

Johan P.N. Freens
Ivo J.B.F. Adan
Alexander Yu. Pogromsky

Eindhoven University of Technology
PO Box 513
Eindhoven, 5600 MB, THE NETHERLANDS

Hugo Ploegmakers

Shapeways
Kanaaldijk-zuid 3a
5613 LE Eindhoven, THE NETHERLANDS

ABSTRACT

To increase a 3D printer's throughput and decrease the print objects' lead times, composing good batches for 3D printers in high-volume 3D printing environments is of great importance. Since manual planners cannot oversee the whole production planning, they tend to make sub-optimal decisions. This paper presents a two-stage procedure that automatically generates batches for multiple 3D printers. The first stage consists of allocating objects to a batch. This problem is an extension of the classical one-dimensional bin packing problem by including lateness and object size mixture requirements. In the second stage, the positions of the print objects in a tray are determined by using third party three-dimensional packing software. By simulating the production planning of Shapeways, a popular 3D printing marketplace, model parameters are calibrated and the performance of the two-stage procedure is evaluated. The results show an increase of the throughput of the 3D-printers by approximately 10% in comparison with manually created batches.

1 INTRODUCTION

Additive manufacturing, or 3D printing, is the process of making three-dimensional objects from a 3D model of virtually any shape. Additive manufacturing systems have been available for industrial use since the 1980s and are widely used for industrial applications, such as rapid prototyping (Lipson and Kurman 2013). With the increase of the computational power in personal computers and the development of new 3D computer graphics software, 3D printing is currently becoming more accessible for the general public. This can be seen in the popularity of relatively cheap consumer type printers and the emergence of 3D printing marketplaces, such as Shapeways.

These 3D printing marketplaces are websites, where users can buy and sell 3D printable files. In other words, people with the ability to design 3D models are able to upload and print their models, which thus allows them the opportunity to sell their models. With the allocation of the purchased objects to one of the many printers that are utilized by these marketplaces, a scheduling problem arises that has not yet been adequately discussed in the current production scheduling literature. Currently these production schedules are made by human planners. The kind and number of objects that are selected, is based on the experience and intuition of the planner. After the planners have selected a batch of objects, the batch is packed by 3D bin packing software before the actual printing can take place. Since the planners cannot oversee the whole production schedule, they tend to make sub-optimal decisions. This makes automating this process very promising. The purpose of this study is to explore the feasibility of automating the production schedule of a network of additive manufacturing systems. An important aspect to explore is whether this automated process is able to compose batches that have a high print density. This will decrease the amount of unused space in the 3D printers, and thus increase throughput and decrease operational cost. At the same time, the due date of orders will also have to be taken into consideration when generating production schedules.

In this study, a method is presented to determine the production schedule for an environment of multiple, high-volume 3D printers. In a more mathematical sense, this method attempts to approximate and to break down a large, intractable 3D bin packing problem into multiple smaller and manageable 3D bin packing problems. The feasibility of automating an entire production schedule and the potential advantages of implementing this method are explored through a simulation study of one-month production data of a 3D printing factory of Shapeways.

High volume 3D printing environments mainly use selective laser sintering (SLS) 3D printers, since these printers have a large building envelope, high throughput, high resolution and need no support structures (Kruth et al. 2003). The current paper is written with SLS printers in mind.

This paper is organized as follows. In Section 2 we briefly describe related work. The translation of the production planning problem to a mathematical problem is presented in Section 3. The performance of the mathematical approach is investigated by means of a simulation study in Section 4. This simulation study is also used for parameter calibration. Finally, Section 5 summarizes our findings.

2 RELATED WORK

In this section we briefly present some related work in order to better understand the mathematical formulation of the production scheduling problem of a 3D printing factory, as described in Section 3.

First, the classical bin packing problem is introduced, and formulated as an integer linear programming problem (ILP). This problem is the starting point of the mathematical formulation of the production scheduling problem. Second, the concept and difficulties of three-dimensional packing are discussed.

2.1 CLASSICAL BIN PACKING PROBLEM

In the classical bin packing problem, a set of items, each with a certain volume, are allocated to a set of bins, each of which has a certain bounded volume capacity. The number of bins, used to pack all the items, has to be minimized. This can be translated to the following mathematical programming formulation (Kantorovich 1960):

$$\text{Minimize} \quad \sum_{j=1}^{m} u_j C_j \tag{1}$$

$$\text{Subject to} \quad \sum_{j=1}^{m} x_{ij} = 1, \qquad i \in N = \{1,\ldots,n\}, \tag{2}$$

$$\sum_{i=1}^{n} v_i x_{ij} \leq u_j V_j^{max}, \qquad j \in M = \{1,\ldots,m\}, \tag{3}$$

$$u_j, x_{ij} \in \{0,1\},$$

where n is the number of items to be packed, m is an upper estimate of the number of bins needed, $u_j = 1$ if bin j is used, and 0 otherwise, and $x_{ij} = 1$ if item i is assigned to bin j, and 0 otherwise. The volume of item i is v_i and the volume of bin j is V_j^{max}. Constraint (2) ensures that all items are packed, and (3) limits the capacity of each bin. The cost of using bin j is denoted by C_j.

Moderately sized one-dimensional packing problems can easily be solved with branch-and-bound algorithms (Land and Doig 1960) and with standard ILP solvers, e.g., Coin-cbc (Forrest and Lougee-Heimer 2005). In the next section we discuss the generalization of one-dimensional packing problems to multiple dimensions.

2.2 PACKING ARBITRARILY SHAPED THREE-DIMENSIONAL OBJECTS

In the multiple-dimensional bin packing problem, an item is not only allocated to a bin, but additionally, also a position in the bin is assigned to the item. The generalization to multiple packing dimensions greatly increases the (computational) complexity of the problem. Nevertheless, with exact branch-and-bound algorithms, this problem can be solved within a reasonable time limit, that is, for a limited number of rectangular-shaped items with fixed orientations (Martello, Pisinger, and Vigo 2000). In the 3D printing industry, however, items are not rectangular-shaped, and objects can, in the case of SLS-printers, be freely orientated around their vertical axis. This makes the problem intractable, and therefore, no optimal solution can be found anymore.

Packing arbitrarily shaped three-dimensional objects without a fixed orientation is studied most notably by Jia and Williams (2001) and Gan et al. (2004). They present a heuristic packing method that works by simulating gravity and object collision. This packing technique is commercially available under the name Digipac-3D; this program is, unfortunately, not available for academic use and is not used in this paper.

Wu et al. (2014) present a packing method that is especially intended for 3D-printing. They first determine the object orientation and then use a modified bottom-left-front packing heuristic to find a near optimal position for the objects in a single rectangular bin. However, they do not deliver a usable software package that implements their method.

In this paper, the program Netfabb is used for three-dimensionally packing print objects. Netfabb is a program that is widely used for preparing 3D-files for 3D printing. It works by loading multiple 3D printable objects and packing these to a minimum amount of space. Although no documentation of the Netfabb's packing algorithm is available, it is probably a so-called random placement algorithm. This is an algorithm where all objects, that need to be packed, are placed in the packing space one by one in randomly selected positions. Netfabb's packing algorithm is known to perform mediocre (Freens 2015). When packing spheres, for example, the maximum density that is reached is 0.46 (i.e., 62 percent of the optimal sphere packing density). However, since no other packing programs are currently available that can easily be used for 3D print objects, Netfabb is adopted in the current paper.

3 MODEL DESCRIPTION

This section explains the production planning process of a 3D printing factory and indicates which attributes need to be considered when automating this process. Subsequently, the production planning problem is translated into a mathematical problem in Section 3.2.

3.1 DESCRIPTION OF THE MANUAL PRODUCTION PLANNING

When a 3D print object is ordered, it is checked for printability and the print orientation is set. The print object will then enter a buffer from where it can be assigned to a batch, which is a group of objects that is to be printed by a specific printer. Before the objects in the buffer are allocated to a batch, the buffer content is checked. If insufficient object volume is present in the buffer, one or more printers are left idle and fewer batches will be needed. The printers that will be utilized, are now individually planned. This planning process is visualized in Figure 1, and is further explained below.

Once the printers are selected, first their batch sizes need to be determined. In industrial SLS 3D printers, e.g., in EOS P100 and P760 printers, a full batch is a print job of approximately two days. Since daily production planning is considered in this paper, choosing the batch size effectively means choosing between a one-day job or a two-day job.

Since larger batch sizes will result in taller printer trays, a planner will choose for a two-day batch if larger quantities of large objects are in the buffer that do not fit in a small tray. However, smaller batches are preferred, since this decreases the lead time of all objects in the batch. Furthermore, the material in an SLS 3D printer deteriorates while being in a printer, so in smaller batches more unsintered material can be reused.

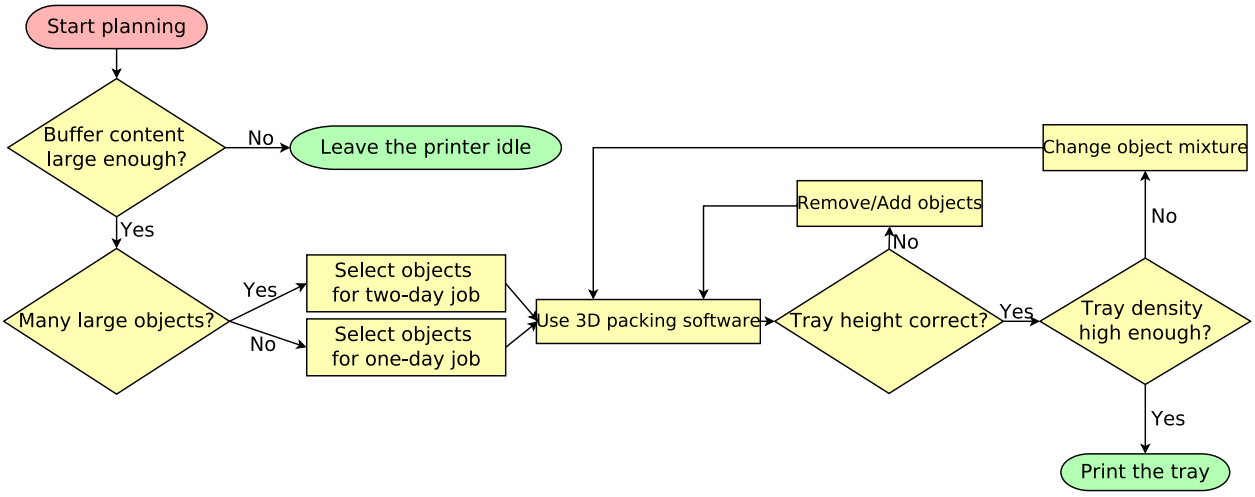


Figure 1: Manual production planning for a single printer.

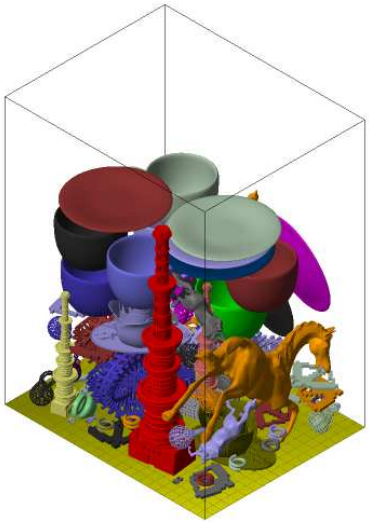


Figure 2: Batch of objects in a tray that is ready to be printed.

After the batch size is set, i.e., a one-day job or a two-day job, the actual allocation of objects to this batch is initiated. Objects close to their due date have priority and are chosen first. Subsequently, this group of objects is then fitted into a tray with 3D packing software. This software minimizes the used space in a printer tray by three-dimensionally packing the selected objects. Figure 2 displays an example of a batch of objects packed into a tray. Depending on the number of objects, running a packing algorithm typically takes ten to thirty minutes.

If the tray does not yet have the desired height after packing, objects are added or removed, and if the tray density is insufficient, large objects are removed and smaller objects are added. This last step is necessary, since large objects split a tray into smaller compartments that can contain small objects only.

After the batch composition is adjusted, the packing algorithm is executed again. How often this procedure is repeated, depends on the planner. It can take between five and ten iterations before the desired batch height and density is reached. This manual procedure of composing a printer tray takes approximately three hours. After the final trays are generated, they are ready to print.

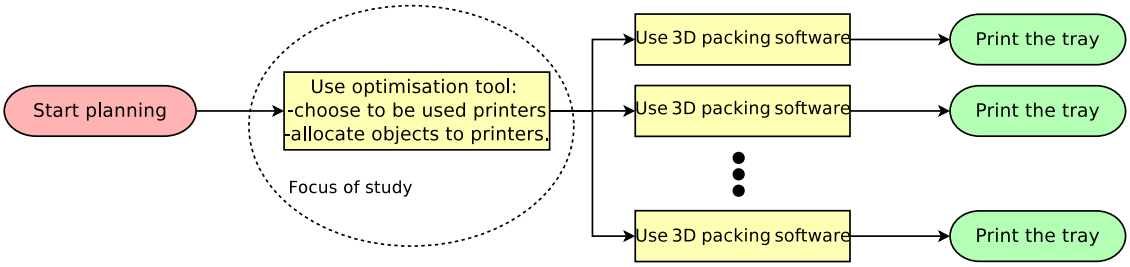


Figure 3: Automated production planning for multiple printers.

After a tray is printed, a printer becomes available again for the next planning cycle. The printed objects will have to be cooled down before they can be further processed. This can take up to one to four days for small and large batches, respectively.

3.2 MATHEMATICAL MODEL

In this section the planning problem described in the previous section is translated into an integer linear programming problem. A solution to this kind of problems can be obtained with standard ILP solvers.

The solution to the problem modeled in this section, prescribes which objects need to be allocated to which batch. Subsequently, these objects can be packed with 3D packing software to create the actual tray. The automated production planning will thus work for multiple printers at the same time, without the need to iterate as done for manually generated production schedules. The automated planning procedure is visualized in Figure 3.

First, the basic optimization problem is presented, which is then extended to include the main aspects of the planning problem of a 3D printing factory.

3.2.1 ADJUSTED VARIABLE SIZE BIN PACKING PROBLEM

In essence the production planning problem is based on the aforementioned one-dimensional size bin packing problem in Section 2.1. In the problem discussed here, the items are called objects and the bins are called batches or trays. Additionally, constraint (2) which ensures that all objects are allocated, is removed. To provide an incentive to still pack objects, costs are added for objects that are not packed, the so-called object left-over cost. An optimization problem is now formulated where the machine utilization costs (C_j^{tray}) are balanced to the left-overs cost (C^{left}). The cost function now is:

$$\sum_{j=1}^{m} u_j C_j^{tray} + \sum_{i=1}^{n} a_i v_i C^{left},$$

where set $N = \{1,...,n\}$ contains all objects that are to be allocated to a tray, and set $M = \{1,...,m\}$ contains all available trays. The variable u_j is 1 if tray j is used, otherwise it is zero, and the variable a_i is 1 if object i is left in the buffer and 0 if it is allocated to a tray. The variable a_i is set by:

$$a_i = 1 - \sum_{j=1}^{m} x_{ij}, \qquad i \in N = \{1,...,n\},$$

where x_{ij} is 1 if object i is assigned to tray j, otherwise it is zero. Additionally, every object can only be packed once:

$$\sum_{j=1}^{m} x_{ij} \leq 1, \qquad i \in N.$$

(a) Original object (b) Shrink-wrapped object

Figure 4: Shrink-wrap generation of a 3D-printable object.

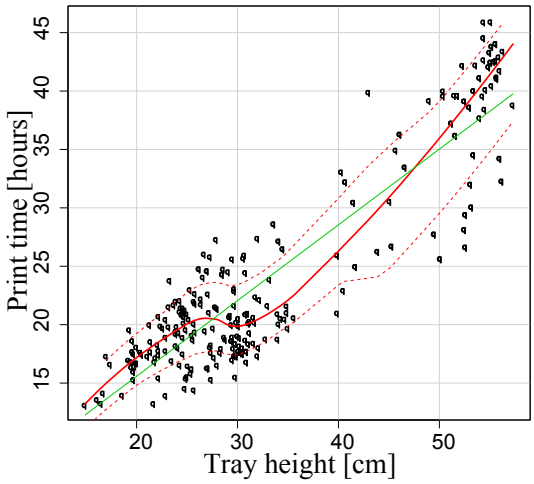


Figure 5: Tray print time versus the tray height (based on historical data from an EOS P760 printer). The green line is a linear regression model; the continuous red line is a locally weighted regression model (Cleveland 1979) and the dotted red line is the spread to this.

The tray print time is largely determined by the height of a tray. This can be seen in Figure 5, where tray print time versus the tray height is shown. The tray height depends to a large extent on the amount of object volume allocated to a tray. Since it is desired to set (or predict) the print time of a tray, every tray has its own maximum volume capacity:

$$\sum_{i=1}^{n} v_i x_{ij} \leq u_j V_j^{max}, \qquad j \in M = \{1,...,m\},$$

where the volume of object i is given by v_i, and the maximal allowed volume of tray j is given by V_j^{max}.

The volume of an object is defined by calculating the volume of a virtual balloon that is wrapped around the object from which all air is drawn. This produces the so-called shrink-wrap volume (Freens 2015). An example of a shrink-wrapped object is given in Figure 4b.

The basic optimization problem is summarized below:

$$\text{Minimize} \quad \sum_{j=1}^{m} u_j C_j^{tray} + \sum_{i=1}^{n} a_i v_i C^{left} \tag{4}$$

$$\text{Subject to} \quad \sum_{j=1}^{m} x_{ij} \leq 1, \qquad i \in N = \{1,\ldots,n\}, \tag{5}$$

$$\sum_{i=1}^{n} v_i x_{ij} \leq V_j^{max} u_j, \qquad j \in M = \{1,\ldots,m\}, \tag{6}$$

$$a_i = 1 - \sum_{j=1}^{m} x_{ij}, \qquad i \in N = \{1,...,n\}, \tag{7}$$

$$a_i, u_j, x_{ij} \in \{0,1\}.$$

In the following sections this basic problem is extended by including job prioritization, one- and two-day jobs and size category constraints.

3.2.2 PRIORITIZING LATE OBJECTS

To prevent long lead times, objects close to their due date should be given higher priority. In this section it is explained how this requirement can be included in the basic problem.

The lateness L_i of object i is defined as the number of days before or after its due date. This value is negative before the due date and positive after the due date. The due date is the point at which an object should be ready for shipment or post processing. In other words, it is the date at which an object's tray should be finished printing and should be cooled down.

The expected tardiness T_i of object i is defined as its lateness L_i plus its expected processing time R_i. If an object can reach the post processing stage before its due date, the tardiness is zero. The tardiness is thus defined as $T_i = \max(0, L_i + R_i)$. To maintain linearity of the optimization problem, the tardiness is modeled slightly differently, but equivalently, by the following constraints:

$$L_i + R_i \leq T_i, \qquad i \in N, \tag{8}$$

$$T_i \geq 0, \qquad i \in N, \tag{9}$$

and the following tardiness cost component is added to the objective function:

$$\sum_{i=1}^{n} T_i v_i C^{tardy}, \tag{10}$$

where C^{tardy} is the cost of tardiness per volume unit. Now the calculation of the expected processing time R_i will be explained now.

Since large trays (with a large V_j^{max}) have significantly longer cooling-down times, objects that are close to their due date should be packed in a small tray (with a small V_j^{max}). The print time of a tray plus its cooling-down time is given by the parameter P_j^{tray}. The print time and cooling times that are used here, are only rough estimates, which can be rounded off to whole days. Depending on the tray an object is allocated to, its time until post-processing can be calculated. This is defined by the following constraint:

$$P_i^{object} = \sum_{j=1}^{m} P_j^{tray} x_{ij}, \qquad i \in N. \tag{11}$$

However, if R_i is set equal to P_i^{object}, this creates the possibility that no objects are selected from the buffer, because of the additional tardiness penalty for allocated objects, for instance, when all objects have a lateness of minus one-day. This problem can be solved by adding a fixed time γ to the lateness of objects that are not allocated and are left in the buffer (i.e., for which $a_i = 1$):

$$E_i = a_i \gamma \qquad i \in N, \tag{12}$$

where the parameter γ should be greater than the process time (i.e., tray print time plus cooling time) of the slowest printer, i.e., $\gamma \geq \max_{1 \leq j \leq m} \left(P_j^{tray} \right)$. In this way, a fair comparison can be made, while still preferring to allocate late objects to small trays. The expected processing time R_i of object i can now be defined as:

$$R_i = P_i^{object} + E_i, \qquad i \in N. \tag{13}$$

Hence, the estimated time until post-processing is:

$$R_i = \begin{cases} P_i^{object} & \text{if object } i \text{ is allocated,} \\ E_i & \text{if object } i \text{ is } \textit{not} \text{ allocated,} \end{cases} \qquad i \in N. \tag{14}$$

3.2.3 CHOOSING ONE- OR TWO-DAY JOBS

The batch size V_j^{max} is not only determined by the type of the assigned printer, but also by the duration of the print job that is chosen. It will now be explained how the decision of the batch size per printer can be modeled. We will only consider full batches and half batches, or, translated to most industrial 3D printers, one- or two-day jobs.

The set $M = \{1, ..., m\}$ contains all trays that are available for planning. It has two entries per printer, that is, it contains an entry for a one-day job and a two-day job. Both tray entries have different values for the maximal allowed volume V_j^{max}, and both trays will also have different values for the process time parameter P_j^{tray}, mentioned in Section 3.2.2. However, if no additional constraints are added, the solution to the optimization problem can contain both the one-day job tray and the two-day job tray, while only one printer is available. To solve this issue, a parameter A_{jk} is introduced that specifies which tray j in set M belongs to printer k in the set $Q = \{1, ..., q\}$. The following constraint will then prevent both a one-day job and a two-day job being planned for printer k:

$$\sum_{j=1}^{m} A_{jk} u_j \leq 1 \qquad k \in Q = \{1, ..., q\}, \tag{15}$$

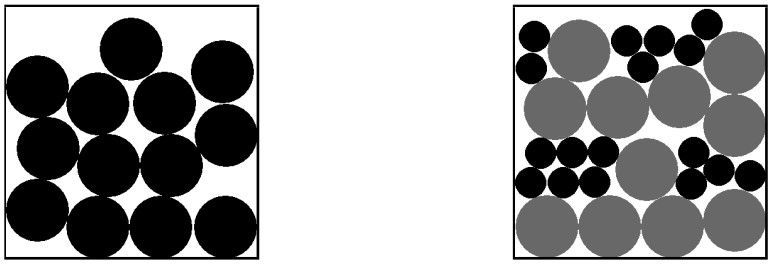
where:

$$A_{jk} = \begin{cases} 1 & \text{if tray } j \text{ corresponds to printer } k, \\ 0 & \text{otherwise} \end{cases} \qquad j \in M, k \in Q. \tag{16}$$

For example, if there are two printers in set Q, there will be four different trays in set M. By setting A_{11}, A_{21}, A_{32} and A_{42} to 1, Constraint (15) will ensure that at most two trays can be selected, that is, for each printer in set Q at most one tray is in set M.

3.2.4 SIZE CATEGORY CONSTRAINTS

The solution of the problem formulated so far, produces a list of objects for each available printer. For each printer, this batch is then three-dimensionally packed in a tray. However, even if an object fits in a



(a) (b)

Figure 6: Two trays with different object size mixtures. Packing (b) is packed denser.

tray volume-wise, it can happen that an object will not fit, because it is longer in one of its size-dimensions (i.e., depth, width or height) than the tray it is allocated to. This problem is also closely related to the density that the 3D packing software will achieve after packing a batch of objects. Relatively large objects that fit in a tray, split the tray into smaller compartments. These compartments will stay unoccupied if no sufficiently small objects are included in the batch that is 3D packed. This phenomenon is visualized in Figure 6. Since a high tray density will increase the capacity of a printer, the amount of volume coming from relatively large objects will be constrained in this section. These size constraints are modeled by assigning an integer size category to all objects and trays. The higher the category number, the larger the tray or the object.

If the category c_i^{object} of object i is higher than the category c_j^{tray} of tray j, object i is not allowed in tray j. This is modeled as follows:

$$c_i{}^{object} x_{ij} \leq c_j^{tray}, \qquad i \in N,\ j \in M, \tag{17}$$

and if object i is more than two categories lower than the category of tray j, it is also not allowed in tray j:

$$c_j{}^{tray} x_{ij} \leq c_i^{object} + 2, \qquad i \in N,\ j \in M. \tag{18}$$

This constraint will preserve small objects for lower categorized (i.e., smaller) trays. A tray will now consist of objects of three size categories. Note that finer object-size mixtures can be obtained by increasing the number of size categories.

Since every tray can only contain a limited number of relatively large objects, an additional volume constrained is added for each category. By adding constraints (19)-(23), undesired combinations of objects can be prevented. The amount of volume allowed per category is specified by the fractions r^0,r^{-1} and r^{-2}, which are multiplied with the maximal allowed volume V_j^{max} per tray. The superscript in r is the categorical difference of an object and a tray. The category fractions of a batch do not necessarily have to add up to 1. If, for example, the fractions would add up to 1 and the buffer does not contain an object of a specific size category, the batch cannot reach its volume capacity (V_j^{max}) anymore. Therefore, some slack is created by setting the fractions such that they will add up to more than 1.

A binary variable z_{ij}^0 is introduced, indicating whether object i is allocated to a tray j that has the same category number:

$$z_{ij}^0 \geq (c_i^{object} - c_j^{tray} + 1) x_{ij}, \qquad i \in N,\ j \in M. \tag{19}$$

By using the fraction r^0, the volume occupied by objects of the same category as the tray is constrained:

$$\sum_{i=1}^{n} z_{ij}^0 v_i \leq u_j r^0 V_j^{max}, \qquad j \in M. \tag{20}$$

The same can be done for objects of one category lower than the tray. First a binary variable z_{ij}^{-1} is introduced, which is set to 1 if object i is of the same or one category lower than tray j.

$$z_{ij}^{-1} \geq (c_i^{object} - c_j^{tray} + 2)x_{ij} - z_{ij}^0, \qquad i \in N,\ j \in M, \tag{21}$$

Volume used by objects that are one category lower than the tray (i.e., $z_{ij}^{-1} - z_{ij}^0 = 1$) can now be limited:

$$\sum_{i=1}^{n} \left(z_{ij}^{-1} - z_{ij}^0\right) v_i \leq u_j r^{-1} V_j^{max}, \qquad j \in M; \tag{22}$$

and for objects that are two categories lower ($x_{ij} - z_{ij}^{-1} - z_{ij}^0 = 1$) than the tray's category:

$$\sum_{i=1}^{n} \left(x_{ij} - z_{ij}^{-1} - z_{ij}^0\right) v_i \leq u_j r^{-2} V_j^{max} \qquad j \in M. \tag{23}$$

In this section we described how to include the requirement that only objects of three size categories can be allocated to a tray. This can of course be extended to finer object size mixtures, if necessary.

4 SIMULATION STUDY

The approach presented in the previous sections is now applied and tested on a factory of Shapeways, which is a key player in the world of additive manufacturing and owns two of the largest 3D printing factories in the world. To test the automated production planning method, a month of production planning of one of Shapeways' 3D printing factories is simulated. This is done for five 3D-printers, i.e., two large EOS P760 printers and three smaller EOS P100 printers. Since every production order placed at Shapeways is tracked and saved in a database, it is possible to exactly regenerate the order buffer of a specific day. From this buffer, objects are allocated to the available printers by using the automated production planning method. Subsequently, the objects are 3D packed with Netfabb. Then the objects ordered for the next day are added to the buffer and the daily printer schedule is generated automatically again.

The volume capacity V_j^{max} and object size mixture are based on the historical averages that Shapeways has used per type of printer. The other model parameters, including C^{left}, C^{tardy}, P_j^{tray}, C_j^{tray} and γ, are set empirically by simulating multiple model configurations and then choosing the configuration yielding satisfactory performance. In future work, calibration of these parameters can be improved by utilizing simulation-based optimization techniques (Deng 2007, Law and McComas 2000).

The resulting tray densities and object lead times are listed in Table 1. This table shows that the automated production planning produces trays that are approximately 10% denser than the manually composed trays. Additionally, it can be seen that the time that objects are waiting to be printed is reduced by almost half a day. However, the larger trays for the EOS P760 printer are less dense. This can be explained by the higher number of large objects that are allocated to this type of printer. This can potentially be solved by changing the object size mixture.

Table 1: Comparison of a manual and automated planning, based on simulation of one-month production.

	Automated				Manual			
	Tray density		Days in buffer		Tray density		Days in buffer	
Printer type	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
P100 - 1 day	0.252	0.060	0.350	0.400	0.220	0.072	0.774	0.703
P760 - 1 day	0.153	0.030	0.360	0.328	0.104	0.025	1.026	0.629
P760 - 2 day	0.170	0.075	1.330	0.461	0.155	0.062	1.239	0.527
All [1]	0.182		0.552		0.170		0.898	

[1] Averages are weighted by the shrink-wrap volume per tray

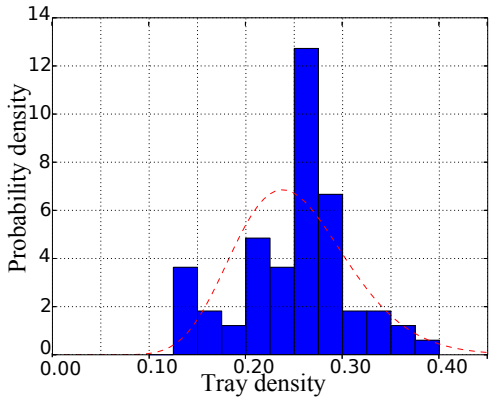


Figure 7: Packing density distribution

The density distribution of the generated trays for the EOS P100 printers is depicted in Figure 7. Clearly, the tray densities fluctuate significantly, with a coefficient of variation (i.e., $std./mean$) of 0.23. This makes it hard to estimate the tray height and hence, to accurately estimate the expected print time of a tray before the 3D-packing of the objects has taken place. This can be improved in future work by investigating which object size mixture will result in more consistent tray densities.

5 CONCLUSION

The aim of this study has been to show the feasibility of automating the production planning of a 3D printing factory. To do this, the production planning of a 3D printing factory has been modeled as an optimization problem and tested in a factory of Shapeways.

The production planning of a 3D printing factory consists of composing printer trays, that is, it needs to be determined which objects will be printed on which printer. Composing these trays consists of two stages. First, sufficiently many objects are allocated to a batch, and secondly, the position of the objects in a tray is determined by applying a 3D packing algorithm.

This paper described how the first stage of this problem can be modeled as an extension of the classical one-dimensional bin packing problem. For the second stage of this problem, that is, the efficient three-dimensional packing of multiple objects in a printer tray, third party packing software has been adopted. The extension of the classical bin packing problem covers specific requirements that appear in high-volume 3D printing environments, including: prioritizing the allocation of late items, choosing the correct batch size, and choosing the right object size mixture, which is a combination of large and small objects. This method has subsequently been tested in a simulation study based on one month of production planning data of Shapeways. This provided a good and fair strategy to test the performance of the proposed method with regard to manually generated production planning. Automating the planning at Shapeways resulted in an 10% increase of printer capacity together with significantly shorter waiting times before printing.

The main drawback of the proposed method is the difficulty in accurately estimating the print time of an automatically generated batch. This difficulty is mostly due to the high variability of tray densities. Future work should therefore focus on the design of a more elaborate simulation study to determine object size mixtures yielding batches, the print time of which can be more accurately estimated. This problem of parameter calibration can be formulated as a simulation based optimization problem, in which the output response of object size mixtures under stochastic input of incoming orders can be systematically explored.

REFERENCES

Cleveland, W. S. 1979. "Robust locally weighted regression and smoothing scatterplots". *Journal of the American statistical association* 74 (368): 829–836.

Deng, G. 2007. *Simulation-Based Optimization*. Ph. D. thesis, University of Wisconsin–Madison.

Forrest, J., and R. Lougee-Heimer. 2005. "CBC User Guide". *INFORMS Tutorials in Operations Research*:257–277.

Freens, J. 2015. "Automating the Production Planning of a 3D Printing Factory". Master's thesis, Eindhoven University of Technology.

Gan, M., N. Gopinathan, X. Jia, and R. A. Williams. 2004. "Predicting Packing Characteristics of Particles of Arbitrary Shapes". *Kona* 22:82–93.

Jia, X., and R. Williams. 2001. "A Packing Algorithm for Particles of Arbitrary Shapes". *Powder Technology* 120 (3): 175–186.

Kantorovich, L. V. 1960. "Mathematical Methods of Organizing and Planning Production". *Management Science*:366–422.

Kruth, J.-P., X. Wang, T. Laoui, and L. Froyen. 2003. "Lasers and Materials in Selective Laser Sintering". *Assembly Automation* 23 (4): 357–371.

Land, A. H., and A. G. Doig. 1960. "An Automatic Method of Solving Discrete Programming Problems". *Econometrica: Journal of the Econometric Society*:497–520.

Law, A. M., and M. G. McComas. 2000. "Simulation-Based Optimization". In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. Joines, R. Barton, K. Kang, and P. Fishwick, 46–49: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lipson, H., and M. Kurman. 2013. *Fabricated: The New World of 3D Printing*. John Wiley & Sons.

Martello, S., D. Pisinger, and D. Vigo. 2000. "The Three-Dimensional Bin Packing Problem". *Operations Research* 48 (2): 256–267.

Wu, S., M. Kay, R. King, A. Vila-Parrish, and D. Warsing. 2014. "Multi-Objective Optimization of 3D Packing Problem in Additive Manufacturing".

AUTHOR BIOGRAPHIES

JOHAN P.N. FREENS is an M.Sc. graduate in Mechanical Engineering at the Eindhoven University of Technology. He has studied the production planning of 3D printing factories as his graduation project. His email address is jfreens@gmail.com.

IVO J.B.F. ADAN is a Professor of the Department of Mechanical Engineering at the Eindhoven University of Technology. His current research interests are in the modeling and design of manufacturing systems, warehousing systems and transportation systems, and more specifically, in the analysis of multi-dimensional Markov processes and queueing models. His email address is i.j.b.f.adan@tue.nl.

ALEXANDER YU. POGROMSKY is an Assistant Professor at the Department of Mechanical Engineering at the Eindhoven University of Technology. His research interests include the theory of nonlinear, adaptive and robust control, nonlinear oscillations. His email address is a.pogromsky@tue.nl .

HUGO PLOEGMAKERS is the Director of Operations at Shapeways. He received the M.Sc. degree in Mechanical Engineering from the Eindhoven University of Technology in 2003. His email address is hugo@shapeways.com.