An Efficient Algorithm for Touring *n* Circles

Yuepeng Ding[1], Xiong Xie[1] and Bo Jiang[1,a]

[1]Dalian Maritime University ,Linghai Road 1,Dalian,China

Abstract. This paper proposed an intelligent algorithm, which can build the shortest path of the intersecting circle sequences in the plane. The problem is transformed to a corresponding problem of computing the shortest path of three disjoint circles. And it is also an in-depth study of the traversal problem for the disjoint circle sequences. On the basis of the previous work, the algorithms are developed to construct such shortest path in polynomial time of O(kn) where $1 \leq k \leq n$ under a given computational threshold. Since the algorithm is fully polynomial time consuming, this work can be conducted in layered manufacturing of rapid prototyping, displacement of wireless sensor networks, or other related computer-aided design and manufacturing applications.

1 Introduction

The study on such problems can be applied to layered manufacturing (LM) [1], sensor networks and other related computer-aided design and manufacturing. For example, in the layered manufacturing technology, the required geometry in model is cut with a laser beam on a thin material, and all cut geometric shapes are superimposed to obtain a 3D physical shape. In the production process of layered manufacturing, if we control the laser beams cut each geometric shape along the shortest traversal path, we will improve the productivity. In that case, we assume that the geometry in LM is circle, then the main problem in this paper is to find the shortest path of circles.

This touring problem, to find the shortest path of circular sequences in the plane, is an extension of the TSP problem [2]. The TSP problem is a classic combinatorial optimization problem and it can be described as follow. Travelers need to visit n cities, start from one city, visit each city only one time then return to the starting city, and the access path should be the shortest. When n cities are unordered in the TSP problem, this problem will be NP-hard [3]. Otherwise, the TSP problem can be solved optimally in linear time. Furthermore, if the access targets are abstracted as the circles, then it will be the main problem in this paper, which can be described as follows. The n circles are numbered from 1 to n, i.e., O_1; . . . ; O_n. In these circles, some of them may intersect each other, and the other may not. We need to choose one point on the border of each circle, which can yield the shortest traversal path of all the circles.

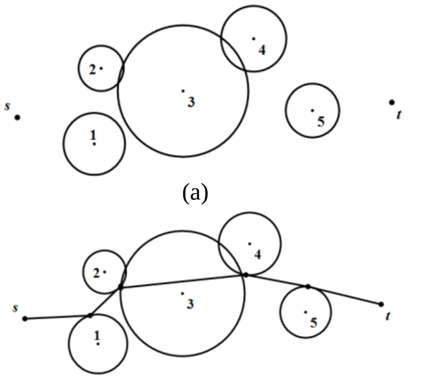
In 2008, Chou et al. proposed a function [4], which can find the shortest path between two points and a circle, by solving the equations. According to the equation, a dynamic point p on the boundary of the circle O can be deduced, which can build the shortest path between two fixed points and the circle O. This function is referred to the "point-circle-point" method, denoted as PCP.

Chou et al combined the PCP method in 2010 and proposed an algorithm, which can find the shortest path of disjoint-circle circular sequences in $O(kn)$, where $1 \leq k \leq n$ is related to the threshold given in the article, called Odd-Even[5].

Chou et al. combined PCP and Odd-Even algorithm in 2012 to design an new algorithm, named DBB (Dominant-Branch-and-Bound) [6], which can build the shortest path of disjoint unordered circles in O($kn*e^n$), where $1 \leq k \leq n$ is related to the threshold given in the article. The DBB algorithm mainly solves the problem of touring unordered circles in order, and it reduces the time complexity from the original $O(kn*n!)$ to $O(kn*e^n)$ by improving the branch-and-bound method.

In summary, the touring problems of disjoint circles, either ordered or unordered, have been solved in previous works. To the author's knowledge, there is no discussion about the problems of joint circles. Therefore, we analyzes this problem and designs an algorithm which can solve the traversing problem of circles in this paper, specifying that the circles are ordered and we give the start and end points to tour these circles. As shown in Figure.1 (a) and (b).

*Corresponding author:[a] Bo Jiang: bojiang@dlmu.edu.cn



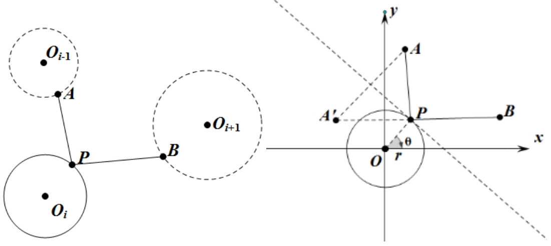
(a)

(b)

Figure 1. An example of the touring problem.

2 The PCP method

This paper will use the PCP method to solve the optimal access point on the circle O_i, so that we can compute the shortest path of three circles, called the partial shortest path, as shown in Figure. 2(a). We use this method to find the partial shortest path between O_{i-1}, O_i and O_{i+1}, then the path of n circles can be optimized and shorten by computing the partial shortest path, until the shortest traversal path be solved. Therefore, PCP is a basic method for this paper. The PCP is shown in Figure. 2(b).



(a) (b)

Figure 2. The reflection path of two points and one circle.

The shortest path of the two interconnecting links $\left|\overline{Ap}+\overline{Bp}\right|$ with length L can be denoted by

$$L = min\{l(\theta_1), l(\theta_2), l(\theta_3), l(\theta_4), l(\theta_5), l(\theta_6), l(\theta_7), l(\theta_8)\} \quad (1)$$

Where

$$l(\theta) = \sqrt{(x_A\text{-}rcos\theta)^2+(y_A\text{-}rsin\theta)^2}+\sqrt{(x_B\text{-}rcos\theta)^2+(y_B\text{-}rsin\theta)^2}$$

$$\theta = \begin{cases} \theta_1 = \text{-}cos^{\text{-}1}(\frac{1}{4}k\text{-}n\text{-}\frac{1}{2}\sqrt{q\text{-}p}) \\ \theta_2 = cos^{\text{-}1}(\frac{1}{4}k\text{-}n\text{-}\frac{1}{2}\sqrt{q\text{-}p}) \\ \theta_3 = \text{-}cos^{\text{-}1}(\frac{1}{4}k\text{-}n+\frac{1}{2}\sqrt{q\text{-}p}) \\ \theta_4 = cos^{\text{-}1}(\frac{1}{4}k\text{-}n+\frac{1}{2}\sqrt{q\text{-}p}) \\ \theta_5 = \text{-}cos^{\text{-}1}(\frac{1}{4}k+n\text{-}\frac{1}{2}\sqrt{q+p}) \\ \theta_6 = cos^{\text{-}1}(\frac{1}{4}k+n\text{-}\frac{1}{2}\sqrt{q+p}) \\ \theta_7 = \text{-}\cos^{\text{-}1}(\frac{1}{4}k+n+\frac{1}{2}\sqrt{q+p}) \\ \theta_8 = cos^{\text{-}1}(\frac{1}{4}k+n+\frac{1}{2}\sqrt{q+p}) \end{cases}$$

Where

$$k = \frac{r(ax_B+bx_A)}{ab},$$

$$n = \frac{1}{2}\sqrt{\frac{l}{2}+\frac{1}{j}(\frac{g}{h}+h)},$$

$$p = \frac{m}{8a^3b^3n}$$

$$q = l-\frac{1}{j}(\frac{g}{h}+h),$$

$$a = x_A^2+y_A^2,$$

$$b = x_B^2+y_B^2,$$

$$c = 2r^2x_Ax_B+r^2y_B^2+x_B^2\left(r^2-4y_A^2\right)+2r^2y_Ay_B+r2y_B^2-4y_A^2y_B^2+x_A^2\left(r^2-4b\right),$$

$$d = 2x_A^2x_B+x_By_A\left(y_A-y_B\right)+x_A\left(2x_B^2-y_B\left(y_A-y_B\right)\right),$$

$$e = x_B^2(-r^2+y_A^2)-2x_Ax_B\left(r^2-y_Ay_B\right)+x_A^2\left(-r^2+y^2{}_B\right),$$

$$f = 36r^2\left(6e\left(ax_B+bx_A\right)^2+cd\left(ax_B+bx_A\right)+6abd^2\right),$$

$$h = \frac{1}{\sqrt[3]{2}}\sqrt{2c^3-288abce+2f+\sqrt{-4g^3+4(c^3-144abce+f)^2}},$$

$$g = c^2+48abe+24r^2d\left(ax_B+bx_A\right),$$

$$j = 12ab,$$

$$m = r\left(r^2\left(ax_B+bx_A\right)^3-abc\left(ax_B+bx_A\right)-4a^2b^2d\right),$$

From the above derivation, an important lemma is given.

Lemma 1. The Euclidean shortest path in the two-dimensional plane, from one point outside the circle via one point on the border of the circle to one point outside the circle, can be computed using *Eq*. (1).

The proofs of the convergence and correctness of Lemma 1 can be found in [4].

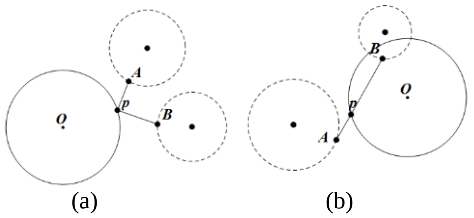


Figure 3. A counter-example for Odd-Even.

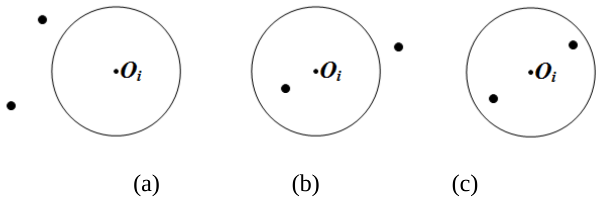
In Odd-Even, after the pre-processing mentioned in this paper, the solution of the partial shortest path is only the case of reflection, and the reflection case mean that the partial shortest path of three circles is formed by the reflection path of light, which can be solved by PCP, as shown in Figure. 3(a). However, when the circles can be intersected, there will be a situation that cannot keep with the reflection situation, as shown in Figure. 3 (b). There are two points *A* and *B*, and *A* is outside the circle *O*. *B* is inside the circle *O*. In this situation, we cannot apply the solution as the reflection situation. And it is easy to find that if we connect the two points *A* and *B* directly, then the intersection *p* between the line segment $\overline{AB}$ and the circle *O* will be the partial shortest path point. This case will be referred to as the crossing situation. Therefore, when the circles can be intersected, there are two ways to solve the partial shortest paths, such as the reflection situation and the crossing situation. So we will analyze the condition of the two situations and design corresponding traversal algorithms.

3 Results and discussion

3.1 Geometric analysis

Theorem 1. There are only the reflection situation and the crossing situation in solving the partial shortest path of the circle sequences, when the circles allow to be intersected.

Proof. This paper allows circles to intersect. The positional relationship of circles is: (a) disjoint, (b) tangent, (c) intersected, (d) contained. Under these four kinds of relationships, we use three circles to analyze the partial shortest path of a circle sequence. Therefore, we divides the partial area of the circle sequence into the outer area of the circle O_i, the inner area of the circle O_i, and the border of the circle O_i. During these divisions, we analyze all the possible positions of the circle O_i, the point p_{i-1} and the point p_{i+1} in the three areas. As shown in the Fig. 4. Since the point p_{i-1} is symmetric to the point p_{i+1}, they can be interchanged without affecting the solution of the partial shortest path, which is to say one point can be the point p_{i-1}, and it also can be the point p_{i+1}.



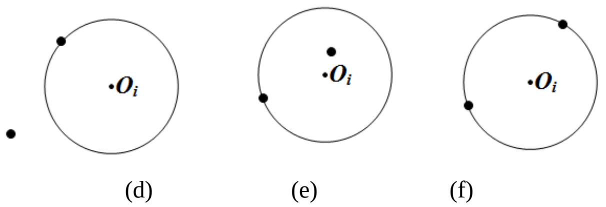


Figure 4. All positions of O_i, p_{i-1} and p_{i+1}.

We discuss the solutions of the partial shortest path based on the positions of the circle O_i and the point p_{i-1} and the point p_{i+1}. During the Figure. 4(a), we can obtain two possible solutions to the partial shortest path under (a) position, as shown in Figure. 5(a) and (b), which are in accord with the reflection situation and the crossing situation.

For the Figure. 4(b), either the point p_{i-1} or the point p_{i+1} is inner the circle O_i, it can be seen that the solution of the partial shortest path only satisfy the crossing situation. As shown in Figure. 5(c).

For the Figure.4(c), the point p_{i-1} and the point p_{i-1} are all inner the circle O_i, and the access path needs to access the circle first, so the solution of the partial shortest path only satisfy the reflection situation. As shown in Figure. 5(d).

For the Figure.4(d)~(f), if one point p_{i-1} or p_{i+1} is on circle O_i, then the access path must pass circle O_i from the point p_{i-1} to the point p_{i+1}, so the solution of the partial shortest path only satisfy the crossing situation. As shown in Figure. 5(e).

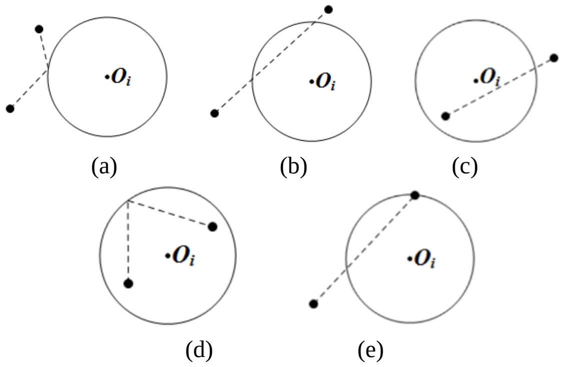


Figure 5. Solutions of all the positions.

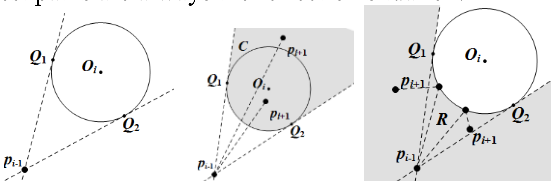
To sum up, all the positions of the circle O_i, the point p_{i-1} and the point p_{i+1} have only the reflection situation or the crossing situation, so Theorem 1 is proved.

3.2 The division of reflection and crossing

According to Theorem 1, we can sure that the solution of the partial shortest path contains only two situations, the reflection and the crossing. According to the division of convex polygons and related problems [7, 8], we divide the plane into two regions, called reflection region and crossing region, denoted by *R* and *C*. According to the position relationship between circle O_i， the point p_{i-1} and the point p_{i+1}, we divide the solution of the shortest path of the circle sequence into two cases.

In the first case, the point p_{i-1} is outside the circle O_i. For this case, we can draw two tangent lines crossed p_{i-1} and touched circle O_i. The tangent points are denoted by

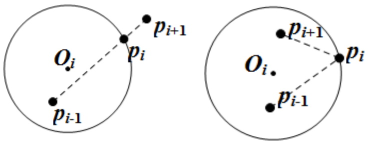
Q_1 and Q_2, as shown in Figure. 6(a). We use C denote the inward area that enclosed by the circle O_i and the tangent lines, and the region of C contains the circle boundary. As can be seen from Figure. 6(b), wherever the point p_{i+1} is at any position in the region of C, the partial shortest paths are all belong to the crossing situation. We use R denote the periphery area of the circle and tangent. As shown in Figure. 6(c), wherever the point p_{i+1} is located at any position of R, the partial shortest paths are always the reflection situation.



(a) (b) (c)

Figure 6. Division of the regions R and C.

In the second case, when the point p_{i-1} is inner the circle. We can see that as long as the point p_{i+1} is outside the circle, there is only the crossing situation, and when the point p_{i+1} is inner the circle, there is only the reflection situation. As shown in Figure.7 (a) and (b).



(a) (b)

Figure 7. Cases of p_{i+1} inner the circle.

3.3 Algorithm

We assume that a set of circle sequences $O = \{O_1, O_2; . . . ; O_5\}$, a starting point s, and an ending point t are on the plane.

Step.1. We initialize an access path for this circle sequences, and denote the length of the path as L_0, then record the initialized path points $\rho = \{p_1, p_2; . . . ; p_5\}$, as shown in Figure. 8(a).

Step.2. After fixing the contact point on each even circle, then we can divide the region R and C for all odd circles so that the solution method (reflection or crossing) can be determined, and finally the partial shortest path point on each odd-numbered circle can be solved by two methods. As shown in Figure. 8(b).

Step.3. After fixing the contact point on each odd circle, then we can divide the region R and C for all even circles so that the solution method (reflection or crossing) can be determined, and finally the partial shortest path point on each even-numbered circle can be solved by two methods. As shown in Figure. 8(c).

Step.4. We calculate the absolute difference between the lengths of two adjacent paths, denoted by d.

Step.5. If $d > \varepsilon$ repeat steps 2~4. If $d \le \varepsilon$ (ε is the maximum numerical accuracy of the given computer), stop iterative calculation and output the sequence of path point ρ. The algorithm ends. As shown in Figure. 8(d).

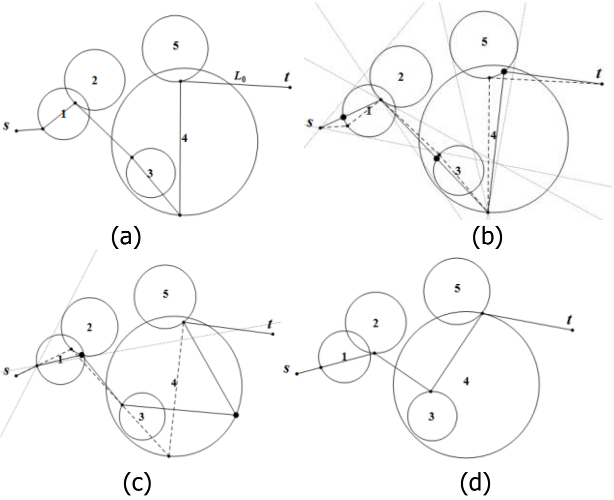


Figure 8. Four steps of Algorithm 2.

With a set of circle sequences $O = \{O_1, O_2; . . . ; O_n\}$, the algorithm for traversing O consists of three parts: preprocessing, main algorithm, and partial shortest path point solution. In order to simplify the description, we separate the solution of the partial shortest path point into an algorithm, denoted as *GetoptPi*.

Algorithm 1 *GetoptPi*

Input: p_{i-1}, p_{i+1}, O_i。

Output: Returns the shortest path point p_i on the circle O_i.

```
if ($p_{i-1}$ is outside circle $O_i$ and not on circle $O_i$) then
    After calculating the two tangents of circle $O_i$ past point $p_{i-1}$, then we use the tangents to divide the regions $C$ and the regions $R$, as shown in Figure. 6 (b) and (c);
    if($p_{i-1}$ in the regions $C$)then
        $p_i$ = The first point of intersection of line $p_{i-1}p_{i+1}$ and circle $O_i$;
    else $p_i$ = PCP($p_{i-1}$, $p_{i+1}$, $O_i$);
else
    if($p_{i+1}$ is outside circle $O_i$ and not on circle $O_i$) then
        $p_i$ = PCP($p_{i-1}$, $p_{i+1}$, $O_i$);
    else $p_i$ = The first point of intersection of line $p_{i-1}p_{i+1}$ and circle $O_i$;
return $p_i$;
```

Algorithm 2

Input: The circle sequence $O = \{O_1, O_2; . . . ; O_n\}$, the starting point s, the ending point t, and the number of circles n.

Output: The shortest traversal path ρ from point s to point t, where $\rho = \{s, p_1, p_2, ..., p_n, t\}$.

```
$\varepsilon = 10^{-10}$, where $\varepsilon > 0$ is the precision;
We should calculate the set of initial path points of all circles in the circle sequence, denoted as $\rho$, and the path length of $\rho$ is denoted as $L_0$;
$j$ = 1, where $j$ is the number of loop times;
$i$ = 1, where $i$ is the number of circles;
do{
while ($i < n + 1$){
After fixing the contact point on each even circle then we apply GetoptPi ($p_{i-1}$, $p_{i+1}$, $O_i$) to update $p_i$ in $\rho$ with
```

```
        parameters on each odd circle;
8       $i = i + 2$; }
9   We calculate the new path length when i is odd, denoted as $L_j^o$;
10  i=2;
11  while ($i < n + 1$){
12      After fixing the contact point on each odd circle then we apply GetoptPi ($p_{i-1}$, $p_{i+1}$, $O_i$) to update $p_i$ in $\rho$ with parameters on each even circle;
13      i = i + 2; }
14  Calculate the new path length when i is even, denoted as $L_j^e$;
15  $j$++;
16  }until $L_j^e - L_j^o \leq \varepsilon$;
17  Output $\rho$;
```

The proofs of the convergence and correctness of Algorithm 2 are given below.

3.4 Algorithm feasibility and time complexity analysis

Theorem 2. Algorithm 2 can yield the shortest path of n circles, which allows circles to intersect each other.

Proof. Light traverses along the least-time path, so a shortest *n-gon* inside another convex *n-gon* can be shown to follow the light fully reflection [5, 11]. The algorithm starts from the starting point s, then it finds the shortest path point on the circle sequence by using solution of the reflection situation or the crossing situation. Finally, it reaches the end point t. For the reflection situation, it satisfies the total reflection of light. For the crossing situation, it also satisfies the linear propagation of light. Therefore, the path ultimately sought by this algorithm satisfies the light propagation path, and it is the shortest path.

Theorem 3. Algorithm 2 is convergent in polynomial time of $O(kn)$, where $1 \leq k \leq n$ under a given computational threshold.

Proof. The initial length L_0 is determined by selecting arbitrary points on each circle. Clearly $L_j^e \leq L_j^o$ holds during the ‘‘Do until” in steps. The convergence criterion now depends on j, and j depends on ε. If ε is specified, then j is bounded. Since $L_j^e \leq L_j^o$ holds in every iteration, and unlimited computing precision is not required, one $j = \omega$ exists in which the computational threshold is reached, i.e.

$$\exists\, j = \omega, \text{ s.t. } L_\omega^e - L_\omega^o \leq \varepsilon.$$

Because for a sequence of n circles, calculating at least one shortest path requires $O(n)$, so $k \geq 1$. It may also require multiple iterations, and each iteration requires $O(n)$, so the total time complexity is $O(\omega n)$，let $O(\omega n) = O(kn)$，since unlimited precision is not required, a computational threshold can be specified to end the process before $O(n^2)$ [5, 9, 10], meaning that $k \leq n$. Therefore, the algorithm is convergent at $O(kn) \leq O(n^2)$, where $1 \leq k \leq n$ under a given computational threshold.

4 Conclusion

The problem of layered manufacturing (LM) is introduced in this work. The studied problem can be simplified as computing the shortest n-gon connecting n circles. Although the shortest path of n disjoint ordered circles can be solved by Algorithm Odd–Even, no algorithm provides the exact steps to build the shortest path of n circles, which allow circles to intersect each other. This work proposed an efficient algorithm to conquer the problem. Algorithm 2 provides the base model to steadily construct the shortest path of n circles in polynomial time of $O(kn)$, where $1 \leq k \leq n$ under a given computational threshold.

This work does not discuss the case where the n circles are the shortest paths in the unordered case, and the time complexity in the unordered case increases by at least $n!$, which is one of the future works. The other future work will consider optimizing the traversal algorithm proposed in this paper, and find a method that can be more concise and universally applicable than Lemma1 for solving the partial shortest path points.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under grant 61702242 and Liaoning Province Natural Science Foundation under grant 20170540147.

References

1. K. Tang and A. Pang. Optimal connection of loops in laminated object manufacturing, Computer-Aided Design, 2003(35): 1011-1022.
2. Dror, M., Efrat, A., Lubiw, A., Mitchell, J.S.B.: Touring a sequence of polygons. In: Proceedings of the STOC 2003, pp. 473–482 (2003)
3. Fellows M, Tan X, Zhu B. Frontiers in Algorithmics and Algorithmic Aspects in Information and Management: Third Joint International Conference, FAW-AAIM 2013, Dalian,China. Computer Science and General Issues)[M]. Springer Publishing Company, Incorporated, 2013.
4. Chou S Y, Chou C C, Chen Y K. A base function for generating contour traversal paths in stereolithography apparatus applications[J]. Expert Systems with Applications, 2008, 35(1–2):235-244.
5. Chou C C. An efficient algorithm for relay placement in a ring sensor networks[M]. Pergamon Press, Inc. 2010.
6. Chou C C. On the Shortest Path Touring n Circles[J]. International Journal of Advancements in Computing Technology, 2012, 4(10):356-364.
7. Tan X, Jiang B. An improved algorithm for computing a shortest watchman route for lines[J]. Information Processing Letters, 2018, 131:51-54.
8. Tan X, Jiang B. The Touring Polygons Problem Revisited[J]. Ieice Transactions on Fundamentals of

Electronics Communications & Computer Sciences, 2018, 101.

9. Tan X, Jiang B. Efficient Algorithms for Touring a Sequence of Convex Polygons and Related Problems[C]// International Conference on Theory and Applications of Models of Computation. Springer, Cham, 2017:614-627.

10. Ahadi A, Mozafari A, Zarei A. Touring a sequence of disjoint polygons: Complexity and extension[J]. Theoretical Computer Science, 2014, 556:45-54.

11. Liu, T. L., Chuang, H. Z., & Wang, W. H. Shortest circumferential inscribed convex n-gon inside a convex n-gon. Research papers from Senior High School version, 29th Taiwan National Primary and High School Science Fair (in Chinese), 1989.