ISSN 1984-820X
Anais do CNMAC
v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010
XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

A heuristic for the nonidentical circle packing problem

Washington A. Oliveira, **Antônio C. Moretti**,
Departamento de Matemática Aplicada, IMECC, Unicamp,
13089-970, Campinas, SP
E-mail: ra019338@ime.unicamp.br, moretti@ime.unicamp.br,

Luiz L. Salles-Neto
Departamento de Ciência e Tecnologia
Campus São José dos Campos - Unifesp
12231-280, São José dos Campos, SP
E-mail: luiz.leduino@unifesp.br.

Abstract: *In this work we studied the NP-hard combinatorial nonidentical circle packing problem. It consists of packing a set of circles of different radius dimensions into a circular region taking in account the design and technological considerations. Our problem is a little bit different, we want to place a circular weighted objects inside a circular container, such that we minimize, both, imbalance of mass and the radius of the container. This problem has a practical importance in industrial applications (such as the design of satellites) and a theoretical interest. We present a heuristic to solve large instances with good outcome and computational time. We compare the heuristic with some of the existing best-known algorithms in literature.*

Palavras-chave: *Circle packing problem, Heuristics, Packing problem.*

1 Introducion

The Nonidentical Circle Packing Problem (NCPP) consists in packing n nonidentical circles without overlap inside the smallest containing circle C, where each circle i is characterized by its radius r_i. The goal is to search for the best packing of the n circles inside C, where the best packing minimizes the space not used.

One variant of NCPP, which we call the Balanced Nonidentical Circle Packing Problem (BNCPP) concerns the physical placement of instruments or pieces of equipment in a spacecraft or satellite. As these objects have mass, the system is also subject to additional constraints. We have studied this problem with two objectives to minimize: (1) the space occupied by a given collection of objects; and (2) the non-equilibrium, i.e. imbalance, of the system. The work is organized as follows. In Section 2 we present a detailed description of the problem. Section 3 describes our heuristic. Section 4 presents the computational results showing that our algorithm is promising compared to the best known algorithms for this problem.

2 Description and problem formulation

Developments in the aerospace industry, and specifically in satellite design, have motivated much work in the past decade on the internal configuration of sophisticated collections of equipment. The BNCPP has significant implications for the cost and performance of devices such as satellites and spacecraft. It concerns the two dimensional physical placement of a collection of objects (instruments or other pieces of equipment) within a spacecraft/satellite cabinet, or container. We assume that each object is circular, with a radius and a mass. The container region is

also assumed to be circular. We present a mathematical model similar to that presented in the literature [5] and [2]. Assume there are n objects to be laid out. The radius of the objects are $r_1, r_2, \ldots, r_n$, and their masses are $m_1, m_2, \ldots, m_n$. The initial radius of the container is r_0. Let the $2n$-dimension vector $z = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$ denote a layout, where (x_i, y_i) is the center of object i in the cartesian coordinate system. The (x, y) is the center of the container C.

BNCPP is equivalent to finding the coordinates (x_i, y_i) of every circle i, $i \in I = \{1, \ldots, n\}$, the radius r and coordinates (x, y) of C, such that there is not pair of circles $(i, j) \in I \times I$ and $i < j$ overlap. Formally, the problem can be stated as finding the optimal level of the decision variables r, (x, y) and (x_i, y_i), $i \in I$, which can be formulated as a bi-criteria programming problem as follows.

(**BNCPP**)

$$\begin{array}{ll} \text{Minimize} & f(z) = (f_1(z), f_2(z)) \\ \text{S.t.} & \\ & z \in X \end{array}$$

where

- $f_1(z) = r \equiv$ The envelopment objective attempts to minimize the radius of the circle containing all of the objects,
- $f_2(z) = \sqrt{\left(\sum_{i=1}^{n} m_i(x_i - x)\right)^2 + \left(\sum_{i=1}^{n} m_i(y_i - y)\right)^2} \equiv$ The imbalance objective attempts to minimize the non-equilibrium the center of mass for a collection of objects around the central point,
- $X = \{z \in \mathbb{R}^{2n} \mid (x_i - x)^2 + (y_i - y)^2 \leq (r - r_i)^2,\ i \in I;$
  $(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2,\ (i, j) \in I \times I,\ i < j;$
  $0 \leq r \leq r_0\}.$

The first set of constraints in X reinforce the complete packing of every circle within C. The second set guarantes the no overlapping of any pair of distinct circles. Finally, the last constraint says that the radius of the containing circle must be nonnegative, the elimination of this constraint from the model makes BNCPP unbounded in $f_1(z)$. For all practical purposes, BNCPP can be solved by setting either $(x, y) = (0, 0)$ or any other pair of coordinates $(x_i, y_i) = (0, 0)$.

3 The algorithm

In this section we are going to present the main ideas of the algorithm. Every time we run the procedure *CONSTRUCTION* we produce a solution that is characterized by a container C and the n circles inserted on it.

In order to minimize $f_2(z)$, during execution of the algorithm, for each circle k chosen to enter in the container C a new center of mass is defined as

$$(x^k, y^k) = \left(\frac{\sum_{i=1}^{k} m_i x_i}{\sum_{i=1}^{k} m_i}, \frac{\sum_{i=1}^{k} m_i y_i}{\sum_{i=1}^{k} m_i} \right).$$

This means that the end of the algorithm we have

ISSN 1984-820X
Anais do CNMAC
v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010
XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

$$(x,y) = (x^n, y^n) = \left(\frac{\sum_{i=1}^{n} m_i x_i}{\sum_{i=1}^{n} m_i}, \frac{\sum_{i=1}^{n} m_i y_i}{\sum_{i=1}^{n} m_i} \right),$$

simply replace this value in the function $f_2(z)$ to obtain $f_2(z) = 0$.

For the layout has not overlap we define

$$c^p_{i,\ell} = \sqrt{(x_i - x_\ell)^2 + (y_i - y_\ell)^2} - (r_i + r_\ell) \quad \text{and} \quad c^p_{i,k} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2} - (r_i + r_k),$$

if the circle i is chosen to enter the container in *contact active* p (this is a pair of coordinates or a point, which will be defined later) and *attraction* (a procedure which will be defined later) in circles ℓ and k. The attraction (the circle i slope in the circles ℓ and k) is made until that $0 \leq c^p_{i,\ell} < 10^{-5}$ and $0 \leq c^p_{i,k} < 10^{-5}$. Thus at the end of the algorithm there is no overlap.

After the construction of the layout, then we try to minimize the value $f_1(z) = r$ current, which can be written as

$$f_1(z) = r = \max_{1 \leq i \leq n} \left(r_i + \sqrt{(x_i - x)^2 + (y_i - y)^2} \right).$$

The pseudo-code below describes the steps of the procedure *CONSTRUCTION*.

Algorithm 1 *CONSTRUCTION HEURISTIC*

Given a sequence of inclusion;

Run the ***Procedure Initial*** *and get the initial design;*

Step 1. While there is circles to include in the container C Do:

- *Step 1.1. Run the Procedure* ***Contacts***
- *Step 1.2. Choose the next circle i to enter in the container*
- *Step 1.3. Run the Procedure* ***Side Allowed***
- *Step 1.4. Run the Procedure* ***Attraction***
- *Step 1.5. Run the Procedure* ***Possible Inclusion***

Step 2. While it is possible to minimize $f_1(z)$ Do

- *Step 2.1. Change a selected circle of contact active and update the value of $f_1(z)$*

Step 3. If $f_1(z) > r_0$ the solution is discarded.

Below, we explain in detail each procedure and steps performed in the main procedure.

3.1 Initial Procedure

This procedure builds an initial design with 4 circles. The first is allocated in coordinates $(0, 0)$ and the second one is allocated in coordinate $((r_1 + r_2)\cos(\theta), (r_1 + r_2)\,\text{sen}(\theta))$, where θ is an angle chosen randomly. Then the coordinate $p = (r_1 \cos(\theta), r_1\,\text{sen}(\theta))$ is the first contact active and the procedure ***Attraction*** allocates the other two circles, limiting the values of $c^p_{3,1}$, $c^p_{3,2}$ and $c^p_{4,1}$, $c^p_{4,2}$.

ISSN 1984-820X
Anais do CNMAC
v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010
XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

3.2 Step 1.1

The procedure called ***Contacts*** stored in a set of coordinates $p^i_{\ell,k} = (p^i_1, p^i_2)$ all positions where two circles were allocated and are in contact in accordance with the procedure ***Attraction***. In Figure 1 the contacts are represented by the letter p. It sets each coordinate $p^i_{\ell,k}$ of contact as active, after the inclusion of a new circle j between the circle ℓ and k the contact $p^j_{\ell,k}$ becomes inactive.

3.3 Step 1.2

The choice of the next circle to enter the container is chosen randomly. Tests show that the best solutions occur when the sequence of inclusion is from the largest to the smallest circle. In our tests the inclusions are made as follows. If $v = (r_{1_k}, r_{2_k}, \ldots, r_{n_k})$ is the vector where each entry represents a circle i of radius r_{i_k}, with $|v| > 10$, the sequence k is defined such that the first $\left\lceil \frac{|v|}{2} \right\rceil$ circles are a random choice among the circles i such that $r_{i_k} \geq \frac{\sum_{i=1}^{n} r_{i_k}}{n}$ and the remaining is a random choice among the circles i such that $r_{i_k} < \frac{\sum_{i=1}^{n} r_{i_k}}{n}$. If $|v| \leq 10$, choose any inclusion at random.

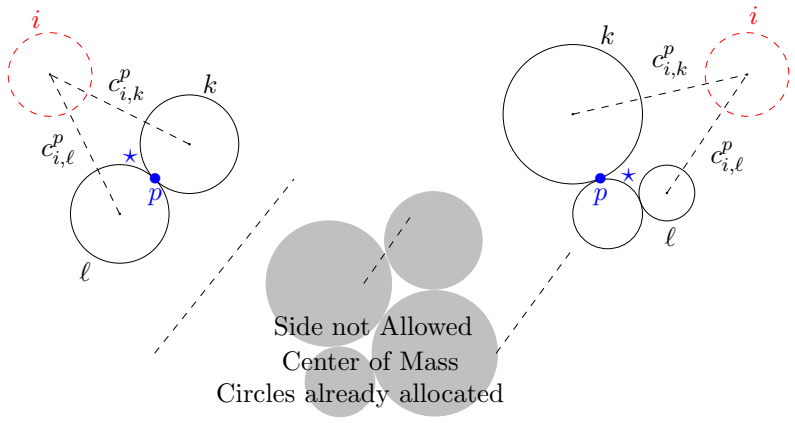

Figure 1: Example of a partial solution

3.4 Step 1.3

After selecting the circle i that will be inserted between the circles ℓ and k (that is, contact active $p^i_{\ell,k}$), the procedure called ***Side Allowed*** determines which side of the circle will be included. There are two choices, one side pointing to the current center of the mass and the other in the opposite direction is always chosen to avoid overlapping. Figure 1 shows a side not allowed to include a new circle in the container, because this side has already circles allocated.

3.5 Step 1.4

Once decided that circle i was selected to be part of the container between the circles ℓ and k the procedure called ***Attraction*** attracts the 3 circles, i, ℓ and k, checking the overlap among the circles until $0 \leq c^p_{i,\ell} < 10^{-5}$ and $0 \leq c^p_{i,k} < 10^{-5}$, where $p = p^i_{\ell,k}$. We can see in Figure 1 the representation of the attraction of a circle i that will be attracted to touch the circles ℓ and k.

ISSN 1984-820X
Anais do CNMAC
v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010
XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

3.6 Step 1.5

The procedure called ***Possible Inclusion*** checks if it is possible to fit a circle among 3 or 4 circles already included in the container. Figure 1 represents two configurations of a free space where it is possible to insert a new circle. The inclusion position is represented by a star, which is the centroid of the 3 circles on the left or the 4 circles on the right in the figure. Among the circles not yet inserted, the most probable inclusion will occur with the circle that has small radius.

3.7 Step 2

In this step the algorithm try to improve the value of the function $f_1(z)$. Step 2 is initialized with the current value of $f_1(z)$ and there are still contacts active. The idea is to change the position of the circle j that determines the value of $f_1(z)$ in the function $\max_{1\leq j\leq n}\left(r_j+\sqrt{(x_j-x)^2+(y_j-y)^2}\right)$ and to check if this value decreases.

3.8 Step 3

This step only checks if the final solution is feasible.

4 Computational Results

In order to test the procedure against the best known algorithms for the BNCCP in literature, we performed numerical experiments. The data used in Examples 1 and 2 were taken from the literature [3]. The results are shown in Table 1. In Table 2 we compare our results with the results presented in the literature [5] and [7], (Comp + SA) and (Comp + PSO) are two heuristcs in the literature [5]. The tables show the best value of f_1 (mm), the imbalance f_2 $(g.mm)$ and running time t (seg). All experiments were carried out on an Pentium 3.40 GHz CPU, 2GB RAM, Win XP and a MATLAB code.

Example 1:

Circle	1	2	3	4	5	6	7
Radius (mm)	10	11	12	11.5	9.5	8.5	10.5
Mass (g)	100	121	144	132	90.25	72.25	110.25

Example 2:

Circle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Radius (mm)	106	112	98	105	93	103	82	93	117	81	89	92	109	104	115	110	114	89	82
Mass (g)	11	12	9	11	8	10	6	8	13	6	7	8	11	10	13	12	12	7	6
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
120	108	86	93	100	102	106	111	107	109	91	111	91	101	91	108	114	118	85	87	98
14	11	7	8	10	10	11	12	11	11	8	12	8	10	8	11	12	13	7	7	9

Algorithm	Example 1			Example 2		
	f_1	f_2	t	f_1	f_2	t
Literature[6]	32.837	0.102000	1735	870.331	0.006000	1358
Literature[1]	32.662	0.029000	1002	874.830	11.39500	1656
Literature[4]	31.985	0.018200	1002	843.940	0.003895	2523
Literature[3]	31.924	0.000014	427	769.819	0.000325	1724
Our algorithm	31.934	**0**	**2**	**751.205**	**0**	**9**
The inclusion sequence	7 2 3 5 4 1 6			37 15 31 17 2 20 16 27 9 36 21 14 28 26 4 13 1 35 6 29 24 33 12 8 23 25 32 3 5 40 7 19 11 34 38 39 18 22 10 30		

Table 1: Results of the Example 1 and 2

ISSN 1984-820X
Anais do CNMAC
v.3
PUBLICAÇÃO DA SBMAC
CNMAC2010
XXXIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL

Size	Zhou et al. [7]			(Comp + SA) [5]			(Comp + PSO) [5]			Our algorithm		
	f_1	f_2	t	f_1	f_2	t	f_1	f_2	t	f_1	f_2	t
10	61.32	0.0002	18401	60.96	0	3237	59.93	0	2898	61.21	0	**3**
15	76.58	0.0002	31816	68.77	0	8320	67.65	0	8659	69.32	0	**3**
20	89.15	0.0002	47496	83.09	0	18431	83.06	0	20035	85.08	0	**5**
25	106.31	0.0002	63201	83.97	0	34032	84.24	0	36815	84.57	0	**7**
30	136.88	0.0004	87985	99.58	0	54565	99.89	0	62360	101.23	0	**6**
35	148.39	0.0004	112144	102.86	0	76760	102.71	0	86537	104.99	0	**8**
40	165.79	0.0004	138030	115.15	0	128112	115.58	0	122390	119.11	0	**10**
45	172.69	0.0004	202446	120.63	0	167484	119.67	0	153006	120.85	0	**11**
50	189.89	0.0005	192479	125.82	0	198071	126.19	0	199050	**125.62**	0	**12**
55	200.82	0.0003	236835	138.22	0	198071	138.89	0	244171	139.17	0	**14**

Table 2: Results of the Literature [5] and [7]

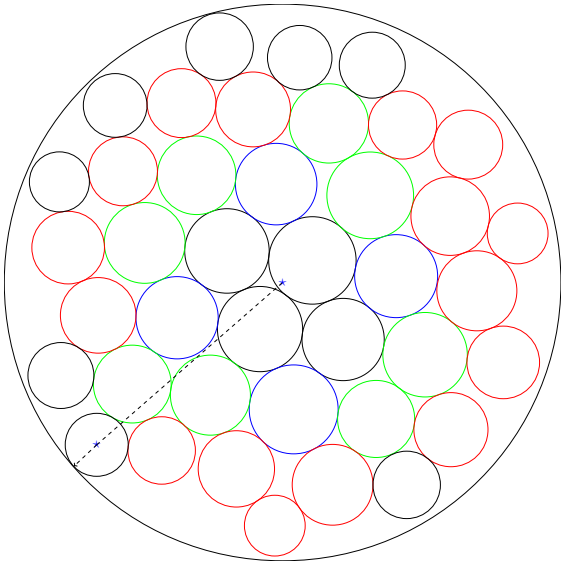

Figure 2: Solution of the Example 2

Each instance was executed 10 times, that is , 10 different including sequences of circles. In the tables, the parameter t represents the total time of the 10 tests. The tests show that our idea is promising. Table 1 and 2 show that the function f_1 was improved in only two instances, but always have the running time significantly lower and the values of f_1 does not exceed 3.1% of the best results. The procedure also entirely eliminate unbalance, i.e., function f_2 was always zero. In our procedure the sequence of circle inclusion is very important, it determines the efficiency of the container packing. Table 1 shows the sequence of circle inclusion to the Examples 1 and 2. And Figure 2 shows the solution for the Example 2.

Conclusion

In this work we studied one variant of the NP-hard combinatorial nonidentical circle packing problem. It concerns the placement of instruments or pieces of equipment in a spacecraft or satellite. We present a promising heuristic algorithm. The computational tests show that our

procedure outperformed the known best ones in terms of running time. In terms of quality of solution we believe that our algorithm is very competitive, since we obtained a low computational cost in several tests that was done.

References

[1] T. Fei, T. Hongfei, A modified genetic algorithm and its application to layout optimization, *Journal of Software (in Chinese)*, Vol. 10, No 10, pp. 1096–1102, 1999.

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

[3] L. Kaiyou, Constrained layout optimization based on adaptive particle swarm optimizer, Springer-Verlag Berlin Heidelberg, pp. 434–442, 2009.

[4] L. Ning, L. Fei, S. Debao, A study on the particle swarm optimization with mutation operator constrained layout optimization, *Chinese Journal of Computers (in Chinese)*, Vol. 27, No 7, pp. 8897–9039, 2004.

[5] R. Xiao, Y.C. Xu, M. Amos, Two hybrid compaction algorithms for the layout optimization problem, *BioSystems*, Vol. 90, pp. 560–567, 2007.

[6] H. F. Teng, S. Shoulin, G. Wenhai, Layout optimization for the dishes installed on rotating table, *Science in China (Series A)*, Vol. 37, No 10, pp. 1272–1280, 1994.

[7] C. Zhou, L. Gao, H. Gao, Particle swarm optimization based algorithm for constrained layout optimization, *Contr. Decis.*, Vol. 20, No 1, pp. 36–40, 2005.