Circle Covering using Medial Axis ⋆

Pedro Rocha [*] **Rui Rodrigues** [*] **Franklina M.B. Toledo** [**]
A. Miguel Gomes [*]

[*] *INESC-TEC, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal, (e-mail: {pro10015, rui.rodrigues, agomes}@fe.up.pt)*
[**] *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Caixa postal 668, São Carlos - SP 13560-970, Brasil, (e-mail: fran@icmc.usp.br)*

Abstract: A good representation of a simple polygon, with a desired degree of approximation and complexity, is critical in many applications. This paper presents a method to achieve a complete Circle Covering Representation of a simple polygon, through a topological skeleton, the Medial Axis. The aim is to produce an efficient circle representation of irregular pieces, while considering the approximation error and the resulting complexity, i.e. the number of circles. This will help to address limitations of current approaches to some problems, such as Irregular Placement problems, which will, in turn, provide a positive economic and environmental impact where similar problems arise.

Keywords: Circle Covering, Medial Axis, Irregular Polygons, Irregular Piece Placement Problem, Geometric Representations, Geometric Approaches.

1. INTRODUCTION

Problems that deal with the positioning of pieces in a given region without overlap, while having the objective of finding the best placement positions in order to minimize waste, require adequate representations for their geometry. Nesting problems are defined as an Irregular Piece Placement problem (IPP), inside one or multiple containers, in a non-overlapping condition, while aiming to reduce waste. The difficulty of these problems grows exponentially with the number and complexity of the pieces. Using an efficient geometrical representation can reduce the complexity of the geometrical component of the problem, allowing for a intensified focus on improving the other components of the problem. An efficient representation might also enable to explore new ways of approaching a problem, such as dealing with free-rotations, or computing overlaps with a simple formula. A geometrical representation, such as the one that is presented on this paper (Circle Covering), can be used to address these limitations (overlap computation and free-rotations) in Nesting. Circle Covering (CC) consists in representing pieces by sets of overlapping circles (whose areas when summed approximate the original piece area), aiming to achieve a certain degree of approximation while minimizing the necessary number of circles, and allows taking full advantage of any possible rotation, enabling easy and fast overlap computation.

The remainder of this section presents a description about the motivation for this work, an overview of the main objectives and ends with the research contributions. The second section, presents a literature review, making reference to various representation forms, the third section relates to the proposed approach for circle covering, and following it, in the fourth section, the computational experiments and the results obtained. The last section is the conclusion and future work.

1.1 Motivation

As previously mentioned, the CC approach is useful to tackle the limitations of problems such as the IPP problem, regarding the admissible orientations and the computational expense of overlaps. Addressing these issues in the geometrical component of the IPP problem allows its combinatorial component to be tackled by alternative approaches. The improvement of current geometric representations will have relevant impact on scientific areas on many industrial applications, both in economic and environmental levels.

1.2 Contribution

Many industrial applications (leather, furniture, garment) aim to find the most efficient placement of a set of pieces with irregular outlines inside a container, in order to minimize waste. While on some industries (furniture and textile) technological constraints and specific characteristics impose limits on the allowable orientations, on other industries (leather, metal sheet cutting) the pieces can rotate freely. The methods that allow the efficient positioning of pieces in order to minimize waste require adequate geometric representations, which can represent pieces with a certain degree of accuracy, and are the least computationally expensive as possible. The CC representation allows alternative approaches to be developed in order to

⋆ This research was partially supported by FP7-PEOPLE-2009-IRSES Project No. 24688 and by Fundação para a Ciência e Tecnologia (FCT) Projects PTDC/EME-GIN/105163/2008 (EaGLeNest) and PTDC/SEN-TRA/121715/2010 (StableCargo).

tackle the complexity of positioning the pieces, addressing current limitations of state of the art algorithms, and improving results. An efficient representation enables problems with greater complexity to be tackled, and explore novel solutions to currently unsolved problems.

This work has a relevant impact at different levels, since all industries with similar problems will benefit from it, having a positive economic impact (cost reduction) and also an environmental impact (less waste, energy and raw material consumption).

2. LITERATURE REVIEW

Problems that need to tackle with geometry require adequate selection of a geometrical representation of pieces. The selected representation should be suited to the specific geometrical characteristics of the main problem. Grid and polygonal representations (Bennell and Oliveira, 2008) are among the most commonly used geometrical representations. Another type of representations is based on decomposing pieces into simpler shapes. An example of this type of representation is the decomposition into primary shapes, which is used together with Phi-Functions (Stoyan et al., 2004) to compute the relative distance between pieces. A less commonly used method is the representation of a piece by a CC having a set of circles with same size or different sizes. Usually CC approaches have equal sized circles, however, since the goal is to achieve the minimum number of circles, while minimizing the approximation error, it can only be achieved by using circles with uneven radius. An example of a CC application can be found in wireless sensor networking (WSN) problems (Huang and Tseng, 2005), where the objective is to ensure that a given region is completely covered by a signal (with a minimum Quality of Service), without weak points. (Huang and Tseng, 2005) also describes related problems to the CC problem and WSN, such as the Art Galery problem, and Robotic Systems Coverage.

Other type of problem dealing with circles is circle packing, where a given set of circles needs to be packed inside a container, without overlap. Although this problem is not directly related to ours, it is presented here to expose its differences. (Huang et al., 2006) refers to using a fixed size circular container while (Kubach et al., 2009) changes to a strip or rectangle, depending on either addressing the Strip Packing problem or Knapsack problem, and (Birgin and Sobral, 2008) addresses the problem of finding the smallest container that can contain the specified items.

One paper that has a problem similar to ours is (Zhang and Zhang, 2009), which presents a method that approximates a component by a set of circles, where three algorithms represent distinct approaches to generate the minimum number of circles, with a given approximation accuracy. The bisection algorithm approximates the outline using circles, not ensuring circle covering of the whole piece, only the outline. The three step algorithm starts from each convex edge, placing the circles until the whole outline and inside of the piece is covered. The last algorithm is an improvement of the second, using a gap between the circles that places them a small distance apart, while ensuring that the gap is small enough to prevent the overlap of the pieces. In (Imamichi, 2009), several packing problems are described, such as circle/sphere packing and its variations, in different containers, rectangles, circles, and spheres in higher dimensions. It is also presented several distinct references to non-linear programming, in order to solve irregular strip packing problem.

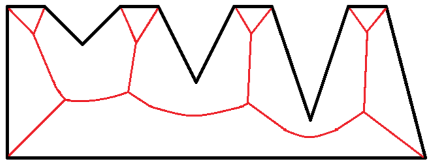


Figure 1. The Medial Axis of a piece.

There are many ways to compute possible placement points for circles inside a polygon, since we have an infinite number of placement positions to chose from. In order to find the best placement positions, it is desired to reduce the region that contains them, as much as possible, by excluding points of inferior quality. Circles have centers in high quality placement positions when those circles cannot be contained by any other circle, and still comply with the requirement of not exceeding the outline. An important tool useful for defining these inner points (that are equidistant to at least two points on the outline of a two-dimensional piece) is the Medial Axis (MA), which is a type of Topological Skeleton. An example of a MA is shown in 1. One method that allows to compute the MA is presented in (Yao and Rokne, 1991), but a more detailed description can be found on a technical paper by (Edwards, 2010). It allows the construction of the MA from simple polygons, returning only straight lines on the skeleton of convex polygons, and returning a set of straight lines and parabolic arcs from irregular polygons, i.e. non regular and non convex polygons. Unfortunately, this method is iterative, leading to increased numerical precision errors with the higher complexity of the pieces.

3. CIRCLE COVERING APPROACH

The proposed approach to compute the CC allows the complete coverage of a simple polygon by a set of uneven circles, which fully covers its outline and its interior.

The main difficulty that arises when generating a circle covering is the difficulty to achieve a low approximation error, with the lowest number of circles. Some details can be deduced beforehand, such as: the position of the center of the circles cannot be outside of the outline of the enclosed polygon, and that better results are achieved if each circle covers the maximum possible area. This means that each circle should have the highest possible radius without exceeding the admissible distance outside of the enclosed piece. For this purpose, placing the circles on the MA will lead to a low number of circles. This allows a significant reduction in the number of possible placement positions for the circle centers. An iterative process derives the circles directly from the MA. The parabolic arcs, which are generated at the concave vertices, are approximated by straight edges to simplify the process. Since a high number of circles has a negative impact on performance, it is desirable to keep the number of circles generated at its minimum.

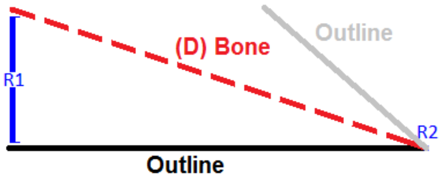


Figure 2. A bone of the skeleton (dashed line) and part of the outline of the piece.

3.1 Iterative Algorithm

The algorithm works upon each bone of the skeleton individually. It operates using the distances to the outline ($R1$ and $R2$, also referred as radius1 and radius2) at the extremities of the bone and the length of each bone (D). Distances $R1$ and $R2$ are measured perpendicularly to the edge of the outline, since a circle that is touching the outline, in only one point, has its center contained in the same line (perpendicular to the outline) that contains the contact point. An example can be seen in Fig. 2.

At the extremities of each bone, we place a circle with the same radius of $R1$ and $R2$ respectively, increased by a given threshold (or tolerance) value that controls our approximation error. The threshold can be seen in Fig. 3. Starting from the biggest circle, the algorithm iteratively computes the next center position, making sure that the circles intersect on the outline, until reaching the opposite circle. The radius of each new circle is easily computed since it is a linear function that depends on the difference of both radius and the distance that separates them. This procedure is the same for every bone except for bones connected to the outline of the piece. If all the placed circles cover the bone, then the algorithm stops generating any more circles, and removes the circle placed at $R2$.

This procedure can be also presented in the following pseudocode :

for all bones **do**
  Expand both circles radius by the threshold value
  Select biggest circle
  repeat
    Compute circle and outline intersection
    Compute next circle center position on the bone*
    Save circle position
  until Circles completely cover the bone
end for

* Taking into account that it must also intersect the previous circle/outline intersection and that the radius of the circle depends on the position of its center on the bone.

After repeating the same operation on every bone of the skeleton, the complete circle covering of a piece can be achieved. Fig. 4 shows the result when all the bones are processed, and the CC is generated. The gray area outside the outline of the piece is the exceeding region that is covered by the circles.

The algorithm computes every bone of the skeleton independently, taking into consideration both radius of the circles at the extremities of the bone, its length and the threshold that each circle is allowed to expand. An example

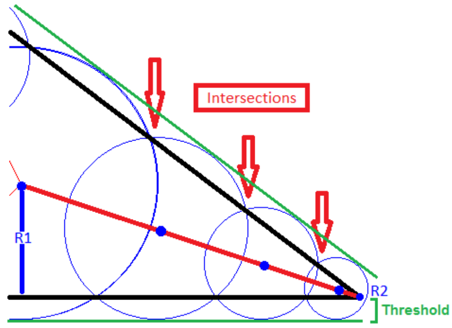


Figure 3. Computed circle placement positions, regarding a specified threshold.

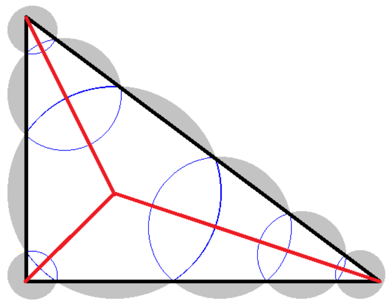


Figure 4. Circle covering of a triangle.

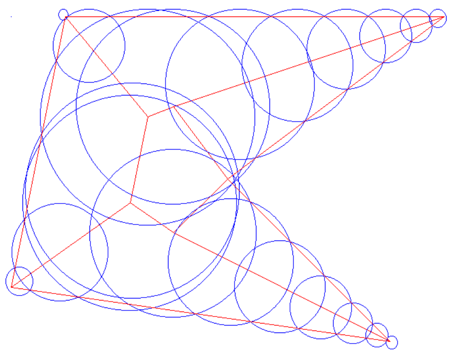


Figure 5. Complete circle covering of a piece.

of a piece that has the complete circle coverage computed, using its skeleton, is shown in Fig. 5.

The method that computes the positions of circles on the skeleton may produce placement positions of circles that are very close to each other. To reduce the total number of circles, in pieces that have either too much detail or very large skeletons, some simplifications were implemented. One of the simplifications is replacing consecutive edges, sharing a concave vertex, with nearly the same tangent, by a single edge. Another simplification, removes inner bones of the skeleton (i.e., not connected to the outline) that are smaller than a given length, and joining all the connections at its extremities into a single common midpoint.

3.2 Partial Circle Covering

An alternative to the complete coverage, the proposed algorithm also has the capability to fill the piece from the inside, without any circle exceeding its outline, and without any uncovered region more distant from the outline than the desired threshold. This is achieved with the same

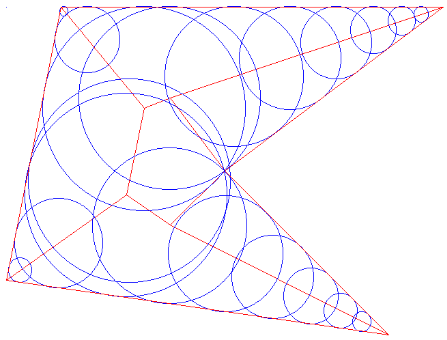


Figure 6. Partial circle covering of a piece.

procedure as the complete circle covering, but in this case, the threshold is set as being the outline, and the outline distance from the MA is reduced by the threshold value. This has the same effect as reducing the outline of the piece by the threshold value, and then computing the complete circle covering. This way, partial circle covering is achieved, guaranteeing that the outline is never trespassed. This may be useful for problems where the original pieces have a perfect fit in the layout. An example of a coverage of this type, that does not exceed the original piece outline, with its skeleton, can be seen in Fig. 6.

4. COMPUTATIONAL EXPERIMENTS AND RESULTS

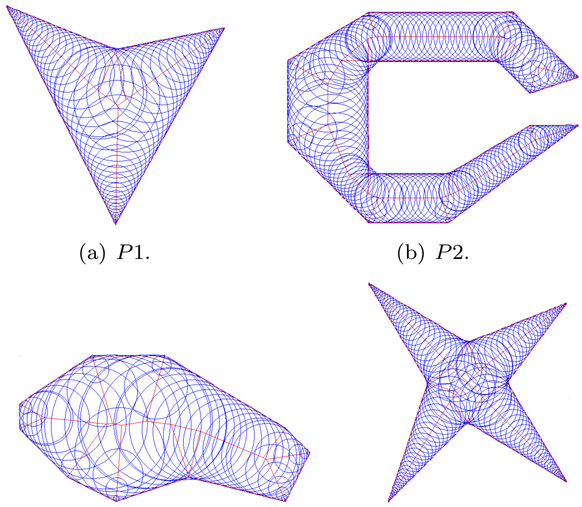
The first experiments that were made are related to the increase of the number of circles when the approximation threshold is low. In order to verify the efficiency of our approach, four pieces were chosen and tested with an approximation threshold from 0.25 to 0.01. The results obtained are summarized on Tab. 1, where for each piece the first column shows the number of circles used in the covering, and the second one shows the additional area related to the original piece area. From this table, it can be observed that a linear reduction in the threshold leads to an exponential increase on the number of circles and an almost linear reduction in the additional area.

	Pieces							
	P1		P2		P3		P4	
TH	#	%	#	%	#	%	#	%
0.25	13	14.1	35	13.2	12	12.1	28	20.6
0.20	14	10.9	37	10.6	13	9.6	29	15.7
0.15	17	8.3	41	7.9	14	7.1	36	12.2
0.10	21	5.5	48	5.2	15	4.4	41	7.9
0.05	31	2.7	70	2.6	25	2.1	61	4.0
0.01	72	0.7	161	0.1	55	0.4	141	0.9

Table 1. Circle number and added area for different threshold values (TH).

The MA+CC (Medial Axis followed by Circle Covering) of these four pieces can be seen on Fig. 7, being $P1$, $P2$, $P3$ and $P4$, with threshold of 0.01. The threshold is expressed in units of length, and is not relative to the size of the piece. As an example, if we use 0.01 units for threshold, it will be the same distance whether the rectangle has 10 or 15 units of length.

An important issue that must be analyzed is the trade-off between the increase in the number of circles and the reduction of the additional area. Tab. 1 already shows this



(a) $P1$. (b) $P2$.

(c) $P3$. (d) $P4$.

Figure 7. CC+MA of $P1$, $P2$, $P3$ and $P4$, with 0.01 threshold, and 72, 161, 55, 141 circles, respectively.

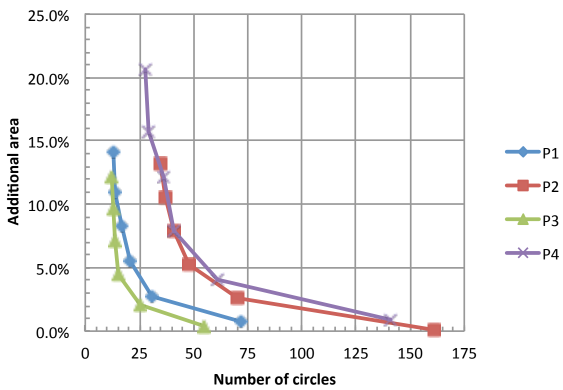
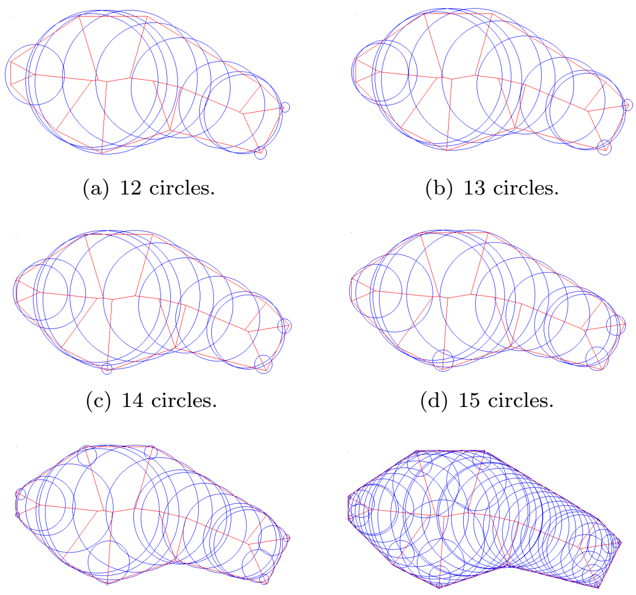


Figure 8. Trade-off between the number of circles and the additional area.

trade-off for pieces $P1$, $P2$, $P3$ and $P4$, where initially a decrease in the threshold value leads to a small increase in the number of circles and a significant reduction in the additional area while later the same reduction in the threshold value leads to large increase in the number of circles and a small reduction in the additional area (Fig. 8). The graph on Fig. 8 shows a similar behavior for all 4 pieces. The evolution of the circle covering of piece $P3$ can be seen in Fig. 9, where the covering with the most amount of circles is also the closest to the real shape of the piece.

The complexity of the piece also has a negative impact on the algorithm, since the numerical precision errors increase at each iteration, due to having to compute multiple intersections, leading to incorrect skeletons being built. For complex pieces where the MA cannot be correctly constructed, convex decomposition is used on the original piece, computing the skeleton for each one, but with the downside of increasing substantially the number of circles.

Comparisons of this MA+CC approach were also made against a set of two pieces, from (Zhang and Zhang, 2009), which were also used to test several distinct methods to



(a) 12 circles. (b) 13 circles.

(c) 14 circles. (d) 15 circles.

(e) 25 circles. (f) 55 circles.

Figure 9. Evolution of the circle covering of piece $P3$.

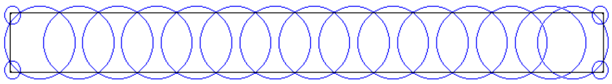


Figure 10. Circle covering the rectangle, with 19 circles.

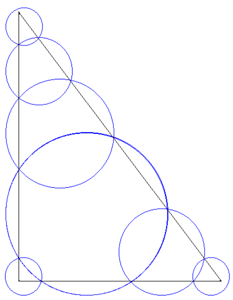


Figure 11. Circle covering of a triangle, with 7 circles.

achieve circle covering. (Zhang and Zhang, 2009) also uses an improved algorithm, that allows a small gap between circles, as long as the salient edge of another piece does not interfere with the current piece. This gap is the allowable distance between circles that still maintain the non-overlapping constraints. To compare the algorithms the same pieces were used, according to the dimensions of the pieces and their threshold, as presented on (Zhang and Zhang, 2009). The pieces that we obtained, with their respective number of circles are presented in Fig. 10 and Fig. 11, having the threshold of 0.1 and 0.2, respectively. These pieces have the circle covering computed by the presented CC+MA approach.

Comparing to the results from (Zhang and Zhang, 2009), Tab. 2 can be presented, with the focus being on the total number of circles used to represent the same pieces, with the same threshold. Although (Zhang and Zhang, 2009) had three distinct algorithms, we compared only against the ones that allowed complete or nearly complete covering of the piece.

	Pieces	
Algorithm	**Rectangle**	**Triangle**
CC+MA	19	7
Three-Step	22	8
Three-Step + Gap	21	6

Table 2. Circle number depending on algorithm.

According to the results, using medial axis and computing the circle placement positions on its skeleton can be competitive when compared to the three step algorithm, for these two pieces. We achieve the best results on one piece. Allowing a small gap prevents a complete covering of the piece, but the overlap computation is still valid as long as the criteria that define the maximum gap is met. Unfortunately, computational results about the other algorithms are lacking, preventing an extensive comparison between our approach and the others. This paper will have as an annex the details about the used pieces, so that further comparisons can be done by other researchers.

5. CONCLUSIONS AND FUTURE WORK

The MA+CC approach appears to be competitive with the presented algorithms. This method to generate circle placement positions by reducing the possible positions to a medial axis is very efficient. The main problem with this approach is that the method that generates medial axis is not very resilient to numerical precision errors. The error has a tendency to grow with the complexity of the piece, and the skeleton bones start appearing with a visible offset. For pieces where its skeleton cannot be constructed due to numerical precision errors, the convex decomposition of the piece is used, but at the expense of a greater number of circles, since the generated skeleton will have a higher amount of bones. The number of circles depends on the approximation error selected, and the lower the desired error, the higher the circle number will be.

Some ideas will be explored in the near future. Our method might achieve better results when computing placement positions of circles taking into account the introduction of a small gap, such as presented on (Zhang and Zhang, 2009), but that is limited to the smallest circle size that can be used on all the pieces, which also depends on the approximation threshold, being the same for all the pieces. Another component that is being implemented is the simplification of the outline of complex polygons, by reducing the number of small concavities, with the idea of reducing the complexity and numerical precision errors and the outline complexity when generating the medial axis, as also simplifying the final skeleton, by eliminating or rearranging some small bone segments while maintaining coherency in the structure. The penalty to pay is that at each modification the size of the circles will grow to ensure that the piece is still fully contained (assuming the complete coverage is desired), but it is expected that the number of circles will be further reduced.

This will help reducing the total number of circles required to represent a piece, within the limits of approximation error to the original piece.

REFERENCES

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.

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.

Huang, W.Q., Li, Y., Li, C.M., and Xu, R.C. (2006). New heuristics for packing unequal circles into a circular container. *Computers & Operations Research*, 33(8), 2125 – 2142.

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.

Kubach, T., Bortfeldt, A., and Gehring, H. (2009). Parallel greedy algorithms for packing unequal circles into a strip or a rectangle. *Central European Journal of Operations Research*, 17, 461–477.

Stoyan, Y., Scheithauer, G., Gil, N., and Romanova, T. (2004). ϕ-functions for complex 2d-objects. *4OR: A Quarterly Journal of Operations Research*, 2, 69–84.

Yao, C. and Rokne, J. (1991). A straightforward algorithm for computing the medial axis of a simple polygon. *International Journal of Computer Mathematics*, 39(1-2), 51–60.

Zhang, W. and Zhang, Q. (2009). Finite-circle method for component approximation and packing design optimization. *Engineering Optimization*, 41(10), 971–987.

Appendix A. PIECE COORDINATES USED IN COMPUTATIONAL EXPERIMENTS

A.1 Piece 1 {x,y}

{0,0} {5,2} {10,1} {5,10}

A.2 Piece 2 {x,y}

{0,0} {9,0} {13,4} {10,5} {8,3} {0,3} {0,10} {5,10} {10,7} {13,7} {5,13} {0,13} {-5,8} {-5,3}

A.3 Piece 3 {x,y}

{3,0} {6,0} {11,3} {12,4} {11,6} {7,5} {4,6} {2,5} {0,3} {0,2}

A.4 Piece 4 {x,y}

{0,0} {5,3} {10,1} {7,5} {10,10} {5,7} {1,11} {3,5}