Bimaterial 3D printing using galvanometer scanners

Daniel Bandeira[1], Marta Pascoal[1,2,3], and Beatriz Santos[1]

[1] *Department of Mathematics, University of Coimbra, EC Santa Cruz, 3001-501 Coimbra, Portugal; Email:* danielf.a.bandeira@gmail.com

[1] *Department of Mathematics, University of Coimbra, EC Santa Cruz, 3001-501 Coimbra, Portugal; Email:* marta@mat.uc.pt

[2] *Centre for Mathematics of the University of Coimbra, Portugal*

[3] *Institute for Systems Engineering and Computers – Coimbra, University of Coimbra, rua Sílvio Lima, Pólo II, 3030-290 Coimbra, Portugal*

[1] *Department of Mathematics, University of Coimbra, EC Santa Cruz, 3001-501 Coimbra, Portugal; Email:* b14796@gmail.com

September 10, 2018

Abstract

In this work a 3D printing system based on the use of stereolithography and able to print parts made of two materials is studied. The 3D printing problem is divided into two subproblems. First, the problem of locating UV light emitters capable of reaching all areas of the polymer that constitutes the part to be printed is analyzed with the goal of minimizing the number of used emitters. Next, for each layer of the part the selected emitters are assigned to the areas of the polymer to be reached, with the goals of maximizing the angle of incidence of the UV light on the printing plane and minimizing the number of emitters that need to be activated. Integer linear formulations were introduced in an earlier work for both problems. Here these models are revisited and heuristic and exact methods are developed. Finally, the formulations and methods are analyzed for a case study. The results of the computational experiments are presented and discussed.

1 Introduction

Three-dimensional printing, or 3D printing, is an additive process for rapid free form manufacturing, where the final object (commonly known as part) is created by the addition of successive thin layers of material. Each layer corresponds to a cross-section of the part to be constructed, and the printer draws each layer as if it was 2D printing [3, 6]. One of the technologies used for 3D printing is stereolithography (SL). In this case, each layer is added by means of a liquid polymer (or resin) which is polymerized (or cured) by exposure to a laser light. Only the zone of the polymer that is reached by the laser beam is cured. Then the platform that supports the model moves to get ready for printing the next layer. Despite being recent, this type of 3D printing has had an increasing popularity and is fairly standard nowadays. This is mainly due to its capacity to produce new parts quickly and at a low cost.

The present work focuses on a process analogous to SL, but with the goal of printing a given part in which the resin coats a previously constructed 3D grid structure made of a different material, like metal or ceramic. The proposed approach can be extended to produce parts with more than two materials, which are added in successive coats to the inner structure. There has been an increasing demand for complex components that combine two or more materials and have added physical and functional characteristics that are too complex to be produced by traditional over-injection or more traditional techniques at reasonable costs. These materials are evolving in terms of applicability and functionality. As such, new manufacturing methods that can produce them in a process which is both feasible and economically sustainable are sought. Examples of its application are custom orthotics, intelligent components, complex or fragile parts and a variety of microstructures that vary in translucency, rigidity, thermal resistance and color where over-injection or others options are not feasible.

Traditional SL 3D printing processes use a single fixed laser, sometimes together with a galvanometer scanner that guides its beam in the desired directions of the printing surface. If a metal structure is pre-installed in the platform, it may block the laser light, thus creating shaded areas and preventing its polymerization. The current work studies a process to implement bimaterial 3D printing, assuming that additional galvanometer scanners are installed on the walls of the printer in order to overcome the issues created by the shaded areas. The galvanometer scanner positions depend on the part to print and are assumed to be fixed from the beginning of the printing process. Nevertheless, the laser beam reflected by each galvanometer scanner can still be directed. In the following, both galvanometer scanners and laser are sometimes simply referred to as emitters.

As explained earlier, the part to print is known in advance and it is discretized into several layers, successively added along the printing process. Additionally, each layer to construct is partitioned into evenly squares, called voxels, and those that must be reached by laser light are known in advance. Naturally, the finer the used discretization is, the more accurate the printing will be and the bigger the problem to be solved.

Assuming that both the voxels that need to be reached by the laser light and the possible locations of the emitters are known, the bimaterial 3D printing problem is split in:

- Emitters location problem (ELP): The goal of which is to find the emitters' positions that allow to minimize the number of emitters required to complete the printing.
- Emitters assignment problem (EAP): Using the emitters output by the ELP, the goal of this problem is to determine which voxels each emitter should reach for each layer.

In a previous work [1] both problems were formulated as linear integer programs. The ELP was modeled as a set covering problem and the EAP was modeled as a general binary problem with two objective functions, which aim at minimizing the distortion of the part while ensuring that the system is as stable as possible. Here the formulations for the problems are reviewed and greedy heuristic methods are presented for the ELP. Additionally, two algorithmic approaches are discussed for the EAP. The heuristics are shown to be polynomial in time, with $O(m^2n)$ if m denotes the number of voxels of the part and n denotes the number of possible positions of the emitters, and with an approximation factor of $1 + \ln(m)$. These methods are complemented with additional rules, for reducing the problem before computing a feasible solution and for eliminating the redundancy of a given feasible solution. One of the approaches introduced for the EAP is based on the weighted sum of the two objective functions, while the other one handles the objectives lexicographically. In general, it is necessary to use a mathematical programming solver to determine Pareto optimal solutions of any of the proposed models. Nevertheless, for a particular case of these models, a greedy exact algorithm with time of $O(m_k n_2)$, where m_k stands for the number of voxels on layer k and n_2 stands for the number of emitters available, is presented.

The rest of the paper is organized as follows. In Section 2 the ELP is addressed and two heuristics are presented for its resolution. Next, in Section 3, the EAP is discussed. After reviewing a formulation for the problem as a biobjective integer linear program, an approach based on a weighted sum of the objective functions and another one based on their lexicographic order are considered. Section 4 is dedicated to the presentation of comparative computational results of the proposed methodologies when applied to a case study. In the last section conclusions and considerations about perspectives for future work are drawn.

2 The emitters location problem

The goal of the ELP is to find the minimum number of emitters that must be installed in order to ensure that a given part can be printed. To do this a matrix, called the emitters coverage matrix, is defined as follows.

Let $M = \{1, \ldots, m\}$ be the set of positions of the voxels that need to be cured and $N = \{1, \ldots, n\}$ be the set of n possible positions for the emitters. We say that the emitter at position j (or simply, the emitter j) covers the voxel at position i (or simply, the voxel i) if it is able to reach it with laser light, $j \in n$, $i \in m$. In order to cure the liquid resin in each voxel at least one emitter needs to cover it. The emitters coverage matrix is a matrix $A = [a_{ij}]_{i=1,\ldots,m; j=1,\ldots,n}$, with entries

$$a_{ij} = \begin{cases} 1 & \text{if the emitter } j \text{ can reach the voxel } i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

for $i \in M$, $j \in N$. The elements of this matrix can be calculated by geometric arguments as shown in [2].

The formulation of the ELP considers the binary decision variables x_j, such that

$$x_j = \begin{cases} 1 & \text{if the emitter } j \text{ is installed} \\ 0 & \text{otherwise} \end{cases}$$

for $j \in N$. Then the ELP can be formulated as

$$\begin{array}{ll} \min & \sum\limits_{j=1}^{n} x_j \\ \text{subject to} & \sum\limits_{j=1}^{n} a_{ij} x_j \geq 1, \quad i \in M \\ & x \in \{0,1\}^n \end{array} \tag{2}$$

where the objective function stands for the total number of emitters to install, whereas the first set of constraints ensures that a solution is feasible if any voxel i can be reached by at least one emitter.

The optimal value of problem (2) is the number of emitters required for the walls to complete the printing. Its optimal solution provides the positions where these emitters should be installed, as these correspond to the indices j such that $x_j = 1$, $j \in N$. As mentioned earlier, this is a set covering problem where each row of matrix A must be covered by at least one column, that is, the voxels of the part need to be covered by at least one emitter. The set covering problem was shown to be NP-complete [7]. Nevertheless, its broad application in many areas has justified the attention of may researchers. A survey on this topic was published by Caprara, Toth and Fischetti [4]. Exact methods can solve this problem, however (due to its complexity) they have limitations as the size of the problem grows. Heuristic methods are described next to provide feasible solutions to this program.

2.1 Heuristic methods

The method for finding solutions of the ELP, is divided into three phases described below:

1. problem reduction,

2. determining a feasible solution, and

3. redundancy elimination.

To simplify the presentation, let $N_i = \{j \in N : a_{ij} = 1\}$ be the set of emitters that cover voxel i, and, analogously, let $M_j = \{i \in M : a_{ij} = 1\}$ be the set of the voxels that are covered by emitter j, for $i \in M$, $j \in N$.

Problem reduction If it is certain that one of the emitters will be used, then the size of problem (2) can be reduced by removing the corresponding column of matrix A. Likewise, all the voxels that are covered by those emitters can also be eliminated.

Further reductions can be achieved by defining a set of dominance rules between rows and columns of the emitters coverage matrix, as follows:

R1 If $M_{j^*} = \emptyset$ for any column $j^* \in N$, then column j^* can be eliminated since it does not cover any row, that is, is replaced N by $N - \{j^*\}$.

R2 If rows $i^*, r^* \in M$ are such that $N_{i^*} \subseteq N_{r^*}$, then row i^* is said to dominate row r^*. In this case row r^* can be eliminated, since by covering row i^* also row r^* is covered. This is achieved by replacing M as $M \backslash \{r^*\}$.

R3 If columns $p^*, q^* \in N$ are such that $M_{p^*} \supseteq M_{q^*}$, then it is said that column p^* dominates column q^*, then the column q^* should be eliminated, since the whole set M_{p^*} covered by column p^* also covers the set M_{q^*}. This is implemented replacing N by $N \backslash \{q^*\}$.

Rule R1 is expected to be less computationally demanding than rules R2 and R3. The rule R1 implies the analysis of all the elements in matrix A, therefore it has a number of operations of complexity $O(mn)$. The rule R2 requires the comparison of each row i with the following, $i+1, \ldots, m$, and each comparison implies checking at most n values. Thus, the number of operations is

$$n \sum_{i=1}^{m} (m-i) = n \sum_{i=0}^{m-1} i = nm \frac{m-1}{2} = \frac{n(m^2-m)}{2}$$

and, in the worst case, it is of order $O(nm^2)$. Similarly, it can be shown that rule R3 can be implemented with time of $O(mn^2)$.

Finding a feasible cover Two greedy heuristic methods are now described. One of them is based on the information obtained from the columns of the emitters coverage matrix, and the other is based on the information obtained from the rows and the columns of that matrix.

The idea behind the first heuristic is to select first the columns that cover the higher number of rows yet uncovered. The algorithm uses two auxiliary variables:

- S, a set that stores the indices of the selected columns, initialized as the empty set,
- R, a set that stores the indices of the rows not yet covered, initialized as M.

The rows that are covered by a selected column are removed from set R. The process ends when all rows are covered. At that moment S is a cover for the set of voxels. This method is outlined in Algorithm 1.

Algorithm 1: Heuristic based on columns

```
1 $S \leftarrow \emptyset$
2 $R \leftarrow M$
3 while $R \neq \emptyset$ do
4     $j^* \leftarrow \text{argmax}\{|M_j \cap R| : j \in N\}$
5     $S \leftarrow S \cup \{j^*\}$
6     $R \leftarrow R \setminus M_{j^*}$
```

In the worst case, the **while** loop in Algorithm 1 has m iterations, whereas line 4 requires nm comparisons. Thus, the algorithm has time of $O(m^2n)$.

The criterion used by the second heuristic to select the matrix columns privileges the rows yet to cover. The idea now is to identify the row that is covered by fewer columns, since this is the hardest to cover. Like in Algorithm 1, of all the columns that cover the chosen row, the one that covers more rows is preferred. This method uses the same sets R and S defined above and is summarized in Algorithm 2.

Algorithm 2: Heuristic based on rows and columns

```
1 $S \leftarrow \emptyset$
2 $R \leftarrow M$
3 while $R \neq \emptyset$ do
4     $i^* \leftarrow \text{argmin}\left\{\sum_{j \in N_i \setminus S} a_{ij} : i \in R\right\}$
5     $j^* \leftarrow \text{argmax}\{|M_j \cap R| : j \in N_{i^*}\}$
6     $S \leftarrow S \cup \{j^*\}$
7     $R \leftarrow R \setminus M_{j^*}$
```

Considering again the worst case, Algorithm 2 performs at most m iterations of the **while** loop. Additionally, line 4 requires at most m comparisons and n calculations, while line 5 requires at most n comparisons and m calculations. Therefore, this method has time of $O(m(mn + nm))$, that is, $O(m^2n)$, like the previous one. Both heuristics have approximation factor of $1 + \ln(m)$, as stated in Theorems 1 and 2. The proofs can be found in appendix.

Theorem 1. *Algorithm 1 has approximation factor of* $1 + \ln(m)$.

Theorem 2. *Algorithm 2 has approximation factor of* $1 + \ln(m)$.

Redundancy elimination After Algorithms 1 and 2 are run, the set of columns stored in the variable S is a cover. However, this set may contain redundant columns. The method presented below seeks to reduce this redundancy by eliminating some columns. To do this we look for columns $j \in S$ such

Algorithm 3: Elimination of redundant columns

```
for $j \in S$ do
    if $S \setminus \{j\}$ is a cover then $S \leftarrow S \setminus \{j\}$
```

that $S \setminus \{j\}$ is still a cover. In this case the column j is redundant and can be eliminated from set S. Otherwise nothing is done. The method is summarized in Algorithm 3.

The **for** loop in Algorithm 3 runs at most n times, given that $|S| \leq n$, and checking whether $S \setminus \{j\}$ is a cover can be done in at most m operations. Thus, Algorithm 3 has time of $O(mn)$.

3 The emitters assignment problem

In the following it is considered that the ELP was solved. As such, a set of emitters was installed in the positions determined by the ELP solution and it is then necessary to select which emitter to use to cure each voxel.

Suppose that, after solving the ELP, it was concluded that n_2 laser light emitters are needed for fully printing an object. These positions are fixed by the solution of the ELP. Also consider that the printing requires p layers, each one with m_k voxels to be reached, $k = 1, \dots, p$. Additionally, it is considered that an emitter can be activated and deactivated along the process, which allows to reach different voxels independently. Thus, the EAP can be formulated independently for different layers.

Let k be a fixed layer to print, $k = 1, \dots, p$. Two sets of decision variables are considered. Since we want to associate the emitters with the voxels to cure, let y_{ij} be binary variables defined by

$$y_{ij} = \begin{cases} 1 & \text{if the emitter } j \text{ is activated to cure voxel } i \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \dots, m_k$, $j = 1, \dots, n_2$. Let also x_j be analogous variables to those used in the ELP, in the previous section, $j = 1, \dots, n_2$.

The two objective functions are z_1, the number of emitters used to construct layer k, and z_2, a measure of the distortion of the laser light on the layer. The latter depends on the angle of incidence of the beam coming from the emitter j when it hits the voxel i, denoted by $\theta_{ij} \in [0, \frac{\pi}{2}]$ and introduced in Section 2, $i = 1, \dots, m_k$, $j = 1, \dots, n_2$.

The EAP can be formulated as a linear integer problem with two objective functions, as follows

$$
\begin{aligned}
\min \quad & z_1(x,y) = \sum_{j=1}^{n_2} x_j \\
\max \quad & z_2(x,y) = \sum_{i=1}^{m_k} \sum_{j=1}^{n_2} \theta_{ij} y_{ij} \\
\text{subject to} \quad & \sum_{j=1}^{n_2} y_{ij} = 1, \ i = 1, \ldots, m_k & \text{(3a)} \\
& \sum_{j=1}^{n_2} a_{ij} y_{ij} = 1, \ i = 1, \ldots, m_k & \text{(3b)} \\
& \sum_{i=1}^{m_k} y_{ij} \leq m_k x_j, \ j = 1, \ldots, n_2 & \text{(3c)} \\
& y_{ij} \in \{0,1\}, \ i = 1, \ldots, m_k, \ j = 1, \ldots, n_2 & \text{(3d)} \\
& x_j \in \{0,1\}, \ j = 1, \ldots, n_2 & \text{(3e)}
\end{aligned}
$$

Constraints (3a) are assignment conditions that ensure the uniqueness of the solution, whereas constraints (3b) ensure its viability. The equalities (3a) and (3b) are not redundant, because only the two sets guarantee that a single emitter reaches each voxel and also that this is possible. Conditions (3c) are covering conditions, to force every voxel to be reached by at least one emitter. Finally, conditions (3d) are aggregated constraints that relate the problem variables y_{ij} and x_j.

In general the objective functions z_1 and z_2 are conflicting and there is no feasible solution that optimizes both simultaneously. As an alternative to the usual concept of optimality in mono-objective problems, when dealing with two objectives we seek to determine efficient solutions. A feasible solution of (3), (x', y'), is said to be an efficient solution if there is no other, (x'', y''), that dominates it, that is, such that

$$
\left\{ \begin{array}{l} z_1(x'', y'') \leq z_1(x', y') \\ z_2(x'', y'') \geq z_2(x', y') \end{array} \right. \quad \text{and } (z_1(x'', y''), z_2(x'', y'')) \neq (z_1(x', y'), z_2(x', y')).
$$

Several methods are known in the literature to determine the efficient solutions of biobjective linear integer problems, usually depending on solving a sequence of mono-objective versions of the same problem or of similar ones. For the application to bimaterial 3D printing it is sufficient to select a single solution, therefore the adaptation of two known strategies for biobjective optimization problems is now described to calculate part of the efficient solutions. The first is to aggregate the objectives z_1 and z_2 in the same objective function by means of a weighted sum. The second is based on a lexicographical perspective, in which one of the objectives is privileged, leaving the other in the background.

3.1 Weighted sum model

One of the most common ways of dealing with biobjective problems is to aggregate the two objective functions by means of a sum. The relative importance of each function can then be expressed by multiplying it by a well-chosen parameter. In the case of the EAP this consists of considering the problem

$$
\begin{array}{ll} \min & z(x,y) = \lambda_1 z_1(x,y) - \lambda_2 z_2(x,y) \\ \text{subject to} & (3a) - (3e) \end{array} \qquad (4)
$$

with $\lambda_1, \lambda_2 \in [0,1]$ such that $\lambda_1 + \lambda_2 = 1$. According to [5] the solution of (4) is an efficient solution of (3) for weights $\lambda_1, \lambda_2 \in]0,1[$ such that $\lambda_1 + \lambda_2 = 1$. However, in general, not all efficient solutions of an integer linear problem are a linear convex combination of two other [5].

Before aggregating z_1 and z_2 it is convenient to normalize them, to have similar orders of magnitude. That is achieved by multiplying z_1 and z_2 by parameters $w_1, w_2 \in \mathbb{R}$, such that $w_1 = n_2$ and

$$w_2 = \sum_{i=1}^{m_k} \left(\overline{\theta}_i - \underline{\theta}_i\right),$$

where

$$\underline{\theta}_i = \min_{j=1,\ldots,n_2} \{\theta_{ij}\} \quad \text{and} \quad \overline{\theta}_i = \max_{j=1,\ldots,n_2} \{\theta_{ij}\},\ i = 1, \ldots, m_k.$$

Then, given $\lambda_1, \lambda_2 \in [0,1]$ with $\lambda_1 + \lambda_2 = 1$, the objective function is

$$\lambda_1 \frac{w_2}{w_1} z_1 - \lambda_2 z_2$$

and the mono-objective linear binary problem

$$\begin{array}{ll} \min & \lambda_1 \frac{w_2}{w_1} z_1 - \lambda_2 z_2 \\ \text{subject to} & (3a) - (3e) \end{array} \tag{5}$$

should be solved to find an efficient solution of (3). This is a binary linear problem that combines different types of constraints, without a particular structure. Because the set covering problem is an NP-complete problem this one should be difficult to solve as well.

If the decision maker knows the relative importance of functions z_1 and z_2, the weights to use can be chosen. However, these situations are rare. Alternatively, varying these weights arbitrarily allows to find a set of efficient solutions. As already mentioned, increasing λ_1 means giving greater importance to the number of emitters that are used in each layer, while increasing λ_2 implies giving greater importance to the deformation of the beam that reaches the layer. In particular, the case where $\lambda = (0,1)$ corresponds to minimizing the laser distortion only, whereas $\lambda = (1,0)$ corresponds to minimizing the number of emitters. In the first case, the objective function z_1 does not depend on the variables x_j, so the problem can simply be written as

$$\begin{array}{ll} \max & z_2(x,y) \\ \text{subject to} & (3a), (3b), (3d) \end{array}$$

and can be easily solved exactly. In fact, the two initial sets of constraints require that, for each voxel i only one of the variables y_{ij} is 1, while the rest are 0. Since the values θ_{ij} are known, z_2 is maximum when

$$y_{ij^*} = \begin{cases} 1 & \text{if } j^* = \text{argmax}\{\theta_{ij} : a_{ij} = 1 \wedge j = 1, \ldots, n_2\} \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, m_k$. Therefore, Algorithm 4 can be applied for each layer $k = 1, \ldots, p$.

Algorithm 4 is polynomial in terms of time, of $O(m_k n_2)$.

If otherwise it is sought to minimize the number of emitters only, when considering the use the weights $\lambda = (1,0)$ in problem (4), the problem can have multiple optimal solutions, which, in practice, corresponds to several possibilities of assigning emitters to some voxels. Then the problem (4) does not specify a criterion for selecting the emitter, although this is needed for the automatization of printing the part. The model proposed below addresses this question.

Algorithm 4: Minimizing layer distortion k

```
for $i = 1, \ldots, m_k$ do
    for $j = 1, \ldots, n_2$ do $y_{ij} \leftarrow 0$
    $j^* \leftarrow \text{argmax}\{\theta_{ij} : a_{ij} = 1 \wedge j = 1, \ldots, n_2\}$
    $y_{ij^*} \leftarrow 1$                    /* Emitter $j^*$ reaches voxel $i$ */
```

3.2 Lexicographic model

Sometimes the greater importance of one objective over another can be assumed, without expressing a clear dependence between the two, as is the case when using weights. If one objective has priority over the other, two consecutive mono-objective problems can be solved, one considering only the most important objective function and the other optimizing the other objective function, restricted to the solutions of the first. An order is thus established between the two functions and these are dealt with lexicographically.

Consider that z_1 and z_2 are the first and the second objective functions, respectively. The lexicographic method applied to the EAP consists of first solving problem (3) but minimizing only the function z_1. Afterwards, only function z_2 is maximized in the set of the all optimal solutions of the previous problem, that is,

$$\begin{array}{ll} \max & z_2(x, y) \\ \text{subject to} & (3a) - (3e) \\ & \sum\limits_{j=1}^{n_2} x_j = z_1^* \end{array} \tag{6}$$

if z_1^* denotes the optimal value for the first problem. The optimal solutions of (6) are efficient solutions of (3) [5].

The first problem can be simplified, since the objective function z_2 depends only on the variables x_j, $j = 1, \ldots, n_2$. Let (x, y) be a feasible solution to this problem. Then, using (3c),

$$\sum_{j=1}^{n_2} a_{ij} x_j \geq \sum_{j=1}^{n_2} a_{ij} \sum_{i=1}^{m_k} \frac{y_{ij}}{m_k} = \frac{1}{m_k} \sum_{i=1}^{m_k} \sum_{j=1}^{n_2} a_{ij} y_{ij},$$

for any $i = 1, \ldots, m_k$. Now using (3b),

$$\frac{1}{m_k} \sum_{i=1}^{m_k} \sum_{j=1}^{n_2} a_{ij} y_{ij} = \frac{1}{m_k} \sum_{i=1}^{m_k} 1 = 1,$$

and hence

$$\sum_{j=1}^{n_2} a_{ij} x_j \geq 1.$$

Therefore, for $k = 1, \ldots, p$, the optimal value of the first problem can be found by solving the problem

$$\begin{array}{ll} \min & \sum\limits_{j=1}^{n_2} x_j \\ \text{subject to} & \sum\limits_{j=1}^{n_2} a_{ij} x_j \geq 1, \ i = 1, \ldots, m_k \\ & x_j \in \{0, 1\}, \ j = 1, \ldots, n_2 \end{array} \tag{7}$$

It should be noted that (7) is a set covering problem analogous to problem (2) defined for the ELP when restricted to a specific layer, k. The solution of this problem provides the number of emitters to use on layer k, z_1^*, needed to solve problem (6) for the same layer.

4 Computational experiments

In this section the models and methods proposed for the ELP and the EAP are tested for a case study, with the goal of assessing them in terms of the solutions they output and of the run times. Nevertheless, the bimaterial 3D printing problem serves planning purposes, therefore, the time needed to solve it is of secondary importance.

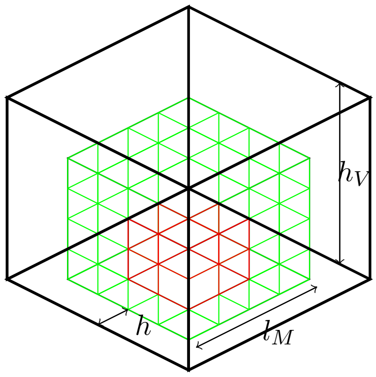


Figure 1: Printing area and part to print

The case study is depicted in Figure 1. It consists of constructing a cube with a hole at its bottom face. This hole is an inner cube. The grid in the image represents the existing metal grid previously produced and which must be coated with resin.

The thickness of the metal grid is considered to be equal to the thickness of the resin layers, that is, 1. This is also the value used as the width of the voxels. The variable parameters for printing the cube are:

- The length of each segment of the metal grid, l_M.
- The thickness of the resin added on each side of the metal grid, l_P.
- The number of divisions of the metal grid, which is assumed to be uniform, n_M.
- The distance between the cube to be printed and the side walls of the printer, where the emitters can be installed, h.
- The height of the printing area, h_V.
- The side of the hole, n_{hole}.

Note that for all the faces of the cube, the length of the edge of the final cube to be printed, denoted by n_V, corresponds to the addition that is made of resin, with length l_P, on each side to the length of the edges of the cube of metal given by l_M, that is, we have $n_V = 2l_P + l_M$. Therefore, the object is formed by $2l_P + l_M$ layers. The two upper layers are excluded from the tests presented. When printing on the top of the grid all the metal structure is below the printing surface, so an emitter on the top of the printer can be used directly. For this reason, in the tests performed, and with no loss of generality, it is assumed that there is a laser emitter at the center of the top wall of the printer. This corresponds to fixing $x_1 = 1$, if that emitter is represented by 1. It is also assumed that the part to print is centered on the printing platform.

In what follows the used length unit corresponds to 0.2 millimeters, the length of the side of the voxels. In addition, the printer dimensions were fixed to $h_V = 1250$ units. The value h varies, given that it depends on the size of the object to print. However, h is always chosen so that the printing platform also has size of 1250×1250 units. We also consider fixed the value $l_P = 1$. The number of voxels at each layer is $n_V \times n_V$. The characteristics of the solved instances are summarized in Table 1.

Table 1: Test parameters

Test	n_V	n_M	n_{hole}	h
T1	300	10	100	475
T2	300	10	166	475
T3	300	12	82	475
T4	300	12	136	475
T5	498	12	136	376
T6	498	12	226	376
T7	498	16	100	376
T8	498	16	232	376

The package CPLEX 12.7 was used for solving the linear problems, whereas MATLAB R2016b was used to implement the heuristic methods. The presented run times are mean values obtained for 30 repetitions of the methods. The tests run on an Intel i7-6700 Quadcore of 3.4GHz, with 8Mb of cache and 16 Gb of RAM.

4.1 The emitters location problem

The first step for solving the bimaterial 3D printing problem in the given case study is to solve the ELP. Besides the parameters mentioned earlier, it was assumed that there is always an emitter installed at the center of the top wall of the printer. Moreover, 60 other possible locations were considered for additional emitters, evenly distributed along the side walls of the printer. An additional constraint was imposed to formulation (2), in order to ensure a minimum height of 500 for the emitters position and thus preventing large distortions of the laser beam on the printing layer.

The results obtained applying CPLEX to the ELP formulation are summarized in Table 2. The optimal value of the ELP is denoted by z^*. The fixed top position, which appears in all solutions, is omitted in table. According to this table, a total of 5 to 6 emitters are required for completing the printing. As expected the run times to solve the instances exactly increase with the size of the formulation. Nevertheless, for all cases an optimal solution was found in an average time of less than 6 minutes.

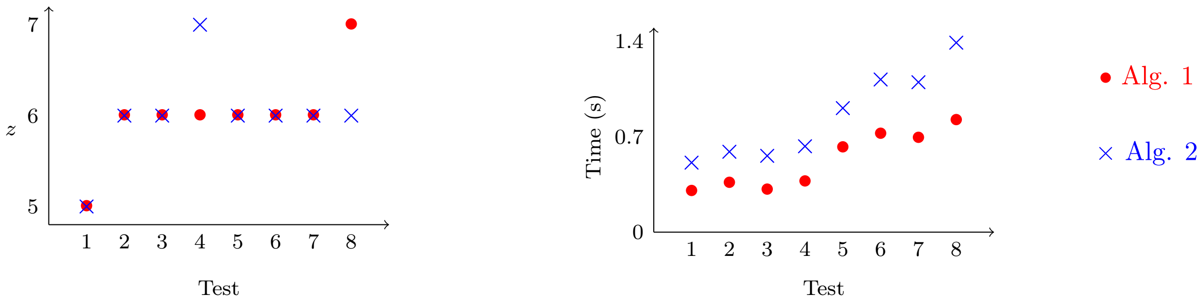
Table 2: ELP solutions and run times (seconds) with CPLEX

Test	z^*	Emitters' position	Time
T1	**5**	(1, 1250, 1000), (1250, 501, 750), (1,1,500), (1250,1250,500)	40.69
T2	**5**	(251, 1, 750), (1, 1,500), (1250, 1000, 500), (501, 1250, 500)	72.64
T3	**6**	(1250, 251, 1000), (1, 1000, 750), (1, 1, 500), (1250, 1, 500), (1250,1250,500)	186.53
T4	**6**	(1, 1000, 1000), (750, 1, 1000), (1, 1, 500), (1250, 501, 500), (1000, 1250, 500)	199.50
T5	**5**	(1, 1000, 500), (1250, 251, 500), (750, 1, 500), (1000, 1250, 500)	42.88
T6	**5**	(1, 1250, 1000), (1000, 1, 750), (1250, 501, 500), (1000, 1250, 500)	71.09
T7	**5**	(1, 251, 1000), (251, 1250, 1000), (1000, 1, 750), (1250, 750, 500)	51.52
T8	**6**	(1, 251, 1000), (750, 1, 1000), (1, 501, 500), (1250, 1000, 500), (251, 1250, 500)	323.26

The heuristic methods outlined in Algorithms 1 and 2 were applied to the same set of tests. The problem was reduced in advance, because of the automatic selection of the first emitter. In the following this will be called the reduction rule R1' (a particular case of reduction rule R1). In a first phase no other reduction rules or redundancy elimination were used. The results for these tests are summarized in Table 3 and Figure 2. Now the number of emitters is denoted by z. The values of z are shown in bold whenever $z = z^*$, that is, when an optimal solution was found.

Table 3: ELP solutions and run times (seconds) of the Algorithms 1 and 2

Test	Alg. 1		Alg. 2	
	z	Time	z	Time
T1	**5**	0.30	**5**	0.51
T2	6	0.36	6	0.59
T3	**6**	0.31	**6**	0.56
T4	**6**	0.37	7	0.63
T5	6	0.62	6	0.91
T6	6	0.72	6	1.12
T7	6	0.69	6	1.10
T8	7	0.82	**6**	1.39



(a) Number of emitters

(b) Run times (s)

Figure 2: ELP solutions and run times of the Algorithms 1 and 2

According to the results, both Algorithms 1 and 2 are able to find the optimal solution for 3 out of the 8 tests, even though not for the same instances for the two methods. Both heuristics produced solutions with the same number of emitters for all other tests, and required 1 additional emitter than the optimal solution. In terms of the run times, Algorithm 1 was always faster than Algorithm 2.

After the application of Algorithms 1 and 2 only, new tests were run including the reduction rules R2 and R3 in the former codes. Additionally, redundant emitters in the provided solutions were eliminated using the Algorithm 3. To simplify the exposition we refer to the combination of Algorithm 1 (Algorithm 2) and remaining rules by `H1` (`H2`). The results for these tests are summarized in Tables 4 and 5, respectively. Both partial and total run times are presented.

According to Tables 4 and 5, both heuristics `H1` and `H2` have found a worse solution than before regarding test T8. Still, the heuristic `H2` performed better than Algorithm 2, as it was able to reach an optimal solution for 6 out of the 8 tests. The run times of the heuristics `H1` and `H2` are fairly higher that those of Algorithms 1 and 2 (between around 4 and 19 minutes), mainly due to the application of the rule R2.

To assess the impact of rules R2 and R3 and of Algorithm 3 when combined, separately, with Algorithms 1 and 2. The results are reported on Tables 6 and 7, respectively. For these instances, Algorithm 1 does not benefit from including reduction rules, altogether – Table 6 – or separately – Table 4, as both the objective function values and the run times worsen compared to the values in Tables 3. Likewise, rule R3 and Algorithm 3 do not change the objective function value obtained by Algorithm 2 only, and

Table 4: ELP solutions and run times (seconds) for heuristic `H1`

Test	z	Time					
		R1'	R2	R3	Alg. 1	Alg. 3	Total
T1	**5**	0.07	234.37	0.27	0.01	0.19	235.21
T2	6	0.08	620.43	0.48	0.03	0.27	621.29
T3	**6**	0.07	323.02	0.28	0.01	0.23	323.62
T4	7	0.07	570.70	0.45	0.03	0.33	571.58
T5	6	0.21	269.86	0.24	0.01	0.71	271.03
T6	6	0.22	391.31	0.30	0.02	0.78	392.63
T7	6	0.21	489.75	0.32	0.02	0.65	490.94
T8	7	0.22	1115.30	0.50	0.03	0.94	1117.00

Table 5: ELP solutions and run times (seconds) for heuristic `H2`

Test	z	Time					
		R1'	R2	R3	Alg. 2	Alg. 3	Total
T1	**5**	0.07	234.13	0.27	0.02	0.19	234.68
T2	**6**	0.08	616.88	0.48	0.04	0.27	617.76
T3	**6**	0.07	321.40	0.27	0.03	0.23	322.00
T4	**6**	0.07	574.72	0.44	0.04	0.27	575.56
T5	6	0.21	267.82	0.24	0.02	0.71	269.00
T6	**5**	0.26	483.06	0.35	0.03	0.75	484.45
T7	**5**	0.20	432.63	0.29	0.03	0.54	433.69
T8	7	0.22	1086.20	0.50	0.05	0.94	1087.90

slow down the process. However, the combination of rule R2 and Algorithm 2 is able to find an optimal solution for test T4, which does not happen when only Algorithm 2 is used. Still, according to the results in Table 7 using Algorithm 2 with all reduction rules allows to find more optimal solutions than using rule R2 only. Additionally, it can be noted that this rule is more time demanding than CPLEX for this set of instances.

4.2 The emitters assignment problem

Solving the ELP provides the location of the emitters for curing a given part. This is followed by solving the EAP, in order to determine which voxels each emitter should cure. In practice it is enough to know a single efficient solution of the problem, therefore the methods described in Section 3 were implemented with that purpose. Considering the previous case study and the exact solutions for the ELP shown in Table 2, and assuming that the angles α and θ, introduced earlier, are known, the following methods for the EAP were implemented:

- the weighted sum model, with weights $(1-\lambda_i, \lambda_i)$, $\lambda_i = 0.25(i-1)$, represented in the rest of the text by `Ai`, $i = 1, \dots, 5$;
- the lexicographic model, represented by `A6`.

For the various approaches, the average number of emitters activated per layer, denoted by μ_1, and the average sum of the incidence angles per layer, denoted by μ_2, were registered, as well as the average run

Table 6: ELP solutions and run times (seconds) of Algorithm 1 combined with the reduction rules separately

Test	R2		R3		Alg. 3	
	z	Times	z	Time	z	Time
T1	**5**	233.97	**5**	8.53	**5**	0.49
T2	6	628.71	6	10.55	6	0.63
T3	**6**	324.00	**6**	9.67	**6**	0.54
T4	7	573.95	**6**	11.25	**6**	0.64
T5	6	266.06	6	14.76	6	1.34
T6	6	462.36	6	19.26	6	1.56
T7	6	473.10	6	18.75	6	1.36
T8	7	1086.30	7	24.29	7	1.75

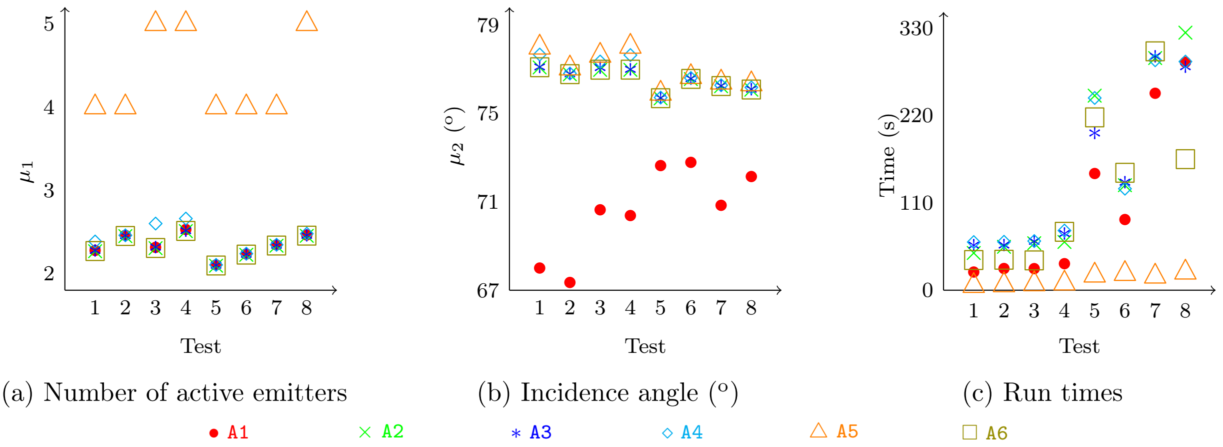
Table 7: ELP solutions and run times (seconds) of Algorithm 2 combined with the reduction rules separately

Test	R2		R3		Alg. 3	
	z	Time	z	Time	z	Time
T1	**5**	232.34	**5**	8.74	**5**	0.71
T2	6	621.57	6	10.69	6	0.86
T3	**6**	321.00	**6**	9.87	**6**	0.80
T4	**6**	573.49	7	11.42	7	0.98
T5	6	269.07	6	15.07	6	1.63
T6	6	419.91	6	18.44	6	1.78
T7	**5**	428.83	6	18.91	6	1.78
T8	7	1101.50	**6**	24.89	**6**	2.15

time required to solve the EAP for all layers. The results are summarized in Figure 3.

As expected, Figure 3 shows that the approach A1 is the best regarding the average number of emitters activated per layer, while approach A5 is the best regarding the angle of incidence of the beam over the printing layer. The remaining approaches find solutions that lie between those determined by A1 and A5. Moreover, for the current tests the approaches A2 and A6 always find the same solutions, because both give high priority to the number of emitters used for printing, and these are better than the approach A1, since μ_1 always has the same value, while μ_2 is better when using the approaches A2 and A6. The results obtained by the remaining approaches differ slightly according to the weights given to z_1 and z_2. As mentioned earlier, the choice of weights to use should depend on the importance that the two functions have in practice. Regarding the run times required to solve the problem, the approach A6 is always the fastest, in general, followed by the approach A1. The other approaches present similar run times, which do not exceed 6 minutes.

Algorithm 4 is an alternative to approach A5. In order to compare their efficiency, their run times were also registered when applied to all layers. Table 8 presents these run times, which show that Algorithm 4 is always considerably faster than approach A5.



(a) Number of active emitters

(b) Incidence angle (°)

(c) Run times

Figure 3: EAP solutions and run times (seconds)

Table 8: EAP run times (seconds) of approach A5 and Algorithm 4

Test	A5	Alg. 4	Test	A5	Alg. 4
T1	5.71	0.10	T5	18.76	0.24
T2	6.56	0.10	T6	20.79	0.25
T3	7.80	0.10	T7	17.48	0.23
T4	8.70	0.11	T8	22.53	0.30

4.3 Printing quality

The previous sections focused on the evaluation of the models and methods for bimaterial 3D printing in terms of the output solutions, namely the number of emitters and the incidence angles, as well as of their run times (even though these are not a main concern for this application). Another aspect that should be taken into account is the quality of the produced part. At the origin of the emitter, the laser beam used for printing has the shape of a circle. Yet, in general, it reaches the printing surface as an ellipse, resulting from the distortion caused by the incidence angle θ.

When considering that the center of the laser points towards the center of the voxel, two situations may occur that affect the quality of the printed part: a region may be cured beyond the target voxel, and part of the target voxel may be left uncured because it is not reached by the laser light. These two regions are illustrated in Figure 4. Their areas are called the outer area, shown in green and denoted by A_{out}, and the inner area, shown in yellow and denoted by A_{in}, respectively. These two values were calculated in [2]. The summation of each of these values for all the voxels in the part is used to estimate the overall quality of the printed part. In general these provide overestimations of the actual inner and outer areas, in the first case because the uncured area of a voxel may still be reached by a beam pointing at another voxel, and in the second because the outer area/distortion is only relevant for the voxels on the border of the part to print.

The standard laser for SL has a radius of 0.05 millimeters, so, taking the considered unit of measurement (the side of the voxel) into account, the laser has a radius of 0.25 units. For each test performed, the values of A_{in} and A_{out} are calculated for all voxels. Figure 5 shows the mean value of the percentage relative to the voxel area of A_{in}, μ_{in}. The mean value of the percentage relative to the voxel area of A_{out} was always 0.

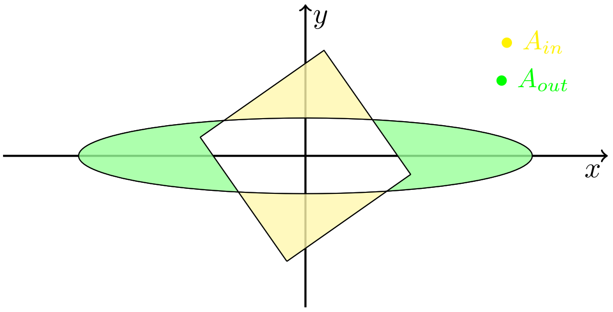


Figure 4: Areas to calculate

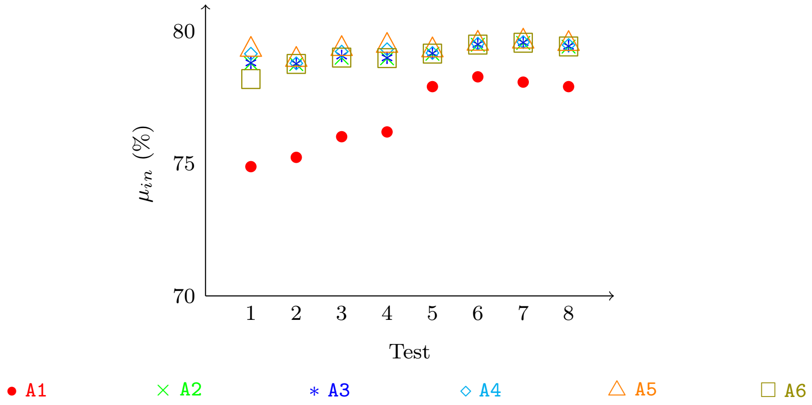


Figure 5: Distortion areas

According to Figure 5, part of the voxels remains liquid, although μ_{in} is greater than 70% in all tests. The main reasons are the assumptions that the laser reaches only the center of the voxels and that the voxel sides are twice the diameter of the laser beam. Working with smaller voxels would result in a reduction of this area, however the average value of A_{out} can increase. Nevertheless, as mentioned earlier, situations like these may be overcome with the use of a post-finishing phase.

5 Conclusions

In this work the bimaterial 3D printing problem was studied, assuming the instalation of galvanometer scanners on the walls of a printer. The problem is split in two: where to locate the emitters (the ELP) and how to assign them to the voxels of the part (the EAP). The first was modeled as a set covering problem and solved by means of constructive heuristics combined with some enhancement rules. The second was formulated as an integer linear program considering two objective functions. Two approaches were used to solve the biobjective problem. The first one was based on the weighted sum method and the second based on a lexicographic criterion. Finally, a case study was described and the corresponding computational results were presented in terms of the solution found and in terms of run times. For evaluating the solutions the area reached by the laser light was calculated.

Several tests were performed for the case study. Both the ELP and the EAP could be solved by the introduced methods. In the first case the exact solution was to use 5 or 6 emitters on the printer walls. The exact solutions were found by CPLEX in less than 6 minutes. Heuristic methods, Algorithms

1 and 2, were able to compute those solutions on 37.5% of the instances in less that 1 second and 2 seconds, respectively, while requiring an additional emitter for the remaining cases. When combined with reduction/simplification rules, the second heuristic found the optimal solution for 75% of the instances, although begin slower than CPLEX. The EAP was formulated considering the minimization of the number of active emitters per layer and the maximization of the sum of the angles of incidence of the laser beam over the printing layer. Methods for finding efficient solutions for the problem were introduced. The tests performed showed that computing one such efficient solution can be done in a short run time per each layer (and less than 6 minutes for the whole part).

To theoretically assess the quality of the solutions found, the outer and the inner areas were calculated for all the voxels in the part. Both depend on the angle of incidence of the laser beam when reaching the voxel. The values obtained for the outer area were close to 0%, however the inner area indicates that between 70% and 80% of the polymer would actually be uncured after the printing is over. These values are overestimations and can be improved by a standard post-printing finishing phase where the whole part goes through an overall cure. The ideal would be to test the production of this and other parts empirically, which should be done in the future. In theoretical terms, the inner area can be reduced if a finer discretization of the part is considered. Naturally, this will increase the size of the problems to solve. Finally, it is important to add that this approach can be used to print parts with more than 2 materials, by handling the result of the first print as the inner structure of the next one.

Acknowledgments This work was developed within the Project PT2020-POCI-SII & DT 17963: NEXT.Parts, Next-Generation of Advanced Hybrid Parts, from the program Portugal 2020, through the COMPETE 2020 – Programa Operacional Competitividade e Internacionalização. The work was also partially supported by the Portuguese Foundation for Science and Technology (FCT) under project grants UID/MAT/00324/2013 and UID/MULTI/00308/2013.

References

[1] D. Bandeira and M. Pascoal. Modeling bimaterial 3D printing using galvanometer scanners. *Submitted for publication*, 2018. Available at http://www.mat.uc.pt/∼marta/NextM/BandeiraPascoal18a.pdf.

[2] D. Bandeira, M. Pascoal, A. Mateus, and M. Reis Silva. Multi-material 3D printing using stereolithography: an optimization approach. *Submitted for publication*, 2017. Available at http://www.mat.uc.pt/∼marta/NextG/BandeiraEtAl17.pdf.

[3] M. Burns. *Automated fabrication: improving productivity in manufacturing.* Prentice-Hall, Inc., 1993.

[4] A. Caprara, P. Toth, and M. Fischetti. Algorithms for the set covering problem. *Annals of Operations Research*, 98(1):353–371, 2000.

[5] M. Ehrgott. *Multicriteria optimization.* Springer Science & Business Media, Berlin, Heidelberg, 2006.

[6] I. Gibson, D. Rosen, and B. Stucker. *Additive manufacturing technologies.* Springer, New York, 2015.

[7] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

A Approximation factor for the emitters location problem

Algorithm 1 is based on a set covering problem. To calculate its approximation factor, two lemmas must be proved. To do so, the set covering problem will be interpreted as a maximal covering problem. For completeness, we start by defining these two types of problems.

Definition 1. *Given a set of $V = \{e_1, \dots, e_m\}$ elements, a list of sets $S_j \subseteq V$ (not necessarily disjoint), $j = 1, \dots, n$, and a limit K, the maximal covering problem consists of finding K sets, $S'_1, \dots, S'_K$, such that $\left|\bigcup_{j=1}^{K} S'_j\right|$ is maximized. The set covering problem consists of finding the least number of sets that, together, cover the entire set V.*

For the ELP, we have $V = M$ and $S_j = \{j\}$, with $j \in N$. Consider also that C^* is the optimum value of the maximal covering problem. In this case, $K = |S|$. Let C_k denote the rows that have already been covered at the end of the k-th iteration of the **while** loop in Algorithm 1. By Algorithm 1, K_j is the largest number of uncovered rows that can be covered by column j, in the k-th iteration of the loop, for $j \in S$. Thus, the following lemma holds.

Lemma 1. *At the k-th iteration of Algorithm 1 we have $K_j = |C_k| - |C_k - 1| \geq \dfrac{C^* - |C_{k-1}|}{|S|}$, for any $j \in S$.*

Proof. Let us divide the proof in two parts.

1. At the end of the k-th iteration, the number of rows yet to be covered is the difference between the number of covered rows at the end of the k-th iteration, $|C_k|$, and the number of rows covered after the $(k-1)$-th iteration, $|C_{k-1}|$. As such, we have $K_j = |C_k| - |C_{k-1}|$.

2. Note that the number of rows covered in the optimal solution at the beginning of the k-th iteration, but not in the algorithm, is greater than or equal to the difference between the optimal value of the maximal covering problem and the number of covered rows in the algorithm at the beginning of the k-th iteration, i.e., at the end of the $(k-1)$-th iteration. That is, it is greater than or equal to $C^* - |C_{k-1}|$.

   Let $S^* = \{j^*_1, \dots, j^*_{|S|}\}$ be an optimal solution for the maximal covering problem using $|S|$ columns. At the beginning of the k-th iteration we have $R = M \setminus C_{k-1}$. Thus, $\bigcup_{k=1}^{|S|} (M_{j^*_k} \cap R) = \left(\bigcup_{k=1}^{|S|} M_{j^*_k}\right) \setminus C_{k-1}$, and therefore,

   $$\sum_{k=1}^{|S|} |M_{j^*_k} \cap R| \geq |\bigcup_{k=1}^{|S|} \left(M_{j^*_k} \cap R\right)| = |\left(\bigcup_{k=1}^{|S|} M_{j^*_k}\right) \setminus C_{k-1}| \geq C^* - |C_{k-1}|.$$

   Then,

   $$\sum_{k=1}^{|S|} |M_{j^*_k} \cap R| \geq C^* - |C_{k-1}| \Rightarrow \sum_{k=1}^{|S|} \max_{i=1,\dots,|S|} |M_{j^*_i} \cap R| \geq \sum_{k=1}^{|S|} |M_{j^*_k} \cap R| \geq C^* - |C_{k-1}|$$

   $$\Rightarrow |S| \max_{i=1,\dots,|S|} |M_{j^*_i} \cap R| \geq C^* - |C_{k-1}| \Leftrightarrow \max_{i=1,\dots,|S|} |M_{j^*_i} \cap R| \geq \frac{C^* - |C_{k-1}|}{|S|}.$$

Because, by definition, $K_j \geq \max_{i=1,\ldots,|S|} |M_{j_i^*} \cap R|$, for any $j \in S$, then, $K_j \geq \dfrac{C^* - |C_{k-1}|}{|S|}$.

□

Lemma 2. *In the k-th iteration of Algorithm 1 we have* $|C_{k-1}| \geq \dfrac{C^*}{|S|} \displaystyle\sum_{n=0}^{k-1} \left(1 - \frac{1}{|S|}\right)^n$.

Proof. Consider $j \in S$ and proceed by induction on k.

For $k = 1$ the first choice is such that $K_j \geq \dfrac{C * -|C_0|}{|S|} = \dfrac{C*}{|S|}$, given that before the loop all the rows are still uncovered, thus $|C_0| = 0$. On the other hand, $K_j = |C_1| - |C_0| = |C_1|$. Therefore, $|C_1| \geq \dfrac{C*}{|S|} = \dfrac{C*}{|S|}\left(1 - \dfrac{1}{|S|}\right)^0$.

Suppose now that the formula holds for k and let us prove that it still holds for $k+1$. In the $(k+1)$-th iteration K_j has the form

$$K_j = |C_{k+1}| - |C_k| \geq \frac{C^* - |C_k|}{|S|} \Rightarrow |C_{k+1}| \geq |C_k| + \frac{C^* - |C_k|}{|S|} = \frac{C^*}{|S|} + \left(1 - \frac{1}{|S|}\right)|C_k|.$$

Then, by the induction hypothesis, one has

$$|C_{k+1}| \geq \frac{C^*}{|S|} + \left(1 - \frac{1}{|S|}\right)\frac{C^*}{|S|}\sum_{n=0}^{k-1}\left(1 - \frac{1}{|S|}\right)^n = \frac{C^*}{|S|} + \frac{C^*}{|S|}\sum_{n=0}^{k-1}\left(1 - \frac{1}{|S|}\right)^{n+1} =$$

$$= \frac{C^*}{|S|} + \frac{C^*}{|S|}\sum_{n=1}^{k}\left(1 - \frac{1}{|S|}\right)^n = \frac{C^*}{|S|}\left(1 + \sum_{n=1}^{k}\left(1 - \frac{1}{|S|}\right)^n\right) = \frac{C^*}{|S|}\sum_{n=0}^{k}\left(1 - \frac{1}{|S|}\right)^n.$$

□

Theorem 1. *Algorithm 1 has approximation factor of* $1 + \ln(m)$.

Proof. Assume that $|S| = opt$, that is, $|S|$ is the optimal number of columns to cover all the rows. Given that all the rows must be covered, it is known that the the maximal covering with $|S|$ columns has $C^* = m$ elements.

We want to find α such that, if l denotes the number of iterations of the **while** loop, then $l \leq \alpha|S|$. It is known that

$$|C_{l-1}| \geq \frac{m}{|S|}\sum_{n=0}^{l-2}\left(1 - \frac{1}{|S|}\right)^n = \frac{m}{|S|}\frac{1 - \left(1 - \dfrac{1}{|S|}\right)^{l-1}}{\dfrac{1}{|S|}} = m\left(1 - \left(1 - \frac{1}{|S|}\right)^{l-1}\right).$$

Because $1 + x \leq e^x$, then $1 - \dfrac{1}{|S|} \leq e^{\frac{-1}{|S|}}$. Therefore, $\left(1 - \left(1 - \dfrac{1}{|S|}\right)^{l-1}\right) \geq 1 - e^{-\frac{l-1}{|S|}}$.

Note also that $|C_{l-1}| \leq m - 1$, given that the set of rows covered in the $(l-1)$-th iteration does not contain all rows, otherwise the **while** loop would have only $l - 1$ iterations (and not l). Hence,

$$m\left(1 - \left(1 - \frac{1}{|S|}\right)^{l-1}\right) \leq |C_{l-1}| \leq m - 1 \Rightarrow m - m\,e^{-\frac{l-1}{|S|}} \leq m - 1 \Leftrightarrow$$

$$\Leftrightarrow m\, e^{-\frac{l-1}{|S|}} \geq 1 \Rightarrow -\frac{l-1}{|S|} + \ln\left(m\right) \geq 0 \Leftrightarrow \ln(m) + \frac{1}{|S|} \geq \frac{l}{|S|}.$$

Therefore, $|S|\left(\ln(m) + \frac{1}{|S|}\right) \geq l$. Taking, $\alpha \leq \ln(m) + \frac{1}{|S|} \leq \ln(m) + 1$, we conclude that the approximation factor is indeed $\ln(m) + 1$. □

Theorem 2. *Algorithm 2 has approximation factor of* $1 + \ln(m)$*.*

Proof. The proof is similar to the previous. It is enough to take K_j as the largest number of rows yet to cover that can be covered by column $j \in S$, which in turn covers the row $i \in R$ that is covered by fewer columns. □