



Nesting algorithm for optimization part placement in additive manufacturing

Maurizio Calabrese¹ · Teresa Primo¹ · Antonio Del Prete¹ · Giuseppe Filitti²

Received: 4 May 2021 / Accepted: 24 September 2021 / Published online: 13 January 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

Significant savings in cost and time can be achieved in additive processes by manufacturing multiple parts in a single setup to obtain efficient machine volume utilization. In this paper, the authors have developed a previsional model able to evaluate the potential performance of various printing technologies for the execution of a given job. This model aims to support technicians in choosing the best solution starting from a specific machine architecture and printing volume. In particular, the model is able to evaluate, from a qualitative and quantitative point of view, the performance of each technology in a transversal manner, taking into consideration the aspects connected to printing: costs, time, and technological parameters. Within the core of the previsional model, there are multiple algorithms able to compute different key performance indicators (nine KPIs). For the computation of some of them, it was necessary to quantitatively evaluate aspects related to nesting operations or to the arrangement of several components within the printing base depending on the dimensional characteristics of the component, the printing direction, and its dimensional and geometric characteristics (rectangular or circular). Starting from this need, the developed nesting algorithm has given a specific answer.

Keywords Nesting · Additive manufacturing · Key performance indicator algorithm · Previsional model

1 Introduction

Additive manufacturing (AM), derived from rapid prototyping (RP), has been investigated and developed for more than three decades. It cannot only provide prototypes rapidly to support the product development, but also produce functional or end-use parts for different application areas [1–5].

Due to its unique processing manner, layer by layer, it owns a great advantage of manufacturing customized parts

with extremely complex geometries against traditional processes. Furthermore, AM technologies can realize manufacturing a group of parts with the same or different geometries in the same build platform simultaneously without using any tools or fixtures since multiple contours of different parts can be placed within one common slice/layer to be built. Therefore, it is a real and ideal technology for the “concurrent manufacturing” [6]. Significant savings in cost and time can be achieved in rapid prototyping (RP) by manufacturing multiple parts in a single setup to achieve efficient machine volume utilization [7]. Intuitively, to improve the machine utilization, more parts should be placed as compactly as possible to harness the build volume so as to reduce the total build time and cost per machine run. It seems that this is a classical nesting or packing problem. However, due to the special constraints of AM, the placement problem is different from other classical nesting or packing problems. When placing multi-parts into a build volume, not only the compactness should be maximized to reduce the total build time and cost, but also the part’s production quality should be guaranteed. In addition, the characteristics of AM processes, the features of part group, the production contexts of AM service bureaus, and the specific preferences and

✉ Teresa Primo
teresa.primo@unisalento.it

Maurizio Calabrese
maurizio.calabrese@unisalento.it

Antonio Del Prete
antonio.delprete@unisalento.it

Giuseppe Filitti
g.filitti@ingeniaepartners.com

¹ Department of Engineering Innovation, Università del Salento, Complesso Ecotekne—Edificio “Corpo O”—Via per Monteroni, 73100 Lecce, Italy

² Ingenia Research Institute, Via Marco Biagi, 73100 Lecce, Italy

requirements of users should be taken into consideration when doing the multi-part placement. Hence, these factors form the customized constraints of AM to make this problem a special variant of classical nesting or packing problem. Currently, due to the insufficient maturity of manufacturing functional parts and little research attention paid on the process planning or scheduling in AM, only a few solutions were proposed in literature to deal with the part placement problem. Till now, in AM service bureaus, the problem is mainly solved manually by skilled technicians who place parts as many as possible [8].

However, doing the part placement manually in a graphic environment is time-consuming, and it becomes more complicated when placing a batch of parts with a large quantity and very complex geometries [9]. Obviously, it is very difficult or even impossible for an operator to find an optimal part placement solution manually when facing such a nesting or packing (NP)-hard problem.

In many production processes, optimization of material usage by means of nesting plays an important role. Traditionally, 2D nesting has been an aspect of sheet metal manufacturing. With the increasingly mature application of layered manufacturing, 3D nesting has become an essential part of the workflow. The reason for this is that the quality of the 3D nest not only influences the required amount of raw material and the quality of the individual products, but also has a direct impact on the throughput time of a production batch [10].

Canellidis et al. studied the platform layout optimization problem for the simultaneous fabrication of different parts, which is addressed in the batch planning of stereolithography additive manufacturing technology. The methodology proposed employs a genetic algorithm technique for the 2D nesting of parts on the platform of the stereolithography machine [11].

Zhang et al. proposed a strategy composed of two main steps, applying an “AM feature-based orientation optimization method” to optimize each part’s build orientation to guarantee the production quality and applying a designed “parallel nesting” algorithm for increasing the compactness of placement by using the parts’ projection profiles so as to decrease the total build time and cost [12].

Bernard et al. introduce a new nesting scheme to more exactly describe this AM industry problem and then propose a parallel nesting method for both improving machine utilization and guaranteeing production quality [13]. Angelo et al. report a collection of the most significant published methods for the estimation of the factors that most influence the build direction is presented [14].

The build time and cost for each build cycle depend on nesting issues (build orientations, placement locations, and geometries of input objects) [15]. Nesting and scheduling problems in the AM literature can be categorized into three types: nesting for AM (NfAM) ([13, 16]); scheduling

for AM (SfAM) ([15, 17]); and nesting and scheduling for AM (NSfAM) ([18–21]). Compared with NfAM problems, SfAM and NSfAM problems have been highlighted recently.

In this paper, a new algorithm for 3D printing nesting has been presented, and it has been developed within the definition of a previsual model that aims to evaluate the performance to manufacture a particular component comparing different available 3D printing technologies.

For this reason, the main purpose of this innovative algorithm is providing to previsual model a worthy solution to compute a possible nesting configuration for a given platform. Therefore, this algorithm has been developed considering the production of medium/large lots of components in which these last have the same dimensions. In particular, it does not contemplate the possibility of engaging components of different dimensions because this is a typical issue that the technicians face during the job setup considering a particular technology and production needs in terms of queue to manage.

The presented algorithm aims to provide a possible layout available to the components on the printing surface based on the geometric characteristics without the use of iterative processes that characterize the most common nesting techniques.

Therefore, what distinguishes this algorithm is the possibility to estimate a feasible number of printable components (for a given printed bed) without having to resort to iterative processes.

The proposed methodology aims to be integrated within innovative systems that exploit the cloud manufacturing paradigm. Specifically, the algorithm was specially developed with technologies integrable within different software structures.

In fact, for system oriented to cloud manufacturing, where there are several technological resources (manufacturing resources, IT resources, etc.), for additive manufacturing service, it is extremely important to have some integrated tool/applications that support the technicians to define a quick quotation for a given order. In this context, the developed tool provides a great advantage because it provides, in quick times, an estimation of printable components for a given 3D printing technology. This parameter is extremely important for aspects concerning the economy of scale. Indeed, this algorithm is integrated in a previsual model that provides information about technological and economic aspects.

The algorithm was designed to be implemented within DSS-oriented software architectures or decision support systems designed for the context of additive manufacturing. Specifically, these solutions aim to assist technicians in product–process development.

In this context, the integration of the developed algorithm would allow technicians to have a function able to estimate, during the preliminary phase of product development, the number of components that can be printed in a single job and, therefore, also evaluate the connected aspects to costs.

The objective was, therefore, to obtain an effective tool able to compute a possible nesting configuration (number of components that it is possible to print with a given 3D printing technology and with a single job) to feed the prevision model in order to compute several key performance indicators (KPIs) that allow to evaluate the available technologies in terms of cost estimation and technological compatibility.

2 AMSA methodology

The prevision model and the nesting algorithm have been implemented as part of a research project Additive Manufacturing Spare parts market Application (AMSA). This project has developed a tool, called AMSA, having as main goal the definition of a common platform to supply different kinds of services: product development oriented to the additive manufacturing (DFAM, design for additive manufacturing), production of prototype or small series with additive manufacturing, and reverse engineering activities to obtain CAD models starting from a physical object. The definition of different kinds of services allows to satisfy several client needs such as necessity to define an innovative product characterized by high performance in terms of stiffness/weight ratio, possibility to manufacture small series, such as in the motorsport field, and possibility to define CAD models for the obsolete parts for which the geometrical information are missed. AMSA platform relies on the reconfigurable supply chain that is dynamic, and it depends on the particular client needs. For example, when the client requires the manufacture of a small series of a particular component, AMSA allows the technicians to choose the best solutions in terms of price and distance. Therefore, the suppliers that contribute to the definition of the dynamic supply chain have an important role. The AMSA platform, for these reasons, represents an important tool able to link the suppliers to the customers in the best manner in order to obtain services characterized by a high performance level.

In fact, it provides support to the AMSA operator in appropriately evaluating supplier selection logics based on the characteristics of the geometry to be built.

Therefore, a methodology has been developed that is able to extrapolate and to cross a series of information able to provide, for each component to be produced through additive manufacturing, a series of key information:

- The most suitable technology
- The suggested machine
- The production time
- The production cost

Considering the large number of variables to be managed, it was decided to propose a methodology based on the selection

of a series of KPIs. The purpose of the latter is to provide a compatibility index of each machine (no technology) with respect to the component requirements to be produced.

The KPIs have percentage values from 0 to 100% and different calculation methods depending on the considered case. Once the KPIs have been calculated, the platform has a list of solutions ordered according to a compatibility KPI obtained as an average (appropriately weighted) of the other available KPIs.

3 Key performance indicators

Key performance indicators represent a synthetic method to collect information and correlate it with each other. The developed prevision model allows to filter the technologies according to the appropriate performance indexes (KPIs) reported below [21, 22]:

1. CST: production cost
2. MAT: material
3. TMP: production time
4. ING: overall dimensions of the component in the machine
5. PRE: technology precision
6. RIS: technology resolution
7. STQ: undercuts management
8. RGS: technology roughness
9. CBA: technology compatibility with the component to produce

The indexes listed above, as it can be seen from the synthetic description, analyze different technological aspects related both to the technical characteristics of the machines but also to the qualitative aspects of the components that must be realized.

The objective of this prevision model is to assist AMSA technicians in choosing, among accredited suppliers to the platform, the most advantageous solutions considering the aspects that distinguish the production of a particular component.

The main KPIs are certainly CST, TMP, and CBA; however, even the geometric KPIs could be useful to allow the AMSA operator to make the necessary evaluation. For a detailed description of KPI, see Appendix 1.

4 Nesting algorithm

Within the prevision model definition, aimed at evaluating the performance of printer technologies, based on the characteristics of the component to be manufactured, it was necessary to develop an algorithm for the management of the aspects connected to the nesting of the components on the printer plan produced by additive manufacturing.

In particular, this need emerges in the calculation of the following indices:

1. CST: production cost
2. TMP: production time
3. ING: overall dimensions of the component in the machine
4. CBA: technology compatibility with the component to produce

5 Consideration on INPUT parameters for index calculation

In the listed indices the variable “ m ” appears, which represents the number of components that can be positioned on the printer plan. This variable represents, precisely, the nesting activity and requires the development of an appropriate algorithm that mainly defines the following data:

- Printer plan type: rectangular or circular
- Printer plan dimensions
- Component plan dimensions

This allows the component number calculation that can be positioned on the printer surface (Fig. 1).

The rectangular and circular case studies concern the printing plans that can be found in the main commercial 3D printing systems; therefore, the cases analyzed may well be representative of the possible solutions that the algorithm must be able to manage.

Therefore, based on the previously mentioned need, to quantify the number of components that can potentially be positioned on a printer plan, taking into account the component dimensions, an algorithm has been developed that has been implemented in the previsual model which in turn has

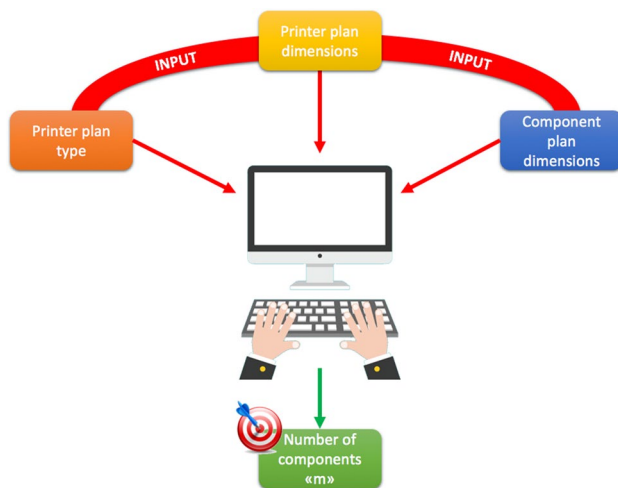


Fig. 1 Definition of the nesting algorithm

been integrated into the AMSA platform. The algorithm has a top-down structure as it is not an iterative algorithm but is based on the use of parameterized geometric data, with the aim of providing an estimate quickly and sufficiently reliable. Therefore, there are no iterations on the parameters as the definition of the latter is made upstream of the algorithm. Furthermore, the algorithm does not deal with managing the qualitative aspects of the printed components.

In fact, the technicians, to use the application, will insert some input parameters such as distance between the parts and distance from the edges of the printing surface. This technological information are available to technicians because they are linked to the production context know-how.

6 Nesting algorithm definition

The Nesting algorithm can be briefly summarized with the following diagram (Fig. 2).

Obviously, the scheme shown represents a simplification but allows to effectively understand the layout of the same and what are the inputs in detail.

In particular, it appears evident that the nesting algorithm can be considered as composed of sub-algorithms that allow, in turn, to compute the number of components in the following cases: rectangular and circular printer plans.

The input data shown in the diagram are detailed below:

- $x = X$ dimension of the printer plane.
- $y = Y$ dimension of the printer plane.
- $gapB$ = offset with respect to the printer plane border.
- $gapX = X$ distance among the components.
- $gapY = Y$ distance among the components.
- $xc = X$ dimension of the component bounding box.
- $yc = Y$ dimensions of the component bounding box.

As can be seen from the input data, there are information about:

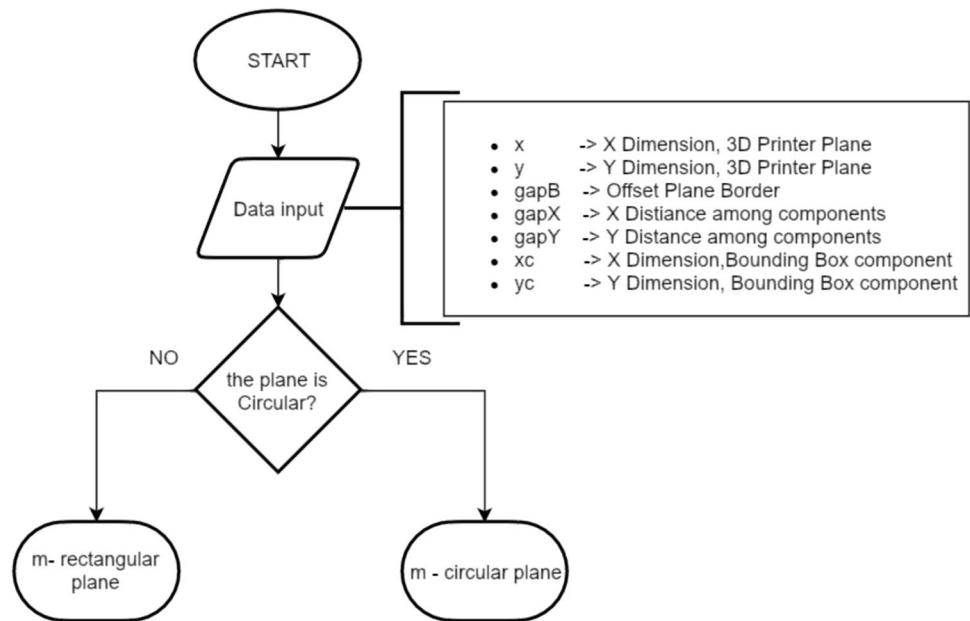
- Printer plane dimensions
- Component dimensions
- Positioning distances

Below will be described in more detail the two sub-algorithms that make up the general nesting algorithm.

7 Nesting algorithm definition for CIRCULAR printing plan

This section reported the description of the computation algorithm of the component number in the case of a circular printing plan. Figure 3 shows the summary workflow.

Fig. 2 Nesting algorithm diagram



The circular printing plan is obviously identified by a radius (Fig. 4), and, for this reason, when the data of the printing plan are inserted, in the case of a circular plan, the following condition will be obtained: $X = Y = r * 2$ ($r = \text{radius}$). Figure 5 also shows the previously indicated offset “ $gapB$ ” (Fig. 3) and the difference “ $r - gapB$ ” which represents the radius for the calculation of the useful surface for the components positioning.

In the positioning calculation, the algorithm assumes that the first component is positioned in the center of the printing plane as shown in Fig. 5.

Further, the algorithm considers that the bounding box dimensions are parallel to the axes. Considering the printing plane symmetry (being circular) the same result is obtained for the following scenarios:

- xc parallels to the x axis and yc parallels to the y axis.
- xc parallels to the y axis and yc parallels to the x axis.

Therefore, the positioning shown in Fig. 5 is considered as default. Starting from this assumption, the algorithm calculates a potential number of components that can be inserted to cover the entire printing plan, from the dimensions in plan of the component and from the printing plan dimensions considering a determinate extrusion direction. In particular, the technicians, for a given component, can choose a particular extrusion direction, and the algorithm considers the other directions to define the dimensions in plan.

$$P1ij = \left(\left[\left(\frac{xc}{2} + \Delta x \right) + (xc + \Delta x) * (Nx - 1) \right], \left[\left(\frac{yc}{2} + \Delta y + yc \right) + (\Delta y + yc) * (Ny - 1) \right] \right)$$

Therefore, the following functions are present:

$$NumberCWX = int\left(\frac{\left(r - gapB - \frac{xc}{2}\right)}{(xc + gapX)}\right) + 1$$

$$NumberCWY = int\left(\frac{\left(r - gapB - \frac{yc}{2}\right)}{(yc + gapY)}\right) + 1$$

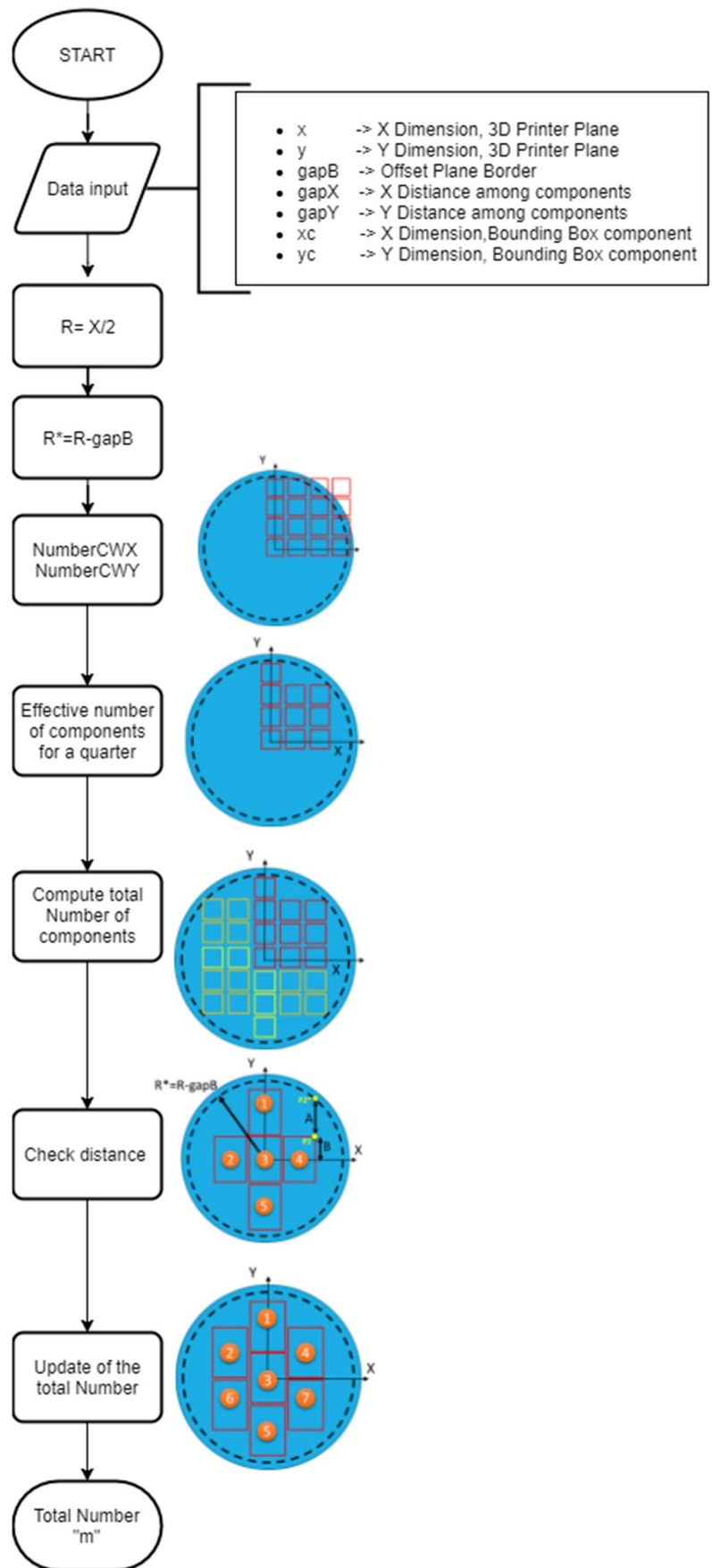
These functions allow you to calculate the maximum number of components in the X and Y directions (Fig. 6).

With the calculation of the components in the two directions, the algorithm, in the preliminary phase, computes the scenario shown in Fig. 7. The figure shows how there are components, according to this computation, which are obviously not in the print domain as the plan is circular. Another aspect that is highlighted in this case is how, in the first phase, the calculation concerns only a quarter of the printing plan. This facilitates the algorithm in terms of calculation times.

In order to understand what the components are actually contained within the printing plan, in the algorithm, a set of parametric equations have been integrated in order to monitor each vertex of the rectangles that identify the plan surface of the component overall volume and to identify if a component is within the printing area or not.

Below are the formulations of these equations:

Fig. 3 Circular nesting workflow



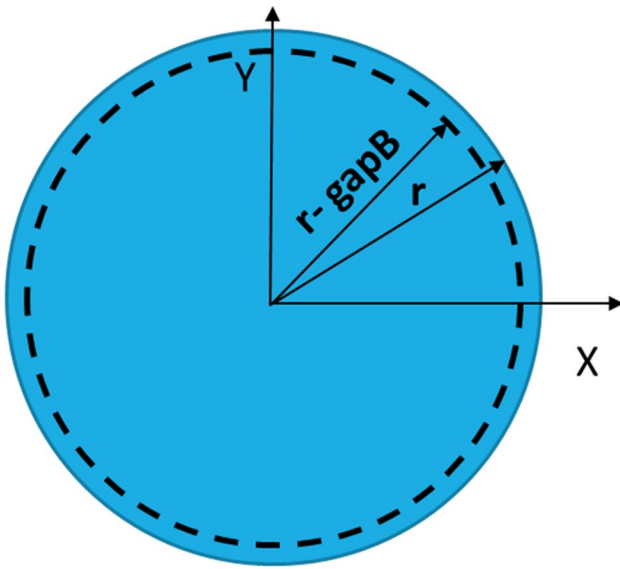


Fig. 4 Circular printing plan

from the center lower than the difference between the radius and the offset “gapB”; rather, the following conditions are valid simultaneously:

- $D1i \leq (r - gapB)$
- $D2i \leq (r - gapB)$
- $D3i \leq (r - gapB)$
- $D4i \leq (r - gapB)$

where $D1i$ represents the distance of vertex 1 belonging to component i . After checking how many components with respect to the conditions listed above, the result obtained is reported in Fig. 9.

After the components have actually been computed within the printing plan for the quarter surface, the components for the entire surface are then evaluated as shown in Fig. 10. In particular, these components are highlighted in yellow to indicate the rows belonging to the x and y axes, and the components replicated in the respective quadrants are indicated in orange.

This calculation has been further refined if there is a sin-

$$P2ij = \left(\left[\left(\frac{xc}{2} + \Delta x + xc \right) + (xc + \Delta x) * (Nx - 1) \right], \left[\left(\frac{yc}{2} + \Delta y + yc \right) + (\Delta y + yc) * (Ny - 1) \right] \right)$$

$$P3ij = \left(\left[\left(\frac{xc}{2} + \Delta x + xc \right) + (xc + \Delta x) * (Nx - 1) \right], \left[\left(\frac{yc}{2} + \Delta y \right) + (\Delta y + yc) * (Ny - 1) \right] \right)$$

$$P4ij = \left(\left[\left(\frac{xc}{2} + \Delta x \right) + (xc + \Delta x) * (Nx - 1) \right], \left[\left(\frac{yc}{2} + \Delta y \right) + (\Delta y + yc) * (Ny - 1) \right] \right)$$

The equations refer to the vertices shown in Fig. 8.

In particular, the equations listed above describe the coordinates of all 4 vertices for each rectangle positioned on the printing plane. For each vertex, it is necessary to calculate the distance. A rectangle (component) is inside the printing plane, if and only if, all 4 vertices have a distance

gle row of components in the x direction of the printing plane. It is necessary to consider this scenario in order to make the algorithm more robust.

In fact, if it is considered the scenario of Fig. 11, the solution described above considers the insertion of only 5 components.

However, this solution does not appear to be robust as, as can be seen in Fig. 11, if components 2 and 4 were properly translated, it would be possible to insert further components.

Figure 12 shows the A and B dimensions defined as:

- A: distance between point $P2^*$ (projection in the vertical direction of point $P2$ on the circumference identified by $R^* = (R - gapB)$) and point $P2$, rather

$$A = dist(P2^*P2)$$

- B: half dimension in Y direction of the component overall volume

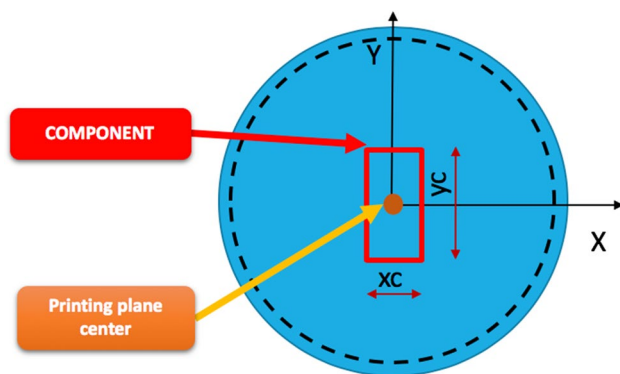
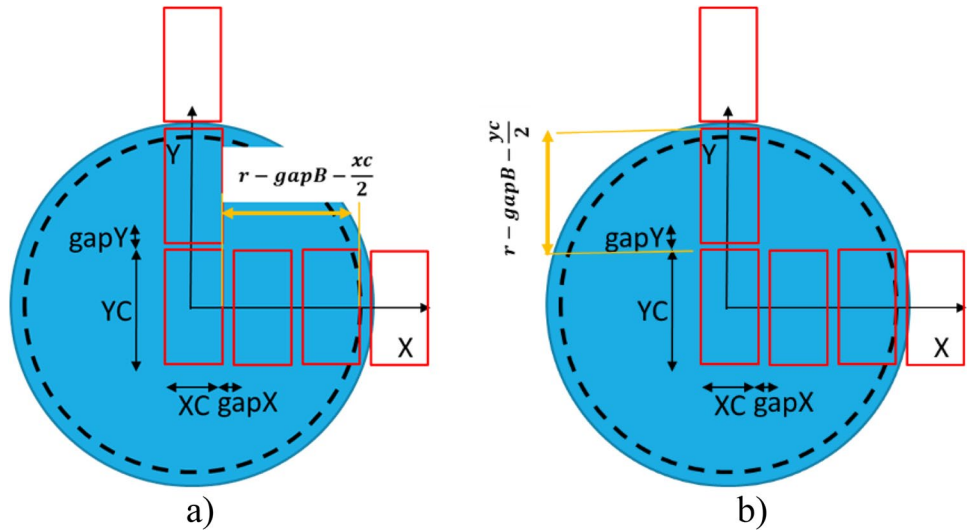


Fig. 5 Circular printing plan — positioning

Fig. 6 a NumberWX calculation; b number WY calculation



$$B = \frac{yc}{2}$$

In order to insert an additional component, distance A must be greater than B, rather

$$A \geq B$$

$$dist(P2*P2) \geq \frac{yc}{2}$$

If this condition is verified, the algorithm requires the insertion of 2 components instead of 1 as shown in Fig. 12.

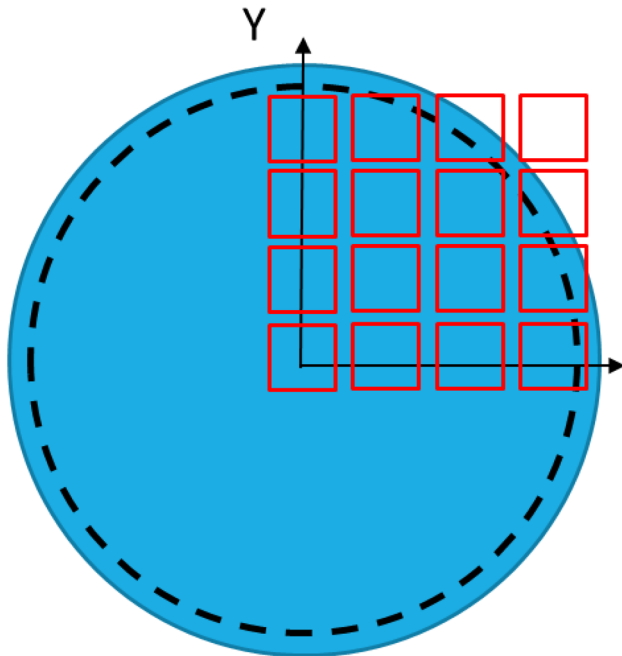


Fig. 7 Circular printing plan — maximum number of components

8 Nesting algorithm definition for RECTANGULAR printing plan

This section reported the description of the computation algorithm of the component number in the case of a rectangular printing plan. Figure 13 shows the summary workflow.

The algorithm is more efficient in the potential component number definition for the rectangular printing plan.

Figure 14 shows the offset “gapB,” previously indicated as in the case of the circular printing plan, and the axes origin necessary for the number of component calculation.

It is assumed that the bounding box dimensions are parallel to the axes. The scenarios that can be presented are the following:

- xc parallels to the x axis and yc parallels to the y axis.

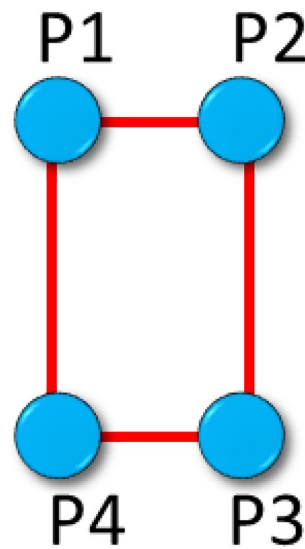


Fig. 8 Control vertices

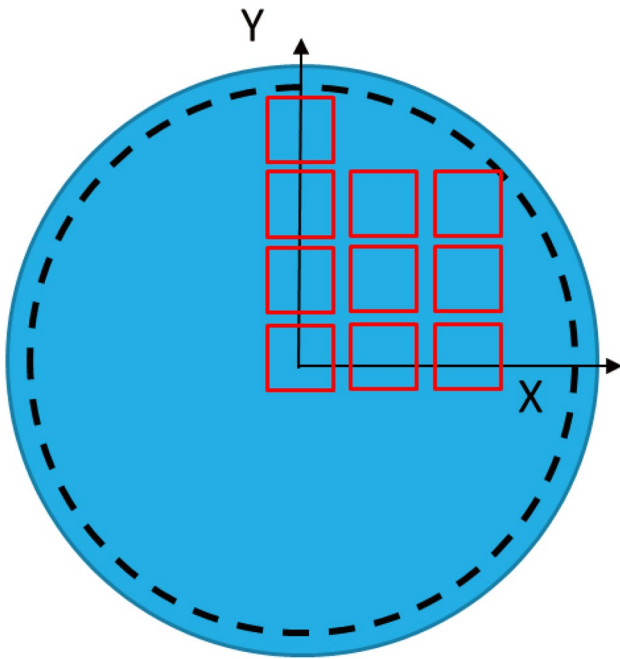


Fig. 9 Result of the check of the components inside the circular printing plan

- x_c parallels to the y axis and y_c parallels to the x axis.

The choice of the scenario to be considered in the component number definition is defined by the algorithm based on the potential number of allocable components. In particular,

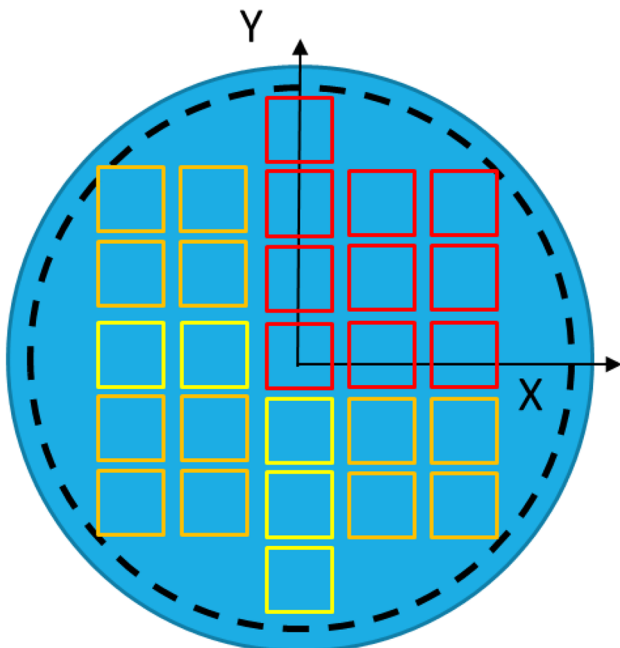


Fig. 10 Result of the check of the components inside the whole circular printing plan

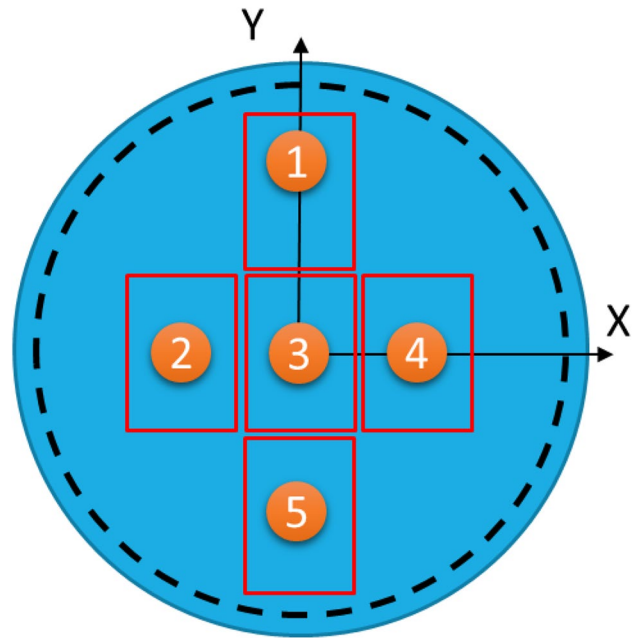


Fig. 11 Scenario for which further implementation is needed

it is considered as a reference condition that between the two, it is possible to insert a greater number of components.

Starting from this assumption, the algorithm calculates the maximum number of components that can be inserted, to cover the entire printing plan, for both scenarios, from the dimensions in plan of the component overall volume and from the printing plan dimensions, using the following functions:

$$NumberRWX = \text{int}\left(\frac{(x - (gapB * 2))}{(xc + gapX)}\right)$$

$$NumberRWY = \text{int}\left(\frac{(x - (gapB * 2))}{(xc + gapY)}\right)$$

These functions allow to calculate the maximum number of components in the x and y directions for the two orientations suggested above (1 and 2).

The possible cases are reported in Table 1.

The component number will therefore be equal to

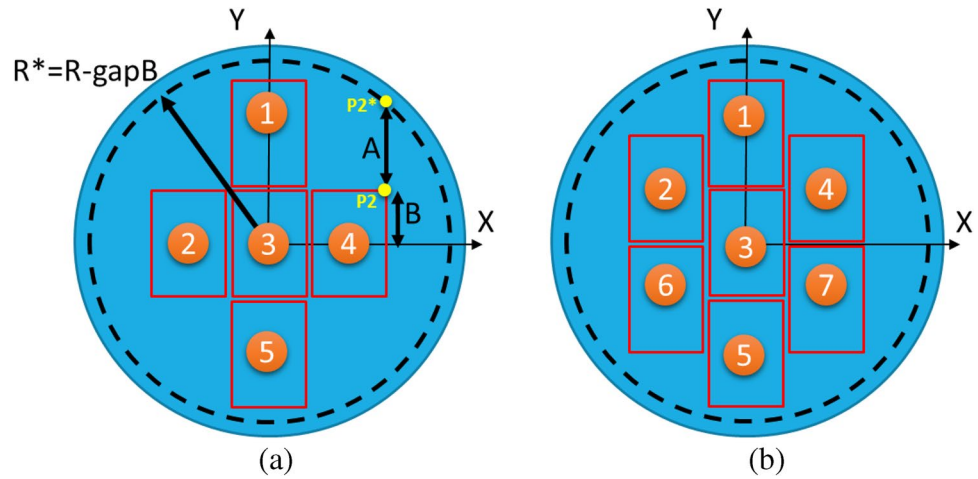
$$N = NumberRWX * NumberRWY$$

The number of components is calculated for both scenarios and is equal to

$$Numerodeicomponenti = \text{Max}(N_A, N_B)$$

The positioning of the single overall volumes in plan is governed by the following parametric equations:

Fig. 12 **a** Size check. **b** New layout



$$P1ij = \{(xc + \text{gap}X + xc) + [(xc + \text{gap}X) * (i - 1)]\},$$

$$\{(yc + \text{gap}Y + yc) + [(\text{gap}Y + yc) * (j - 1)]\}$$

$$P2ij = \{(xc + \text{gap}X) + [(xc + \text{gap}X) * (i - 1)]\},$$

$$\{(yc + \text{gap}Y + yc) + [(\text{gap}Y + yc) * (j - 1)]\}$$

$$P3ij = \{(xc + \text{gap}X) + [(xc + \text{gap}X) * (i - 1)]\},$$

$$\{(yc + \text{gap}Y) + [(\text{gap}Y + yc) * (j - 1)]\}$$

$$P4ij = \{(xc + \text{gap}X + xc) + [(xc + \text{gap}X) * (i - 1)]\},$$

$$\{(yc + \text{gap}Y) + [(\text{gap}Y + yc) * (j - 1)]\}$$

The equations refer to the vertices shown in Fig. 15.

In particular, the equations listed above describe the coordinates of all 4 vertices for each rectangle positioned on the printing plane.

Figure 16 shows the component positioning phase on the rectangular printing plane considering the mutual distances between the components themselves in the x and y direction.

Nesting management in the case of the rectangular printing plan entails the need to consider two specific cases within the algorithm.

The first case, reported in Fig. 17, foresees, as orientation, the following scenario (scenario 1):

- xc in the X direction
- yc in the Y direction.

In this case, it could be verified that the insertion of a further row of components in the Y direction is not physically possible. However, this scenario does not allow a priori to exclude the possibility of inserting additional components by rotating them. It is possible to rotate the components and insert additional files. The condition for this event to occur is described below.

First of all, the number of possible rows, with components rotated with respect to the general orientation, defined according to the following formula, is calculated:

$$\text{Numerodifile}(Nf) = \text{int}\left(\frac{A}{(xc + \text{gap}Y)}\right)$$

$$A \geq (xc * Nf) + (\text{gap}Y * (Nf - 1))$$

Figure 18 shows the completion of the nesting phase with all the possible components that can be inserted on the printing table according to the defined parameters.

The second case, shown in Fig. 19, provides, as a guideline, the following scenario (scenario 2):

- xc in the Y direction
- yc in the X direction

In this case, it could be verified that the insertion of a further row of components in the X direction is not physically possible. However, this scenario does not allow a priori to exclude the possibility of inserting further components by rotating them. It is possible to rotate the components and insert additional files. The condition for this condition to occur is described below.

First of all, the number of possible files with components rotated with respect to the general orientation, defined according to the following formula, is calculated:

$$\text{Numerodifile}(Nf) = \text{int}\left(\frac{A}{(yc + \text{gap}X)}\right)$$

$$A \geq (yc * Nf) + (\text{gap}Y * (Nf - 1))$$

Figure 20 shows the completion of the nesting phase with all the possible components that can be inserted on the printing table according to the defined parameters.

Fig. 13 Rectangular nesting workflow

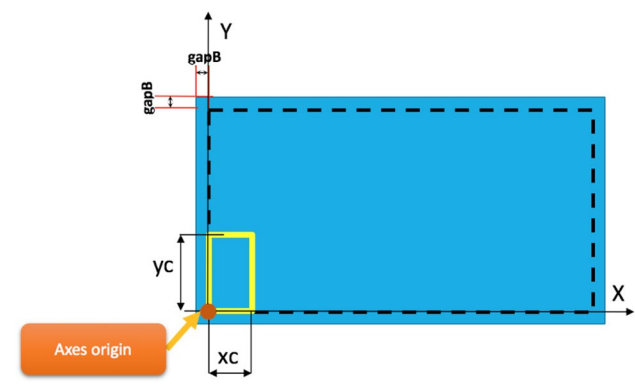
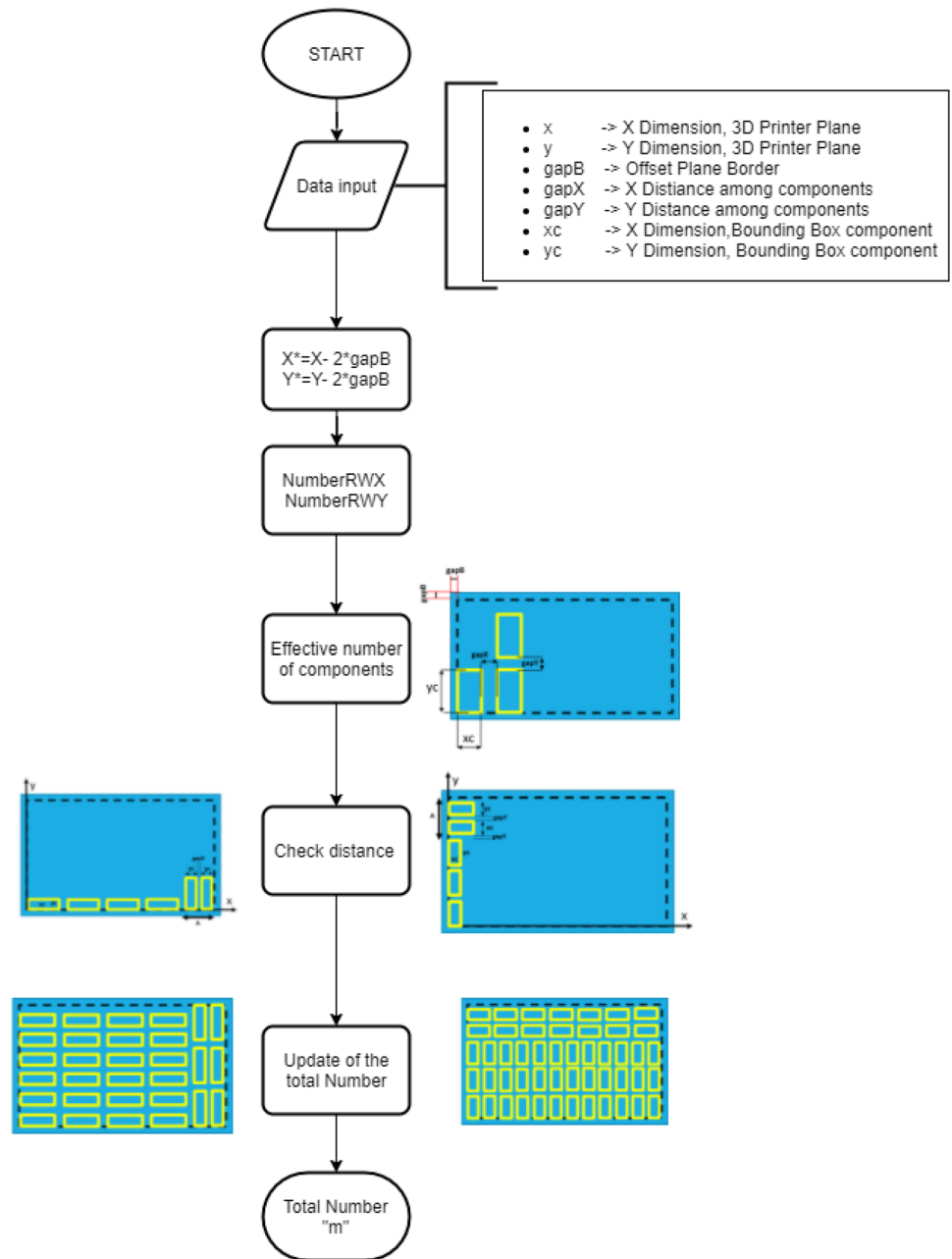


Fig. 14 Rectangular printing plan

9 Results and discussion

The implementation of the nesting algorithm has been structured in two different steps:

Table 1 Number of component calculation for scenarios 1 and 2

Scenario 1	Scenario 2
$NumberRWX = \text{int}\left(\frac{(x - (gapB * 2))}{(xc + gapX)}\right)$	$NumberRWX = \text{int}\left(\frac{(x - (gapB * 2))}{(yc + gapX)}\right)$
$NumberRWY = \text{int}\left(\frac{(x - (gapB * 2))}{(xc + gapY)}\right)$	$NumberRWY = \text{int}\left(\frac{(x - (gapB * 2))}{(xc + gapY)}\right)$
xc parallel to the x axis and yc parallel to the y axis	xc parallel to the y axis and yc parallel to the x axis

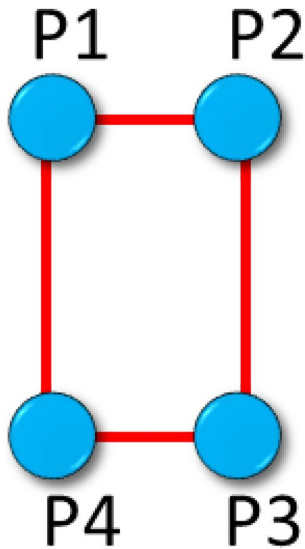


Fig. 15 Control vertices

- Step 1, algorithm implementation using the TCL language (tool command language)
- Step 2, porting of the algorithm to the Java language (Appendix 2)

The choice to use the TCL language was made due to the availability of a development environment, which provides tools able to plot the results giving, therefore, the possibility to graphically verify the quality of the solution (Fig. 21). The figure also shows an example of application of the algorithm on a real mechanical component characterized by dimensions in plan equal to 100×200 mm (<https://grabcad.com/>) [23].

In Fig. 21d it is highlighted the output of the algorithm for the considered real component.

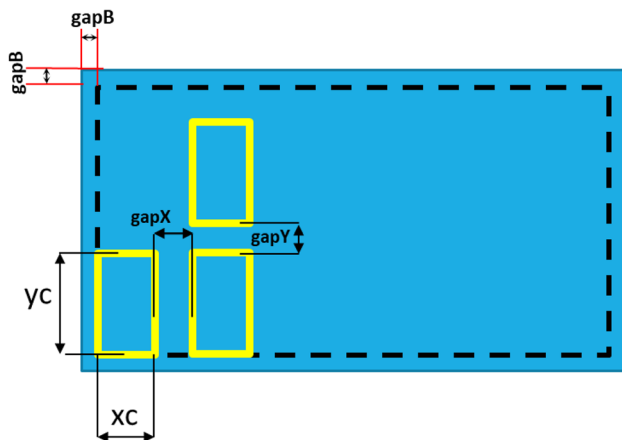


Fig. 16 Rectangular printing plan — positioning

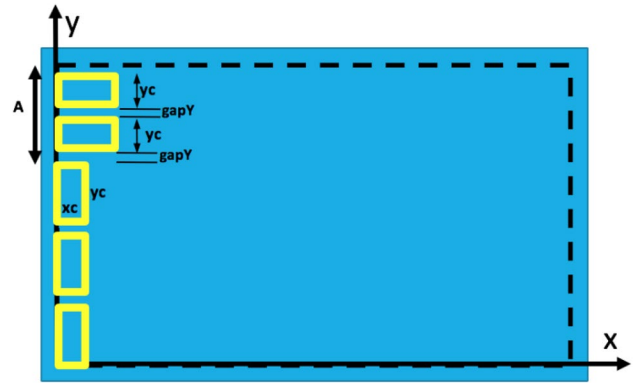


Fig. 17 New positioning, scenario 1

This development environment is a CAE tool, Altair® HyperMesh®, which allows to literally design the component layout on the printing plan through appropriate APIs available. The language with which it is possible to interact with the APIs present is TCL [24].

In step 2, after the development and testing of the algorithm, thanks also to the visualization of the results in the HyperMesh® environment, they were ported to Java. This choice was dictated by the implementation requirements for its integration into the AMSA platform.

The developed algorithm aims to address the problems associated with nesting operations for 3D printing technologies. Several nesting algorithms are present in the literature, many of which are based on optimization algorithms that often require different iterations to obtain the final configuration.

The present algorithm, although it is not based on optimization algorithms, turns out to be a technologically solution as it allows to provide an estimate of the components that can be positioned on the printing plan (circular or rectangular) which is for the purposes of estimating the production costs.

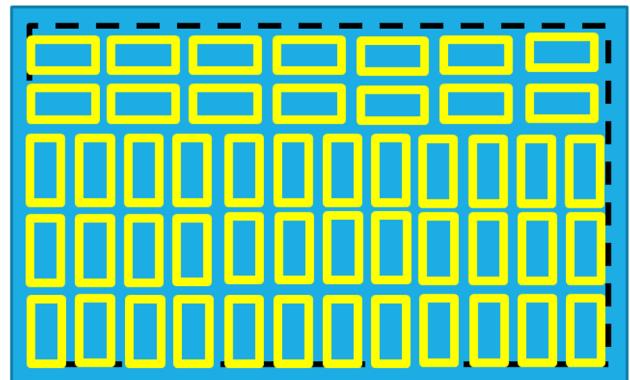


Fig. 18 New complete positioning, scenario 1

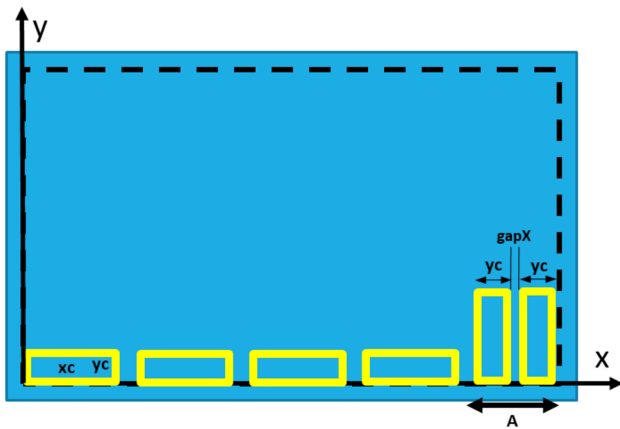


Fig. 19 New positioning, scenario 2

The main advantage, compared to the more expensive optimization algorithms, lies in the fact that, based on geometric considerations, it is able to provide an estimate in a very short time and, therefore, the technician has an available immediate tool for estimating the components that can be placed on any printing plan with the only condition of having the dimensions of the printing plan itself and of the component for a given orientation.

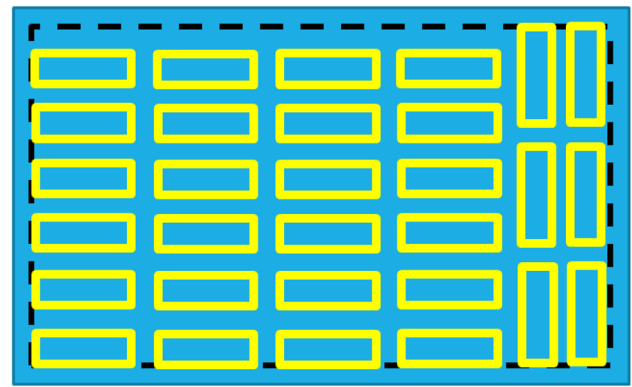


Fig. 20 New complete positioning, scenario 2

The analyzed case studies within the paper refer to the bounding box of the possible components to be printed and not to the actual geometries. It is important to highlight that the bounding box plays an important role in defining the number of components that can be printed on a given printer plane.

Through the bounding box, the maximum dimensions in plan of a given component and the maximum height defined at a given inclination are defined. The technology

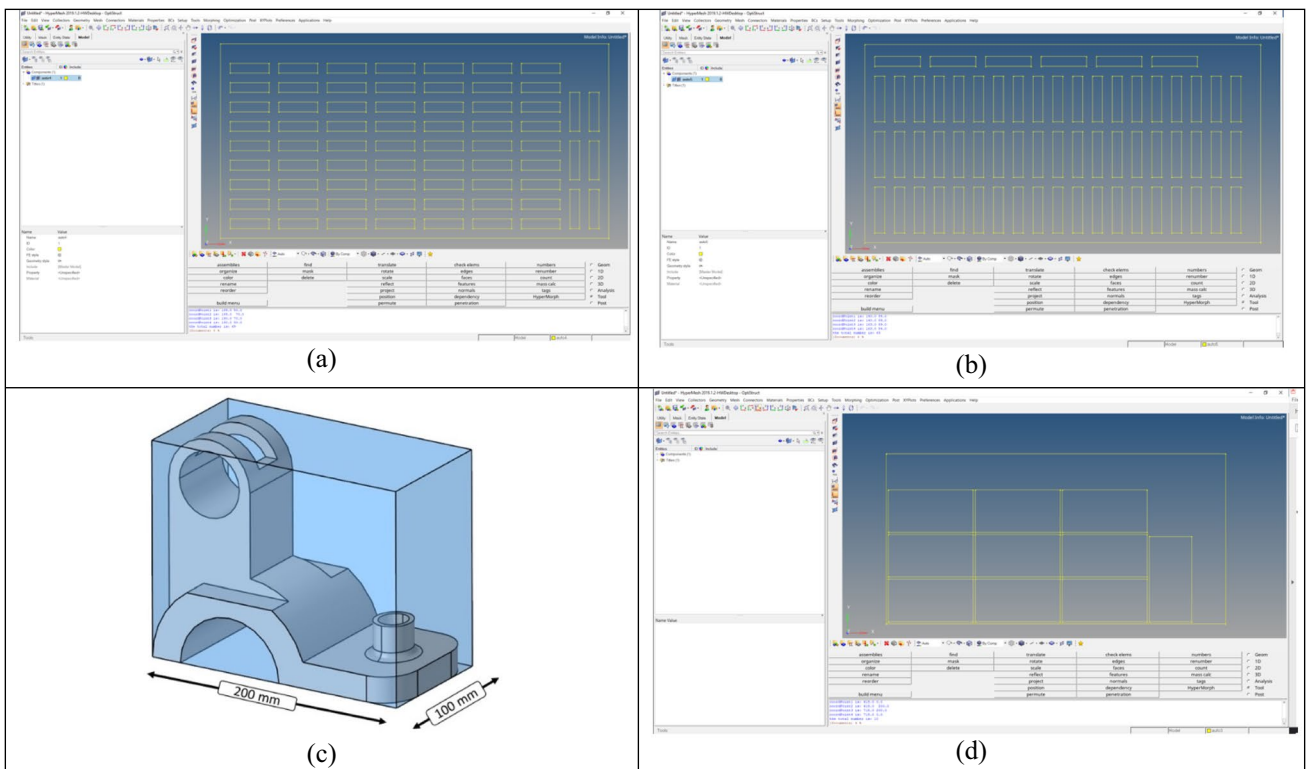


Fig. 21 Graphic visualization in HM environment (a) and (b) example of graphic verification

was also developed for powder bed 3D printing systems, such as SLM, and, specifically, the definition of the reference bounding boxes for testing the algorithm was conducted in collaboration with printing centers. The latter, possessing the technologies at home, have transferred their needs and the most relevant technological information to the authors for the development of a method that allows to obtain, in a very short time, a hypothesis of components distribution on the printer plane.

The algorithm was not designed to be integrated into commercial 3D printing systems, as its goal is to be implemented in the future, in solutions that are placed downstream of the production system and that want to assist the decision-making process in the preliminary product development phases.

It is important to highlight that, as for the classic nesting techniques based on iterative processes, in the case study, in the first analysis, the qualitative aspects were not considered, as the tool was designed to assist technicians in the cost estimation and not in the product–process development phase. However, the authors are conducting additional studies aimed at improving the algorithm by introducing additional technological parameters typical of the most common additive manufacturing processes, specifically SLM. The innovation introduced by this algorithm is to provide a solution which, although based on mathematical and geometric rules, can be integrated into solutions based on cloud manufacturing. In fact, this solution was developed specifically for an application of the cloud manufacturing paradigm, and, compared to classic nesting systems, it is expressly designed for 3D printing.

10 Conclusions

The technical specification definition for the prevision model implementation has led the need to develop a nesting algorithm for the computation of the printable component number on a given printing plan. This variable is necessary for some KPI calculation that allows to evaluate in advance the qualitative aspects of the components obtained for each printing technology. This tool allows AMSA technicians to choose the best performing technologies to request a quotation for a particular component. The objective has not been to provide a tool properly dedicated to the proper nesting of the components on the printing plane (this is a typical issue that the technicians face during the job setup considering a particular technology and production needs in terms of queue to manage), but the objective was to estimate the component number for a consequent estimate of the costs for a given machine park. The algorithm was not designed to be integrated into commercial 3D printing systems, as its goal is to be implemented in the future, in solutions that

are placed downstream of the production system and that want to assist the decision-making process in the preliminary product development phases.

Currently, the algorithm has been developed to work in the XY plane taking into account the needs expressed by industrial partners that collaborate in the research activities. Indeed, the industrial partners work with different 3D printing technologies such as SLM and DMD, and they have a consolidate know-how about the additive manufacturing processes. However, among future developments, the integration of the Z axis within the algorithm is expected.

The objective was, therefore, to obtain an immediate tool at the service of the calculation algorithms of the cost estimate and of the technological compatibility of the available machines for a given component.

This algorithm represents a flexible solution that it is possible to implement within other prevision model or framework to evaluate the 3D printer performance considering as AM becomes increasingly widespread and taking into account to have a useful tool to quickly evaluate a potential number of a particular component to print with a determinate machine and for a given extrusion direction. The developed algorithm is a utility that can be integrated into any platform. In fact, it was created to be integrated into solutions that exploit the cloud manufacturing paradigm and therefore was not designed to be integrated within real CAD systems but rather for web-based platforms that also act as decision support systems (DSS). The software was developed using various technologies, one of which listed in the article is Java which stands out for its extreme flexibility and integrability with any system.

Appendix 1

To better understand the details, the complete formulation of the indexes listed above is shown below.

Material index (MAT)

This index evaluates the effects of compatibility between the material requested by the customer and the analyzed AM machine. The following is the complete formulation of the MAT index with the relative control variables:

$$MAT = \theta_{mat} \cdot \left(\alpha_{appr} \cdot \frac{1}{\sqrt{1 + \tau_{apprw}[g]}} + \alpha_{mt} \cdot \frac{1}{\sqrt{1 + \tau_{AMmt}[g]}} + \alpha_{fr} \cdot (2 - f_r) \right) \quad (1)$$

- θ_{mat} : is the compatibility between the material and machine.
- α_{appr} : is the metal powder supply coefficient.

- $\tau_{appr_{w[gl]}}$: is the supply time for metallic powders [days].
- α_{mt} : is the material change coefficient.
- $\tau_{AM_{ml[gl]}}$: is the material change time [days].
- α_{fr} : is the material processability risk coefficient.
- f_r : is the risk factor of material processability, which represents a factor for increasing the mass of material required by the process, and it is requested to the supplier.

Production cost index (CST)

The objective of this KPI is to provide an indicative quotation of the product according to the indicated technology. The aspects considered by the KPI are costs due to the:

- Material
- Use of the machine
- Operator
- Geometry complexity

Below, we describe the complete formulation of the CST index with the list of related control variables:

$$\begin{aligned}
 CST = & \theta_{mat} \cdot \theta_{ing} \cdot \left\{ \left[(n \cdot \delta_d) \cdot \left(\frac{V_d}{\epsilon_{w_d}} + \frac{S_d \cdot \gamma_{rt} \cdot \Phi_w}{\epsilon_{w_r}} \right) \right] \cdot C_w \right. \\
 & + [n \cdot \delta_t \cdot V_{bound_t} \cdot \Phi_t] \cdot C_t + [n \cdot \mu_g \cdot S_{bound} \cdot (1 + \Phi_g) \cdot c_{dot} \delta_g] \cdot C_g \left. \right\} \\
 & c_{dot} f_r + \left\{ K_d c_{dot} (c_{d_{op}} \cdot (\tau_{d_{or}} + \tau_{d_{sl}} + \tau_{d_{ps}} + \tau_{d_{cm}})) + K_{AM} \right. \\
 & \cdot \left[(c_{AM_{eq}} + c_{AM_{mh}} + c_{AM_{en}} + c_{AM_{op}}) \cdot (c \tau_{AM_{mt}} + (\tau_{AM_{start}} + \tau_{AM_{risc}}) \right. \\
 & \left. \left. \cdot \left[\frac{n}{m} \right] + \frac{V_d}{P_{AM}} \cdot n + \tau_{AM_{cc}} \cdot \left(\left[\frac{n}{m} \right] - 1 \right) + (\tau_{AM_{raffr}} + \tau_{AM_{clean}}) \cdot \left[\frac{n}{m} \right] \right] \right\}
 \end{aligned} \tag{2}$$

- θ_{mat} : is the compatibility between material and machine.
- θ_{ing} : is the compatibility between component and the machine working volume.
- δ_d : is the material density indicated by the customer (kg/m³).
- V_d : is the volume of the component to be produced (m³).
- S_d : is the total area of the component (m²).
- γ_{rt} : is the average machining allowance thickness identified by technology (m).
- Φ_w : is the ratio between the surface subject to machining allowance and the total surface area of the component (%).
- ϵ_{w_r} : is the powder management efficiency in the production of machining allowance (%).
- ϵ_{w_d} : is the powder management efficiency in the production of the component (%).

- C_w : is the unit cost of metal powders (€/kg).
- δ_t : is the material density chosen for the supports (kg/m³).
- V_{bound_t} : is the parallelepiped volume of the containment of support structures (m³).
- Φ_t : is the technological coefficient for the mass of material for the supports (%).
- C_t : is the unit cost of the material required for the supports (€/kg).
- δ_g : is the material density of the anchor plate (kg/m³).
- μ_g : is the thickness of the anchor plate (kg/m³).
- S_{bound} : is the surface of the component’s bounding box (kg/m³).
- Φ_g : is the increase coefficient of the bounding box surface (%).
- f_r : is the material risk coefficient.
- C_g : is the unit cost of the anchor plates (€/kg).
- K_d : is the complexity coefficient for process design.
- K_{AM} : is the complexity coefficient for additive production.
- $c_{d_{op}}$: is the operator hourly cost for process design.
- $\tau_{d_{or}}$: is the operator time to identify component orientation (h).

- $\tau_{d_{sl}}$: is the operator time to identify the optimal slicing strategy (h).
- $\tau_{d_{ps}}$: is the operator time to identify process parameters (h).
- $\tau_{d_{cm}}$: is the operator time to set the tool path (h).
- $C_{AM_{eq}}$: is the hourly cost for additive production due to amortization.
- $C_{AM_{mh}}$: is the hourly cost for additive production due to maintenance.
- $C_{AM_{en}}$: is the hourly cost for additive production due to energy consumption.
- $C_{AM_{op}}$: is the hourly cost for additive production due to the operator.
- $\tau_{AM_{mt}}$: is the material change time (h).
- $\tau_{AM_{start}}$: is the machine start-up time (h).
- $\tau_{AM_{risc}}$: is the machine preheating time (h).

- n : is the components of the lot number.
- m : is the maximum number of components that can be produced within a job.
- $\lceil \frac{n}{m} \rceil$: is the number of jobs required by a machine with a maximum capacity of m components to make n components.
- $\tau_{AM_{cc}}$: is the cycle change time (h).
- P_{AM} : is the machine productivity (m^3/h).
- $\tau_{AM_{raffr}}$: is the chamber cooling time (h).
- $\tau_{AM_{clean}}$: is the excess powders removal time (h).

Production time index (TMP)

It provides an estimate of component delivery time. The aspects considered by the KPI are:

- Powder supply time
- Material change time in the machine
- Time for process design
- Accessory times for heating and cooling of the machine
- Production time
- Machine cleaning time

The following is the complete formulation of the TMP index with the relative control variables:

$$TMP = \theta_{mat} \cdot \theta_{ing} \cdot \left\{ \tau_{appr_{w[h]}} + (\tau_{d_{or}} + \tau_{d_{sl}} + \tau_{d_{ps}} + \tau_{d_{cm}}) + \left[\tau_{AM_{mt}} + (\tau_{AM_{start}} + \tau_{AM_{risc}}) \cdot \left(\frac{n}{m}\right) \right] + \left[\frac{V_d}{P_{AM}} \cdot n + \tau_{AM_{cc}} \cdot \left(\left(\frac{n}{m}\right) - 1\right) \right] + \left[(\tau_{AM_{raffr}} + \tau_{AM_{clean}}) \cdot \left(\frac{n}{m}\right) \right] \right\} \tag{3}$$

- θ_{mat} : is the compatibility between material and machine.
- θ_{ing} : is the compatibility between component and the machine working volume.
- $\tau_{appr_{w[h]}}$: is the powder supply time (h).
- $\tau_{d_{or}}$: is the operator time to identify component orientation (h).
- $\tau_{d_{sl}}$: is the operator time to identify the optimal slicing strategy (h).
- $\tau_{d_{ps}}$: is the operator time to identify process parameters (h).
- $\tau_{d_{cm}}$: is the operator time to set the tool path (h).
- $\tau_{AM_{mt}}$: is the material change time (h).
- $\tau_{AM_{start}}$: is the machine start-up time (h).
- $\tau_{AM_{risc}}$: is the machine preheating time (h).
- m : is the maximum component number that can be produced within a job.
- $\tau_{AM_{raffr}}$: is the room cooling time (h).
- $\tau_{AM_{clean}}$: is the excess powder removal time (h).
- $\tau_{AM_{cc}}$: is the cycle change time (h).

- n : is the number of components of the lot.
- P_{AM} : is the machine productivity (m^3/h).
- V_d : is the total volume to be melted (m^3).

Overall dimensions of the component in the machine (ING)

It evaluates the compatibility between the dimensions of the component(s) to be produced with respect to the AM machine. The aspects considered by the KPI are:

- Compatibility between the component overall dimensions and the machine working volume
- Convenience of work volume saturation in the case of powder bed processes

The following is the complete formulation of the ING index with the relative control variables:

$$ING = \theta_{ing} \cdot \left\{ \left[\left(\frac{[(x_{AM}-x_d)]}{x_{AM}} + \frac{(y_{AM}-y_d)}{y_{AM}} + \frac{(z_{AM}-z_d)}{z_{AM}} \right) / 3 \right] + \left[\frac{n}{m} - (n_{cycle} - 1) \right] \right\} \tag{4}$$

- θ_{ing} : is the compatibility between component and the machine working volume.
- x_{AM} .
- y_{AM} : is the machine working volume.
- z_{AM} .
- z_d .
- y_d : is the volume of the parallelepiped containing the component.
- z_d .
- n : is the number of components of the lot.
- n_{cycle} is the job numbers.
- m : is the maximum number of components that can be produced within a job.

Technology precision (PRE)

It evaluates the compatibility between the precision required by the component and the precision guaranteed by the machine. The aspects considered by the KPI are:

- Reference precision of the component
- Accuracy guaranteed by the machine

The following is the complete formulation of the PRE index with the relative control variables:

$$PRE = \theta_{pre} \cdot \frac{(\zeta_d - \zeta_{AM})}{\zeta_d} \tag{5}$$

- θ_{pre} : is the product—machine coefficient, $\theta_{pre} = 0 \rightarrow \zeta_d < \zeta_{AM}\theta_{pre} = 1 \rightarrow \zeta_d > \zeta_{AM}$.
- ζ_d : is the reference precision of the component.
- ζ_{AM} : is the machine precision.

Technology resolution (RIS)

It evaluates the compatibility between the smallest feature of the component and the machine’s capabilities in making it. The aspects considered by the KPI are:

- The smallest feature dimension in the component
- Machine resolution

The following is the complete formulation of the RIS index with the relative control variables:

$$RIS = \theta_{ris} \cdot \frac{(\xi_d - \xi_{AM})}{\xi_d} \tag{6}$$

- θ_{ris} : is the product—machine coefficient, $\theta_{ris} = 0 \rightarrow \zeta_d < \zeta_{AM}\theta_{ris} = 1 \rightarrow \zeta_d > \zeta_{AM}$.
- ζ_d : is the reference resolution of the component.
- ζ_{AM} : is the machine resolution.

Undercut management (STQ)

It evaluates the compatibility between the undercut surfaces and the capacity of the considered AM machine. The aspects considered by the KPI are:

- Surface of the undercut areas
- Total component surface

The following is the complete formulation of the SQT index with the relative control variables:

$$STQ = \frac{S_d}{\left(\frac{S_t}{\psi^2} + S_d\right)} \tag{7}$$

- S_d : is the total component surface.
- S_t : is the total undercut surface.
- ψ : is the support management coefficient.

Technology roughness (RGS)

It evaluates the compatibility between the roughness required by the component and the roughness obtainable by the AM machine. The aspects considered by the KPI are:

- Component reference roughness
- Roughness guaranteed by the machine

The following is the complete formulation of the RGS index with the relative control variables:

$$RGS = \theta_{rgs} \cdot \frac{(\chi_d - \chi_{AM})}{\chi_d} \tag{8}$$

- θ_{rgs} : is the product—machine coefficient, $\theta_{rgs} = 0 \rightarrow \chi_d < \chi_{AM}\theta_{rgs} = 1 \rightarrow \chi_d > \chi_{AM}$.
- χ_d : is the component reference roughness.
- χ_{AM} : is the roughness obtainable with the machine.

Compatibility index (CBA)

The compatibility coefficient is represented by the appropriately weighted sum of a series of contributions given by specific KPIs. This last can be analyzed separately by the AMSA technician in the analysis of the estimate quotation, but the values will flow into the CBA calculation which will be used to order the list of suppliers to be evaluated.

The following is the complete formulation of the CBA index with the relative control variables:

$$CBA = \theta_{mat} \cdot \theta_{ing} \cdot \left\{ \alpha_{gmt} \cdot \left(\frac{PRE + RIS + RGS + STQ}{4} \right) + \alpha_{cst} \cdot \left(\frac{1 - CST}{CST_{max} \cdot (1 + \alpha_{cst})} \right) + \alpha_{tmp} \cdot \left(\frac{1 - TMP}{TMP_{max} \cdot (1 + \alpha_{tmp})} \right) \right\} \tag{9}$$

The KPI is equal to zero when θ_{mat} or θ_{ing} is null, which are conditions corresponding, respectively, to an incompatibility in terms of material (the material is not supported by the machine) or overall dimensions (the component is too large compared to the machine’s working volume).

If PRE, RIS, RGS, or STQ is equal to zero, CBA does not become null because the component is actually feasible; it requires only additional treatments or workings that cannot be ignored if the customer’s request is to be satisfied:

- θ_{mat} : $\theta_{mat} = 0$ incompatible material $\theta_{mat} = 1$ compatible material.
- θ_{ing} : is the compatibility between the component and the machine working volume.
- α_{gmt} : is the weight associated with geometric KPIs.
- α_{cst} : is the weight associated with cost KPIs.
- α_{tmp} : is the weight associated with time KPIs.
- PRE is the technology precision.
- RIS is the technology resolution.
- RGS is the technology roughness.
- STQ is the undercuts management.

- CST is the production cost.
- TMP is the production time.
- CST_{max} : is the maximum CST value calculated among all available machines.
- TMP_{max} : is the maximum TMP value calculated among all available machines.

Appendix 2

The following is the complete algorithm in Java:

```

package macro;

import macro.enumeration.Layouts;

public class Nesting {

    public Nesting() {
        // TODO Auto-generated constructor stub
    }

    private static int NumberWX(double r, double gapB, double gapX, double xc) {
        double radiusModX;
        double deltaRadiusX;
        double stepX;
        double lsCompXfloat;
        int lsCompX;
        radiusModX = r - gapB;
        deltaRadiusX = Math.abs(radiusModX - (xc / 2));
        stepX = xc + gapX;
        lsCompXfloat = (deltaRadiusX / stepX) + 1;
        lsCompX = (int) lsCompXfloat;
        return lsCompX;
    }

    private static int NumberWY(double r, double gapB, double gapY, double yc) {
        double radiusModY;
        double deltaRadiusY;
        double stepY;
        double lsCompYfloat;
        int lsCompY;
        radiusModY = r - gapB;
        deltaRadiusY = Math.abs(radiusModY - (yc / 2));
        stepY = yc + gapY;
        lsCompYfloat = (deltaRadiusY / stepY) + 1;
        lsCompY = (int) lsCompYfloat;
        return lsCompY;
    }

    private static int NumberOfPartsCirc(double r, double gapB, double gapX, double gapY, double xc, double yc) {
        if (xc == 0.0 || yc == 0.0) {
            return 0;
        } else {
            int lsNumberWY;
            int lsNumberWX;
            int lsresult = 0;
            int lsresult1 = 0;
            double cX = 0.0;
            double cY = 0.0;
            double radiusMod = r - gapB;
            int index1 = 0;
            int index2 = 0;
            lsNumberWY = NumberWY(r, gapB, gapY, yc);
            lsNumberWX = NumberWX(r, gapB, gapX, xc);
            int j = 0;
            while (j <= lsNumberWY) {
                int i = 0;
                while (i <= lsNumberWX) {
                    double point1[] = { ((xc / 2 + gapX) + ((xc + gapX) * (i - 1))),
                        ((yc / 2 + gapY + yc) + ((gapY + yc) * (j - 1))) };
                    double point2[] = { ((xc / 2 + gapX + xc) + ((xc + gapX) * (i - 1))),
                        ((yc / 2 + gapY + yc) + ((gapY + yc) * (j - 1))) };
                    double point3[] = { ((xc / 2 + gapX + xc) + ((xc + gapX) * (i - 1))),
                        ((yc / 2 + gapY) + ((gapY + yc) * (j - 1))) };
                }
            }
        }
    }
}

```

```

double point4[] = { ((xc / 2 + gapX) + ((xc + gapX) * (i - 1))),
                   ((yc / 2 + gapY) + ((gapY + yc) * (j - 1))) };
double distPoint1;
distPoint1 = Math.sqrt((Math.pow((point1[0] - cX), 2)) + (Math.pow((point1[1] - cY), 2)));
double distPoint2;
distPoint2 = Math.sqrt((Math.pow((point2[0] - cX), 2)) + (Math.pow((point2[1] - cY), 2)));
double distPoint3;
distPoint3 = Math.sqrt((Math.pow((point3[0] - cX), 2)) + (Math.pow((point3[1] - cY), 2)));
double distPoint4;
distPoint4 = Math.sqrt((Math.pow((point4[0] - cX), 2)) + (Math.pow((point4[1] - cY), 2)));
if (distPoint1 <= radiusMod && distPoint2 <= radiusMod && distPoint3 <= radiusMod
    && distPoint4 <= radiusMod) {
    if (j == 0 && i != 0) {
        index1 = index1 + 1;
        lsresult = ModNumber(xc, yc, gapX, gapY, radiusMod, i, index1);
        if (lsresult != 0) {
            index1 = lsresult;
        }
    } else if (j != 0 && i == 0) {
        index1 = index1 + 1;
    } else if (j != 0 && i != 0) {
        index2 = index2 + 1;
    } else {
        index1 = index1 + 1;
        lsresult1 = ModNumber(xc, yc, gapX, gapY, radiusMod, i, index1);
        if (lsresult1 != 0) {
            index1 = lsresult1;
        }
    }
}
}
i = i + 1;
}
j = j + 1;
}
int middleParts1;
if (lsresult1 != 0) {
    middleParts1 = (index1 * 2) - 2;
    if (middleParts1 < 0) {
        middleParts1 = 0;
    }
} else {
    middleParts1 = (index1 * 2) - 1;
    if (middleParts1 < 0) {
        middleParts1 = 0;
    }
}
int peripheralParts1;
int totalParts1;
peripheralParts1 = index2 * 4;
totalParts1 = middleParts1 + peripheralParts1;
return totalParts1;
}
}

private static int ModNumber(double xc, double yc, double gapX, double gapY, double radiusMod, int i, int k) {
    double cX = 0.0;
    double cY = 0.0;
    double stepX = (xc / 2 + gapX + xc) + ((xc + gapX) * (i - 1));
    double cosalpha = stepX / radiusMod;
    double alpha = Math.acos(cosalpha);
    double sinalpha = Math.sin(alpha);
    double bDimension = radiusMod * sinalpha;
    double bDimensionFilter = bDimension - (yc / 2);
    double check1 = (yc / 2) + (gapY / 2);
    double check2 = yc + gapY;
    double checkVertexX = (xc / 2 + gapX + xc / 2) + ((xc + gapX) * (i - 1)) + (xc / 2);
    double checkVertexY = yc + gapY;
    double checkVertex[] = { checkVertexX, checkVertexY };
    double distPointCheck;
    distPointCheck = Math.sqrt((Math.pow((checkVertex[0] - cX), 2)) + (Math.pow((checkVertex[1] - cY), 2)));
    if (bDimensionFilter > check1 && bDimensionFilter < check2 && distPointCheck < radiusMod) {
        k = k + 1;
        k = (int) k;
        return k;
    } else {
        return 0;
    }
}
}

```

```

}

private static int NumberOfPartsRect(double x, double y, double gapB, double gapX, double gapY, double xc,
double yc) {
    int number1;
    int number2;
    int number;
    int nx;
    int ny;
    double xcL;
    double ycL;

    int numberResWorkpiece = 0;
    double stepCheck;
    int newNumber = 0;

    if (xc == 0.0 || yc == 0.0) {
        return 0;
    } else {
        double xmod = x - (gapB * 2);
        double ymod = y - (gapB * 2);
        double nx1d = (xmod / (xc + gapX));
        int nx1 = (int) nx1d;
        if (nx1 > 0) {
            double distCalcX1 = (nx1 * xc) + ((nx1 - 1) * gapX);
            if ((xmod - distCalcX1) >= (xc + gapX)) {
                nx1 = nx1 + 1;
            }
        }
        double ny1d = (ymod / (yc + gapY));
        int ny1 = (int) ny1d;
        if (ny1 > 0) {
            double distCalcY1 = (ny1 * yc) + ((ny1 - 1) * gapY);
            if ((ymod - distCalcY1) >= (yc + gapY)) {
                ny1 = ny1 + 1;
            }
        }
        double nx2d = (xmod / (yc + gapX));
        int nx2 = (int) nx2d;
        if (nx2 > 0) {
            double distCalcX2 = (nx2 * yc) + ((nx2 - 1) * gapX);
            if ((xmod - distCalcX2) >= (yc + gapX)) {
                nx2 = nx2 + 1;
            }
        }
        double ny2d = (ymod / (xc + gapY));
        int ny2 = (int) ny2d;
        if (ny2 > 0) {
            double distCalcY2 = (ny2 * xc) + ((ny2 - 1) * gapY);
            if ((ymod - distCalcY2) >= (xc + gapY)) {
                ny2 = ny2 + 1;
            }
        }
        if (nx1 >= 1 && ny1 >= 1) {
            number1 = nx1 * ny1;
        } else {
            number1 = 0;
        }
        if (nx2 >= 1 && ny2 >= 1) {
            number2 = nx2 * ny2;
        } else {
            number2 = 0;
        }
        if (number1 > number2) {
            number = number1;
            nx = nx1;
            ny = ny1;
            xcL = xc;
            ycL = yc;
        } else {
            number = number2;
            nx = nx2;
            ny = ny2;
            xcL = yc;
            ycL = xc;
        }
    }
}

```

```

        if (number != 0) {
            if (xcL < ycL) {
                stepCheck = (ycL * ny) + (gapY * (ny - 1));
                double differenceStep = ymod - stepCheck;
                double ycL1 = xcL;
                double xcL1 = ycL;
                double newDimension = ycL1 + gapY;
                double newNumberd = (differenceStep / newDimension);
                newNumber = (int) newNumberd;
                double newStepCheck = xcL1 + gapX;
                double numberResWorkpieced = (xmod / newStepCheck);
                numberResWorkpiece = (int) numberResWorkpieced;
            } else if (xcL > ycL) {
                stepCheck = (xcL * nx) + (gapX * (nx - 1));
                double differenceStep = xmod - stepCheck;
                double ycL2 = xcL;
                double xcL2 = ycL;
                double newDimension = xcL2 + gapX;
                double newNumberd = (differenceStep / newDimension);
                newNumber = (int) newNumberd;
                if (newNumber != 0) {
                    double newStepCheck = ycL2 + gapY;
                    double numberResWorkpieced = (ymod / newStepCheck);
                    numberResWorkpiece = (int) numberResWorkpieced;
                }
            }
        }
        int totalNumber;
        if (numberResWorkpiece != 0) {
            totalNumber = number + (numberResWorkpiece * newNumber);
        } else {
            totalNumber = number;
        }
        return totalNumber;
    }
}

/**
 * Funzione che calcola il numero di pezzi stampabili data la superficie del pezzo e del piano di stampa
 * e piano
 *
 * @param x      larghezza piano di stampa
 * @param y      altezza piano di stampa
 * @param gapB   Offset dal bordo
 * @param gapX   Offset in direzione x del pezzo
 * @param gapY   Offset in direzione y del pezzo
 * @param xc     dimensione x del componente
 * @param yc     dimensione y del componente
 * @param formFactor form factor del piano
 * @return numero di pezzi stampabili.
 */
public static int NestingFunctionDef(double x, double y, double gapB, double gapX, double gapY, double xc,
    double yc, Layouts formFactor) {
    if (formFactor == Layouts.CIRCLE) {
        double r = x / 2;
        return NumberOfPartsCirc(r, gapB, gapX, gapY, xc, yc);
    } else {
        return NumberOfPartsRect(x, y, gapB, gapX, gapY, xc, yc);
    }
}
}

```

Funding This research was funded by the Ministry of Economic Development, Bando Horizon 2020-PON 2014/2020.

Data availability Not applicable.

Code availability Appendix 2 reports the complete algorithm in Java.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication All authors have agreed to authorship, read, and approved the manuscript and have given consent for submission and subsequent publication of the manuscript. The authors guarantee that the contribution to the work has not been previously published elsewhere.

Conflict of interest The authors declare no competing interests.

References

- Zhang Y, Gupta RK, Bernard A (2016) Two-dimensional placement optimization for multi-parts production in additive manufacturing. *Robot Comput Integr Manuf* 38:102–117
- Bartolo P, Kruth JP, Silva J, Levy G, Malshe A, Rajurkar K, Leu M (2012) Biomedical production of implants by additive electrochemical and physical processes. *CIRP Ann Manuf Technol* 61(2):635–655
- Kruth JP, Leu MC, Nakagawa T (1998) Progress in additive manufacturing and rapid prototyping. *CIRP Ann Manuf Technol* 47(2):525–540
- Levy GN, Schindel R, Kruth JP (2003) Rapid manufacturing and rapid tooling with layer manufacturing (LM) technologies, state of the art and future perspectives. *CIRP Ann Manuf Technol* 52(2):589–609
- Wohlers T (2013) Wohlers report 2013, Wohlers association, USA
- Zhang Y, Bernard A (2014) Grouping parts for multiple parts production in additive manufacturing. *Procedia CIRP* 17:308–313
- Gogate AS, Pande SS (2008) Intelligent layout planning for rapid prototyping. *Int J Prod Res* 46(20):5607–5631
- Dickinson JK (1999) Packing subsets of arbitrary three-dimensional objects. University of Western Ontario, London, Ontario, Canada: Doctoral dissertation, Ph. D. Thesis, Faculty of Graduate Studies
- Zhang X, Zhou B, Zeng Y, Gu P (2002) Model layout optimization for solid ground curing rapid prototyping processes. *Robot Comput Integr Manuf* 18(1):41–51
- Luttersa E, ten Dama D, Fanekera T (2012) 3D nesting of complex shapes. 45th CIRP Conference on Manufacturing Systems 2012. *Procedia CIRP* 3:26–31
- Canellidis V, Giannatsis J, Dedoussis V (2013) Efficient parts nesting schemes for improving stereolithography utilization. *Comput Aided Des* 45:875–886
- Zhang Y, Kumar R, Bernard GA (2016) Two-dimensional placement optimization for multi-parts production in additive manufacturing. *Robot Comput Integr Manuf* 38:102–117
- Zhang Y, Bernard A, Harik R, Fadel G. A new method for single-layer-part nesting in additive manufacturing. *Rapid Prototyp J* 24(5):840–854
- Angelo L, Stefano P, Guardiani E (2020) Search for the optimal build direction in additive manufacturing technologies: a review. *J Manuf Mater Process* 4:71. <https://doi.org/10.3390/jmmp4030071>
- Kucukkoc I (2019) MILP models to minimise makespan in additive manufacturing machine scheduling problems. *Comput Oper Res* 105:58–67. <https://doi.org/10.1016/j.cor.2019.01.006>
- Araújo JP, Panesar A, Özcan E, Atkin J, Baumers M, Ashcroft I (2019) An experimental analysis of deepest bottom-left-fill packing methods for additive manufacturing. *Int J Prod Res* 1–17. <https://doi.org/10.1080/00207543.2019.1686187>
- Kim H-J (2018) Bounds for parallel machine scheduling with predefined parts of jobs and setup time. *Ann Oper Res* 261(1):401–412. <https://doi.org/10.1007/s10479-017-2615-z>
- Chergui A, Hadj-Hamou K, Vignat F (2018) Production scheduling and nesting in additive manufacturing. *Comput Ind Eng* 126:292–301. <https://doi.org/10.1016/j.cie.2018.09.048>
- Dvorak F, Micali M, Mathieug M (2018) Planning and scheduling in additive manufacturing. *Intel Artif* 21(62):40–52. <https://doi.org/10.4114/intartif.vol21iss62pp40-52>
- Zhang J, Yao X, Li Y (2019) Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing. *Int J Prod Res* 1–20. <https://doi.org/10.1080/00207543.2019.1617447>
- Oha Y, Witherellc P, Luc Y, Sprockc T (2020) Nesting and scheduling problems for additive manufacturing: a taxonomy and review. *Addit Manuf* 36:101492
- Del Prete A, Primo T (2021) Innovative methodology for the identification of the most suitable additive technology based on product characteristics. *Metals* 11:409. <https://doi.org/10.3390/met11030409>
- <https://grabcad.com/>
- HyperWorks (2020) User's Manual, Altair Corporation

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.