International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 3 (2018) pp. 1784-1790

Linear Models for Non-Congruent Circle Packing in a Rectangular Container

Rafael Torres-Escobar[1], José Antonio Marmolejo-Saucedo[2]* and Igor Litvinchev[3]

Center of Top Management in Engineering and Technology, Anahuac University Mexico North Campus, 55786 Huixquilucan, México.

Universidad Panamericana. Facultad de Ingeniería. Augusto Rodin 498, México, Ciudad de México, 03920, México.

Complex Systems Department, Computing Center, Russian Academy of Sciences, 119333 Moscow, Russia.

**Correspondence author*

Abstract

The problem of packing non-congruent circles within bounded regions is considered. The aim is to maximize the number of circles placed into a rectangular container or minimize the waste. The circle is considered as a set of points that are all the same distance (not necessary Euclidean) from a given point. An integer programming model is proposed using a dotted-board approximating the container and considering the dots as potential positions for assigning centers of the circles. The packing problem is then stated as a large scale linear 0–1 optimization problem due to the number of variables generated mainly in the overlap constraints. Binary decision variables are associated with each discrete point of the board (a dot) and with each object. The resulting binary problem is then solved by using a commercial solver and a general-purpose programming language. Nesting circles inside one another is considered. Numerical results on packing circles, rhombuses and squares are presented to demonstrate the efficiency of the proposed approach.

Keywords: Circle Packing, Integer Programming, Optimization

INTRODUCTION

Packing problems can be important components in other studies such as cutting patterns, area coverage, the design of facilities, loading of vehicles, assignment, sequencing of equipment and chain management of supply [1–4]. A Packing problem consists in the best arrangement of several objects inside a bounded area named as the container. This arrangement must fulfill with technological constraints, for example, objects should not be overlapping. The shape of the container may vary from a circle, a square, a rectangle, etc. An extension of the packing problem can be incorporated into the following problems [5–7].

This paper focuses on problems dealing with circular objects in two dimensions. Circular packing problems are difficult problems, owing to the combinatorial nature in the arrangement of such objects. In fact, they are NP-hard combinatorial optimization problems [8,9]; this means that there is not exists a procedure to exactly solves them in deterministic polynomial time [3].

The circle packing problem has been extensively documented [10–13], the aim is to place a certain number of circles, each one with a fixed known radius inside a container. The circles must totally lie inside the container without overlapping. In some cases, the objective of the problem becomes the maximization of the profits of the items placed (known as the weighted version of the problem).

Depending on the situation it may be more advantageous to maximize the weight of the items or other aspects of the items in a specific container as minimizing of wasting [14]. This type of problem has many applications, mainly in the packing industry, although there are also applications in the timber industry to optimize cuts of wood or metal in small pieces, textile, naval, automobile, aerospace and food industries and computer-aided design like image stippling [2,3,15,16].

Some packing models for circular objects are typically formulated as non-convex optimization problems; where the continuous variables are the coordinates of the objects, so they are limited to not finding optimal solutions [17–19]. The nonconvexity is mainly provided by non-overlapping conditions among circles. These conditions typically state that the Euclidean distance separating the centers of the circles is greater than a sum of their radii [20]. For this reason, the present work proposes large scale integer linear models with assignment constraints assuring the convergence to an optimal solution. It should be clarified that the optimal solutions for these models are achieved with the discretization of space, may not be the optimal solutions for the original problem with continuous variables.

In this article, we study the packing of circular objects using a regular grid (see Figure 1) to approximate the shape of the rectangular container. The circular object is considered in a general form, as the set of points that are at the same distance from a center. Different shapes of objects like squares, circles, rectangles or rhombuses are treated in the same way, only the standard used to define the distance is changed. Nodes of a dotted-board are considered as potential assignments for the centers of each object. It is also considered the placement of small objects within larger ones. Numerical results with circles, rhombuses and squares are shown to demonstrate the efficiency of the proposed models.

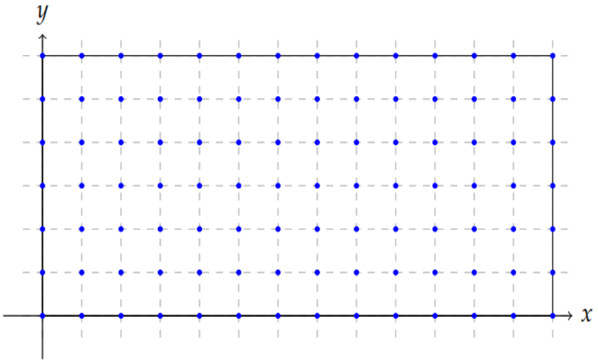


Figure 1. Example of grid covering a rectangular container.

The rest of the article is structured as follows, Section 1 explains the entire programming model for the problem of packing circular-like objects. Section 2 shows the results of the packing of circles, rhombuses and squares. In the final section the conclusions of this article are shared.

THE MODEL

The main idea of this model was first implemented in [21], later a similar approach was used in [22,23] for objects with irregular shape. This article is a continuation of the work proposed by [24,25]; where they only reported results for equal circles, rhombuses and octagons.

Suppose we have non-identical circles C_k of known radius $R_k, k \in K = \{1,2,\dots,K\}$. Here we consider the circle as a set of points that are all the same distance R_k (not necessary Euclidean) from a given point. In what follows, we will use the same notation C_k for the figure bounded by the circle $C_k = \{z \in \mathbb{R}^2 : \|z - z_{0k}\| \le R_k\}$, here, z_{0k} is the center of the object. Denote W_k by the area of C_k. Let b_k be the number of circles C_k are available for packing and at least a_k of them must be packed. Denote by $i \in I = 1,2,\dots,n$ the node points of a regular grid covering the rectangular container. Let $F \subset I$ be the grid points lying on the boundary of the container. Denote by d_{ij} the distance (in the sense of norm used to define the circle) between points i and j of the grid. Define binary variables $x_i^k = 1$ if a center of a circle C_k is assigned to the point i; $x_i^k = 0$ otherwise.

In order to the circle C_k assigned to the point i be non-overlapping with other circles being packed, it is necessary that $x_j^\ell = 0$ for $j \in I, l \in K$ such that $d_{ij} < R_k + R_l$. For fixed i,k let $N_{ik} = \{j,l : i \ne j, d_{ij} < R_k + R_l\}$. Then the problem of maximizing the area covered by circles can be stated as follows

$$\max \sum_{i \in I} \sum_{k \in K} W_k x_i^k \tag{1}$$

subject to

$$a_k \le \sum_{i \in I} x_i^k \le b_k, k \in K \tag{2}$$

$$\sum_{k \in K} x_i^k \le 1, \quad i \in I \setminus F, \tag{3}$$

$$R_k x_i^k \le \min_{j \in F} d_{ij}, \quad i \in I, \quad k \in K, \tag{4}$$

$$x_i^k + x_j^\ell \le 1, \ for \ \ i \in I; \ \ k \in K; \ \ (j,\ell) \in N_{ik}, \tag{5}$$

$$x_i^k \in \{0,1\}, \quad i \in I, k \in K \tag{6}$$

Constraints (2) ensure that the number of circles packed is between a_k and b_k; constraints (3) that at most one center is assigned to any grid point; constraints (4) that the point i cannot be a center of the circle C_k if the distance from i to the closest grid point of the boundary is less than R_k; pair-wise constraints (5) guarantee that there is no overlapping between the circles; constraints (6) represent the binary nature of variables. In (4) R_k is compared with the distance from the center to grid points of the boundary, i.e. the boundary is represented by its grid points only. Thus, in general constraints (4) do not guarantee that a circle fits into the container. However, if the point of the boundary closest to is a grid point, then (4) ensure that there are no intersections between circles and the boundary.

In many applied problems nesting circles inside one another is permitted. For example, in [26,27] nesting is considered in the context of packing pipes of different diameters into a shipping container. Nesting is also essential for storing different cylinders one over another in the form of cylindrical towers. To consider nesting circles inside one another, we only need to modify the non-overlapping constraints. In order to the circle C_k assigned to the point i be non-overlapping with other circles being packed (including circles placed inside this circle), it is necessary that $x_j^\ell = 0$ for $j \in I, \ell \in K$, such the $R_k - R_\ell < d_{ij} < R_k + R_\ell$ for $R_k > R_\ell$. Let $\Omega_{ik} = \{j,\ell : i \ne j, R_k - R_\ell < d_{ij} < R_k + R_\ell, R_k > R_\ell\}$. Then the non-overlapping constraints for packing circles with nesting can be stated in the form

$$\boldsymbol{x_i^k + x_j^\ell \le 1, \ for \ \ i \in I; \ \ k \in K; \ \ (j,\ell) \in \Omega_{ik}} \tag{7}$$

Constraints (3) have to be relaxed in case of nesting. The model for considering nesting is shown below

$$\max \sum_{i \in I} \sum_{k \in K} W_k x_i^k \tag{8}$$

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 3 (2018) pp. 1784-1790

subject to

$$a_k \leq \sum_{i \in I} x_i^k \leq b_k, k \in K \tag{9}$$

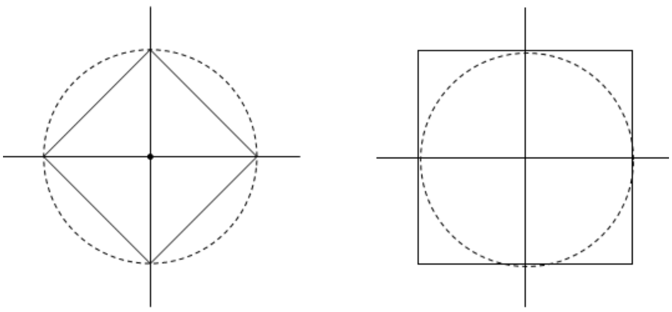
$$R_k x_i^k \leq \min_{j \in F} d_{ij}, \quad i \in I, \quad k \in K, \tag{10}$$

$$x_i^k + x_j^\ell \leq 1, \; for \;\; i \in I; \;\; k \in K; \;\; (j, \ell) \in \Omega_{ik}, \tag{11}$$

$$x_i^k \in \{0,1\}, \;\; i \in I, k \in K \tag{12}$$

Note that all constructions proposed above, remain valid for any norm used to define the circular-like object. In fact, changing the norm affects only the distance d_{ij} used in the definitions of the sets N_{ik}, Ω_{ik} in the non-overlapping constraints (5) and (11). That is, by simple pre-processing we can use the basic model (1) -(6) for packing different geometrical objects of the same shape.

It is important to note that the non-overlapping condition has the form $d_{ij} \geq R_k + R_\ell$ no matter which norm is used. For example, a circular object in the maximum norm $\|z\|_\infty := max_i\{|z_i|\}$is represented by a square, taxicab norm $\|z\|_1 := \sum|z_i|$ yields a rhombus, as we can see in Figure 2. In a similar way we may manage rectangles, ellipses, etc. Using a superposition of norms, we can consider more complex circular objects. For $\|z\| := max_i\{|z_i|, \gamma\sum|z_i|\}$ and suitable $0.5 < \gamma < 1$ we get an octagon, an intersection of a square and rhombus.



(a) Rhombus. **(b)**Square.

Figure 2. Circular-like objects.

METHOD

In this section we present a numerical study. To use objects having a circle-like definition, that is, sets of points equidistant from a reference point (center) we use three different norms (for packing non-congruent circles, rhombuses) to measure the distance between two points that allow us to detect overlaps. A rectangular grid of size Δ determined by the $M = N$ number of equidistant grid points on both the horizontal and vertical edge of the container, respectively. With previous experimentation we determined that evenly distributed points between 30 and 35 were enough to carry out tests.

Furthermore, in [24] they use similar parameters for the grid points with congruent (equal) circles. The test set of 9 instances was used for packing maximal number of equal circular objects into a rectangular container of width 300 and height 300. The values of radii were generated at random with a uniform distribution $U(2,40)$. The size of the container was generated with a uniform distribution $U(253,300)$ for both Length and Width. The models were codified in Python 3.5 with the optimizer (solver) Gurobi 6.5.1 on a laptop computer with 8Gb in RAM, AMD A8-5545M APU processor at 1.7 GHz with Ubuntu/GNU Linux 64-bit operating system.

Numerical Results

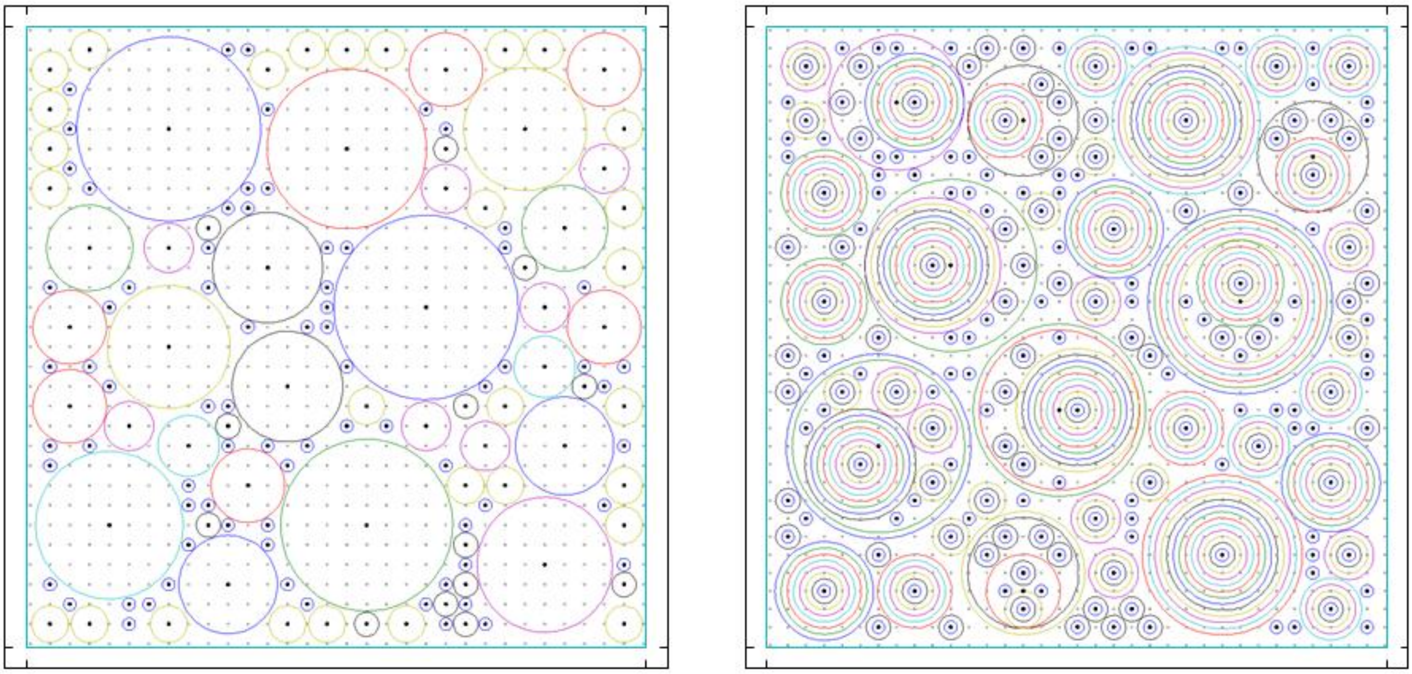
For the development of this work it is proposed to use a general-purpose language that is easy to use and that is current in the academic and practical sphere, as well as an optimizer that incorporates algorithms present in the state of the art. We use Python as a programming language and Gurobi as optimizer. Python is well known in the field of web programming with systems such as Django and CherryPy, it also serves as a platform for sites with a lot of information traffic. But in addition to web programming, extensions are being created for disciplines from other areas, such as linear algebra, data visualization and automated learning. For these applications, Python is the program that allows to easily gather the exchange of information of data and data through routines written in programming languages like Fortran and C.

Results obtained with circles, rhombuses and squares are shown in Tables 1-3. The tables show the instances ordered by occupancy percentage (column labeled as % occupancy), this occupancy percentage is calculated by dividing the total area of the packed objects by the total area of the container; in the case of nesting is convenient to see the number of packed objects inside the container, since the occupancy percentage in some cases may exceed the 100%. We see, for example, that for Table 1 the instances with the highest occupancy percentage were "Circle 5 2a.txt" and the one with the lowest performance was "Circle 4.txt", these results are presented under the model without nesting. Packing for instance "Circle 5 2a.txt" is shown in Figure 3a.

It is important to note that in Tables 1 to 3 the Gap is greater than 100%, but this is related to the instance. According to the typology proposed by [14] our work is classified as Single Large Object Placement Problem which is characterized by the assignment of a weakly heterogeneous assortment of small items to a single large object. For this reason, if the maximum number of items is very large, the problem is impossible to solve. This behavior can be seen in the columns labeled as %Packaged objects and %occupancy, while we have only 19% of packaged objects we can reach the 76% of the total area, so there is no room for more objects, at the same time we can verify this results with the example shown in the Figure 3.

Table 1. Results of Circles ordered by occupancy in container.

Circle	Gap %	Objective function	Total of objects	% Packed objects	% occupancy	Nesting
Circle 5 2a.txt	1E+102	61674.71	752	19	76	0
Circle 5.txt	1E+102	61674.71	1167	12	76	0
Circle 6 2a.txt	1E+102	63569.08	752	17	74	0
Circle 1.txt	1E+102	62278.37	704	18	74	0
Circle 6.txt	1E+102	63569.08	1167	11	74	0
Circle 6 2a.txt	1E+102	63569.08	752	17	74	0
Circle 4 2a.txt	1E+102	45868.68	752	15	72	0
Circle 4.txt	1E+102	45868.68	1167	10	72	0
Circle 4.txt	1E+102	45868.68	1167	10	72	0
Circle 6 tel.txt	222	83745	1167	37	97	1
Circle 5 tel.txt	224	74698.87	1167	42	93	1
Circle 1 tel.txt	142	68370.53	704	49	81	1
Circle 3 tel.txt	1E+102	66827.64	1167	33	77	1
Circle 1 tel.txt	144	64380.04	704	51	76	1



(a) Instance “Circle 5 2a.txt” **(b)** Instance “Circle 6 tel.txt”

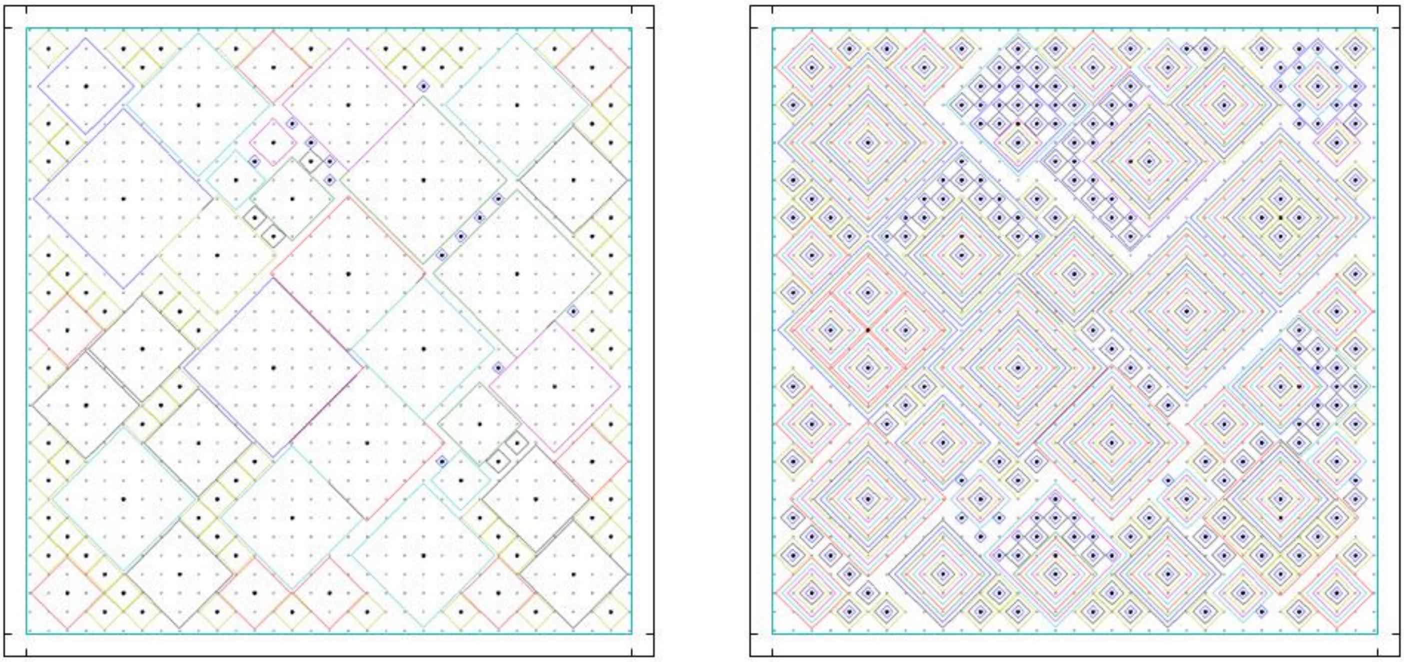
Figure 3. Packing for circles

In Table 2 the instance with the highest occupancy percentage was Rhombus 6.txt and the one with the lowest performance was “Rhombus 2.txt". These results are presented under the model without nesting. For the nesting case the best results were obtained with the instance "Rhombus 1 tel.txt". If other performance measures are observed for the instances, it is observed, for example, that the instance "Rhombus 1 2a.txt" is the one that more objects have accommodated inside the container (see column % of packaged objects of Table 2) and secondly the instances “Rhombus 1.txt" and " Rhombus 2.txt" with a percentage of objects packed of 8% are observed. Packing for instance "Rhombus 6.txt" is shown in Figure 4a.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 3 (2018) pp. 1784-1790

Table 2. Results of Rhombuses ordered by occupation in container.

Rhombus	Gap %	Objective function	Total of objects	% Packed objects	% occupancy	Nesting
Rhombus 6.txt	641	62403.38	1837	7	82	0
Rhombus 1.txt	649	61286.5	1840	8	81	0
Rhombus 1 2a.txt	1E+102	59426.13	1185	14	79	0
Rhombus 3.txt	645	64379.61	1839	7	78	0
Rhombus 5.txt	587	51222.94	1839	6	78	0
Rhombus 5.txt	593	50816.61	1839	7	78	0
Rhombus 4.txt	66	61844.73	1840	7	77	0
Rhombus 2.txt	1E+102	51453.76	1839	8	73	0
Rhombus 1 tel.txt	1E+102	73462.13	1840	34	97	1
Rhombus 3 tel.txt	1E+102	74807.53	1839	35	91	1
Rhombus 3 tel.txt	1E+102	74807.53	1839	35	91	1
Rhombus 2 tel.txt	1E+102	63708.7	1839	33	90	1



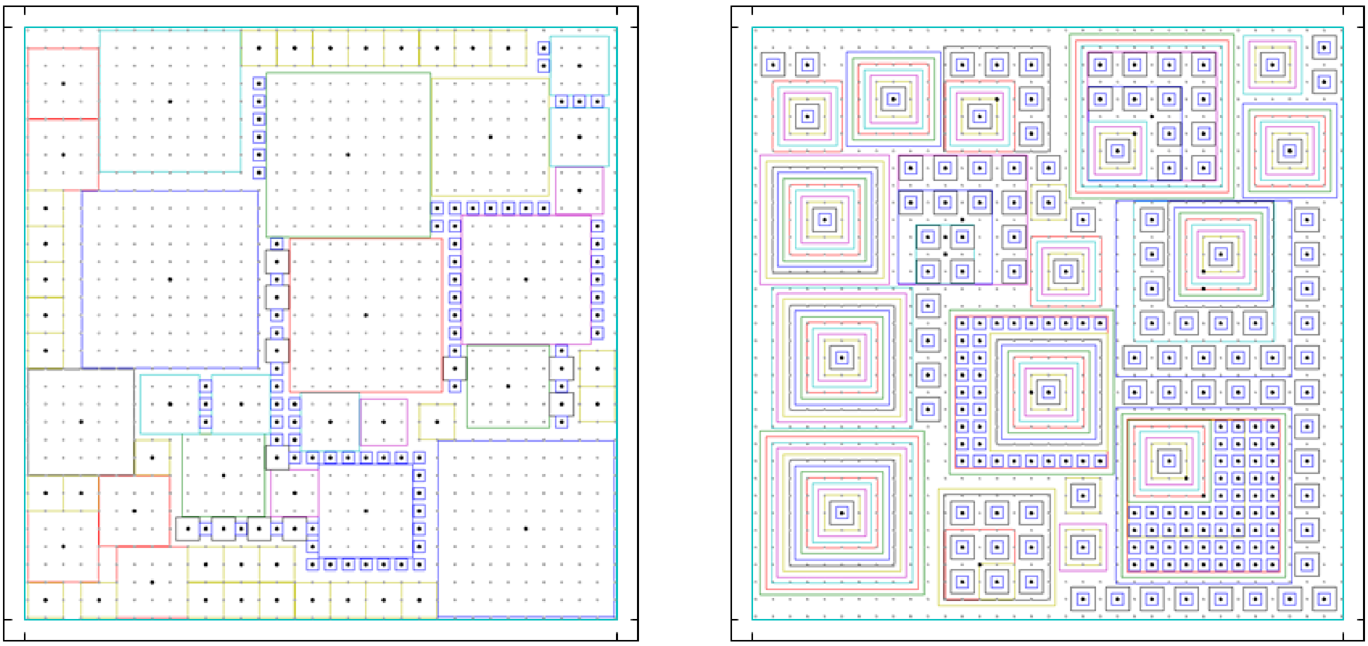
(a) Instance "Rhombus 6.txt" **(b)** Instance "Rhombus 1 tel.txt"

Figure 4. Packing for rhombuses

In Table 3 the instance with the highest occupancy percentage was "Square 12a.txt" and the lowest performance was "Square 3.txt", these results are presented under the model without nesting. For the case of nesting the best results were obtained with the instance "Square 3 tel.txt". If other performance measures are observed for the instances, it is observed that the instance "Square 1 2a.txt" is the one that more objects have accommodated inside the container (see column % of objects packed in the Table 6) and secondly the instance "Square 2 3a.txt" is observed with a percentage of packaged objects of 17%. Packing for instance "Square 1 2a.txt" is shown in Figure 5a.

Table 3. Results of Squares ordered by occupation in container

Squares	Gap %	Objective function	Total of objects	% Packed objects	% occupancy	Nesting
Square 1 2a.txt	708	71726.93	600	24	87	0
Square 2.txt	715	66326.4	912	10	85	0
Square 6.txt	1E+102	68235.83	916	14	85	0
Square 1.txt	1E+102	70178.39	915	14	85	0
Square 4.txt	701	70759.44	914	6	84	0
Square 5.txt	701	66450.25	916	6	84	0
Square 2 3a.txt	647	64821.12	590	17	83	0
Square 4 2a.txt	714	69607.78	601	8	82	0
Square 3 3a.txt	1E+102	59420.33	579	14	75	0
Square 3 2a.txt	1E+102	59420.33	601	14	75	0
Square 3.txt	1E+102	59420.33	916	9	75	0
Square 3 tel.txt	176	96096.8	916	43	121	1
Square 2 tel.txt	164	83637.12	912	45	107	1
Square 4 tel.txt	163	88779.56	914	40	105	1
Square 1 tel.txt	174	85630.81	915	43	104	1



(a) Instance "Square 1 2a.txt" **(b)** Instance "Square 3 tel.txt"

Figure 5. Packing for squares

CONCLUSIONS

This article proposes a linear model using a discretization of the space of solutions through a dotted-board(grid) covering the container. With this proposal it is tried to avoid the non-linearity and the non-convexity of these problems due to the overlap constraints. Another advantage of this model is that the binary decision variables only depend of the different item type this is suitable for problems with few piece types. A general-purpose programming language is used in conjunction with Gurobi. It was demonstrated that by simply changing the definition of the distance (preprocessing) it is possible to use the same basic models for packing different circular-like objects such as circles, rhombuses and squares, etc. To cope with large dimension of arising problems it is interesting to study the use of corresponding heuristics [4,28–30].

FUTURE WORK

A future research is to study the use of evolutionary computation to cope with large dimension of arising problems. Another area for future study is to use a hybrid solution strategy combining integer formulations and heuristics as good initial solutions.

REFERENCES

[1] Birgin, E.G.; Gentil, J.M. New and improved results for packing identical unitary radius circles within triangles, rectangles and strips. Computers & Operations Research **2010**, 37, 1318 – 1327.

[2] Castillo, I.; Kampas, F.J.; Pintér, J.D. Solving circle packing problems by global optimization: Numerical results and industrial applications. European Journal of Operational Research **2008**, 191, 786–802.

[3] Hifi, M.; M'Hallah, R. A literature review on circle and sphere packing problems: Models and methodologies. Advances in Operations Research **2009**, 2009.

[4] Martínez, L.; Amaya, C. A vehicle routing problem with multi-trips and time windows for circular items. Journal of the Operational Research Society **2013**, 64, 1630–1643.

[5] Gracia, M.D.; González-Ramírez, R.G.; Mar-Ortiz, J. The impact of lanes segmentation and booking levels on a container terminal gate congestion. Flexible Services and Manufacturing Journal **2017**, 29, 403–432.

[6] Casas-Ramírez, M.S.; Camacho-Vallejo, J.F.; Martínez-Salazar, I.A. Approximating solutions to a bilevel capacitated facility location problem with customer 's patronization toward a list of preferences. Applied Mathematics and Computation **2018**, 319, 369 – 386. Recent Advances in Computing.

[7] Dubey, R.; Bag, S.; Ali, S.S. Green supply chain practices and its impact on organizational performance:an insight from Indian rubber industry. International Journal of Logistics Systems and Management **2014**, 19, 20–42.

[8] Huang, W.; Xu, R. Two personification strategies for solving circles packing problem. Science in China Series E: Technological Sciences **1999**, 42, 595–602.

[9] Lenstra, J.; Kan, A.; Econometric institute, E.u.R. Complexity of packing, covering and partitioning problems.; Econometric Institute, 1979.

[10] Fasano, G. A global optimization point of view to handle non-standard object packing problems. Journal of Global Optimization **2012**, 55, 279–299.

[11] Fasano, G. A Modeling-Based Approach for Non-standard Packing Problems. In Optimized Packings with Applications; Fasano, G.; Pintér, J.D., Eds.; Springer International Publishing, 2015; pp. 67–85.

[12] Szabó, P.G.; Markót, M.C.; Csendes, T.; Specht, E.; Casado, L.G.; García, I. New approaches to circle packing in a square. With program codes.; Springer, 2007.

[13] Tarnai, T.; Miyazaki, K. Circle Packings and the Sacred Lotus. Leonardo **2003**, 36, 145–150.

[14] Wäscher, G.; Haußner, H.; Schumann, H. An improved typology of cutting and packing problems. European Journal of Operational Research **2007**, 183, 1109–1130.

[15] Al-Mudahka, I.; Hifi, M.; M'Hallah, R. Packing circles in the smallest circle: an adaptive hybrid algorithm. Journal of the Operational Research Society **2011**, 62, 1917–1930.

[16] Liang, G.; Lu, L.; Chen, Z.; Yang, C. Poisson disk sampling through disk packing. Computational Visual Media **2015**, 1, 17–26.

[17] Birgin, E.G. Applications + Practical Conceptualization + Mathematics = fruitful Innovation. In Applications + Practical Conceptualization + Mathematics = fruitful Innovation: Proceedings of the Forum of Mathematics for Industry 2014; Anderssen, R.S.; Broadbridge, P.; Fukumoto, Y.; Kajiwara, K.; Takagi, T.; Verbitskiy, E.; Wakayama, M., Eds.; Springer Japan: Tokyo, 2016; Vol. 11, pp. 31–39.

[18] Huang, W.; Ye, T. Quasi-physical global optimization method for solving the equal circle packing problem. Science China Information Sciences **2011**, 54, 1333–1339.

[19] Miyazawa, F.K.; Pedrosa, L.L.C.; Schouery, R.C.S.; Sviridenko, M.; Wakabayashi, Y. Polynomial-time approximation schemes for circle packing problems. In Algorithms - ESA 2014: 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings; Springer Berlin Heidelberg, 2014; pp. 713–724.

[20] He, Y.; Wu, Y. Packing non-identical circles within a rectangle with open length. Journal of Global Optimization **2013**, 56, 1187–1215.

[21] Beasley, J.E. An exact two-dimensional non-guillotine cutting tree search procedure. Operations Research **1985**, 33, 49–64.

[22] Galiev, S.I.; Lisafina, M.S. Linear models for the approximate solution of the problem of packing equal circles into a given domain. European Journal of Operational Research **2013**, 230, 505–514.

[23] Toledo, F.M.; Carravilla, M.A.; Ribeiro, C.; Oliveira, J.F.; Gomes, A.M. The Dotted-Board Model: A new MIP model for nesting irregular shapes. International Journal of Production Economics **2013**, 145, 478–487.

[24] Litvinchev, I.; Ozuna Espinosa, E.L. Integer Programming Formulations for Approximate Packing Circles in a Rectangular Container. Mathematical Problems in Engineering **2014**, 2014, 1–6.

[25] Litvinchev, I.; Infante, L.; Ozuna Espinosa, E.L. Packing circular-like objects in a rectangular container. Journal of Computer and Systems Sciences International **2015**, 54, 259–267.

[26] George, J.A.; George, J.M.; Lamar, B.W. Packing different-sized circles into a rectangular container. European Journal of Operational Research **1995**, 84, 693–712.

[27] George, J.A. Multiple Container Packing: A Case Study of Pipe Packing. Journal of the Operational Research Society **1996**, 47, 1098–1109.

[28] He, Z.; Liu, J. A sharp estimate for the hexagonal circle packing constants. Geometriae Dedicata **2010**, 146, 193–210.

[29] López, C.; Beasley, J. A heuristic for the circle packing problem with a variety of containers. European Journal of Operational Research **2011**, 214, 512–525.

[30] Flores, J.J.; Martínez, J.; Calderón, F. Evolutionary computation solutions to the circle packing problem. Soft Computing **2015**, 20, 1521–1535.