

Circle Covering Representation for Nesting problems with continuous rotations [★]

Pedro Rocha ^{*} Rui Rodrigues ^{*} A. Miguel Gomes ^{*}
Franklina M.B. Toledo ^{**} Marina Andretta ^{**}

^{*} *INESC-TEC, Faculdade de Engenharia, Universidade do Porto*
(e-mail: {pro10015, rui.rodrigues, agomes}@fe.up.pt)

^{**} *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo* (e-mail: {fran, andretta}@icmc.usp.br)

Abstract

This paper analyses distinct methods to represent a polygon through circle covering, which satisfy specific requirements, that impact primarily the feasibility and the quality of the layout of final solution. The trade-off between the quality of the polygonal representation and its derived number of circles is also discussed, showing the impact on the resolution of the problem, in terms of computational efficiency. The approach used to tackle the Nesting problem in strip packing uses a Non-Linear Programming model. Addressing these problems allows to tackle real world problems with continuous rotations.

Keywords: Circle Covering, Irregular Piece Placement Problem, Geometric Representations, Non-Linear Programming Models.

1. INTRODUCTION

The Nesting problem arises on many industries where the main objective is to place a set of pieces in the most efficient configuration, in order to minimize waste. It is also known as an Irregular Piece Placement problem (IPP) and it is defined as a positioning problem of irregular pieces into a region without overlap, which can be configured as a closed container, or it may have some infinite length in some of its dimensions. The characteristics of this problem make it very complex, and its difficulty grows exponentially with the increase in the number of pieces and their geometric complexity. Depending on the industry, technological constraints or specific characteristics define the shape of the pieces to be placed and the restrictions on their orientation. In furniture and textile industries the orientations are usually restricted to a discrete set, while on other industries that deal with leather or metal sheet cutting, any orientations are accepted. Selecting a proper geometric representation might reduce or increase the difficulty of the solution approach, since each one has specific inherent limitations, such as admissible orientations. A good geometric representation allows reducing the complexity of the geometrical component of the problem. The geometrical representation discussed in this paper is the Circle Covering (CC) representation, which allows for

simple overlap computation and continuous rotations, due to the simplicity of operations with circles. The pieces are represented through coverings with sets of overlapping circles, aiming for the most accurate representation with the least amount of circles. The movement and adjustment of the pieces into a feasible layout is done with the assistance of a Non-Linear Programming (NLP) model and a specific solver.

The remaining of this section describes the motivation for this work, its main objectives and its expected contributions. The second section is focused on the literature review, considering brief mentions of alternative geometrical representations, and some variations of problems related to CC. The third section present the details about our approach to CC specifically adapted to the characteristics of the Nesting problem, while the fourth section discusses a generic NLP model developed for Nesting. The fifth section presents the computational tests and results, and finally, the sixth section concludes the article, with the summary of the developed work, conclusions and future work.

1.1 Motivation

Many approaches to the Nesting problem limit themselves to fixed or discrete rotations, due to the difficulty imposed by continuous rotations. To be able to tackle problems with continuous rotations one needs to use an adequate geometric representation, like the CC representation. This representation naturally leads to the use of NLP models to solve this problem. The aim of this work is to develop an approach to the Nesting problem with continuous rotations based on the CC representation and NLP models. This approach allows more efficient solutions that have a relevant impact on many industrial applications where this problem arises.

^{*} This research was partially supported by Fundação para a Ciência e Tecnologia (FCT) Project PTDC/SEN-TRA/121715/2010 (StableCargo) and Project "NORTE-07-0124-FEDER-000057". Project "NORTE-07-0124-FEDER-000057" is financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

1.2 Contribution

The main contribution of this work is an approach based on CC representation together with a NLP model to tackle Nesting problems with continuous rotations. We analyze the impact that different types of covering and approximation quality have on the compaction of a layout using a NLP model. The geometric aspects of the CC representation are based on the previous work (Rocha et al., 2013). The different types of covering allow dealing with problems with different characteristics, while the controllable approximation quality enables controlling the computational. Both of these aspects influence the layout compaction quality. Using the NLP model with pieces represented by a CC enables the generation of compact layouts without any human intervention, only requiring the instance data sets and the configuration parameters. A full automatic process with an efficient approach to determine piece placement positions allows reducing the setup time, and improving efficiency in industrial applications. This NLP model formulation is most useful in industrial applications where the layout is a strip, and continuous rotations are supported. However, with other formulations, the different coverings can be used to solve other types of Nesting problems. The increase in the quality of representation of the pieces, maintaining the same number of circles, increases the computational efficiency, thus enabling problems with greater complexity to be tackled.

2. LITERATURE REVIEW

The specific characteristics of the Nesting problem, in terms of orientation, piece placement and quality of representation require an adequate geometrical representation. Several representations are discussed in (Bennell and Oliveira, 2008), such as grid, where the placement region and pieces are discretized into blocks and only orthogonal rotations are supported, and polygonal representations, where the pieces are represented through polygons, with discrete or continuous rotations. Regarding polygonal representation, several techniques exist to reduce the computational cost required to compute overlaps, such as No-Fit Polygons (NFP), which transform the comparison between two polygons into a comparison between a point and a polygon, with the advantage of being much faster, although only for discrete orientations. Due to most approaches dealing with fixed or discrete rotations, the usual geometric representation is based on the use of NFPs. Stoyan et al. (2004) presents the representation through Phi-Functions, where pieces are represented through unions of primary shapes, composed by line and arc segments, circles and other basic primitives. Phi-Functions allow to compute the relative distance between two pieces, while supporting free-rotations. The approximation of objects by circles are commonly found among many distinct areas. Huang and Tseng (2005) presents cases for wireless sensor networks, art gallery problems and robotic systems coverage, among others, where overlap between circles is allowed. Another variation can be found in Circle Packing problems, that consists in packing circles into a container, without overlap. Birgin and Sobral (2008) tackles the packing problem that aims to find the smallest container for a specific set of circles or spheres. While these

problems are not directly related to CC, their end result can be considered a representation of a piece by circles. Since all of them have some important similarities, they are presented here. Regarding applications of CC, our previous work (Rocha et al., 2013) introduces a method to generate a CC representation with controllable approximation error, obtained from a topological skeleton, the Medial Axis (MA) (Edwards, 2010). In it, we compare our approach to (Zhang and Zhang, 2009), which approximates a piece by a set of circles, with three distinct algorithms: one that approximates the piece outline, other that completely covers a piece, and finally, another that covers a piece but keeps a small gap between the circles to try to reduce their number. Imamichi (2009) describes several packing problems, among them circle/sphere packing, while also considering different containers, such as rectangles, circles, etc. It discusses applications of NLP models to solve irregular strip packing problems, including optimizations, such as spatial partition schemes to improve performance.

3. CIRCLE COVERING REPRESENTATION

The CC representation algorithm consists in a method that reduces the possible placement points for the circle centers inside a polygon into a small set, represented by a topological skeleton, the Medial Axis (MA). This structure consists of the collection of points where the circles of maximum radius can be placed without exceeding the outline of the piece, but with two or more contact points on the outline. By managing the distance that the circles can exceed the outline, or the distance of the gaps that other circles can penetrate the piece, can one can control the approximation error and the number of circles generated. Finding the right balance is critical to solve Nesting problems with continuous rotations.

3.1 Previous work

Our approach to compute the CC is based on our previous work, presented in (Rocha et al., 2013). It consists in creating a MA skeleton of each piece, and using an iterative heuristic to place unequal circles with their center on the MA of each piece, and with a radius equal to the distance to the nearest edge plus the value defined by the protrusion distance allowed (maximum distance from outside the piece outline that the circles can cover). Depending on the types of covering desired, the values for the protrusion distance and penetration depth (maximum distance from the outline that points are uncovered by circles) change, being their difference defined as the approximation threshold. The CC can produce three different types of covering: Complete Circle Covering (*CCC*) (Fig.1(a)), where the set of circles completely covers the piece, Inner Circle Covering (*ICC*) (Fig.1(b)), where the set of circles is completely contained inside the piece, and Partial Circle Covering (*PCC*) (Fig.1(c)) where the set of circles does not completely contain the piece, nor is completely contained by it. An analysis was done in (Rocha et al., 2013) to expose the trade-off between the quality of the approximation and the number of circles produced, using different values for the threshold, also with several pieces. The variation of the threshold value can produce distinct levels of approximation quality, such as low and high resolution (*LR* and *HR*,

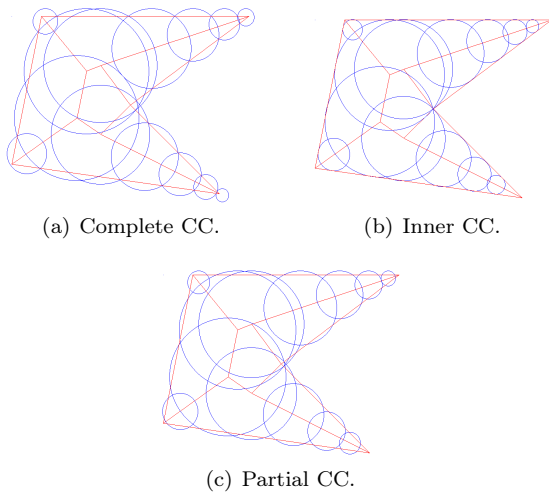


Figure 1. Three types of CC representation.

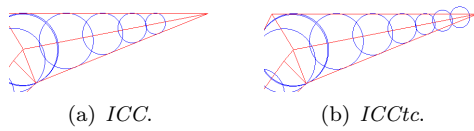


Figure 2. Correction of covering for acute angles for *ICC*.

respectively), with different numbers of circles. *LR* coverings have significantly less circles than the *HR* covering at the expense of a lower quality of representation. While having the representation with highest possible quality is desirable, supporting multiple CC levels of quality enables the computational cost to be adjusted as required, and at the same time also considering the quality of the solutions generated. These basic types of covering are useful for solving the Nesting problem with continuous rotations while considering different characteristics.

3.2 Enhancements and Simplifications to Circle Covering

The CC algorithm has some downsides, considering coverings with penetration depth. Pieces whose vertices have very acute angles may not be covered by circles (Fig.2(a)). This is because the penetration depth is measured perpendicularly to the edges of the outline, causing in these specific cases a distance higher than the penetration depth to be uncovered from the vertex to the closest circle. In order to correct this, circles are added until the distance from the last placed circle to the vertex is within the defined value for the penetration depth. If the circles added have the same size (in order to limit the number of additional circles placed) the covering will exceed the outline of the piece (Fig.2(b)), thus preventing solving problems with perfect fits. Considering this correction for the tips of the polygons, we can distinguish between five types of covering: *CCC*, *PCC*, *ICC*, *PCC* with tip covering (*PCCTc*) and *ICC* with tip covering (*ICCTc*).

A simplification approach can also be used to reduce the complexity of the representation by replacing two adjacent edges that have highly obtuse angles on the vertex that connects them. This simplification also takes into account the additional area that the piece will have, since it is always oriented to reduce the concavities of the piece. Another simplification, that also has an impact on the

final number of circles, is the simplification of the skeleton which consists in removing the smallest bones of the skeleton, defined by a distance, and connecting the bones next to it together. The remaining option to reduce the number of circles consists in merging pairs of circles whose resulting circle would not be larger than the biggest circle of the pair, by a certain distance. These simplifications are used mostly on pieces that have very complex outlines, where minor simplifications can significantly reduce the total number of circles produced.

4. NLP MODEL FOR NESTING PROBLEM WITH CONTINUOUS ROTATIONS

The CC approach was developed for solving Nesting problems with continuous rotations. The non-linearity of the non-overlapping constraints of the CC leads to the formulation of a NLP model. The presented model focuses on solving Irregular Piece Strip Packing problems, thus the container is a rectangle with fixed width and infinite length, with the aim to minimize total length used.

$$\text{minimize } l \tag{1}$$

subject to:

$$x_{k_i} + R_{k_i} \leq l, \quad \forall i \in C_k, \forall k \in \mathbb{N} \tag{2}$$

$$-x_{k_i} + R_{k_i} \leq 0, \quad \forall i \in C_k, \forall k \in \mathbb{N} \tag{3}$$

$$y_{k_i} + R_{k_i} \leq W, \quad \forall i \in C_k, \forall k \in \mathbb{N} \tag{4}$$

$$-y_{k_i} + R_{k_i} \leq 0, \quad \forall i \in C_k, \forall k \in \mathbb{N} \tag{5}$$

$$(x_{k_i} - x_{h_j})^2 + (y_{k_i} - y_{h_j})^2 \geq (R_{k_i} + R_{h_j})^2, \quad \forall i \in C_k, \forall k \in \mathbb{N}, \forall j \in C_h, \forall h \in \mathbb{N}, k \neq h \tag{6}$$

$$x_{k_i} - x_{k_0} = \cos(A_{k_0,i} + \theta_k) \times D_{k_0,i}, \quad \forall i \in C_k, \forall k \in \mathbb{N} \tag{7}$$

$$y_{k_i} - y_{k_0} = \sin(A_{k_0,i} + \theta_k) \times D_{k_0,i}, \quad \forall i \in C_k, \forall k \in \mathbb{N} \tag{8}$$

$$x_{k_i}, y_{k_i}, \theta_k, l \in \mathbb{R} \tag{9}$$

The NLP model consists of variables that relate to the coordinates of each piece, their orientation and the value of the objective function. Regarding the constraints, there are three types: non-overlapping constraints, piece integrity constraints and containment constraints. The objective function of the model, as seen in (1), is represented by the variable l . The objective is to minimize it, therefore reducing the occupied length of the strip. Each piece k is composed by a set of C_k circles. For each circle, two variables are defined: x_{k_i} is a variable that defines the position of circle i of piece k on the x -axis, and y_{k_i} is a variable that defines the position of circle i of piece k on the y -axis. Each set of circles has an associated variable that determines their current orientation, defined as θ_k , that is the angle of piece k . The non-overlapping constraints, defined in (6), being this equation a comparison between the distance between each pair of circles i, j from pieces k, h and the radius of both circles R_{k_i} and R_{h_j} . The containment constraints, defined from (2-5) ensure that the x and y positions of each circle do not exceed the admissible placement area. Equations (7-8) are used to maintain the relative position between circles of the same piece, ensuring the piece integrity. This means that when the piece is rotated, the relative positions between all

Instance	Number of pieces	Total number of vertices	Total number of concavities	Strip width
<i>jakobs1</i>	25	150	22	40
<i>polyKa</i> *	15×k	69×k	6 × k	40

* with $K = \{1, 2, 3\}$

Table 1. Test instances used for computational experiments.

circles from the same piece are maintained. The position of each circle i is fixed relative to the circle 0 of the piece k . $A_{k_0,i}$ and $D_{k_0,i}$ consist, respectively, of the initial angle and distance between circle i and circle 0 of piece k . In (9) the domain of the variables is defined.

5. COMPUTATIONAL EXPERIMENTS AND RESULTS

The computational tests were selected with the objective of comparing various CC, regarding the number of circles and the quality of the solutions achieved when solving the Nesting problem with continuous rotations through a NLP model. These computational experiments allow to verify the behavior of each covering, and determine if some coverings are more suited to a specific instance than other coverings. This was done using the NLP model with the solver, to compact the layout of each instance. After selecting one type of covering that achieves good quality solutions, the impact of the number of circles on the NLP model was analyzed with both *LR* and *HR* versions of the selected CC.

5.1 Test Settings

The tests were done on a computer with two Intel Xeon E5-5690@3.46Ghz processors and 48Gb Ram@1333Mhz. The operating system is Ubuntu 12.04 LTS x86-64, and the Non-Linear solver is Algencan¹ v2.37 (Andreani et al., 2007), (Andreani et al., 2008). The covering algorithms were developed and implemented in C++. The four instances used, *jakobs1*, *poly1a*, *poly2a* and *poly3a*, were taken from the ESICUP² website. The details about the instances can be seen in Table1. When we refer to instance *polyKa*, the k is a multiple of the *poly1a* instance.

The selection of these instances is due to their geometry. The instance *jakobs1* has many pieces with concavities and square angles and the instances *poly1a*, *poly2a* and *poly3a* are composed by mixed types of pieces, but mostly convex, some being small, others big. This has the benefit of testing the coverings in instances with different characteristics, thus better understanding its efficiency, and also allowing a comparison with other results in the literature. The initial solutions were generated by placing the pieces into a grid, with a random sequence, and random orientations. For the tests with each type of covering and resolution, the same set of 30 initial solutions were used. All pieces have been represented with their maximum quality, only being configured the types of coverings and their correction for the tip coverings.

¹ <http://www.ime.usp.br/~egbirgin/tango/>

² EURO Special Interest Group on Cutting and Packing (www.fe.up.pt/esicup)

Instance	Res.	Total # of circles for <i>HR</i> coverings				
		<i>CCC</i>	<i>PCC</i>	<i>ICC</i>	<i>PCctc</i>	<i>ICctc</i>
<i>jakobs1</i>	<i>LR</i>	208	202	198	202	198
	<i>HR</i>	368	368	340	368	340
<i>polyKa</i>	<i>LR</i>	154×k	149×k	145×k	149×k	155×k
	<i>HR</i>	260×k	254×k	250×k	254×k	261×k

Table 2. Number of circles depending on *LR* and *HR* resolutions and types of CC.

5.2 Model behavior with different covering types

The number of circles derived from the different types of covering, with different approximation quality (*LR* and *HR*), for each instance, can be seen in Table 2. The *LR* covering uses an approximation threshold of 0.25 units while *HR* uses 0.10 units. Table 2 shows that the total number of circles increases proportionally to the complexity of the instance (total number of vertices), and that no significant difference exists between different types of covering, although there is a noticeable difference between both resolutions used. The *HR* coverings have many more circles than the equivalent *LR* covering.

The impact on the NLP model of the different types of covering is presented in Table 3. Considering the compaction using the *ICC* and *PCC* variants, the layouts are infeasible but have the potential to achieve tighter layouts than *CCC*, if the overlaps are corrected. The average infeasibilities are determined by the average number of overlapping events (considering the polygonal representation of the pieces) for every layout produced. The approach used to solve the overlap may not be able to eliminate all overlaps if the pieces are packed to tightly and solving the overlap usually degrades the compaction quality. Thus, using a *ICC* type of covering is only beneficial if the layouts have configurations that allow perfect fits between pieces. The *CCC* achieves a feasible compaction but with significant waste due to the excess coverage of the pieces. The *PCC* has a much lower number of average infeasibilities than *ICC*, which can be easier to correct due to low number and for its much smaller penetration depth. Its compaction values on Table 3 also show potential tighter layouts than *CCC*, thus allowing us to conclude that the *PCC* is the most balanced covering to be used.

The use of tip covering (*ICctc*) leads to a small reduction in the potential compaction of the layout, but also a lower number of infeasibilities, which improve the chance of successfully correcting the layout. The *PCctc* is not compared since the *PCC*, with the used parameters, already satisfies the requirements of the tip covering, thus not requiring additional circles (*PCC* is the same as *PCctc*). A compaction and separation approach based on a linear programming model ((Gomes and Oliveira, 2006)) can be used to remove infeasibilities. Preliminary tests showed that the length l of the layout can increase up to 3%. This method can also be used to improve the compaction quality when using the *CCC*, due to the reduction of wasted space caused by the circles that exceed the outline of the pieces.

Instance	Cover	Obj. Function (l)			Avg. Infeas.	Avg. Time (s)
		Min	Avg.	Max		
<i>jakobs1</i>	<i>CCC</i>	14.50	16.45	21.76	0.0	209.8
	<i>PCC</i>	13.27	15.60	20.27	5.7	369.4
	<i>ICC</i>	12.89	14.79	17.95	39.2	175.1
	<i>ICCTc</i>	12.89	14.87	17.95	35.2	171.8
<i>poly1a</i>	<i>CCC</i>	14.93	16.24	17.83	0.0	53.6
	<i>PCC</i>	13.87	15.79	17.32	5.7	53.3
	<i>ICC</i>	13.90	15.02	16.74	25.2	59.7
	<i>ICCTc</i>	14.19	15.23	17.90	35.1	60.2
<i>poly2a</i>	<i>CCC</i>	28.36	30.73	32.73	0.0	885.0
	<i>PCC</i>	27.44	28.91	31.05	14.1	735.7
	<i>ICC</i>	26.24	27.69	29.17	61.4	1455.6
	<i>ICCTc</i>	25.95	28.12	30.62	54.9	2321.1

Table 3. Solution analysis for CC types (using a *HR* covering).

Instance	Solution Approach	Obj. Function (l)			Avg. Time (s)
		Min	Avg.	Max	
<i>jakobs1</i>	<i>LR</i>	14.03	15.83	18.85	49.0
	<i>HR</i>	13.27	15.60	20.27	369.4
	<i>LR+HR</i>	13.49	15.14	18.36	170.9
<i>poly1a</i>	<i>LR</i>	14.30	16.32	18.90	18.1
	<i>HR</i>	13.87	15.79	17.32	53.4
	<i>LR+HR</i>	13.93	15.69	17.64	33.6
<i>poly2a</i>	<i>LR</i>	27.37	29.85	32.56	270.8
	<i>HR</i>	27.44	28.91	31.05	735.7
	<i>LR+HR</i>	26.50	28.95	31.87	420.6
<i>poly3a</i>	<i>LR</i>	42.12	43.76	45.31	2869.3
	<i>HR</i>	41.09	42.69	44.37	6638.7
	<i>LR+HR</i>	39.52	41.37	44.13	3223.4

Table 4. *LR*, *HR* and *LR+HR* scaling with *PCC* and NLP model.

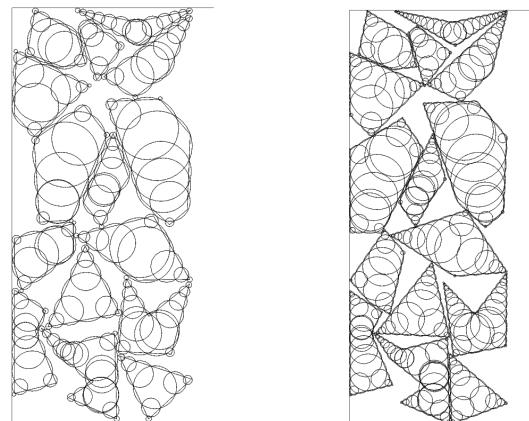
5.3 Model behavior with different circle covering resolutions

The impact that each resolution has, together with the NLP model, using *PCC*, on the compaction quality of the layout is shown in Table 4.

The feasibility of the layout regarding polygonal pieces is not considered, since the aim is to analyze behavior of the NLP model with different levels of approximation quality. Table 4 shows that using a *LR* covering has a significant impact on the quality of the layout compaction, considering the minimum and average compaction length, by having worse results than the *HR* covering. The compaction quality of *HR* causes a negative impact on the computational time, being significantly higher than *LR*. This behavior is similar in all instances. Using the NLP model with *HR* achieves better quality layouts, while compacting with *LR* is much faster (which can be very useful when dealing with large instances). The large size of *poly3a* instance increases the computational cost significantly when compared to other instances, and its difficulty further increases when using *HR* covering.

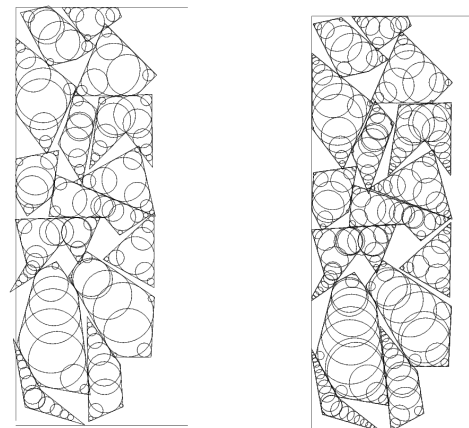
5.4 Multi-resolution and post-optimization approach

The advantages of both *LR* and *HR* coverings can be combined into a two-step compaction approach (called *LR+HR*), where the first step uses *LR* to compact the pieces, while second step uses *HR* to adjust pieces to each other. This procedure tries to combine the fast compaction speed of one with the high compaction quality of the other. Table 4 shows the results when compared to the



(a) *LR CCC* in Step 1.

(b) *HR CCC* in Step 2.



(c) *LR ICC* in Step 1.

(d) *HR ICC* in Step 2.

Figure 3. Compaction of *CCC* & *ICC* coverings with step 1 and step 2 of *LR+HR*.

single-step compaction using *LR* and *HR*. The *LR+HR* compaction achieves layouts with very good quality, when compared with the *HR* compaction, while significantly reducing the computational cost. In *opoly2a*, it is able to achieve the best quality layout due to its re-compaction, by leaving the first local minimum found on the first step. In *poly3a* this effect is much more noticeable, where the use of *LR+HR* increases the compaction quality and also reduces the computational cost.

The effect of combining different types of covering and resolutions with a two-step approach can be seen in Fig.3. Fig. 3(a) uses *CCC* with *LR*, achieving a feasible layout, but with some additional waste between pieces. Re-compacting using a *HR* covering produces the layout in Fig. 3(b), which is feasible, and improves the length to 15,22 from 16,73. Using a *LR* compaction with *ICC* produces the layout in Fig.3(c), with overlaps between the pieces. The quality of the compaction is degraded when using *HR* with *ICC*, as seen in Fig. 3(d), although the amount of overlap between pieces is significantly reduced. The *ICC* with *LR* has a length of 13,39 which is degraded to 13,49 with *HR*.

Since coverings with higher resolution have lower values for the penetration depth, when their CC is feasible the number of infeasibilities will be reduced (considering

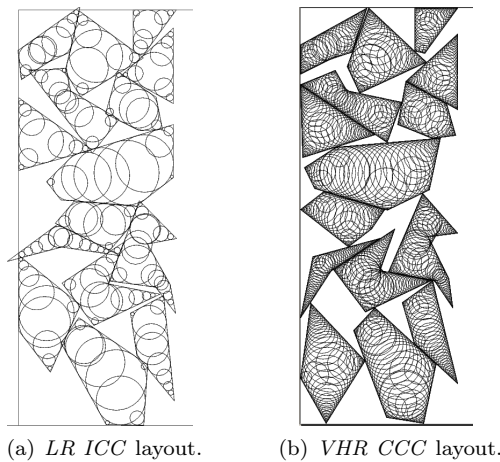


Figure 4. Using Post-Optimization to solve an *ICC* layout.

the piece outlines), compared to coverings with lower resolution. This effect allows addressing infeasible layouts, in a post-optimization phase, to improve a current layout, by re-compacting using a higher resolution with *CCC*, to ensure a feasible solution. This approach is being worked upon, thus the results for the instances presented in this paper are not shown. An example of this process can be seen in Fig. 4, where a *LR ICC* layout (Fig. 4(a)) is re-compacted with a very high resolution *CCC* producing a *VHR CCC* layout (Fig. 4(b)) that is feasible. In this example, to make the layout feasible the layout compaction quality was reduced.

6. CONCLUSIONS AND FUTURE WORK

The different types of covering, and multiple resolutions have a significant impact on the feasibility and quality of the layout. The *CCC* is able to produce feasible layouts but with wasted space between the pieces, due to its excess covering. The *ICC* allows for potentially tighter layouts, at the expense of a higher number of infeasibilities, which may not be possible to correct. The *PCC* is a trade-off between both *CCC* and *ICC*, where while the layouts are still usually infeasible, the number of infeasibilities is much lower than *ICC*, being much easier to correct, while still allowing for potentially tighter layouts than the *CCC*. Regarding the different resolutions, *LR* increases the effects of each of the covering type (higher excess covering in *CCC* and higher penetration depth in *ICC*) while allowing for a significant reduction in computation time, comparatively to the *HR* covering, which produces the best layout quality. The two-step *LR+HR* approach produces great improvements compared to the single-step *LR* and *HR* compaction, by successfully combining the strengths of each resolution to achieve a good quality compaction, at a reduced computational cost. One of the effects of using a higher resolution on the second step was the reduction of the number of infeasibilities on the layout, which, if combined with a higher resolution *CCC* can be an effective approach to correct layouts with infeasibilities. No results using this method are available, since this approach is currently being worked upon. Another approach that is also being explored to tackle the infeasibilities consists in using a linear programming model to move pieces into feasible positions. Further developments of the *CC*

approach may consist in adjusting its parameters to better tackle specific characteristics of real world problems, such as the requirement of a minimum distance between pieces placed into a layout, due to the cutting procedure, such as when cutting wood pieces from a board. Regarding the future development of the *NLP* model, improvements may be achieved by reducing the number of variables and constraints, and also changing the objective function in order to improve its compaction quality and computational cost.

REFERENCES

- Andreani, R., Birgin, E., Martínez, J., and Schuverdt, M. (2007). On augmented lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, 18(4), 1286–1309.
- Andreani, R., Birgin, E., Martínez, J., and Schuverdt, M. (2008). Augmented lagrangian methods under the constant positive linear dependence constraint qualification. *Mathematical Programming*, 111(1), 5–32.
- Bennell, J.A. and Oliveira, J.F. (2008). The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2), 397–415.
- Birgin, E. and Sobral, F. (2008). Minimizing the object dimensions in circle and sphere packing problems. *Computers & Operations Research*, 35(7), 2357–2375.
- Edwards, R.G. (2010). Determining the skeleton of a simple polygon in (almost) linear time. Technical report, Oak Ridge, Tennessee.
- Gomes, A. and Oliveira, J. (2006). Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research*, 171(3), 811–829.
- Huang, C.F. and Tseng, Y.C. (2005). A Survey of Solutions to the Coverage Problems in Wireless Sensor Networks. *Journal of Internet Technology*, 6(1), 1–8.
- Imamichi, T. (2009). *Nonlinear Programming Based Algorithms to Cutting and Packing Problems*. Ph.D. thesis, Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto.
- Rocha, P., Gomes, A., Rodrigues, R., and Toledo, F. (2013). Circle covering using medial axis. *Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems*, 11, 402–407.
- Stoyan, Y., Scheithauer, G., Gil, N., and Romanova, T. (2004). ϕ -functions for complex 2d-objects. *4OR: A Quarterly Journal of Operations Research*, 2(1), 69–84.
- Zhang, W. and Zhang, Q. (2009). Finite-circle method for component approximation and packing design optimization. *Engineering Optimization*, 41(10), 971–987.