Stock Cutting Of Complicated Designs by Computing Minimal Nested Polygons

By: J. Bhadury and R. Chandrasekaran

J. Bhadury, R. Chandrasekaran (1995). "Stock Cutting of Complicated Designs By Computing Minimal Nested Polygons", *Engineering Optimization*, Vol. 25, pp 165-178. DOI: 10.1080/03052159508941261

Made available courtesy of Taylor & Francis: http://dx.doi.org/10.1080/03052159508941261

Abstract:
This paper studies the following problem in stock cutting: when it is required to cut out complicated designs from parent material, it is cumbersome to cut out the exact design or shape, especially if the cutting process involves optimization. In such cases, it is desired that, as a first step, the machine cut out a relatively simpler approximation of the original design, in order to facilitate the optimization techniques that are then used to cut out the actua1 design. This paper studies this problem of approximating complicated designs or shapes. The problem is defined formally first and then it is shown that this problem is equivalent to the Minima1 Nested Polygon problem in geometry. Some properties of the problem are then shown and it is demonstrated that the problem is related to the Minimal Turns Path problem in geometry. With these results, an efficient approximate algorithm is obtained for the origina1 stock cutting problem. Numerica1 examples are provided to illustrate the working of the algorithm in different cases.

Key Words: stock cutting, design approximation, minima1 nested polygon, approximate algorithm.

Article:

1 INTRODUCTION

Stock cutting refers to the operation of cutting out a given design or shape from parent material. Most of it is done by automated machines nowadays. Problems arising in stock cutting have been studied for some time but traditionally, most of the literature on stock cutting in the OR/OM community has focused on minimizing the amount of parent material wasted in cutting—for example Gilmore and Gomory[10,11], Christofides and Whitlock', Dori and Ben-Assat[6], Venkateswarlu and Martyn[15]. There are, however, other problems of theoretical and practical interest in this area. One such problem arises when it is required to cut out shapes or designs that are very complicated, and hence it is difficult, error prone (and therefore expensive) to cut out such shapes or designs. In additions, almost all cutting processes utilize some optimization techniques—for instance, these might be intended to minimize the trim losses encountered in the cutting process (as in Venkateswarlu and Martyn[15]) or the time required to cut out the design (as in Bhadury and Chandrasekaran[2]). As all of these optimization procedures involve extensive computations, the time taken by them depends, to a large extent, on the number of edges in P_{in} and P_{out} – thus the more complicated the design or the parent material, the more time consuming it is to apply optimization techniques. Therefore, when the designs and shapes that are to be cut out are very complicated it is desirable to approximate them first, by simpler designs or shapes.

There are many ways to define "simpler" – however, one of the most popular ways to define it is to approximate the shape with another one that has the fewest number of edges. While the intuitive reasoning behind such an approximation is clear, there is also a practical reason for using it. Fewer edges mean fewer turns to be taken by the cutting machine and that is desirable because of the standard assumption in robotics, that it is more expensive for any automatic machine to stop and turn than it is for it to travel straight.

It is this problem of design approximation in stock cutting of complicated designs, that will be studied in this paperr. A formal description of the problem is as follows: given a piece of material, (assume that it is a polygon and therefore its boundary is made up of straight line edges) it is required to cut out a design that is drawn upon it (also assumed to be polygonal). Because of the complexity of the design, it is desired to approximate the design instead by the simplest design possible, i.e. by another design with the minimum number of sides. This can be seen equivalent to the following problem in geometry, known as the Minimal Nested Polygon problem: given two simple polygons P_{in} and P_{out} with P_{in} completely contained in P_{out}, it is required to find another polygon, P^*, that is contained in the annular region between P_{in} and P_{out}, completely contains the polygon P_{in} and has the fewest number of sides.

Once the approximation is completed, the design that is to be cut out is enclosed in the simplest possible shape. Then, in the second stage of the cutting process, the desired design is cut out from this approximation after applying the required optimization techniques. Thus by replacing the parent material P_{out}, by as simple a shape as possible, this approximation facilitates this optimization process. For the same reasons, this approximation is also helpful when the given design is not complicated, but the parent material, i.e. P_{out} is.

Two other interesting applications of this problem are in facility layout and robotic path planning respectively. In facility layout problems, this problem arises when the annulus between P_{in} and P_{out} represents a geographical region and it is required to cover the entire region by locating the fewest number of facilities while ensuring that line-of-sight communication is maintained between adjacent facilities. In robotic path planning this problem has applications when the annulus between P_{in} and P_{out}, represents the space that has to be traversed by a robot, and it is required to design a path that makes the minimum number of turns. The Minimal Nested Polygon problem has therefore been extensively studied, especially in the computer science community-see Aggarwal *et al*.[1], Ghosh and Maheshwari[9], Chandru *et al*.[4]-all of whom give algorithms for different cases of this problem.

In this paper however, some different aspects of this problem have been explored. Some properties of the Minimal Nested Polygon problem are shown first, and then it is shown that the design approximation problem being considered in this paper is related to another well known problem in geometry, namely, the Minimal Turns Path problem. The Minimal Turns Path problem is stated as follows: given a simple polygon P and any two points x and v in it, find a path between x and y that is fully contained in P and makes the minimum number of turns (see Suri[14], Reif and Storer" for algorithms for the problem). With the results obtained, an efficient algorithm is given which, given a pair of polygons P_{in} and P_{out}, outputs a polygon that is contained in the annulus between the two, contains P_{in}, and has at most two sides more than P^*.

The remainder of the paper is divided into four sections. The second section defines all the terms and notation used in the paper and reviews some existing results that are necessary for the analysis. Some results are then stated and proved in the third section—they help characterize the shape and relative position of the Minimal Nested Polygon P^* and also relate this problem to the Minimal Turns Path problem. With the aid of the results obtained, the fourth section gives an efficient approximate algorithm that gives an answer to the design approximation problem that is very close to the optimal answer. Finally, the fifth section summarizes the conclusions of the paper and discusses some open problems.

2 DEFINITIONS AND EX1ST1NG ALGORITHMS

This paper requires several definitions. All of the definitions required by the paper and the existing results in the literature that are used here are now defined below.

- The polygons P_{in} and P_{out} are assumed to have p and q vertices respectively. Hence a linear time algorithm for this problem would take $O(p + q)$ time.
- A polygon is said to be *simple* if and only if no two of its edges intersect at any point other than a vertex of the polygon—see Figure 1, where the polygon S is a simple polygon. If two edges of a polygon intersect at a point other than a vertex, (obviously these edges would then have to be non-adjacent), the polygon is called a non-simple polygon- as shown by the polygon in Figure 2.

- Two points in a polygon are said to be *visible* to each other if and only if the line joining these two points is completely inside the polygon, i.e. it never leaves the interior of the polygon. For example in Figure 1, points x and y are visible to each other but points x and z are not.

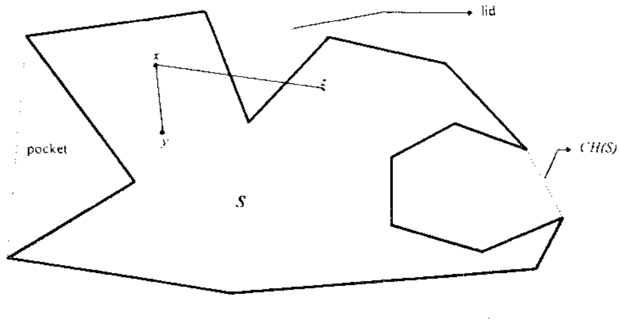

Figure 1

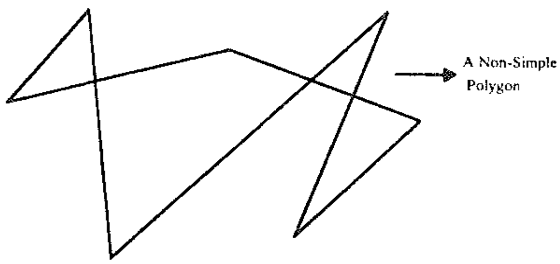

Figure 2

- A polygon is said to be *convex* if and only if all of its interior points are visible to each other. One of the properties of a convex polygon is that all its internal angles are less than 180°.
- The *convex hull* of a simple polygon *S*, denoted by CH(*S*), is the smallest convex polygon that contains *S* – see Figure 1 for the convex hull of the simple polygon *S*. As shown in the figure, "convexifying" a simple polygon by constructing its convex hull, identifies *pockets* and *lids* in the original simple polygon. It has been shown in Preparata and Shamos[12] that given a simple polygon, its convex hull can be found by a linear time algorithm.
- A polygon is said to be *star shaped* if there exists at least one point in it from which the entire polygon is visible. The set of all such points in a star shaped polygon, from where the entire polygon is visible is called the *kernel* of this star-shaped polygon—see Figure 3 for a star shaped polygon and its kernel. It is clear that a convex polygon is also star shaped and its entire interior is its kernel.
- Given two polygons *P* and *Q*, where *P* is simple and *Q* is convex and *Q* is completely contained in *P*, the *complete visibility polygon* of *Q* in *P* is the set of all points in *P* such that from each of these points the entire polygon *Q* is visible, i.e. from each of these points, it is possible to draw both the anti-clockwise and the clockwise tangent to P_{in}. For instance, if *P* is star shaped and *Q* is its kernel, then

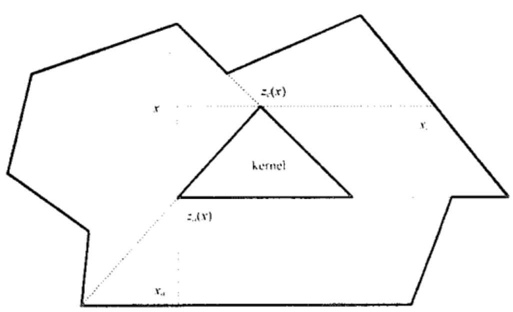

Figure 3

- P is the complete visibility polygon of Q in P. Given P and Q, the complete visibility polygon of Q in P can be found in linear time as shown in Ghosh[7].
- Given a polygon S and two points x and y in it, the algorithm in Suri[14] computes the Minimum Turns Path between x and y in linear time.

Consider any point x in the annulus between P_{in} and P_{out} such that CH(P_{in}) is completely visible from it, i.e. it is possible to draw both the clockwise and the anti-clockwise tangents from this point to P_{in}. Let the two vertices of P_{in} that these two tangents meet be denoted by $z_c(x)$ for the clockwise tangent and $z_a(x)$ for the anti-clockwise one respectively (in case any one (or both) of these tangents intersects P_{in} at two vertices choose any vertex arbitrarily to call $z_c(x)$ or $z_a(x)$). The point where the directed line segments $[\overrightarrow{x, z_c(x)}]$ and $[\overrightarrow{x, z_a(x)}]$ leave the interior of P_{out} for the first time is denoted by x_c and x_a respectively. See, for example, Figure 3 where the points x, $z_c(x)$, $z_a(x)$, x_c and x_a are shown.

3 MINIMAL NESTED POLYGONS AND MINIMAL TURNS PATHS

This section contains the lemmas that prove some properties of the Minimal Nested Polygon problem and establish its relationship with the Minimal Turns Path problem. Given the shapes and the relative positions of P_{in} and P_{out}, the shape and the position of P^* is characterized in Lemmas 1 and 2 and Corollaries 1 and 2. The final results from these lemmas and corollaries are listed in Table 1. The question to be investigated in these lemmas is the following: under what conditions will P^* be convex, and if so, what will be its relative position? The first part of the question is answered by Lemma 1 and Corollary 1 and the second part by Lemma 2 and Corollary 2 respectively.

Lemma 1: *For a given pair of polygons P_{in} and P_{out} the Minimal Nested Polygon P^* is convex if and only if* CH (P_{in}) *is fully contained inside P_{out}.*

Proof. In the appendix.

It is interesting to note that a similar property of a nested polygon, but not the Minimal Nested Polygon P^*, has been shown in Ghosh[7].

Corollary 1: *If P_{in} is convex, P^* is convex too.*

Proof. If P_{in} is convex, its convex hull is P_{in} itself, which is known to be contained in P_{out}. Hence the conditions of lemma 1 are automatically satisfied and P^* is convex. ■

Table 1 Shape and Relative Position of P^*.

Case A: *P_{in} is convex*	Case B: *P_{in} is non-convex*	
P^* is convex and contained in the complete visibility polygon of P_{in} in P_{out}.	If CH(P_{in}) is contained in P_{out}, we can replace P_{in} by CH(P_{in}) to get a new P_{in} that is convex. Then it becomes similar to Case A.	If CH(P_{in}) is not contained in P_{out}, then P^* is non-convex.

Lemma 2: (*from Ghosh*[7] *and Ghosh et al.*[8]) *For a given pair of polygons P_{in} and P_{out} if P^* is convex then P^* is fully contained in the complete visibility polygon of P_{in} in P_{out}.*

For a proof of Lemma 2, the reader is referred to the above papers. However based on the previous lemmas and corollary it can be claimed that:

Corollary 2: *If P_{in} is non-convex but* CH (P_{in}) *is contained inside P_{out}, then P^* lies in the annulus between* CH(P_{in},) *and the complete visibility polygon of* CH(P_{in}) *in P_{out}.*

Proof. Because CH(P_{in}) is contained inside P_{out}, it is known from Lemma 1 that P^* is convex. Thus, it can be claimed that P^* does not have any vertex inside a pocket of P_{in} that is formed by constructing CH(P_{in}), because if it did then P^* would not be convex. Therefore, by Lemma 2 above, P^* is inside the complete visibility polygon of CH(P_{in}) in Pout. ■

Hence, by Lemma 2, it can be concluded that if CH(P_{in}) is contained in P_{out}, then P_{in} can be replaced by CH (P_{in}) without changing the problem. The various properties of P^*, depending on the shapes of P_{in} and P_{out} are given in Table 1.

Having characterized the shape and relative position of P^*, it now remains to be shown how the Minimal Nested Polygon problem is related to the Minimal Nested Turns path problem, and how the latter can be used to give an approximate answer to the former. That is now done in Lemmas 3 and 4. These two lemmas will then form the basis for the approximate algorithm to be given in the next section. However the following definition is needed first.

Definition 1 *Choose any point x in the annulus that is outside* CH(P_{in}) *from which it is possible to draw both the clockwise and the anti clockwise tangent to* CH (P_{in}). C*onsider all paths from point x to itself that circumscribe* P_{in} *— of these, the one that makes the minimum number of turns to come back to the point x is called the Minimum Turns Path of x — this path also forms a polygon that is nested in the annulus between* P_{in} *and* P_{out} *and passes through x — this is referred to as the Minimal Turns Polygon of x and denoted by* MT (x).

One problem encountered in computing MT(x) is that the annulus between P_{in} and P_{out} is not a polygon—however this problem can be rectified in a simple way. Given P_{in}, P_{out} and x, MT(x) can be found by introducing an additional edge through x that cuts the annulus between P_{in} and P_{out} and converts it to a simple polygon and then giving this converted polygon as an input to the algorithm in Suri[14]. For example in Figure 4, the edge (7,F) is introduced and that converts the annulus to the polygon whose vertices are, (in order): F,7,1,2,...6, 7, F, E, D,C..., A, F. By the choice of the point x, it can be guaranteed that such an additional edge can always be constructed. Hence given P_{in}, P_{out} and a point x in the annulus with the required property, MT(x) can be found in linear, i.e. $O(p + q)$ time.

However, in the special case where P_{in} is convex and P_{out} is its complete visibility polygon for any point x in the annulus, MT(x) can be constructed by an easy procedure, without resorting to the algorithm in Suri[14]. The procedure is as follows: from x, draw the clockwise tangent from x to P_in and denote the intersection point of this tangent with the boundary of P_{out} as y. From y, repeat this process of drawing successive clockwise tangents to P_{in} until x becomes visible to the last found point on the boundary on P_{out}. This has been shown in Figure 4 whereand P_{out} are both convex and hence the Minimal Turns Path polygon of the point x in the figure is given by the polygon whose vertices, in order, are x, x_1, x_2, x_3 and have been found by drawing clockwise tangents from x successively. It is easy to verify that this procedure to obtain the Minimal Turns Polygon of a point also takes linear time.

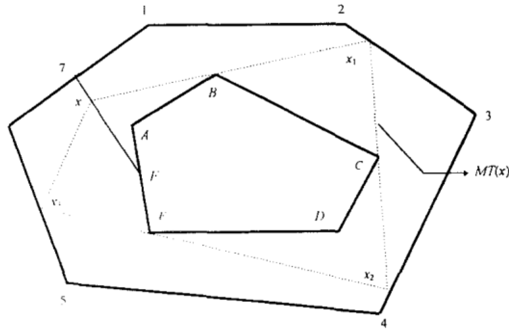

Figure 4

Lemma 3: *If is convex, consider any point .x that is in the complete visibility polygon of* P_{in} *in* P_{out}. *It is always true that* MT(x) *can have at most two more edges than* P^*.

Proof. In the appendix.

Lemma 4: *If* P_{in} *is non-convex and* CH(P_{in})is *contained in* P_{out}, *for any point x that is in the annulus between* CH(P_{in}) *and the complete visibility polygon of* CH(P_{in}) *in* P_{out}, MT(x) *has at most two more edges than* P^*. *If* CH (P_{in}) *is not contained in* P_{out}, *then there always exists a point x on the boundary of* CH(P_{in}) *for which the above is true.*

Proof. In the appendix.

Lemma 5: *For a given pair of polygons* P_{in} *and* P_{out} *if* P^* *is convex and if x is a point in the annulus outside* CH(P_{in}) *such that it is possible to draw both clockwise and anticlockwise tangents from x to* CH(P_{in}), then MT(x) *is also convex.*

Proof. In the appendix.

Hence by lemmas 3, 4 and 5, if one can choose a point x that is outside CH(P_{in}) and from which it is possible to draw both the clockwise and the anti-clockwise tangents to CH (P_{in}), then the Minimal Turns Polygon of x can be used as an approximation of P^* as it will have the same property of convexity as P^* and have at most two more edges. This will form the basis of the algorithm given in the next section.

4 AN EFFICIENT ALGORITHM FOR FINDING P^*

Based on the lemmas and corollaries of the previous section, the following algorithm is proposed to give an approximation of P^* Given P_{in} and P_{out}, the algorithm first determines whether P^* is convex. Then it chooses a point x with the property that for this point x, MT(x) is an approximation of P^*.

Algorithm *Approximate P**
Input: Two polygons P_m and P_{oor}
Output: A Nested Polygon that has the same convexity properties as P* and at most 2 more sides.
1. Check if P_{in} is convex.
2. If P_{in} is convex **then**
{
2.1 Because P_{in} is convex, by Corollary 1 the Minimal Nested Polygon, P^* is convex. It is also known that P^* is contained in the complete visibility polygon of P_{in} in P_{out}, Hence replace P_{out} by the complete visibility polygon of P_{in} in P_{out}. Then choose any point x in the annulus between P_{out} and P_{in}.
} end if
3. If P_{in} is non-convex then
{
3.1 Construct CH(Pin). Check if CH(Pin) is contained in P_{oo},.
3.2 If CH(P_{in}) is contained in P_{out} **then**
{
3.2.1 It is known from Corollary 2 that P* is convex and lies between CH(P_{in}) and the complete visibility polygon of CH(P_{in}) in P_{out}. Hence do the following:
3.2.2 Replace P_{in} by CH(P_{in})
3.2.3 Construct the complete visibility polygon of P_{in} in P_{out} and replace P_{out} with this complete visibility polygon.
3.2.4 choose any point x in the annulus between P_{out} and P_{in}.
} **end if**
3.3 If CH(P_{in}) is not contained in P_{out}, **then**

3.3.1 Choose any point x on the boundary of CH(P_{in}) that is inside P_{out}—in particular, one can choose any vertex of P_{in} that defines a lid in CH(P_{in}). Lemma 4 guarantees that with such a choice of x the algorithm will produce a correct approximation of P^*.

} **end if**

} **end if**

4. For the point x chosen construct MT(x).

Step 1 can be done in $O(p + q)$, i.e. linear time, by checking all the internal angles of P_{in}, and verifying if all are less than 180°. Step 2 requires the construction of the complete visibility polygon of P_{in} in P_{out} by using the algorithm given in Ghosh[7]; this can be done in the $O(p + q)$ time. Step 3.1 finds the convex hull of P_{in}, which can also be done in linear time using the algorithm in Preparata and Shamos[12]. In step 3.1, checking if CH(P_{in}) is fully contained in P_{out} can also be done in linear time using the algorithm given in Preparata and Shamos[12]. In step 4, MT(x) can be found in $O(p + q)$ time by edge through x, and then using the algorithm given in Suri[14]. Therefore the entire algorithm takes linear, i.e. $O(p + q)$ time. It has already been shown why MT(x) is guaranteed to have the same properties of convexity and at most two edges more than P^*. By the linearity of this approximate algorithm, it can be claimed that this is an approximate algorithm that takes optimal time.

To better explain the working of this algorithm, three examples are given—they are shown in Figures 4, 5 and 6 respectively. Table 2 lists gives a description of each these examples and the working of the algorithm in each case.

5 CONCLUSIONS AND FUTURE WORK

This paper has considered the problem of approximating complicated designs for stock cutting and has shown the equivalence of this problem and the well known Minimal Nested Polygon problem. The conclusions form this study can be summarized as.

1. The shape of the Minimal Nested Polygon P* is decided by the shapes and relative positions of Pin and Pou,, as shown in Table L

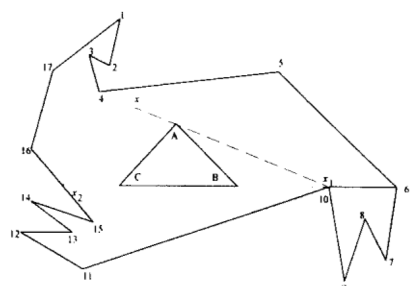

Figure 5

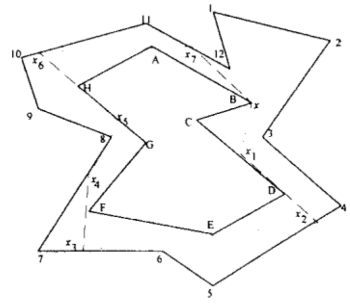

Figure 6

Table 2 Numerical Examples.

	P_{in} and P_{out}	*Properties of* P_{in} *and* P_{out}		*Steps of Algorithm Approximate-P**	
Figure 4	P_{in}: A, B, … E P_{out}: 1, 2, … 6	P_{in} is convex. P_{out} is convex. Hence $CH(P_{in})$ is contained in P_{out}. The complete visibility polygon of P_{in} in P_{out} is thus the entire polygon P_{out} itself.	In Step 1, the algorithm finds that P_{in} is convex. So it moves to Step 2.	In Step 2.1 it finds the complete visibility polygon of P_{in} in P_{out}-which is the entire polygon P_{out} itself. Then it chooses a point-without loss of generality, assume that the point chosen is x shown in figure.	In Step 4, it finds the polygon MT(x). In this case it is given by x, x_1, x_2, x_3. Hence MT(x) has four sides.
Figure 5	P_{in}: A, B, … C P_{out}: 1, 2, … 17	P_{in} is convex. P_{out} is non-convex. Hence $CH(P_{in})$ is not contained in P_{out}. The complete visibility polygon of P_{in} in P_{out} is given by the ordered set of vertices: 4, 5, 6, 10, 11, 15, 16, 17.	In Step 1, the algorithm finds that P_{in} is convex. So it moves to Step 2.	In Step 2.1 it finds the complete visibility polygon of P_{in} in P_{out}. Then it chooses a point-without loss of generality, assume that the point chosen is x shown in the figure.	In Step 4, it finds the polygon MT(x). In this case it is given by x, x_1, x_2. Hence MT(x) has three sides (which by definition is optimal).
Figure 6	P_{in}: A, B, … H P_{out}: 1, 2, … 12	P_{in} is non-convex. P_{out} is also non-convex. Furthermore, $CH(P_{in})$ is not contained in P_{out}- for e.g., the line segment [B, D] forms a lid on $CH(P_{in})$ that intersects with P_{out}.	In Step 3, the algorithm finds that P_{in} is non-convex.	In Step 3.1 it finds out that $CH(P_{in})$ is not contained in P_{out}. Hence it moves to Step 3.3 In Step 3.3.1, it chooses a point x that defines a lid in $CH(P_{in})$. Without loss of generality, assume that it chooses the point x in the figure (which is the same as vertex B of P_{in}).	In Step 4, it finds the polygon MT(x). In this case it is given by $x, x_1, x_2, \ldots, x_7$.. Hence MT(x) has eight sides.

2. This problem is related to the Minimal Turn Path problem by defining the Minimal Turns Polygon of a point x, denoted by MT(x).
3. Regardless of the shape of *Pin* and $_{Po}$ut, there exists a point x in the annulus between the two polygons for which MT(x) can be used as an approximation for *P** as it has the same property of convexity as *P** and has at most two more edges.
4. An algorithm has been presented that uses the property in 3 above to find such a point x and give an approximate *P** in time that is linear in the input.

There are a lot of open problems in this area. For instance, an interesting open problem is to consider the generalization of Lemmas 4 and 3 to any point that is in the annulus between P_{in} and P_{out}, when P_{in} and P_{out} have arbitrary shapes. It may be possible to relate the number of edges in P^* to the number of edges in the Minimal Turns Polygon of any point in the annulus. Another interesting open problem is to partition the annulus between Pin and Pout into disjoint regions such that all points within the same region have the same number of edges in their Minimal Turns Polygons. This problem has been considered in Bhadury and Chandrasekaran3 in the special case when Pm and Pout are both convex. However no result exists about the general case when these two polygons are only required to be simple.

Acknowledqement

The first author was supported by NSERC Grant #OGP 0I2 I689 and UNB Grant #26-7. The second author was supported, in part, by The Morris Hite Center at The University of Texas at Dallas. This support is gratefuIly acknowledged.

References

1. Aggarwal, A., Booth, H., O'Rourke, J., Suri, S. and Yap, C. K. (1989) Finding minimal convex nested polygons. *Information and Computation,* 83, (1), 98-110.
2. Bhadury, J. and Chandrasekaran, R. (1994) Stock cutting to minimize cutting 1ength. *European Journal of Operational Research* (to appear).
3. Bhadury, J. and Chandrasekaran, R. (1993) Finding the set of all minimal nested convex polygons. *Working Paper,* University of New Brunswick, Fredericton, Canada.
4. Chandru, V., Ghosh, S. K., Maheshwari, A., Rajan, V. T. and Saluja, S. (1991) NC-algorithms for minimum link path and related problems. *Technical Report No. CC-91-4,* Institute for Interdisciplinary Engineering Studies, Purdue University.
5. Christofides, N. and Whitlock, C. (1977) An algorithm for two dimensional cutting problems. *Operations Research,* 25, 30--44.
6. Dori, B. and Ben-Assat, M. (1983) Circumscribing a convex polygon by a polygon of fewer sides with minimum area addition. *Computer Vision, Graphics and Image Processing,* 24, 131-159.
7. Ghosh, S. K. (1991) Computing the visibility polygon from a convex set and related problems. *Journal of Algorithms,* 12, 75-95.
8. Ghosh, S. K., Maheshwari, A., Pal, S. P., Saluja, S. and Vent Madhavan, C. E. (1993) Characterizing weak visibility polygons and related problems. *Computational Geometry: Theory and Applications,* 3 (4), 213-233.
9. Ghosh, S. K. and Maheshwari, A. (1990) Optimal algorithm for computing minima1 nested non-convex *polygons. Information Processing Letters,* 36, 277-280.
10. Gilmore, P. C. and Gomory, R. E. (1961) A linear programming approach to the cutting stock problem. *Operations Research,* 9, (6), 849 859.
11. Gilmore, P. C. and Gomory, R. E. (1963) A 1inear programming approach to the cutting stock Problem-Part II. *Operations Research,11,* (6), 863-888.
12. Preparata, F. P. and Shamos, M. I. (1985) *Computational Geometry: An Introduction.* Springer-Verlag.
13. Reif, J. H. and Storer, J. A. (1987) Minimizing turns for discrete movement in the interior of a polygon. *IEEE Journal on Robotics and* Automation, RA-3, 182-193.
14. Suri, S. (1986) A 1inear time algorithm for minimum link paths inside a simple polygon. *Computer Vision, Graphics and Image Processing,* 35, 99-110.
15. Venkateswarlu, P. and Martyn, C. W. (1992) The trim-loss problem in a wooden drum industry. *OR-92: Proceedings of the Convention of Operational Research Society of India.*

APPENDIX

This appendix contains the proofs of some of the lemmas in the paper.

Proof of lemma 1: By definition every vertex of P_{in} is inside P^*. But CH(P_{in}) is the smallest convex set that contains P_{in}. Therefore when P^* is convex, CH(P_{in}) is also inside P^* and therefore, inside Pout. To prove the converse, it needs to be shown that if CH(P_{in}) is contained in P_{out}, P^* is convex.

Suppose not. Then CH(P_{in}) is contained in P_{out} but P^* is non-convex. Then choose a reflex vertex (one whose internal angle is more than 180°) of P^*, say p. Identify the vertex of CH(P_{in}) that is closest to p and call it x. Condition on two possible cases: Case (1): p is outside CH(P_{in}) and Case (2): p is inside CH(P_{in}).

Case (1): This is shown in Figure 7. Note that from x it is possible to draw both the anti clockwise and the clockwise tangents to CH (P_{in}) (in fact these tangents in this case will be edges of CH(P_{in})) and so all the points

x, $z_c(x)$, $z_a(x)$, x_c, x_a exist as shown in the figure. Define $p - 1$ and $p + 1$ as the previous and successive vertices of p and P^*. It is clear that P^* intersects both $[x, x_c]$ and $[x, x_a]$ and let these points be denoted by x' and x'' respectively. Proceeding in a clockwise direction on the boundary of P^* let p, be the first vertex of P^* that occurs prior to the intersection of P^* with $[x, x_a]$ and let p_r be the first vertex of P^* that occurs after its intersection with $[x,x_c]$. Because p is reflex and lies outside CH (P_{in}), at least one of p_i and pr is distinct from p — *1* and p + *1* respectively. By replacing the section of P^* between x' and x'' with two edges $[x', x]$ and $[x, x'']$, a new polygon is obtained, call this the new P^*.

When p_i, p_r, $p - 1$, $p + 1$ are all distinct, the number of edges between p_l and p_r in P^* is at least four and the new P^* has as many edges between them. Now assume that two of these are coincident, without loss, p_i and $p - 1$ as shown in Figure 8. Then it is clear that $p + 1$ and p_r are distinct—as shown in the figure. In this case P^* has at least three edges between p, and pr and the new P^* has at most as many. Therefore, this new P^* has at least as many edges between p_l and p_r as the given P^* and furthermore, this new P^* has one less reflex vertex.

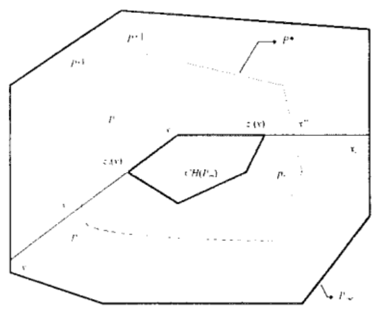

Figure 7

A similar proof holds when p lies on the boundary of CH(P_{in}). Thus the only remaining possibility in Case (2), i.e. the case where p is inside CH (P_{in}).

Case (2): Suppose p is inside CH (P_{in}). Then CH (P_{in}) has pockets and lids and clearly p is inside a pocket of CH (P_{in}) as shown in Figure 9. Let A and B denote the two vertices of P_{in} that define this pocket—i.e. let the line segment $[A, B_c]$ be the lid of this pocket. Because A is on CH(P_{in}), it is possible to draw the two tangents from A to P_{in} (in fact here $z_a(A) = A$ and $z_c(A) = B$, as shown in the figure) and hence, define Ac and A_n similarly. As before, it is evident that P^* intersects both line segments $[A, A_a]$ and $[B, A_c]$ and let the points of intersection be denoted by A' and A'' respectively. Define p_l, p_r, $p - 1$ and $p + 1$ as before. In this case, the section of P^* between A' and A'' will be replaced by the line segments $[A', A]$ and $[A, A'']$, to get the new P^*. Because p is inside CH(P_{in}) all of the points p_i, $p - 1$, $p + 1$ and p_r are distinct. Hence P^* has at least four edges between p_l and p_r, and the new P^* has at most as many. ■

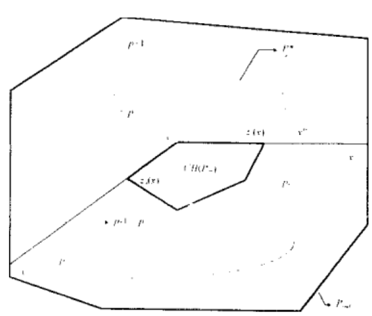

Figure 8

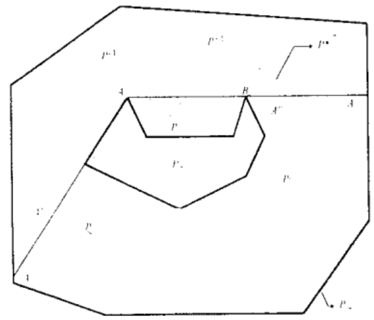

Figure 9

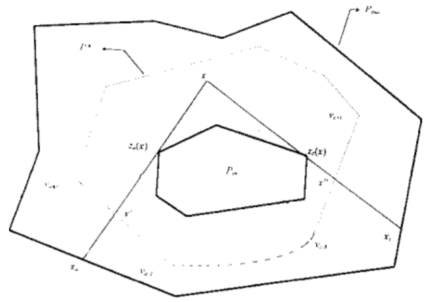

Figure 10

Proof of Lemma 3: The proof is by construction. As P_{in} is convex and x is in the complete visibility polygon of P_{in} in P_{out}, it is possible to draw the two clockwise and anti-clockwise tangents from x to P_{in}. Consider the line segments $[x_a, x]$ and $[x, x_c]$ as shown in Figure 10. It is clear that the Minimal Nested Polygon P^*, will intersect the line segments $[x_a, x]$ and $[x, x_c]$ and let these points of intersection be denoted by x' and x'' respectively. Let the edges of P^* that intersect $[x_a, x]$ and $[x, x_c]$ be $[v_{a+1}, v_{a-1}]$ and $[v_{c+1}, v_{c-1}]$ respectively. Consider the polygon that is obtained by retaining all the edges of P^* between v_{a-1} and v_{c-1} and adding to it the edges (v_{a-1}, x'), (x', x), (x, x'') and (x'', v_{c-1}) in that order. This new polygon thus obtained is certainly a candidate for being a path from x to itself that goes around P_{in}. Furthermore, this new polygon has at most two more edges than P^* because P^* has at least two edges between v_{a-1} and v_{c-1} and this new polygon has four. Since MT(x) has at most as many edges as this new polygon, the lemma follows. ■

Proof of Lemma 4: If CH (P_{in}) is contained in P_{out}, then by Corollary 2, Pin can be rep]aced by CH(P_{in}), without changing the optimal solution to the problem. Doing that, one obtains a P_{in} that is convex and therefore, Lemma 3 becomes applicable. That proves the first statement of this lemma.

To prove the second statement—consider the case where CH(P_{in}) is not inside P_{out} However it is clear that since P_{in} is contained inside P_{out}, there must exist some point on the boundary of CH (P_{in}) that is inside (for example all the vertices of P_{in} that define the lids of CH (P_{in}) will be on CH (P_{in}) and inside P_{out}). Choose some such point as x and repeat the arguments of Lemma 3 above, a similar proof by construction will show that for this point x, MT(x) can have at most two more edges than P^*. ■

Proof of Lemma 5: Since P^* is convex, it can be assumed that CH(P_{in}) is completely contained in P_{out}. Then, assuming that x is a point with the above mentioned properties and MT(x) is non-convex (and so has at least one reflex vertex), a proof similar to the one used in Lemma 1 can show that the reflex vertices of MT(x) can be replaced with non-reflex vertices without increasing the number of sides in MT(x). ■