

# Nesting Algorithm for Optimization Part Placement in Additive Manufacturing

**Maurizio Calabrese**

Universita del Salento

**Teresa Primo** (✉ [teresa.primo@unisalento.it](mailto:teresa.primo@unisalento.it))

Universita del Salento

**Antonio Del Prete**

Universita del Salento

**Giuseppe Filitti**

Universita del Salento

---

## Research Article

**Keywords:** Nesting, Additive Manufacturing, Key performance indicators Algorithm, Previsional model

**Posted Date:** May 27th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-483697/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

**Version of Record:** A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on January 13th, 2022. See the published version at <https://doi.org/10.1007/s00170-021-08130-y>.

# “Nesting algorithm for optimization part placement in additive manufacturing”

Maurizio Calabrese<sup>1,a</sup>, Teresa Primo<sup>1,b</sup>, Antonio Del Prete<sup>1,c</sup>, <sup>2d</sup>Giuseppe Filitti

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

<sup>2</sup>Ingenia Research Institute, Via Marco Biagi, 73100, Lecce.

[maurizio.calabrese@unisalento.it](mailto:maurizio.calabrese@unisalento.it);

[teresa.primo@unisalento.it](mailto:teresa.primo@unisalento.it);

[antonio.delprete@unisalento.it](mailto:antonio.delprete@unisalento.it);

[g.filitti@ingeniaepartners.com](mailto:g.filitti@ingeniaepartners.com)

## 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 Indicator (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 indicators Algorithm, Previsional model

## Introduction

Additive Manufacturing (AM), derived from Rapid Prototyping (RP), has been investigated and developed for more than three decades. It can not only provide prototypes rapidly to support the product development, but also produce functional or end-use parts for different application areas [1][2][3][4][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 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 requirements of users should be taken into consideration when doing the multi- parts 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 NP (nesting or packing)-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].

In this paper a new algorithm for 3D printing nesting has presented and it has been developed within the definition of a previsional 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 previsional model a worth 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 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 KPI that allow to evaluate the available technologies in terms of cost estimation and technological compatibility.

## **AMSA Methodology**

The prevision model and the nesting algorithm have been implemented as part of a research project AMSA (Additive Manufacturing Spare parts market Application). In this project has been developed a tool, called AMSA, having as main goal the definition of a common platform to supply different kind 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 kind of services allow 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 to 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 an 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.

## **Key performance indicators**

KPIs (Key Performance Indicators) represent a synthetic method to collect information and correlate it with each other. The developed previsional model allows to filter the technologies according to the appropriate performance indexes (KPIs) reported below **Error! Reference source not found.:**

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 previsional 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 the appendix A.

## Nesting Algorithm

Within the previsional 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.

## 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

allows the components number calculation that can be positioned on the printer surface (Figure 1).

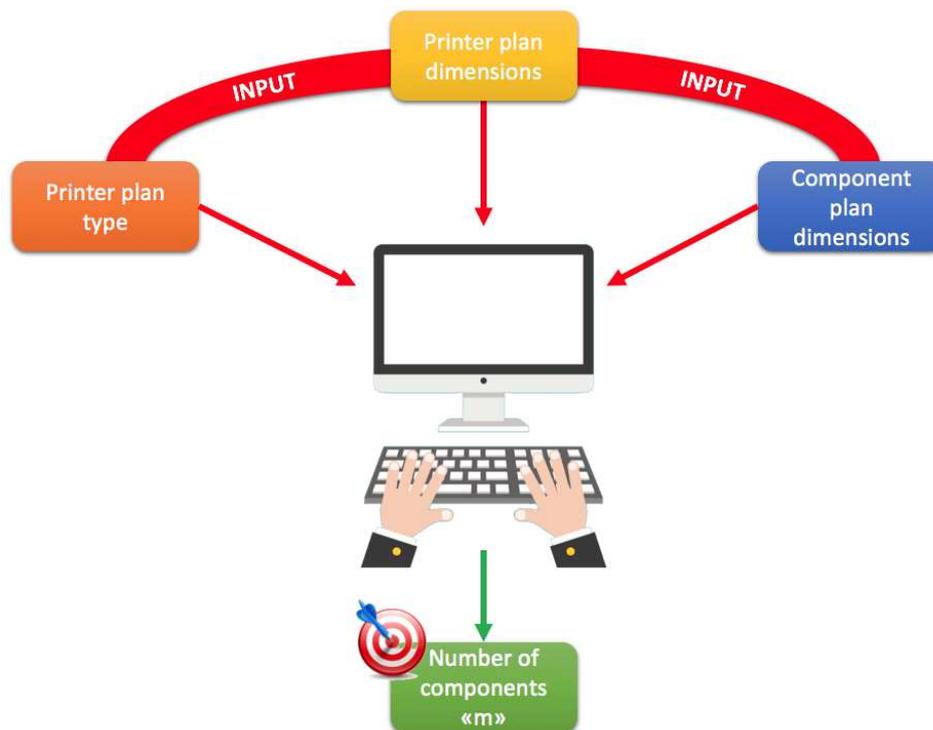


Figure 1 - Definition of the Nesting algorithm

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 components dimensions an algorithm has been developed that has been implemented in the previsual model which in turn has been integrated into the AMSA platform.

### Nesting algorithm definition

The Nesting algorithm can be briefly summarized with the following diagram.

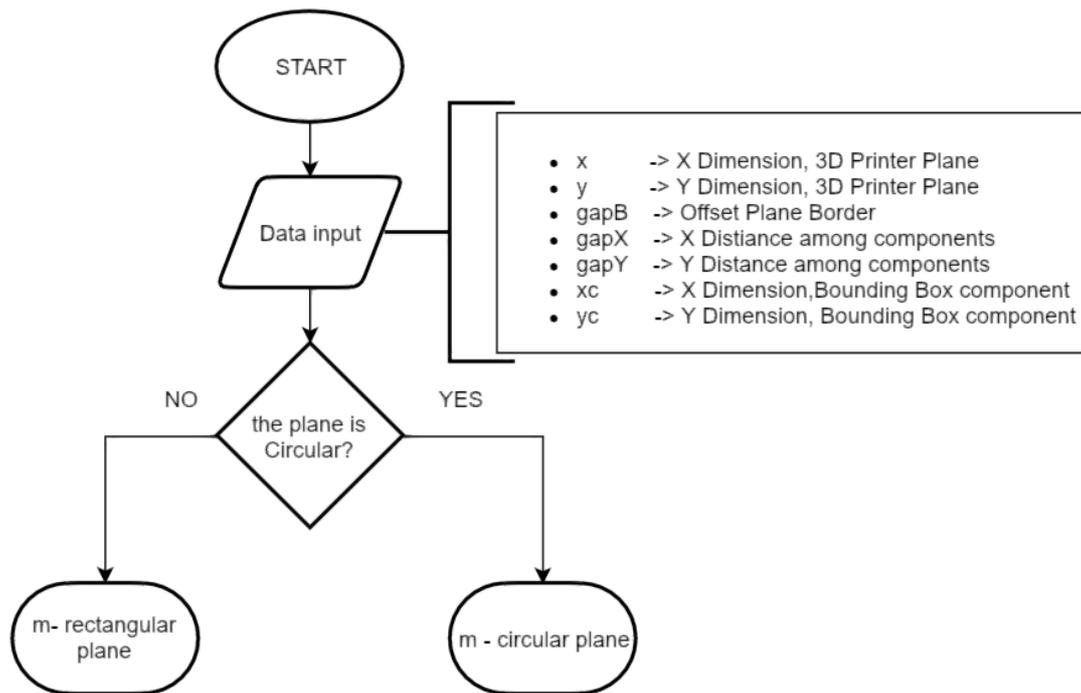


Figure 2 - Nesting algorithm diagram

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 printer plane
- $y$  = Y dimension of printer plane
- $gapB$  = offset respect to printer plane border
- $gapX$  = X distance among components
- $gapY$  = Y distance among components

- $x_c$  = X dimension of component Bounding Box
- $y_c$  = Y dimensions of 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.

### **Nesting algorithm definition for CIRCULAR printing plan**

In this section is reported the description of the computation algorithm of the components number in the case of a circular printing plan. Figure 3 shows the summary workflow.

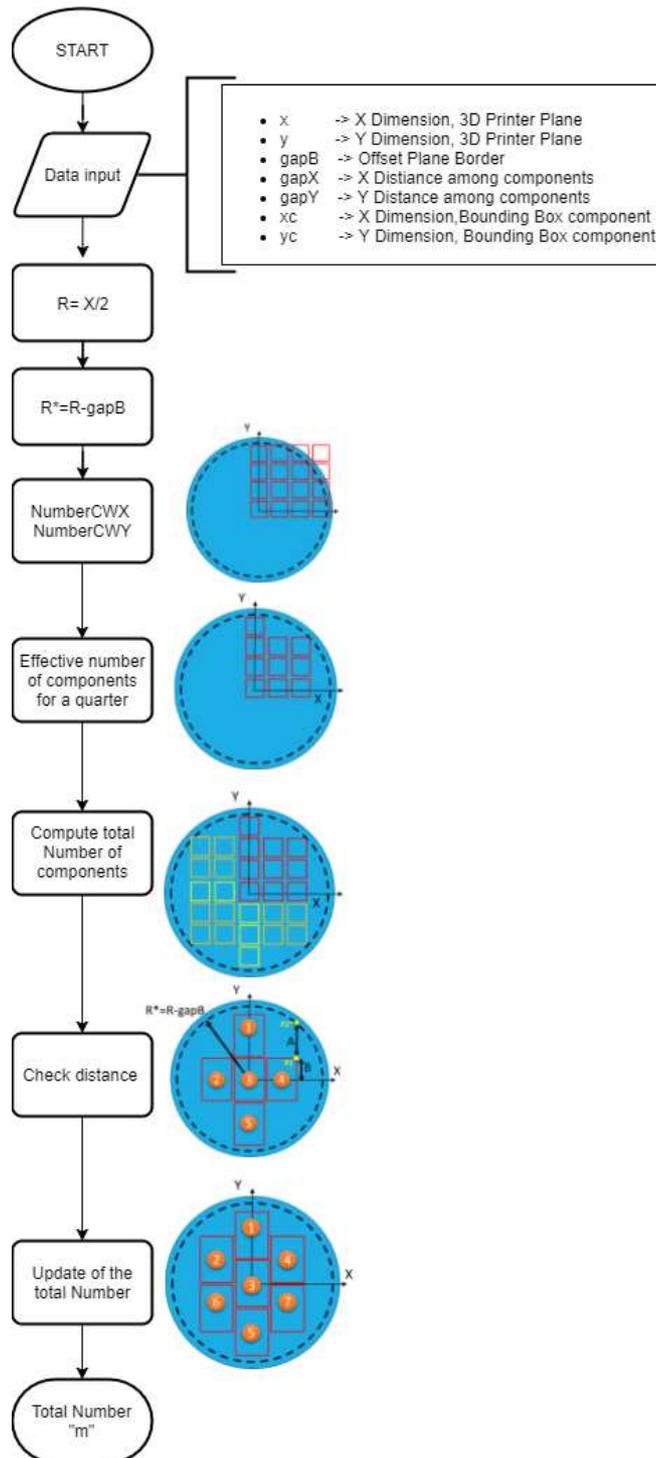
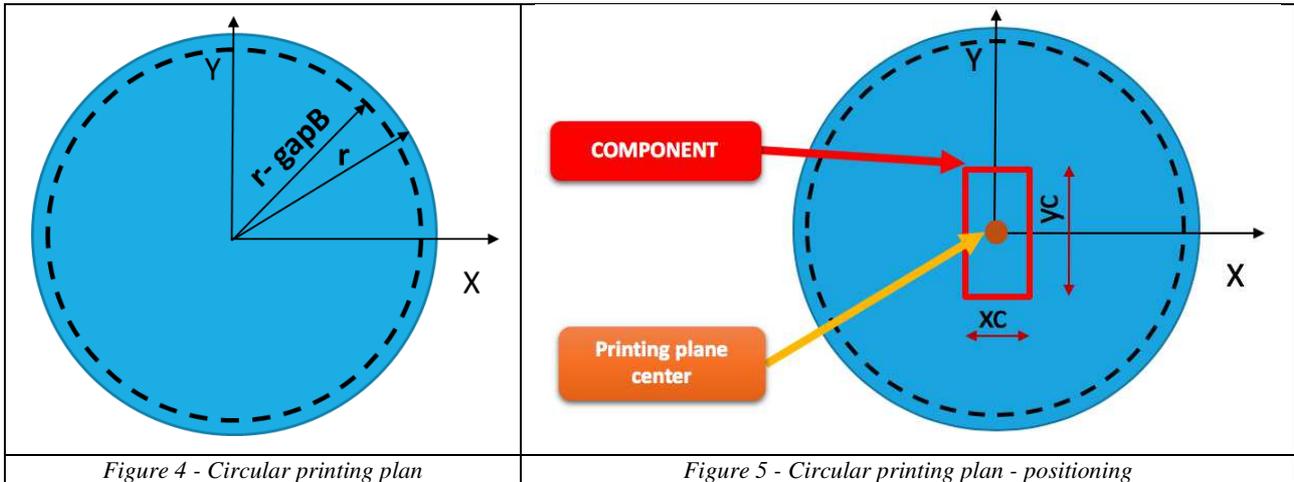


Figure 3 - Circular nesting workflow

The circular printing plan is obviously identified by a radius (Figure 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:  $X = Y = r * 2$  ( $r = \text{radius}$ ). The Figure 5 also shows the previously indicated offset “ $gapB$ ” (Figure 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 (**Error! Reference source not found.**).



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 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.

Therefore the positioning shown in Figure 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 choice a particular extrusion direction and the algorithm considers the other directions to define the dimensions in plan.

Therefore the following functions are present:

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

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

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

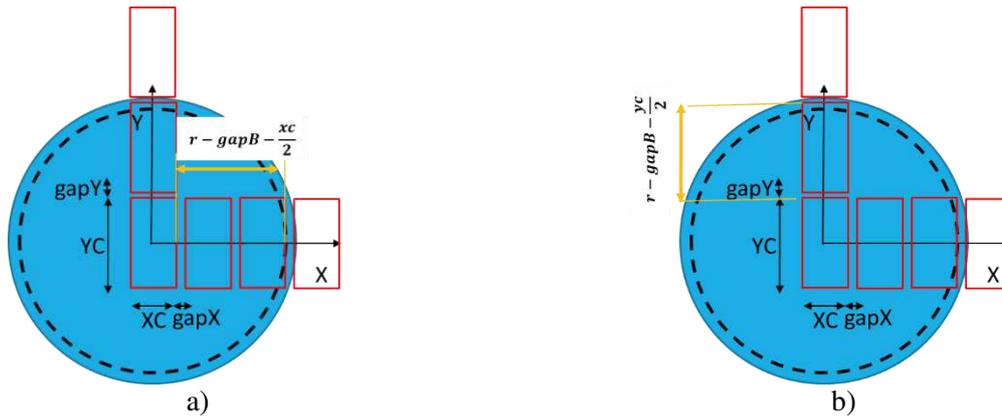


Figure 6 - a) NumberWX Calculation; b) NumberWY calculation

With the calculation of the components in the two directions, the algorithm, in the preliminary phase, computes the scenario shown in Figure 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.

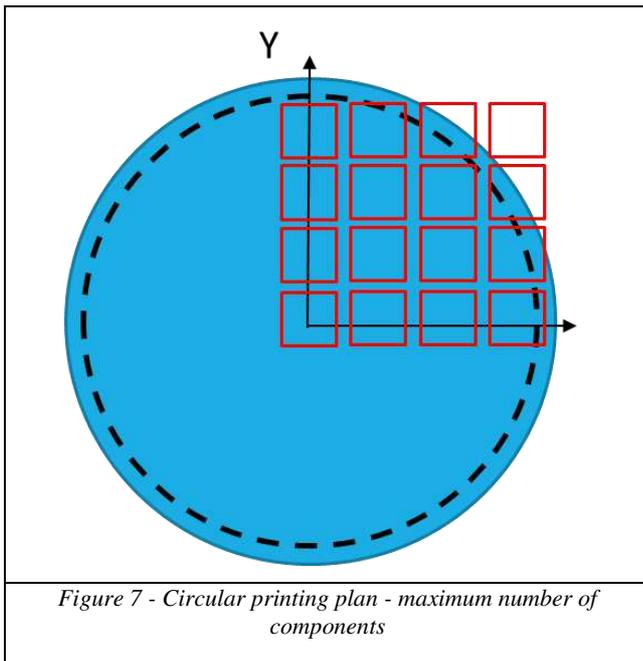


Figure 7 - Circular printing plan - maximum number of components

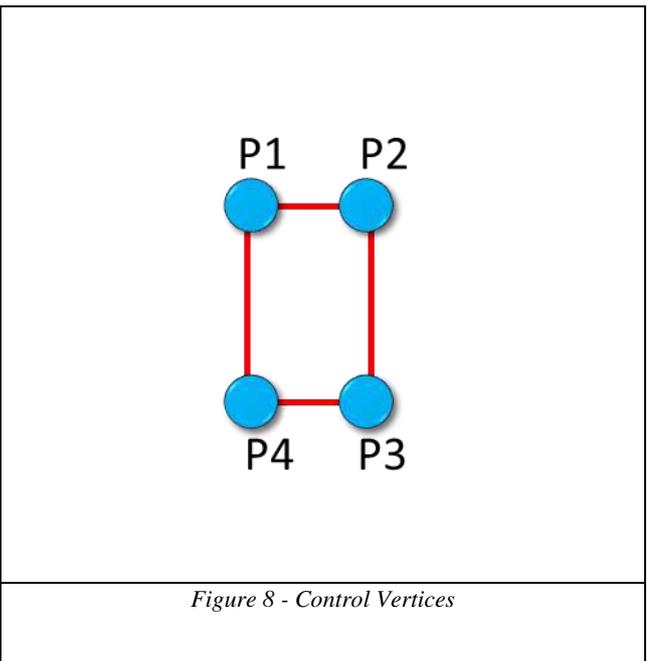


Figure 8 - Control Vertices

In order to understand what the components are actually contained within the printing plan, in the algorithm a set of parametric equations have 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 the formulation of these equations:

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

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

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

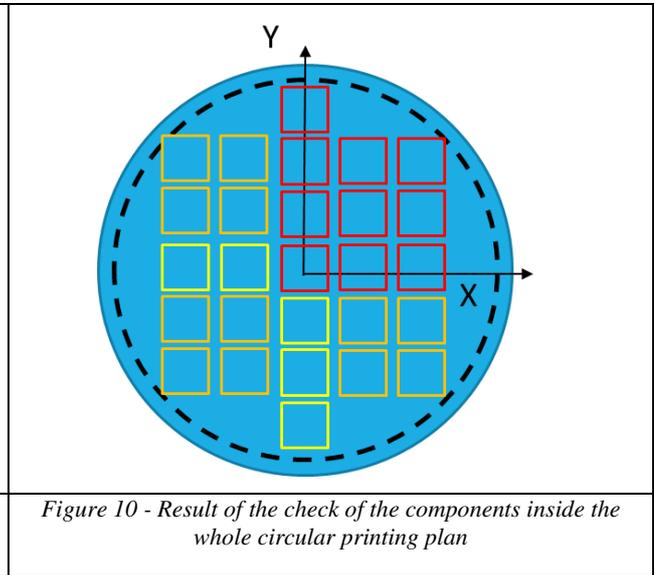
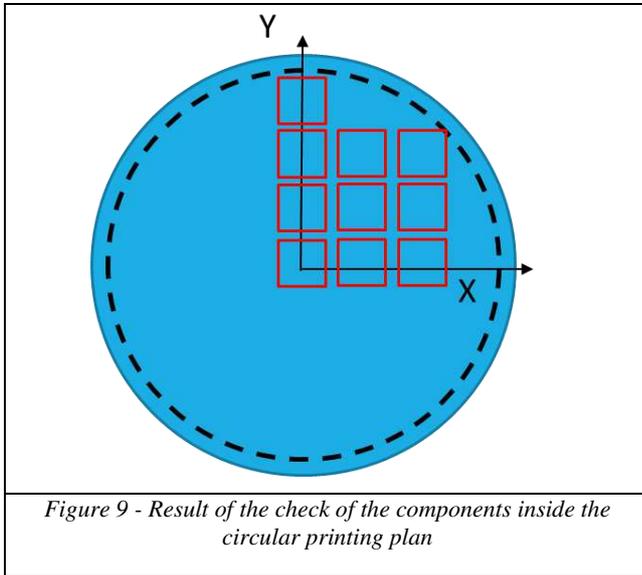
$$P4ij = \left( \left[ \left( \frac{xc}{2} + \Delta x \right) + (xc + \Delta x) * (Nx - 1) \right], \left[ \left( \frac{yc}{2} + \Delta y \right) + (\Delta y + yc) * (Ny - 1) \right] \right)$$

The equations refer to the vertices shown in the Figure 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 from the center lower than to 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 the vertex 1 belonging to the component  $i$ . After the checking of how many components respect the conditions listed above, the result obtained is reported in Figure 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 Figure 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 single 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 Figure 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 the Figure 11, if components 2 and 4 were properly translated, it would be possible to insert further components.

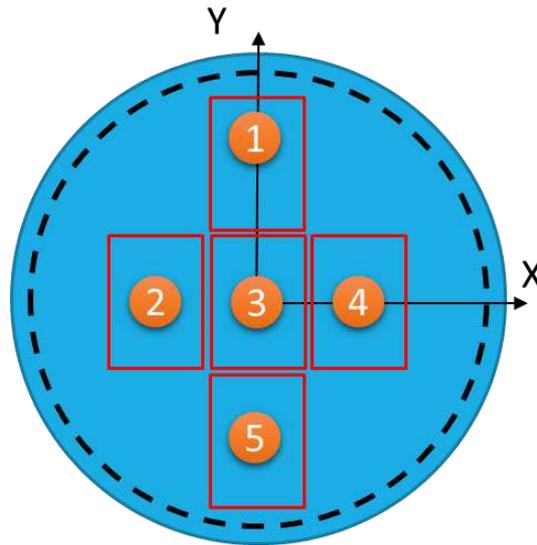


Figure 11 - Scenario for which further implementation is needed

The Figure 12 shows the A and B dimensions defined as:

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

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

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

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

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 the Figure 12.

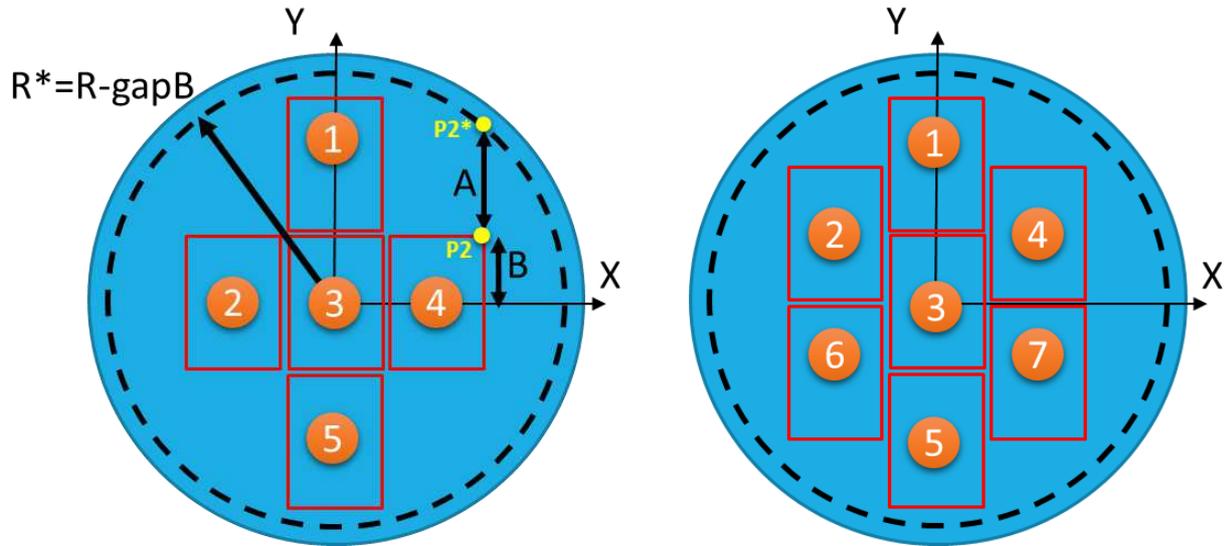


Figure 12 – (a) Size check

(b) New layout

### **Nesting algorithm definition for RECTANGULAR printing plan**

In this section is reported the description of the computation algorithm of the components number in the case of a rectangular printing plan. Figure 13 shows the summary workflow.

The algorithm is more efficient in the potential components number definition for the RECTANGULAR printing plan.

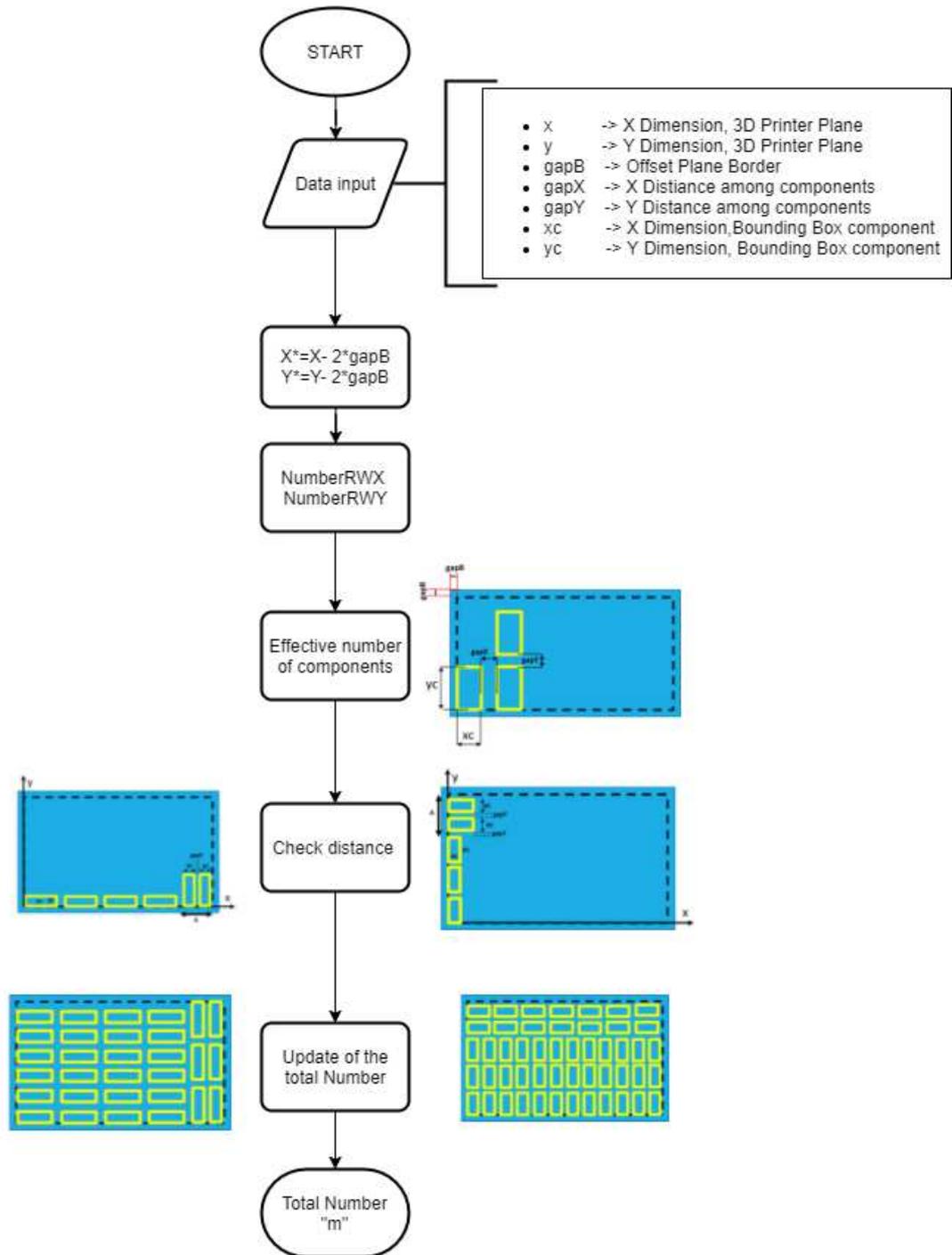


Figure 13 - Rectangular nesting workflow

The 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 components calculation.

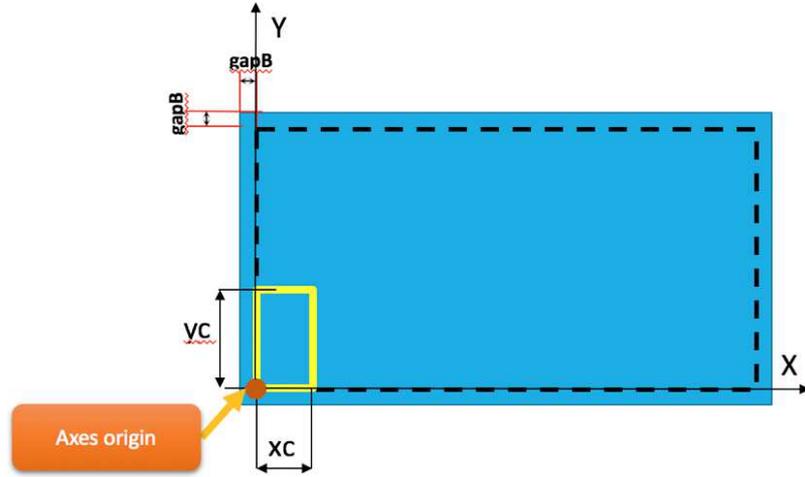


Figure 14 - Rectangular printing plan

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

- 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.

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

Table 1 –Number of components calculation for scenario 1 and 2

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

The components number will therefore be equal to:

$$N = \text{NumberRWX} * \text{NumberRWY}$$

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

$$\text{Numero dei componenti} = \text{Max}(N_A, N_B)$$

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

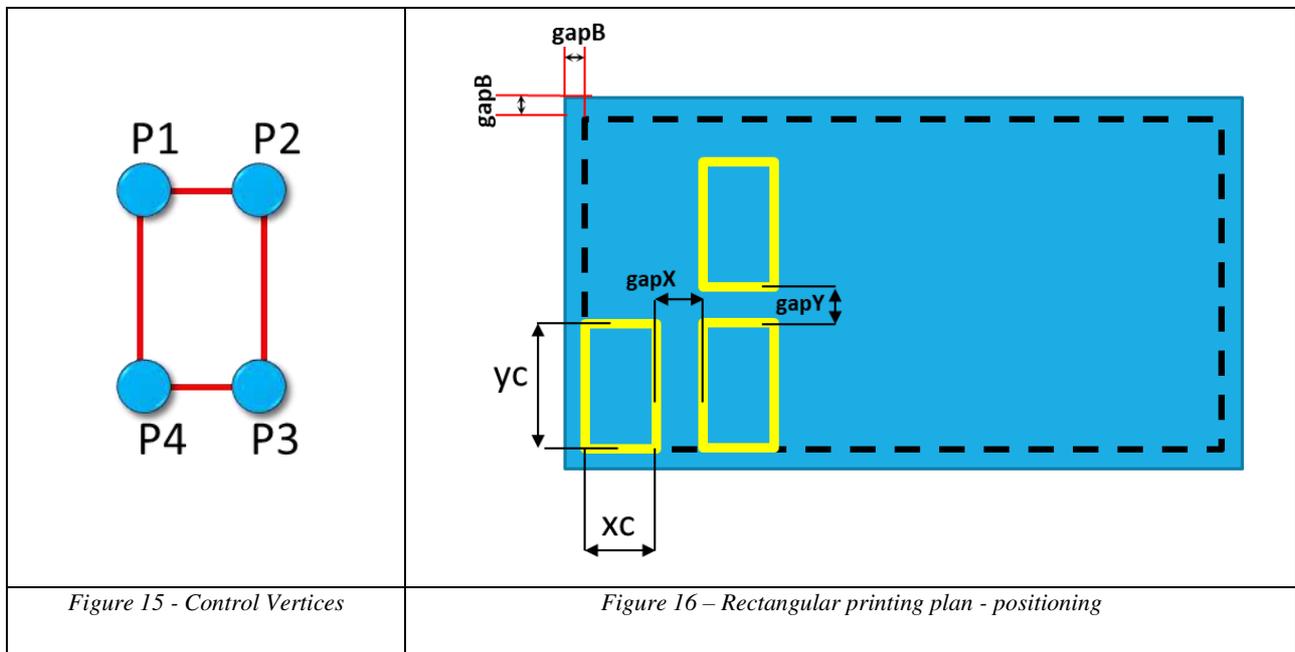
$$P1ij = \{(xc + gapX + xc) + [(xc + gapX) * (i - 1)]\}, \{(yc + gapY + yc) + [(gapY + yc) * (j - 1)]\}$$

$$P2ij = \{(xc + gapX) + [(xc + gapX) * (i - 1)]\}, \{(yc + gapY + yc) + [(gapY + yc) * (j - 1)]\}$$

$$P3ij = \{(xc + gapX) + [(xc + gapX) * (i - 1)]\}, \{(yc + gapY) + [(gapY + yc) * (j - 1)]\}$$

$$P4ij = \{(xc + gapX + xc) + [(xc + gapX) * (i - 1)]\}, \{(yc + gapY) + [(gapY + yc) * (j - 1)]\}$$

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



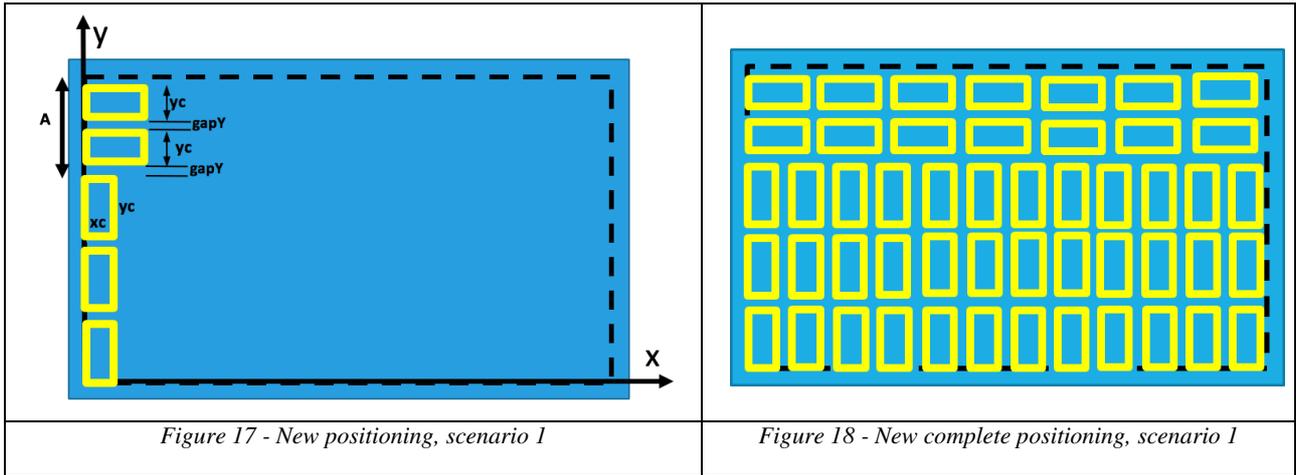
In particular, the equations listed above describe the coordinates of all 4 vertices for each rectangle positioned on the printing plane.

Figure 16 shows the components 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 the Figure 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:

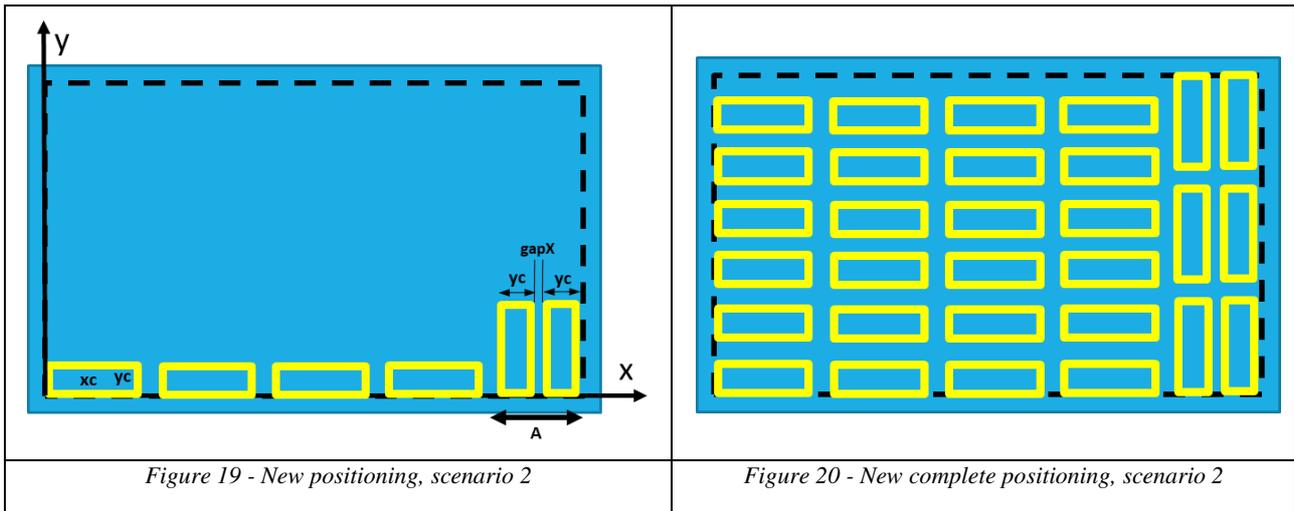
$$\text{Numero di file } (Nf) = \text{int}\left(\frac{A}{(xc + gapY)}\right)$$

$$A \geq (xc * Nf) + (gapY * (Nf - 1))$$

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 the Figure 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:

$$\text{Numero di file } (Nf) = \text{int}\left(\frac{A}{(yc + gapX)}\right)$$

$$A \geq (yc * Nf) + (gapY * (Nf - 1))$$

The 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.

## Results and Discussion

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

- Step 1, algorithm implementation using the TCL language (Tool Command Language);
- Step 2, porting of the algorithm to the Java language (APPENDIX B).

The choice to use the TCL language was made due to the availability of a development environment, that provides tools able to plot the results giving, therefore, the possibility to graphically verify the quality of the solution (Figure 21).

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

In the step 2, after the development and testing of the algorithm, thanks also to the visualization of the results in the HyperMesh® environment, the was 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.

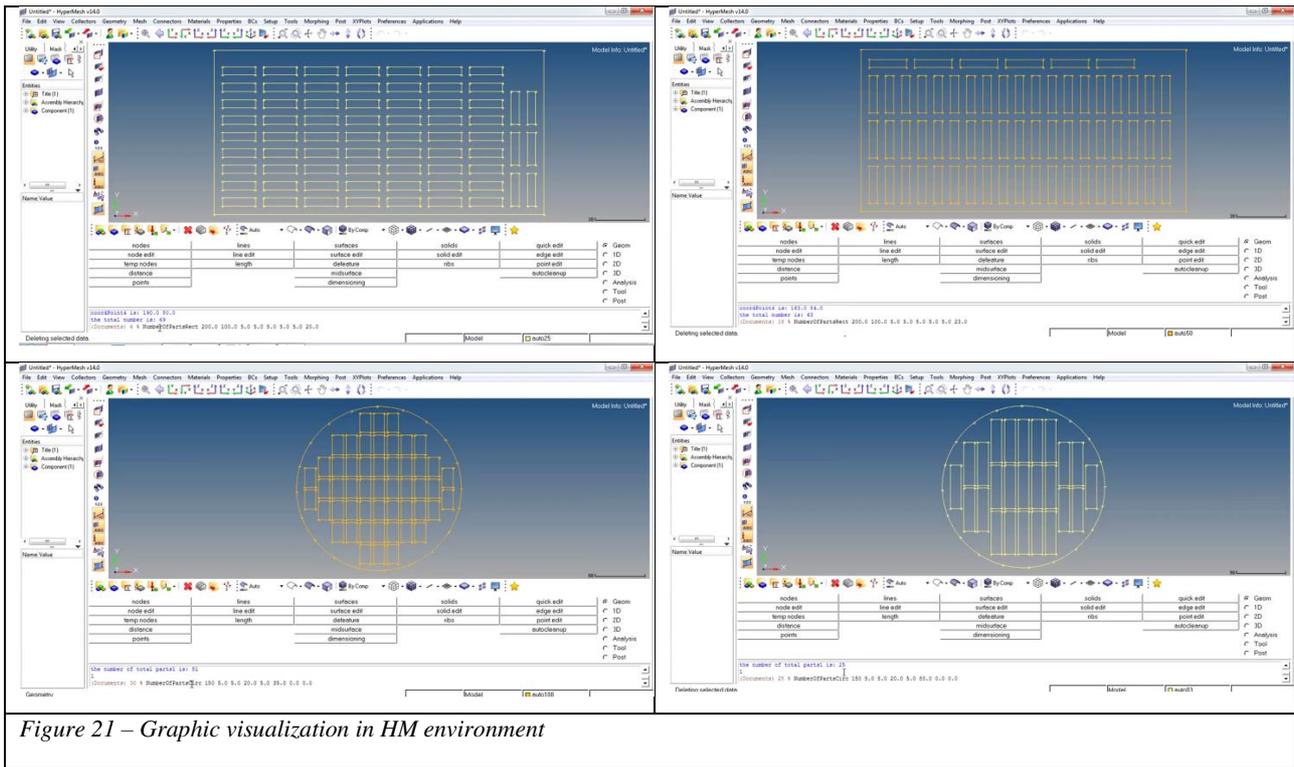


Figure 21 – Graphic visualization in HM environment

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

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 available an 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.

## Conclusions

The technical specifications definition for the previsional model implementation, has led the need to develop a nesting algorithm for the computation of the printable components number on a given printing plan. This variable is necessary for some KPIs calculation that allow 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 components number for a consequent estimate of the costs for a given machine park.

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 solutions that it is possible to implement within other previsional model or framework to evaluate the 3D printer performance considering as AM becomes increasingly widespread and taking into account to have an useful tool to quickly evaluate a potential number of a particular component to print with a determinate machine and for a given extrusion direction.

## Declarations

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

**Conflicts of interest/Competing interests:** The authors declare no conflict of interest.

**Availability of data and material:** Not applicable.

**Code availability:** APPENDIX B reports the complete algorithm in Java.

**Ethics approval:** Not applicable.

**Consent to participate:** Not applicable.

**Consent for publication:** All authors have agreed to authorship, read, and approved the manuscript, and 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.

## References

- [1] Yicha Zhang n, Ravi Kumar Gupta, Alain Bernard, “*Two-dimensional placement optimization for multi-parts production in additive manufacturing*”, *Robotics and Computer-Integrated Manufacturing* 38 (2016) 102–117.
- [2] P. Bartolo, J.P. Kruth, J. Silva, G. Levy, A. Malshe, K. Rajurkar, M. Leu, “*Biomedical production of implants by additive electro-chemical and physical processes*”, *CIRP Ann. – Manuf. Technol.* 61 (2) (2012) 635–655.
- [3] J.P. Kruth, M.C. Leu, T. Nakagawa, “*Progress in additive manufacturing and rapid prototyping*”, *CIRP Ann. – Manuf. Technol.* 47 (2) (1998) 525–540.
- [4] G.N. Levy, R. Schindel, J.P. Kruth, “*Rapid manufacturing and rapid tooling with layer manufacturing (LM) technologies, state of the art and future perspectives*”, *CIRP Ann. – Manuf. Technol.* 52 (2) (2003) 589–609.
- [5] T. Wohlers, Wohlers report 2013, Wohlers association, USA, 2013.
- [6] Y. Zhang, A. Bernard, “*Grouping parts for multiple parts production in additive manufacturing*”, *Procedia CIRP* 17 (2014) 308–313.
- [7] A.S. Gogate, S.S. Pande, “*Intelligent layout planning for rapid prototyping*”, *Int. J. Prod. Res.* 46 (20) (2008) 5607–5631.
- [8] J.K. Dickinson, “*Packing Subsets of Arbitrary Three-dimensional Objects*”, University of Western Ontario, London, Ontario, Canada, 1999, Doctoral dissertation, Ph. D. Thesis, Faculty of Graduate Studies.
- [9] X. Zhang, B. Zhou, Y. Zeng, P. Gu, “*Model layout optimization for solid ground curing rapid prototyping processes*”, *Robot. Comput. – Integr. Manuf.* 18 (1), (2002) 41–51.
- [10] E. Luttersa, D. ten Dama, T. Fanekera, “*3D Nesting of Complex Shapes*”, 45th CIRP Conference on Manufacturing Systems 2012, *Procedia CIRP* 3 (2012) 26 – 31.
- [11] Antonio Del Prete and Teresa Primo, “*Innovative Methodology for the Identification of the Most Suitable Additive Technology Based on Product Characteristics*”, *Metals* 2021, 11, 409. <https://doi.org/10.3390/met11030409>.

## APPENDIX A

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

### A.1. Material Index "MAT"

This index evaluate 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.

$$MAT = \theta_{mat} \cdot \left( \alpha_{appr} \cdot \frac{1}{\sqrt{1 + \tau_{apprw[g]}}} + \alpha_{mt} \cdot \frac{1}{\sqrt{1 + \tau_{AMmt[g]}}} + \alpha_{fr} \cdot (2 - f_r) \right) \quad (A1)$$

- $\theta_{mat}$ : compatibility between material and machine
- $\alpha_{appr}$ : metal powder supply coefficient
- $\tau_{apprw[g]}$ : supply time for metallic powders [days]
- $\alpha_{mt}$ : material change coefficient
- $\tau_{AMmt[g]}$ : material change time [days]
- $\alpha_{fr}$ : material processability risk coefficient
- $f_r$ : 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.

### A.2. 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.

$$CST = \theta_{mat} \cdot \theta_{ing} \cdot \left\{ \left[ (n \cdot \delta_d) \cdot \left( \frac{V_d}{\varepsilon_{wd}} + \frac{S_d \cdot \gamma_{rt} \cdot \varphi_w}{\varepsilon_{wr}} \right) \right] \cdot C_w + [n \cdot \delta_t \cdot V_{boundt} \cdot \varphi_t] \cdot C_t \right. \\ \left. + [n \cdot \mu_g \cdot S_{bound} \cdot (1 + \varphi_g) \cdot \delta_g] \cdot C_g \right\} \cdot f_r \\ + \left\{ K_d \cdot \left( c_{dop} \cdot (\tau_{dor} + \tau_{dsl} + \tau_{dps} + \tau_{dcm}) \right) \right\} + K_{AM} \\ \cdot \left[ (c_{AMeq} + c_{AMmh} + c_{AMen} + c_{AMop}) \right. \\ \cdot \left( \tau_{AMmt} + (\tau_{AMstart} + \tau_{AMrisc}) \cdot \left[ \frac{n}{m} \right] + \frac{V_d}{P_{AM}} \cdot n + \tau_{AMcc} \cdot \left( \left[ \frac{n}{m} \right] - 1 \right) + (\tau_{AMraffr} + \tau_{AMclean}) \right. \\ \left. \left. \cdot \left[ \frac{n}{m} \right] \right) \right\} \quad (A1)$$

- $\theta_{mat}$ : compatibility between material and machine
- $\theta_{ing}$ : compatibility between component and the machine working volume
- $\delta_d$ : material density indicated by the customer (kg/m<sup>3</sup>);
- $V_d$ : volume of the component to be produced (m<sup>3</sup>);
- $S_d$ : total area of the component (m<sup>2</sup>);
- $\gamma_{rt}$ : average machining allowance thickness identified by technology (m)

- $\varphi_w$ : ratio between the surface subject to machining allowance and the total surface area of the component (%);
- $\varepsilon_{w_r}$ : powders management efficiency in the production of machining allowance (%);
- $\varepsilon_{w_d}$ : powders management efficiency in the production of the component (%);
- $C_w$ : unit cost of metal powders (€/kg);
- $\delta_t$ : material density chosen for the supports (kg/m<sup>3</sup>);
- $V_{bound_t}$ : parallelepiped volume of the containment of support structures (m<sup>3</sup>);
- $\varphi_t$ : technological coefficient for the mass of material for the supports (%);
- $C_t$ : unit cost of the material required for the supports (€/kg);
- $\delta_g$ : material density of the anchor plate (kg/m<sup>3</sup>);
- $\mu_g$ : thickness of the anchor plate (kg/m<sup>3</sup>);
- $S_{bound}$ : surface of the component's bounding box (kg/m<sup>3</sup>);
- $\varphi_g$ : increase coefficient of the bounding box surface (%);
- $f_r$ : material risk coefficient;
- $C_g$ : unit cost of the anchor plates (€/kg);
- $K_d$ : complexity coefficient for process design;
- $K_{AM}$ : complexity coefficient for additive production;
- $c_{d_{op}}$ : operator hourly cost for process design;
- $\tau_{d_{or}}$ : Operator time to identify component orientation (h)
- $\tau_{d_{sl}}$ : Operator time to identify the optimal slicing strategy (h)
- $\tau_{d_{ps}}$ : Operator time to identify process parameters (h)
- $\tau_{d_{cm}}$ : Operator time to set the tool path (h)
- $c_{AM_{eq}}$ : hourly cost for Additive Production due to amortization;
- $c_{AM_{mh}}$ : hourly cost for Additive Production due to maintenance;
- $c_{AM_{en}}$ : hourly cost for Additive Production due to energy consumption;
- $c_{AM_{op}}$ : hourly cost for Additive Production due to the operator;
- $\tau_{AM_{mt}}$ : material change time (h);
- $\tau_{AM_{start}}$ : machine start-up time (h)
- $\tau_{AM_{risc}}$ : machine preheating time (h);
- $n$ : components of the lot number;
- $m$ : maximum number of components that can be produced within a job;
- $\left\lceil \frac{n}{m} \right\rceil$ : number of jobs required by a machine with a maximum capacity of m components to make n components;
- $\tau_{AM_{cc}}$ : cycle change time (h);
- $P_{AM}$ : machine productivity (m<sup>3</sup>/h);
- $\tau_{AM_{raffr}}$ : chamber cooling time (h);
- $\tau_{AM_{clean}}$ : excess powders removal time (h).

### A.3. 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.

$$\begin{aligned}
TMP = \theta_{mat} \cdot \theta_{ing} & \cdot \left\{ \tau_{appr_{w[h]}} + (\tau_{dor} + \tau_{d_{sl}} + \tau_{d_{ps}} + \tau_{d_{cm}}) + [\tau_{AM_{mt}} + (\tau_{AM_{start}} + \tau_{AM_{risc}}) \cdot \left(\frac{n}{m}\right)] \right. \\
& \left. + \left[ \frac{V_d}{P_{AM}} \cdot n + \tau_{AM_{cc}} \cdot \left(\left(\frac{n}{m}\right) - 1\right) \right] + [(\tau_{AM_{raffr}} + \tau_{AM_{clean}}) \cdot \left(\frac{n}{m}\right)] \right\} \quad (A2)
\end{aligned}$$

- $\theta_{mat}$ : compatibility between material and machine
- $\theta_{ing}$ : compatibility between component and the machine working volume
- $\tau_{appr_{w[h]}}$ : powder supply time (h)
- $\tau_{dor}$ : operator time to identify component orientation (h)
- $\tau_{d_{sl}}$ : operator time to identify the optimal slicing strategy (h)
- $\tau_{d_{ps}}$ : operator time to identify process parameters (h)
- $\tau_{d_{cm}}$ : operator time to set the tool path (h)
- $\tau_{AM_{mt}}$ : material change time (h)
- $\tau_{AM_{start}}$ : machine start-up time (h)
- $\tau_{AM_{risc}}$ : machine preheating time (h)
- $m$ : maximum components number that can be produced within a job
- $\tau_{AM_{raffr}}$ : room cooling time (h)
- $\tau_{AM_{clean}}$ : excess powder removal time (h)
- $\tau_{AM_{cc}}$ : cycle change time (h)
- $n$ : number of components of the lot
- $P_{AM}$ : machine productivity (m<sup>3</sup>/h)
- $V_d$ : total volume to be melted (m<sup>3</sup>)

#### A.4. Overall Dimensions of the Component in the Machine "ING"

It evaluates the compatibility between the dimensions of the component(s) to be produced 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.

$$ING = \theta_{ing} \cdot \left\{ \frac{\left[ \left( \frac{[(x_{AM} - X_d)]}{x_{AM}} + \frac{(Y_{AM} - Y_d)}{y_{AM}} + \frac{(z_{AM} - z_d)}{z_{AM}} \right) / 3 \right] + \left[ \frac{n}{m} - (n_{cycle} - 1) \right]}{2} \right\} \quad (A3)$$

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

#### A.5. 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.

$$PRE = \theta_{pre} \cdot \frac{(\zeta_d - \zeta_{AM})}{\zeta_d} \quad (A4)$$

- $\theta_{pre}$ : product–machine coefficient,  $\theta_{pre} = 0 \rightarrow \zeta_d < \zeta_{AM}$   $\theta_{pre} = 1 \rightarrow \zeta_d > \zeta_{AM}$ ;
- $\zeta_d$ : reference precision of the component;
- $\zeta_{AM}$ : machine precision.

#### A.6. 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.

$$RIS = \theta_{ris} \cdot \frac{(\xi_d - \xi_{AM})}{\xi_d} \quad (A5)$$

- $\theta_{ris}$ : product–machine coefficient,  $\theta_{ris} = 0 \rightarrow \zeta_d < \zeta_{AM}$   $\theta_{ris} = 1 \rightarrow \zeta_d > \zeta_{AM}$ ;
- $\xi_d$ : reference resolution of the component;
- $\xi_{AM}$ : machine resolution.

#### A.7. Undercuts 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.

$$STQ = \frac{S_d}{\left(\frac{S_t}{\psi^2} + S_d\right)} \quad (A6)$$

- $S_d$ : total component surface;
- $S_t$ : total undercut surface;
- $\psi$ : support management coefficient.

#### A.8. 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.

$$RGS = \theta_{rgs} \cdot \frac{(\chi_d - \chi_{AM})}{\chi_d} \quad (A7)$$

- $\theta_{rgs}$ : product–machine coefficient,  $\theta_{rgs} = 0 \rightarrow \chi_d < \chi_{AM}$   $\theta_{rgs} = 1 \rightarrow \chi_d > \chi_{AM}$ ;

- $\chi_d$ : component reference roughness;
- $\chi_{AM}$ : roughness obtainable with the machine.

#### A.9. 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.

$$CBA = \theta_{mat} \cdot \theta_{ing} \cdot \left\{ \alpha_{gmt} \cdot \left( \frac{PRE + RIS + RGS + STQ}{4} \right) + \alpha_{cst} \cdot \left( \frac{1 - CST}{CST_{max} \cdot (1 + \alpha_{cst})} \right) + \alpha_{tmp} \cdot \left( \frac{1 - TMP}{TMP_{max} \cdot (1 + \alpha_{tmp})} \right) \right\} \quad (A8)$$

The KPI in equal to zero when  $\theta_{mat}$  or  $\theta_{ing}$  are 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 are 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.

- $\theta_{mat}$ :  $\theta_{mat} = 0 \rightarrow$  incompatible material  $\theta_{mat} = 1 \rightarrow$  compatible material;
- $\theta_{ing}$ : compatibility between component and the machine working volume
- $\alpha_{gmt}$ : weight associated with geometric KPIs;
- $\alpha_{cst}$ : weight associated with cost KPIs;
- $\alpha_{tmp}$ : weight associated with time KPIs;
- PRE: technology precision;
- RIS: technology resolution;
- RGS: technology roughness;
- STQ: undercuts management;
- CST: production cost;
- TMP: production time;
- $CST_{max}$ : maximum CST value calculated among all available machines;
- $TMP_{max}$ : maximum TMP value calculated among all available machines.

## APPENDIX B

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)));
                }
                j++;
            }
        }
    }
}
```

```

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;
    }
}

```

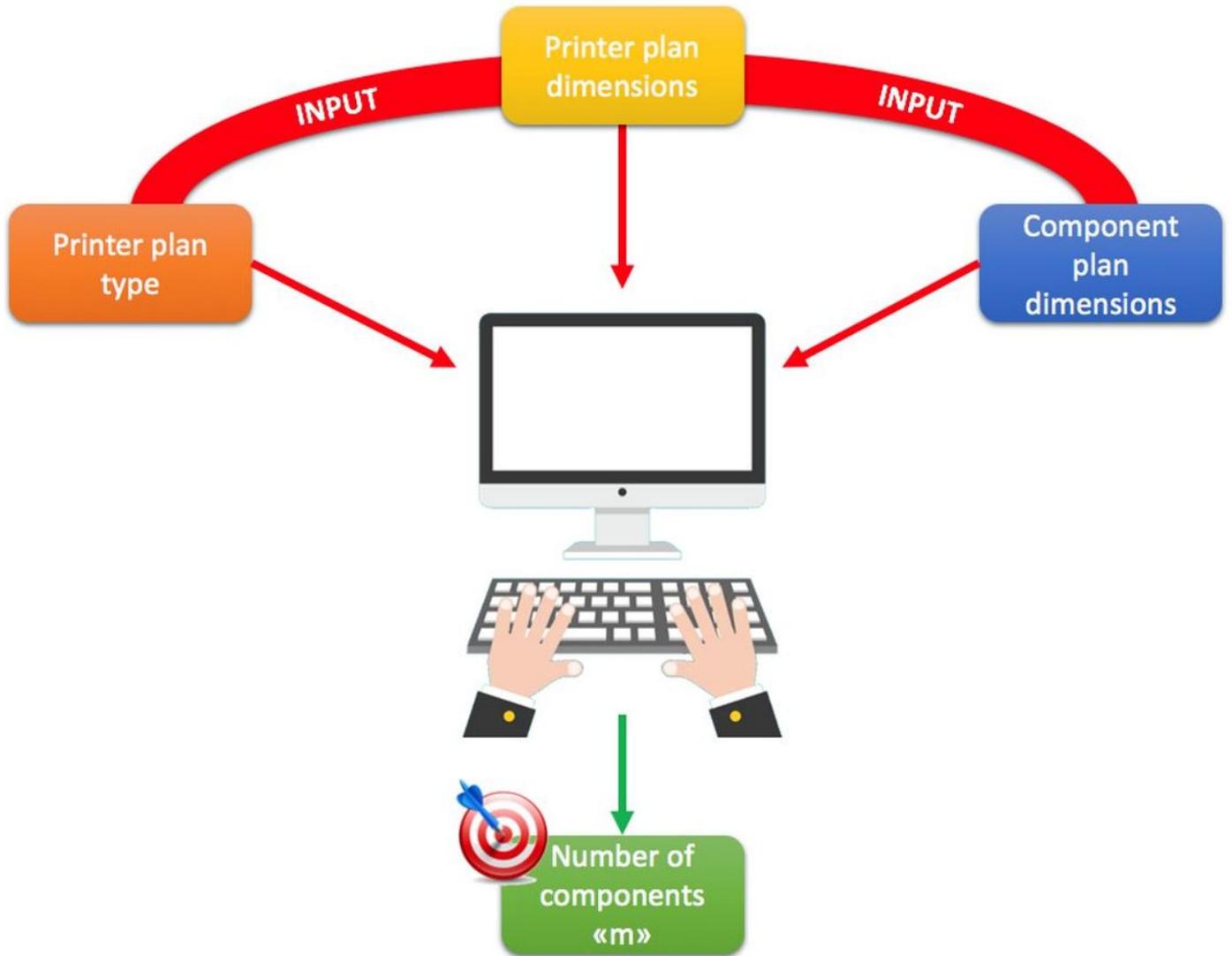
```

        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);
    }
}
}
}

```

# Figures

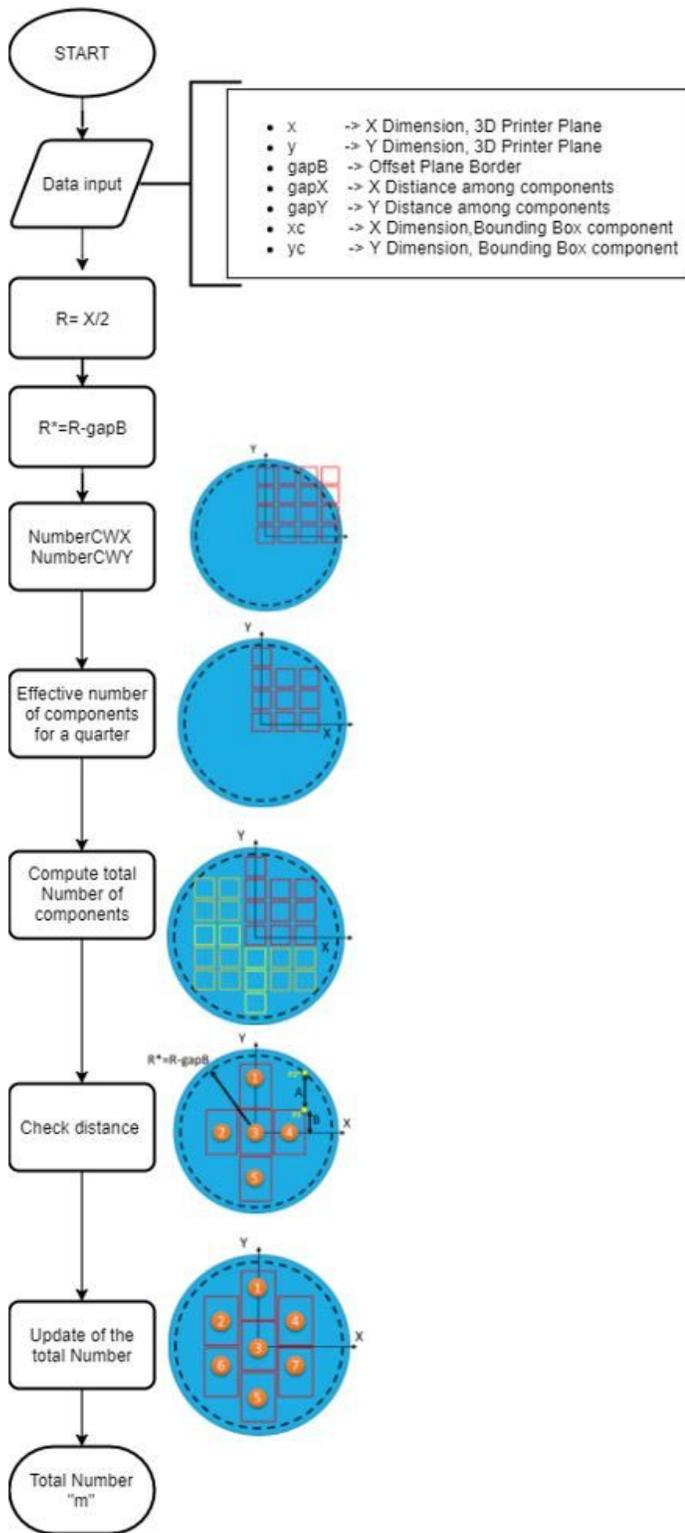


**Figure 1**

Definition of the Nesting algorithm

**Figure 2**

Nesting algorithm diagram



**Figure 3**

Circular nesting workflow

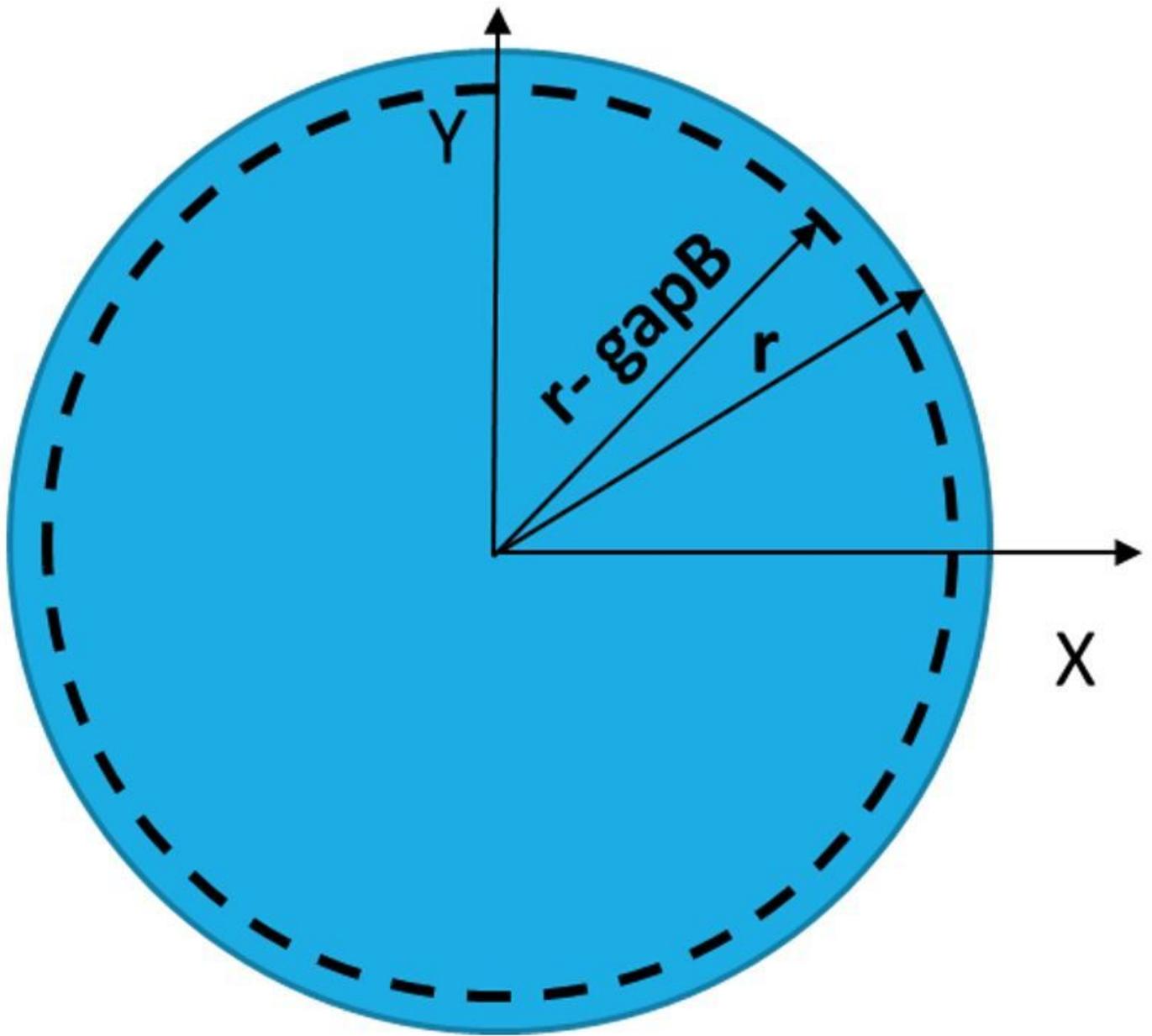


Figure 4

Circular printing plan

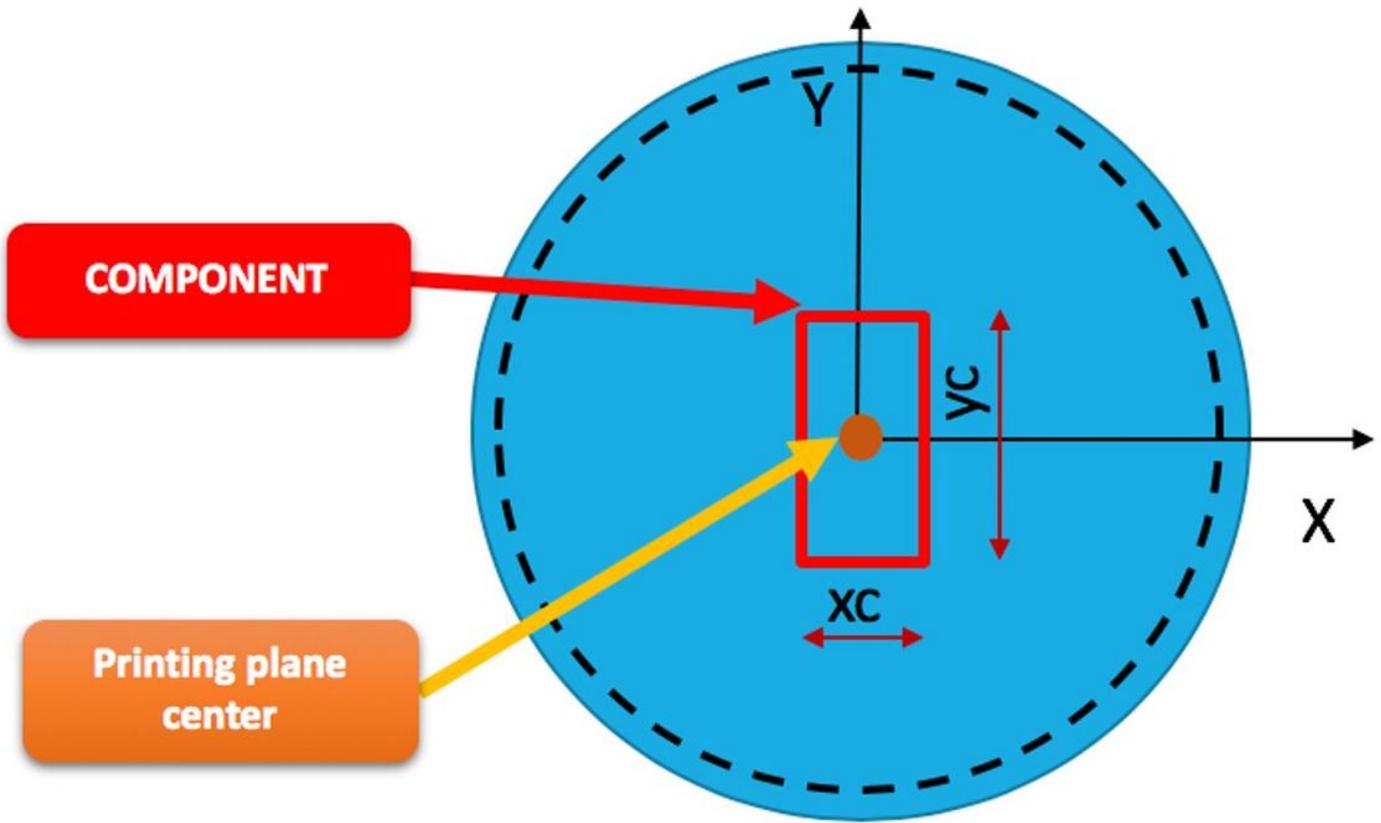


Figure 5

Circular printing plan - positioning

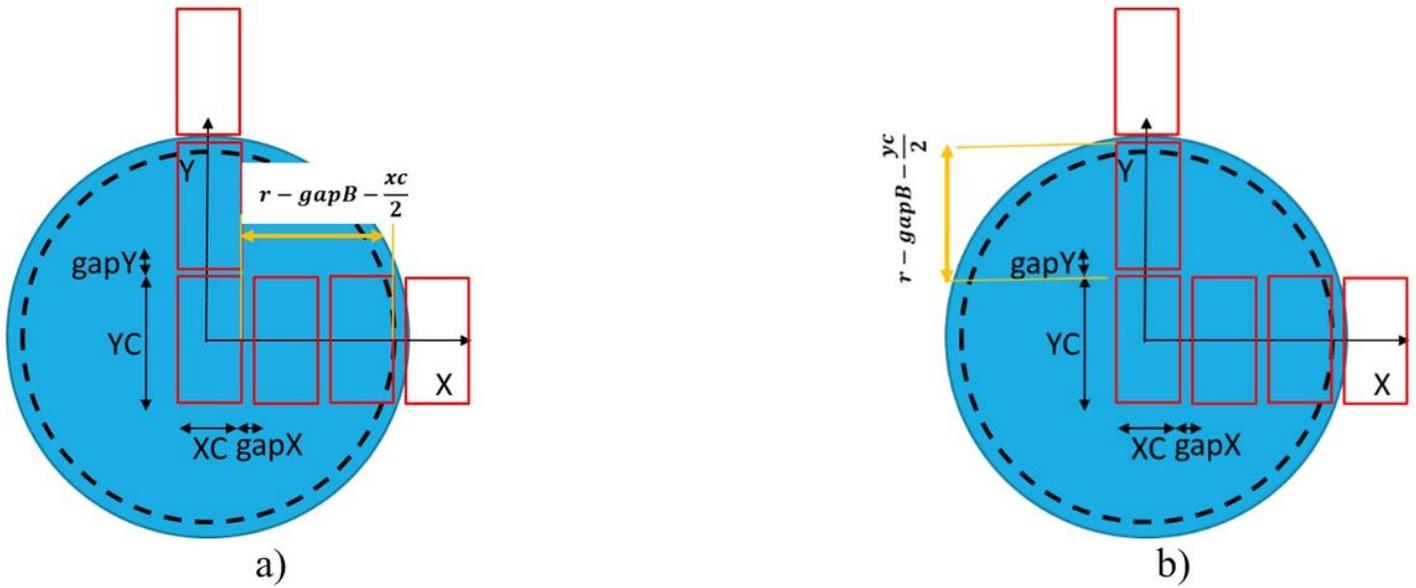


Figure 6

a) NumberWX Calculation; b) NumberWY calculation

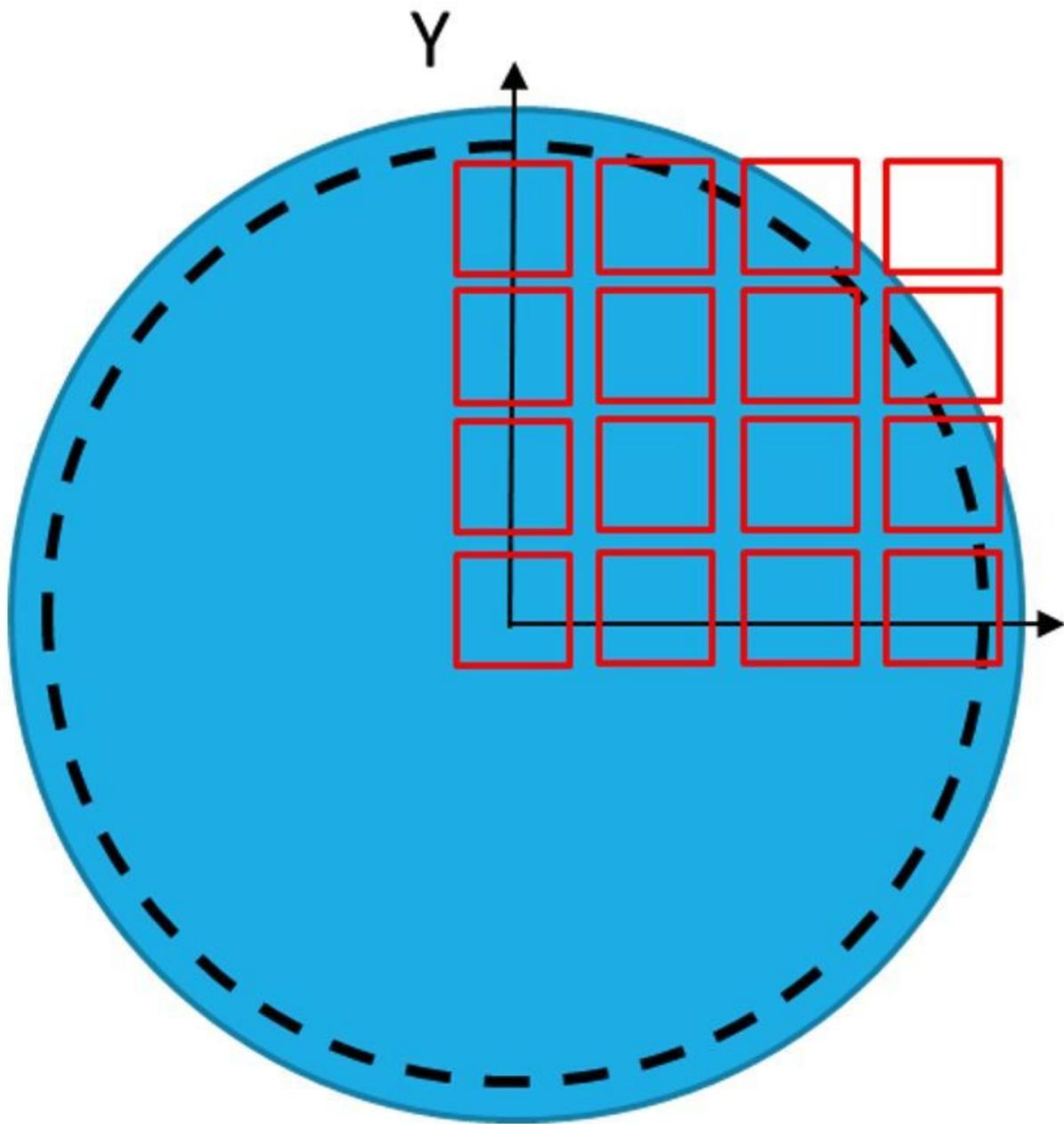
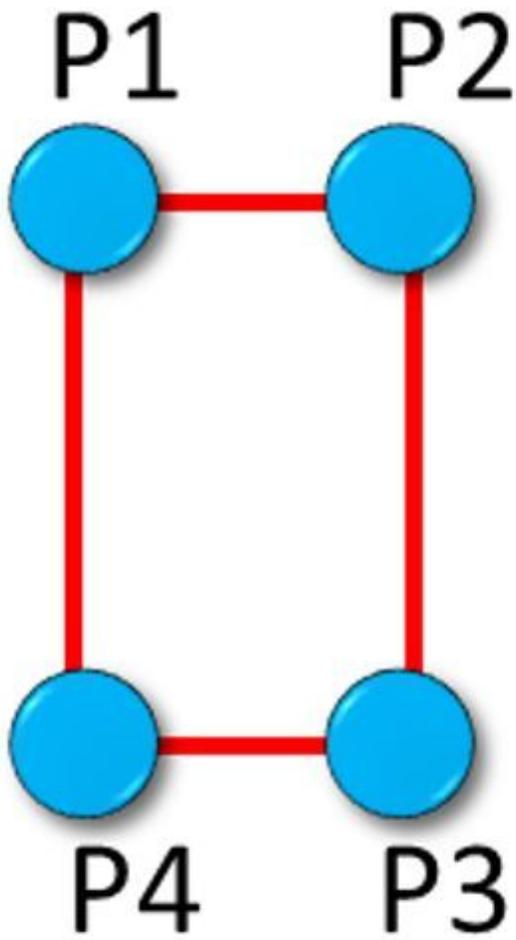


Figure 7

Circular printing plan - maximum number of components



**Figure 8**

Control Vertices

**Figure 9**

Result of the check of the components inside the circular printing plan

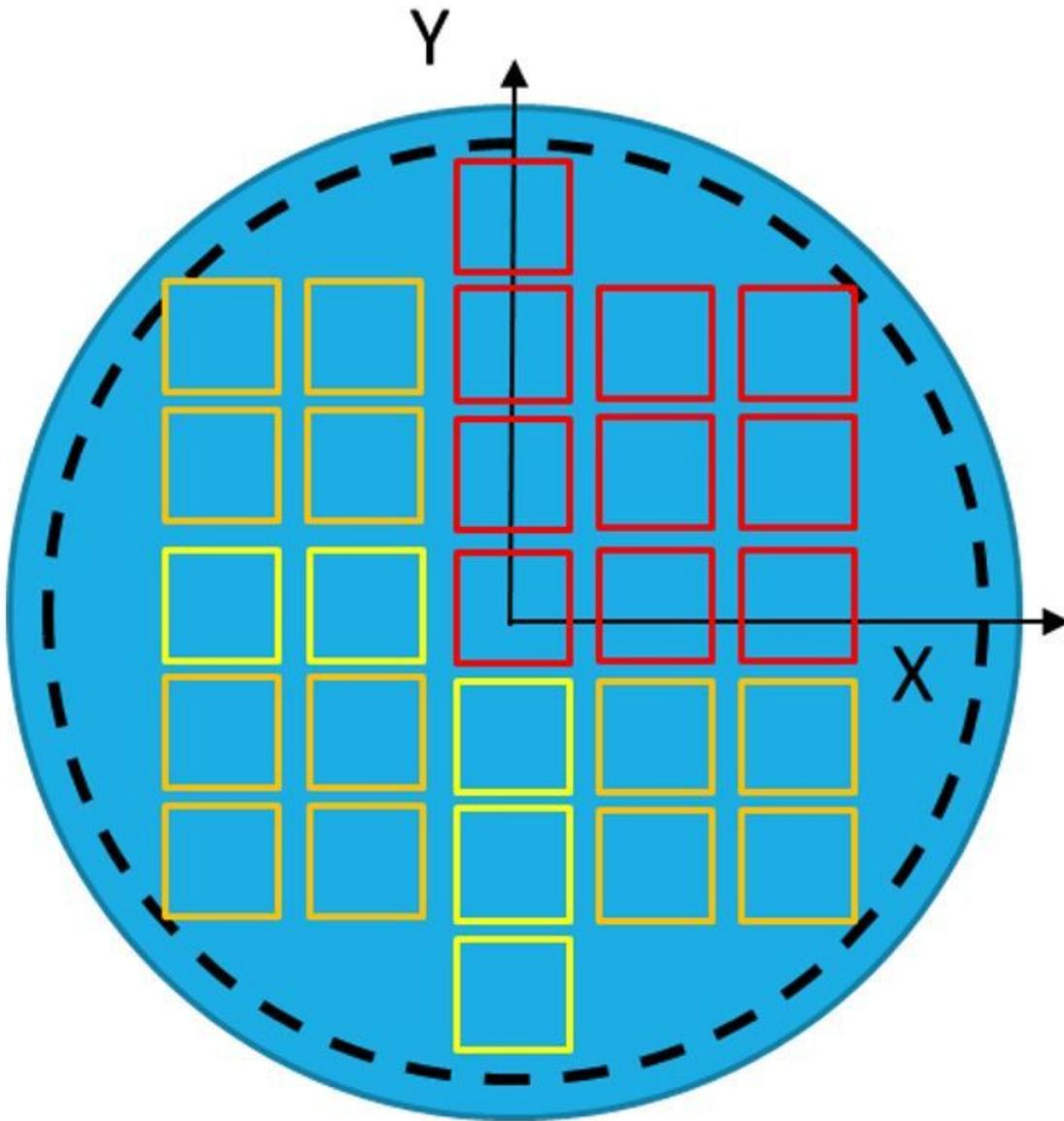


Figure 10

Result of the check of the components inside the whole circular printing plan

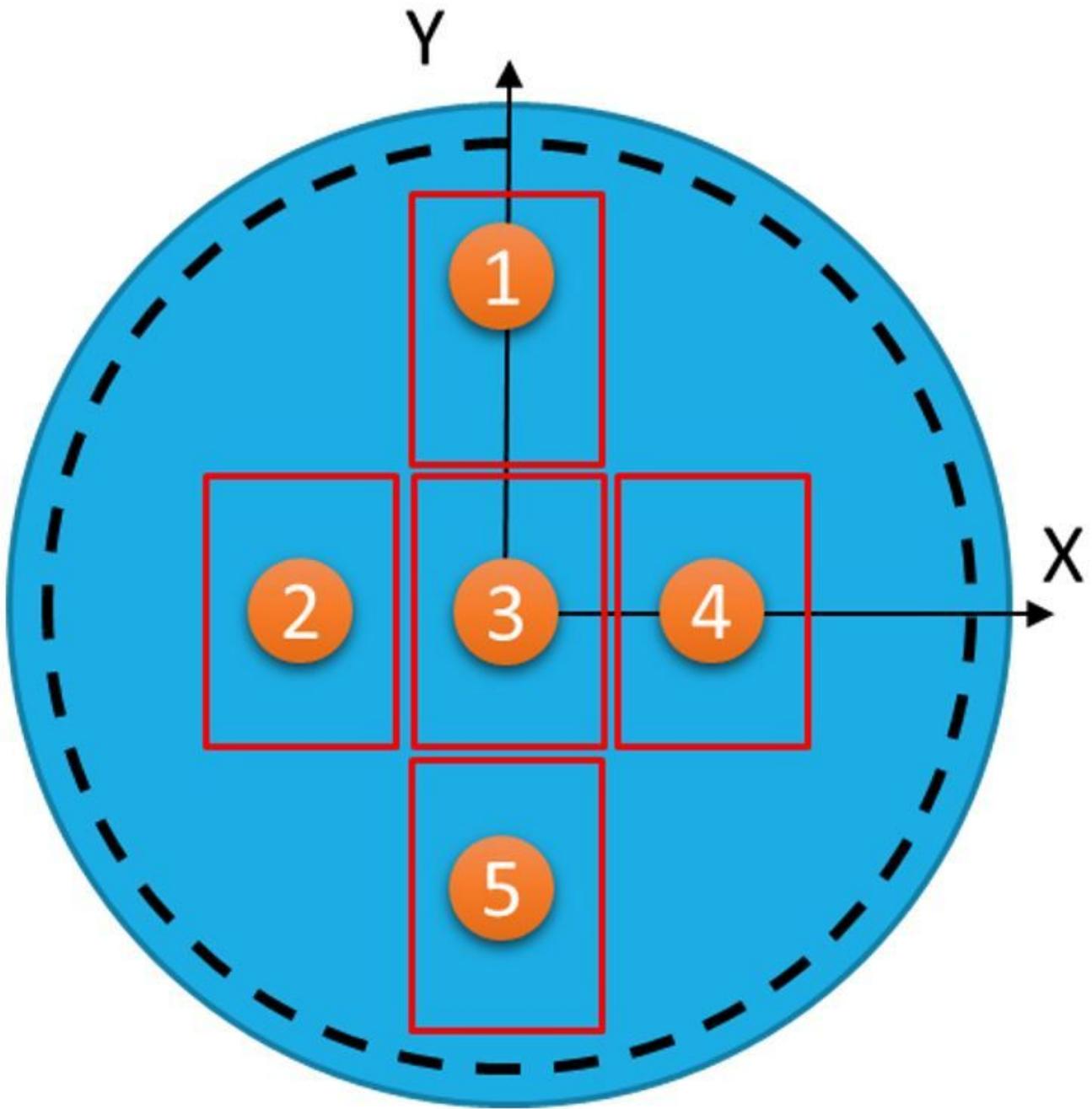


Figure 11

Scenario for which further implementation is needed

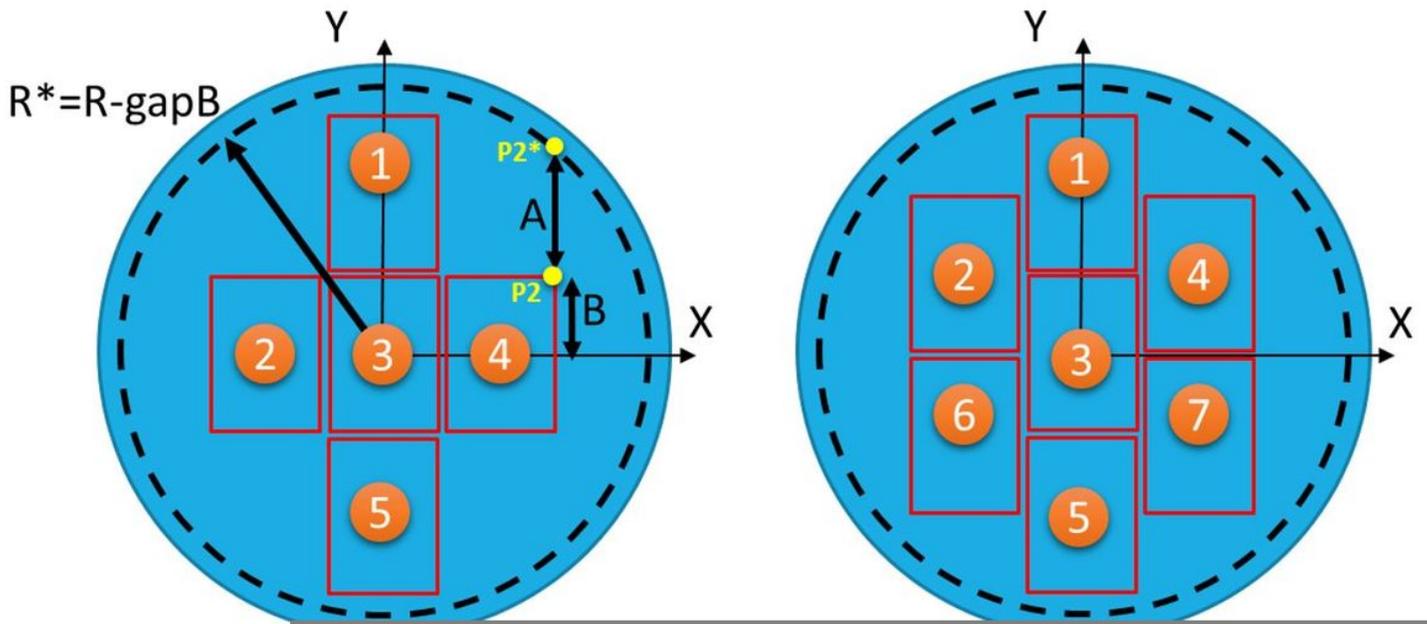


Figure 12

(a) Size check (b) New layout

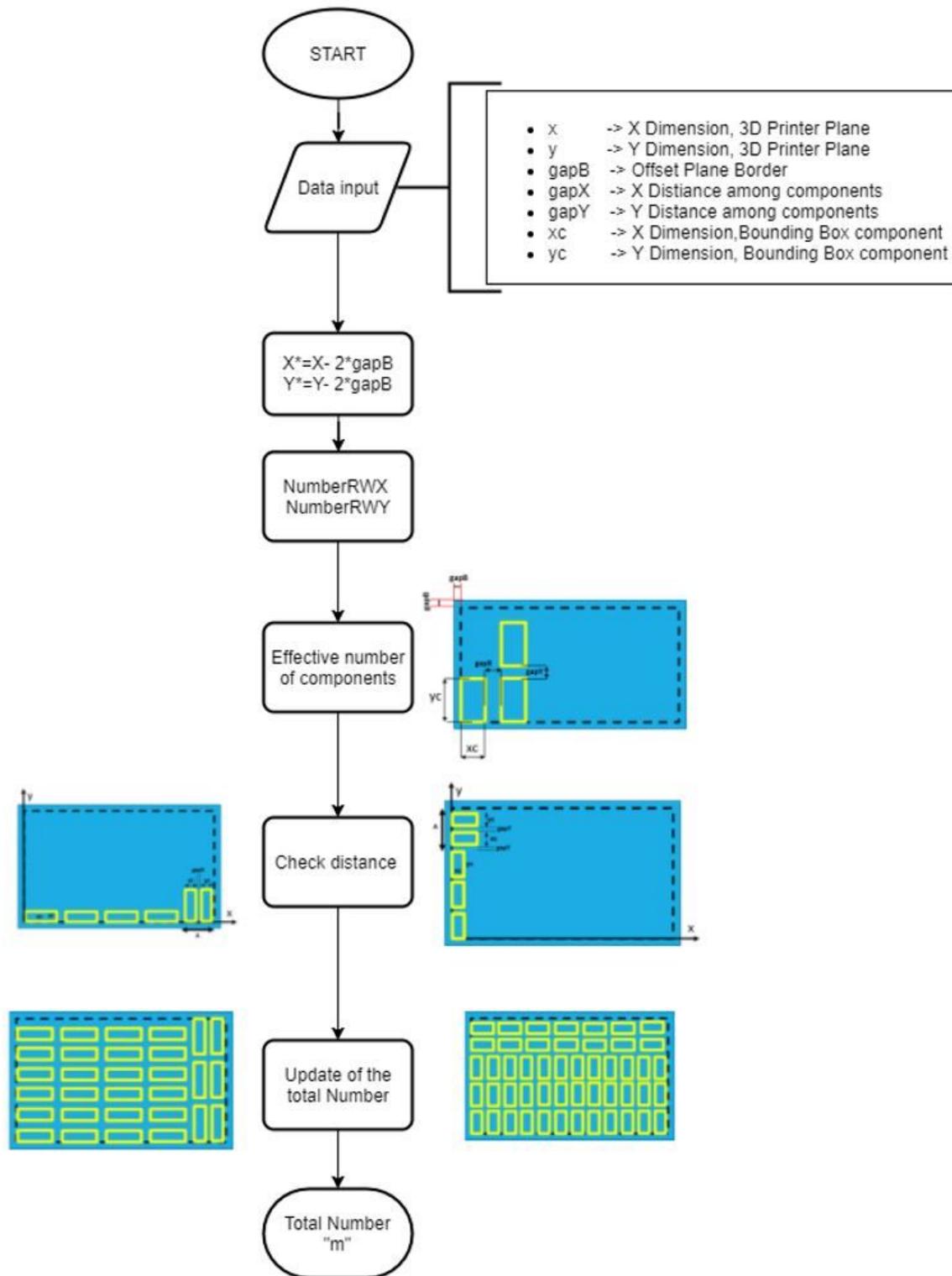


Figure 13

Rectangular nesting workflow

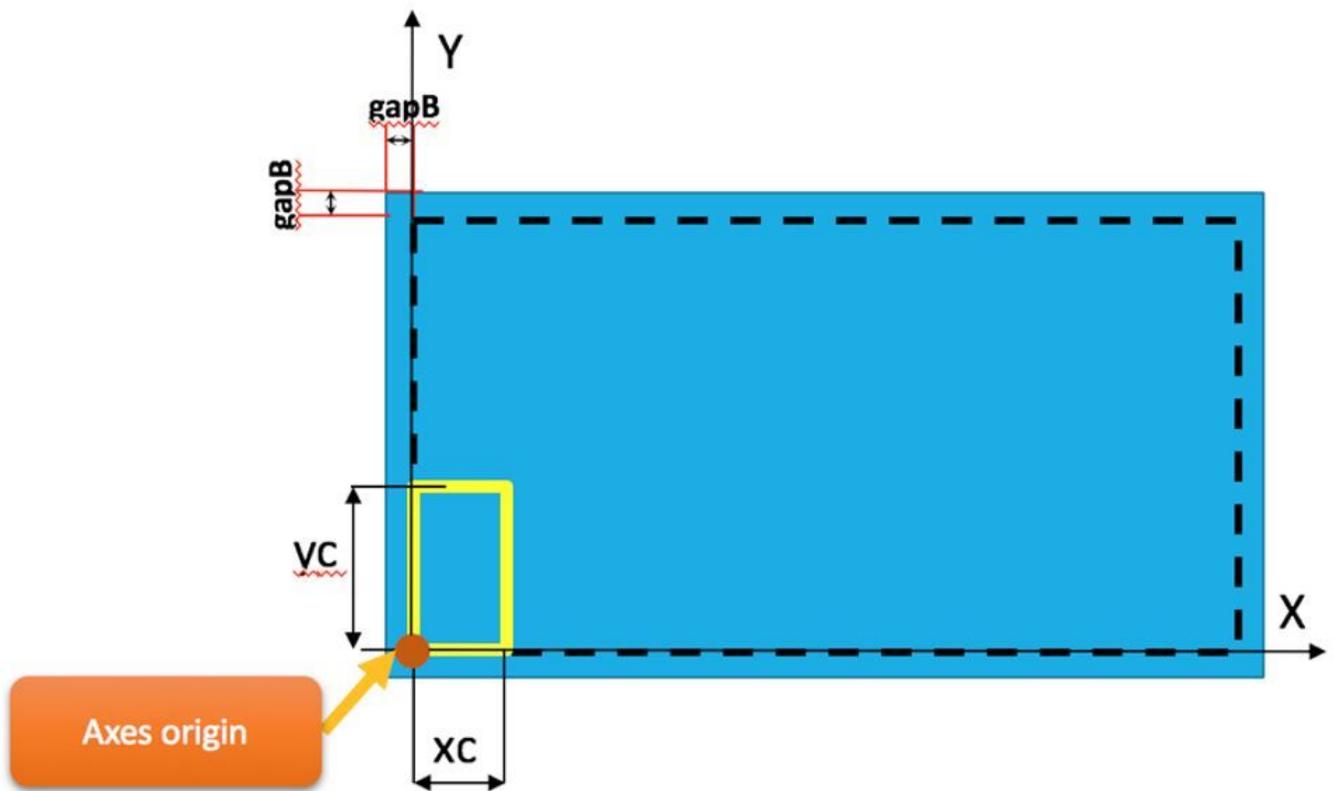


Figure 14

Rectangular printing plan

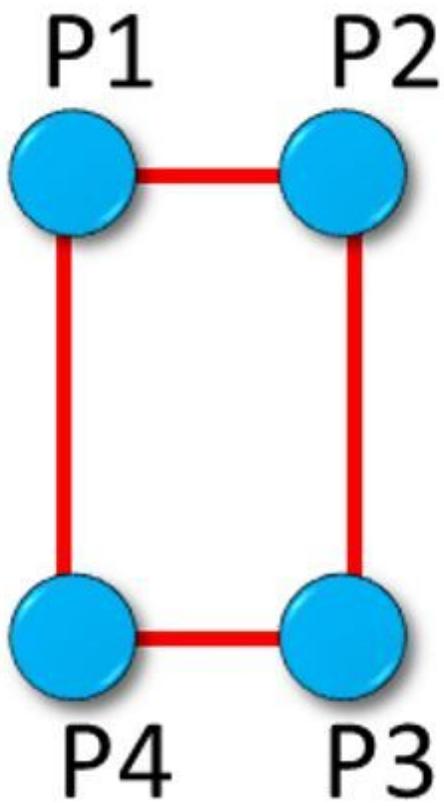


Figure 15

Control Vertices

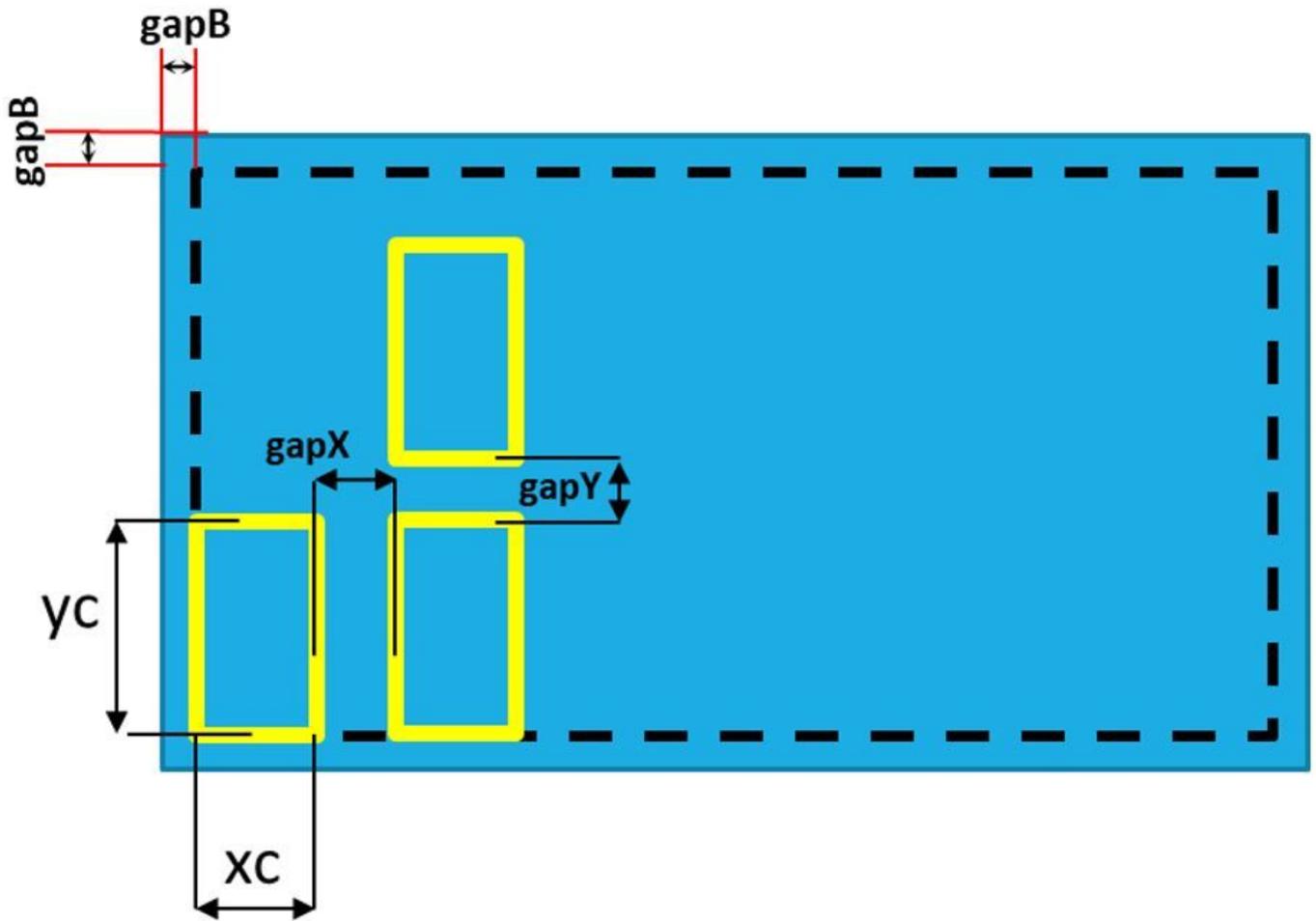


Figure 16

Rectangular printing plan - positioning

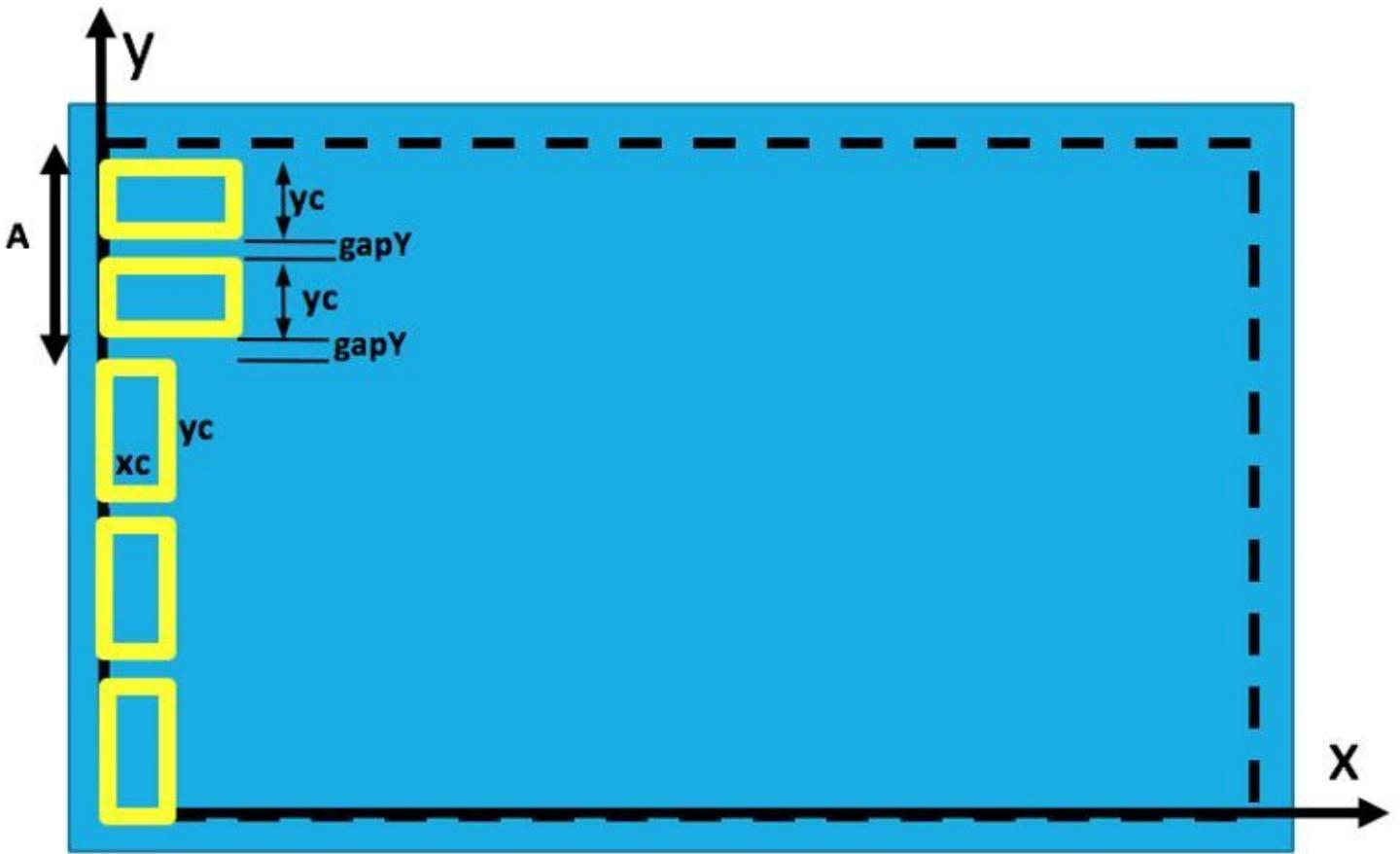


Figure 17

New positioning, scenario 1

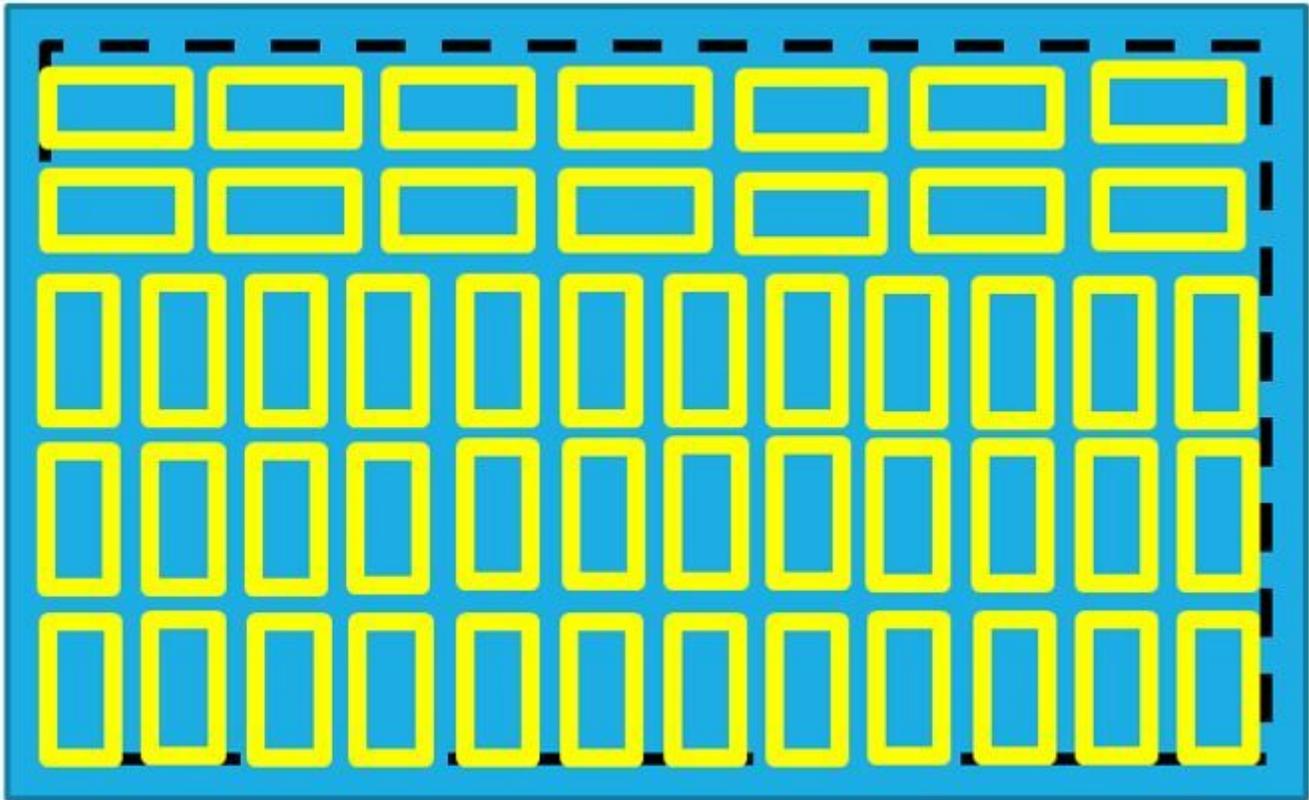


Figure 18

New complete positioning, scenario 1

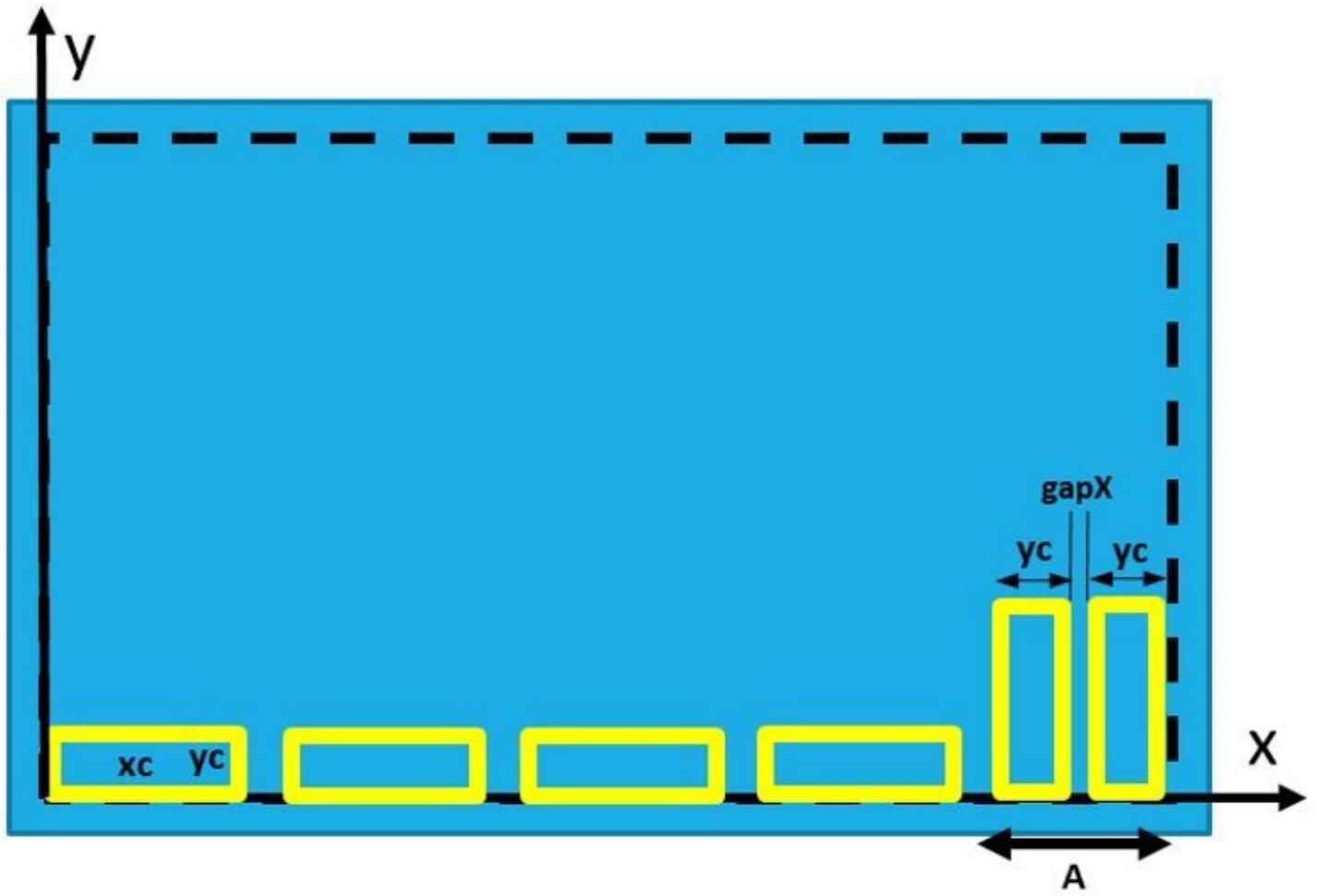


Figure 19

New positioning, scenario 2

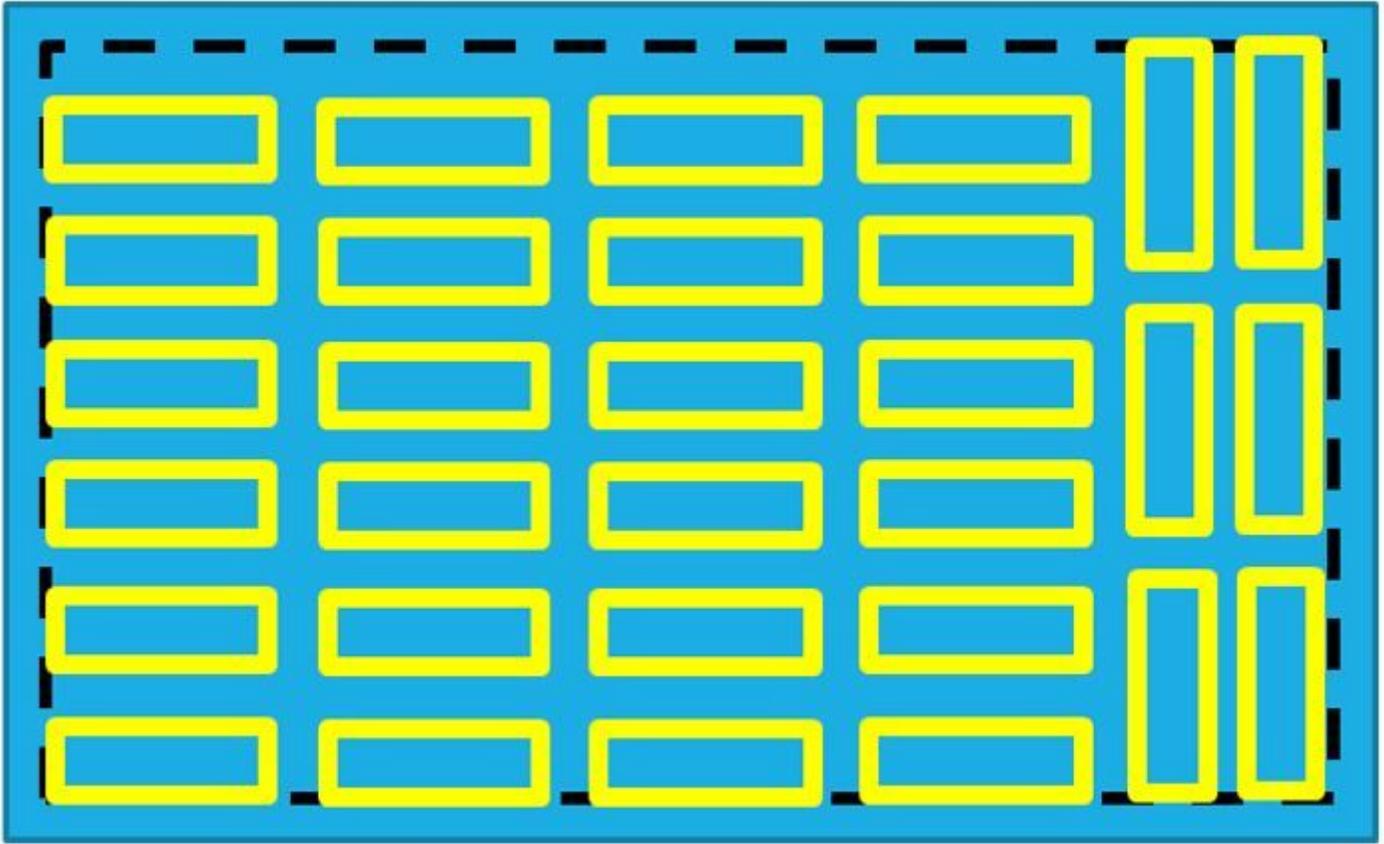


Figure 20

New complete positioning, scenario 2

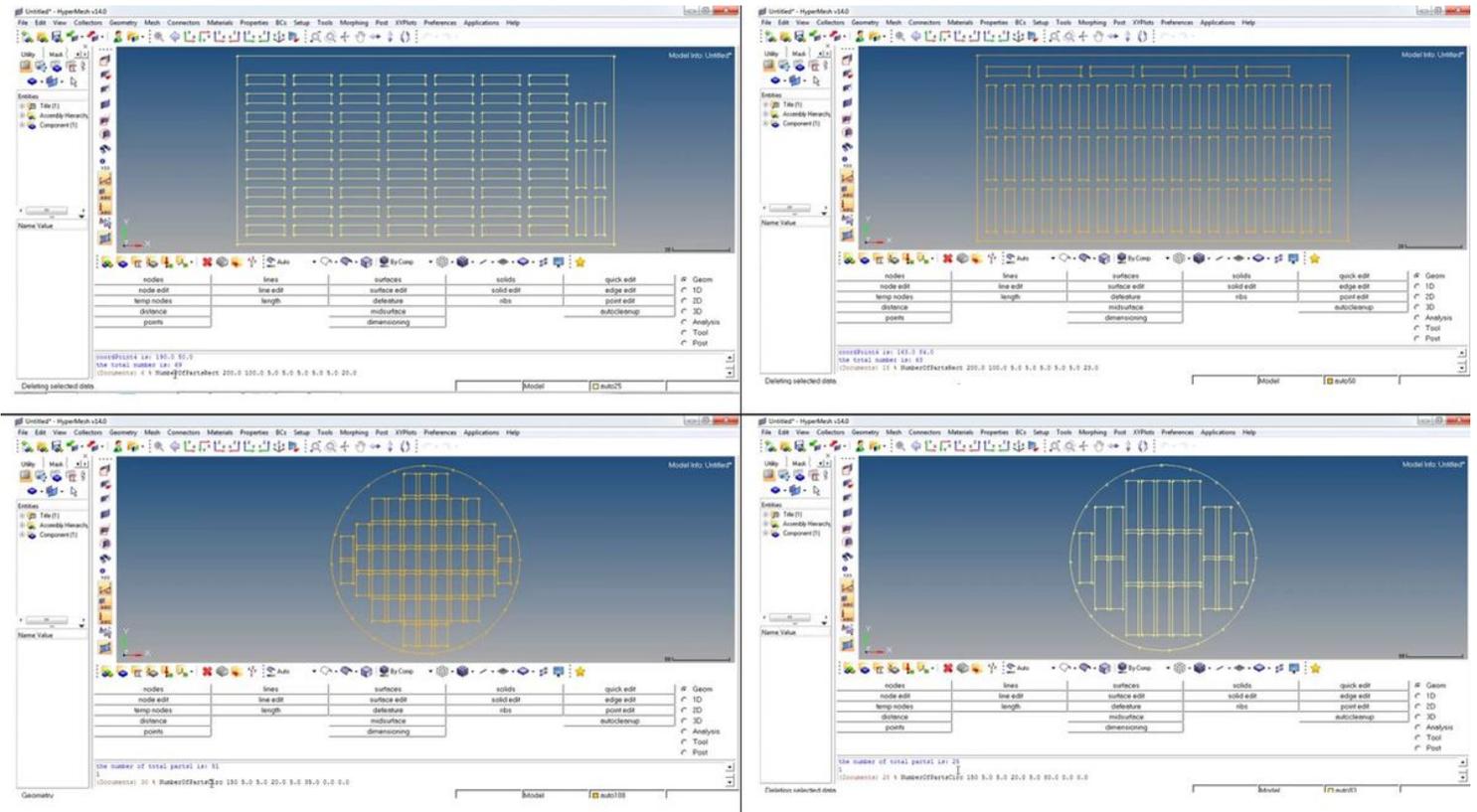


Figure 21

Graphic visualization in HM environment