アルゴリズム 22－3
（1991. 7. 22）

最短ウォッチマン経路を構成するインクリメンタルアルゴリズム

譚 学厚　平田 富夫　稲垣 康善

名古屋大学　工学部

単純な多角形Pの辺上に始点sがある場合の最短ウォッチマン経路を見つける問題を調べる．Pのウォッチマン経路とは，その経路をたどればPの内部の点がすべて見えるような経路のことである．本研究では，n 頂点を持つ単純な多角形に対し，$O(n^3)$ の時間で最短ウォッチマン経路を見つけるインクリメンタルアルゴリズムを提案する．これは以前の $O(n^4)$ 時間のアルゴリズムを改良する．

An incremental algorithm for constructing shortest watchman routes

Xue-Hou TAN, Tomio HIRATA and Yasuyoshi INAGAKI

Faculty of Engineering, Nagoya University, Chikusa-ku, Nagoya 464, Japan

The problem of finding the shortest watchman route in a simple polygon P through a point s on its bounday is considered. A route is a watchman route if every point inside P can be seen from at least one point along the route. We present an incremental algorithm that constructs the shortest watchman route in $O(n^3)$ time for a simple polygon with n edges. This improves the previous $O(n^4)$ bound.

(Extended Abstract)

1 Introduction

The **watchman route problem**, posed by Chin and Ntafos [2], deals with finding the shortest route from a point x back to itself in the given polygon so that every point in the polygon can be seen from at least one point along the route. In [2], Chin and Ntafos gave an $O(n)$ algorithm for the watchman route problem in a simple rectilinear polygon with n edges. They also presented an $O(n^4)$ algorithm for constructing the shortest watchman route in a simple polygon through a starting point s specified on its boundary [3]. In this paper, we present an $O(n^3)$ algorithm for this problem. Both algorithms for simple polygons construct the optimum path by repeatedly adjusting the current one. While Chin and Ntafos' algorithm always adjusts the whole watchman path, our attention is put on the partial watchman path, i.e., our algorithm starts from a "small" watchman path and then increases its visibility incrementally until it becomes the whole watchman path. This incremental method gives an efficient solution to the watchman route problem for simple polygons.

2 Preliminaries

Let P be an n-sided simple polygon with a point s on its boundary. We assume that P is given by the sequence of its vertices in the clockwise order (from s). A vertex is *reflex* if the internal angle is greater than 180^0. P can be partitioned into two pieces by a "cut" that starts at a reflex vertex v and extends either edge incident to v until it first intersects the boundary. We say a cut is a *visibility cut* if it produces a *convex* angle ($< 180^0$) at v in the piece of P containing s. (Some reflex vertices may not contribute to any visibility cut.) Such a visibility cut "resolves" the reflexivity at v; in order to see the edge (or corner) incident to it, a watchman route needs to visit only one point on that cut. Since the shortest path between s and a cut C need not go over C (as viewed from s), the piece of P containing s is called the **essential piece** of C. A cut C is described by a pair of points (l, r), where l (left endpoint) is the endpoint of C that is first visited in a clockwise scan of P and r (right endpoint) is the other endpoint. The orientation of a cut C is supposed to be from l to r. Thus, s always falls to the right side of visibility cuts.

A watchman route must visit all visibility cuts so that each corner of P can be seen. But some of them are not important in determining the shortest watchman route. We say cut C_j **dominates** cut C_i if C_j appears between two endpoints of C_i in a clockwise scan of the boundary. Clearly, if C_j dominates C_i, any route that visits C_j will automatically visit C_i, i.e., C_i can be disregarded in determining the shortest watchman route. A cut is called an **essential cut** if it is not dominated by any other cuts. It is important to observe that any watchman route must visit these essential cuts and any route that visits them is a watchman route.

The essential cuts can be identified in $O(n)$ time by applying the clockwise scanning scheme. Let $C_1, C_2, \cdots, C_m$ be the sequence of essential cuts indexed in the clockwise order of their left endpoints. The set of essential cuts is then partitioned into cut corners. A **cut corner** is a subset of consecutive essential cuts $C_i, C_{i+1}, \cdots, C_j$ such that each C_k intersects with C_{k-1} ($i < k \leq j$), and C_i and C_j do not intersect with C_{i-1} and C_{j+1}, respectively. This partition of essential cuts into cut corners takes $O(n)$ time.

Lemma 1 *(Chin and Ntafos[3]) The shortest watchman route in a simple polygon P through a point s on its boundary is unique.* □

In fact, Lemma 1 can be stated in a more general form: Given a source point s on P's boundary and a target point t on cut C_i, the shortest watchman path from s to t such that (1) $C_1, C_2, \cdots, C_{i-1}$ are visited and (2) it lies in the essential piece of C_i is unique.

Chin and Ntafos [3] also show that the shortest watchman route should visit cut corners in the order in which they appear in the boundary of P and forms a convex chain within each cut corner.

3 Incremental construction of shortest watchman routes

In a cut corner, an essential cut is intersected with at most $m-1$ essential cuts and thus divided into at most $m-1$ **fragments.** We say fragment f **dominates** fragment g if any route that visits f also visits the cut to which g belongs. Each fragment thus has a unique set of essential cuts dominated by it. (Note that two fragments may dominate each other.) A watchman route can then be defined by a set of the fragments such that (i) (completeness) the union of dominances of the fragments is the whole set of essential cuts and (ii) (independence) no one is dominated by any other fragments. Visiting these fragments in any order will give a watchman route. Such a fragment set is called the **watchman fragment set.** With respect to a watchman fragment set, we distinguish a fragment (cut) as *active* or *unactive* according to whether it (some fragment of it) belongs to the fragment set or not. Since the active fragments are not dominated each other, the optimum route for the fragment set need not go over any active cuts. Removing those "non-essential" pieces of polygon P will not affect the optimum route.

Once a watchman fragment set is given, the corresponding (optimum) watchman route can be computed as that is described in [2]. The non-essential pieces of all active essential cuts are first removed. The resulting polygon P' is triangulated and then rolled-out by treating the active fragment on each cut as a mirror and reflecting the sleeve from one active fragment to the next one with respect to these mirrors. The optimum watchman route is then found by constructing the shortest path from s to its image s' in the rolled-out polygon. From these watchman routes, we can find the shortest watchman route. A bit of thought will convince the reader that the number of possible watchman fragment sets can be high to exponential in the number of essential cuts.

We shall present below an $O(n^3)$ algorithm for the watchman route problem. Our algorithm proceeds in an incremental way. That is, the watchman, starting from s, visits the essential cuts one by one in order. When cut C_i is involved, we construct the shortest watchman path P_i from s to a chosen point s_i of C_i in the essential piece of C_i so that all the essential cuts with index less than i are visited along the path. Finally, the shortest watchman route is obtained when the watchman returns to s. For each cut C_i, a list is maintained to hold the intersections with the cuts added by now. The intersections are ordered from l_i to r_i. It takes $O(n^2 log\ n)$ time and $O(n^2)$ space to maintain these lists.

The chosen points s_i's on essential cuts are called "images". For a point p and a segment g, p's **image** on g is the point of g which is closest to p. Images in a cut corner are defined as follows. Let C_i be the first (least indexed) cut in a cut corner. The left endpoint l_i is defined as the current starting image s_i. We then denote s_{i+1} as the image of s_i on cut C_{i+1}, s_{i+2} as the image of s_{i+1} on cut C_{i+2} and so on. In the case that the segment (s_{k-1}, s_k) intersects with cut C_{k+1}, the image s_{k+1} of s_k on C_{k+1} is undefined (since C_{k+1} has already be visited by any watchman path from s to s_k). The next image s_{k+2} of s_k on C_{k+2} is then considered. Observe that the set of the images in a cut corner is the vertex set of a convex polygon. For completeness, we set $s_{m+1} = s$. The computation of images in polygon P takes linear time.

For simplicity, we assume that all of the cuts have the images defined on them, i.e., image s_i is on cut C_i. A watchman path P_i from s to s_i can also be defined by a set of the fragments with index less than i (the index of a fragment is that of the cut to which it belongs). But now, the union of dominances of not only the fragment set but also the end point s_i is the set $\{C_1, C_2, \cdots, C_{i-1}\}$. We say image s_i dominates cut C_h if s and s_i lie to the different sides of C_h.

Consider how the shortest watchman paths can come in contact with the essential cuts. A shortest watchman path makes a **reflection contact** with an essential cut if the path reflects on some point of the cut (there exists only one point in common between the path and the cut). The reflection is *perfect* when the incoming angle of the path with the cut is equal to the outgoing angle. A shortest watchman path makes a **crossing contact** with an essential cut if the path crosses the cut once or twice. The degenerate case of a crossing contact where the path crosses the cut twice but does not properly cross the cut is called a **tangential contact**. In this case, they share a line segment. For a watchman fragment set, the corresponding optimum path makes reflection contacts with the active cuts and crossing contacts with the unactive cuts.

At the initial step of our algorithm, we look for the watchman path P_k with the greatest index k which is just the line segment (s, s_k). This path can be found by connecting s to s_k in order until the segment (s, s_k) does not intersect all the cuts with index less than k. This initial step takes $O(k^2)$ time. The watchman fragment set for P_k is initially set to be empty.

Assume inductively that P_l is given. Now we shall describe how P_{l+1} is obtained when C_{l+1} is added. The procedure consists of (1) finding an initial path P^0_{l+1} which visits all of the cuts with index less than $l+1$ and (2) adjusting the current path P^m_{l+1} until it becomes optimal. The path obtained at the end is P_{l+1}. We shall describe these two steps in detail.

Finding an initial path P^0_{l+1}

When C_l and C_{l+1} are in the different cut corners, P^0_{l+1} is simply found by adding the shortest path ([4]) between s_l and s_{l+1} to P_l, where s_{l+1} is the starting image of the next cut corner. The fragment of C_l containing s_l is then inserted into the current watchman fragment set. When C_l and C_{l+1} belong to the same cut corner, we consider the configurations between the last segment of P_l and s_{l+1}. With respect to cut C_l, s_{l+1} might be to the left or right of it (see Fig. 1a-1b). If s is to the right of C_l (Fig. 1a), P^0_{l+1} is simply formed by adding the segment (s_l, s_{l+1}) to P_l. The fragment of C_l containing s_l is inserted into the current watchman fragment set. If s_l is to the left of C_l (Fig. 1-b1,b2), we first look for the unactive cuts (with index less than l) which have s and s_{l+1} in the right side of them and are crossed exactly once by the last segment of P_l. Among these cuts, we select one whose intersection with P_l is nearest to s_l. If there exists such cut, P^0_{l+1} is obtained by replacing the segment of P_l from the intersection to s_l with the segment from the intersection to s_{l+1} (Fig. 1-b1). The fragment of the selected cut containing that intersection is inserted into the watchman fragment set. Otherwise, P^0_{l+1} is obtained by replacing the last segment of P_l with the segment which starts at the contact point of P_l with the last active cut and ends at s_{l+1} (Fig. 1-b2). In this case, there is no change in the watchman fragment set. In the special case where the last segment of P_l overlaps with C_l (see Fig. 1c), P^0_{l+1} is obtained by replacing the last segment of P_l with the segment which starts at the last reflect contact point of P_l and ends at s_{l+1}. The current watchman fragment set remains unchanged.

In the above way, we obtain an initial path P^0_{l+1} that remains convex within each cut corner and visits all of the cuts with index less than $l+1$. Suppose that the cut which has the point connected to s_{l+1} in P^0_{l+1} has index l' (l' coincides with l in Fig. 1a). It is important to observe that the part of P^0_{l+1} that is completely overlapped with P_l (from s to the point of $C_{l'}$ which is connected to s_{l+1}) has been already adjusted, and is optimal as it is. This step also takes linear time.

Adjusting the current path P^m_{l+1}

We say a watchman path P is *adjustable* on a cut C if P makes a reflection contact with C and the incoming angle of P with C is not equal to the outgoing angle. Thus, the path P^0_{l+1} is adjustable on $C_{l'}$, and it is not optimal with respect to the current fragment set. Using the current active fragments, we optimize the path P^0_{l+1} into the path P^1_{l+1} by the rolled-out method (see [2] for detail). Since P^0_{l+1} is very important in proving our main theorem (see Section 4), we treat it as a special path. The shortest watchman path P_{l+1} is then found by adjusting the current path P^m_{l+1}, where $m \geq 1$. Each adjustment involves a change in the watchman fragment set and results in a shorter path. If there are several possible candidates we can take any one of them.

An adjustment can only occur at the intersection of two essential cuts. As Chin and Ntafos showed in [3], there are three types of adjustments on an active cut C_i (Fig. 2). In Fig. 2, the incoming angle of P_{l+1}^m with C_i is assumed to be smaller than the outgoing angle. The bold and discontinuous segments in Fig. 2 stand for the active fragments before and after an adjustment, respectively. A possible next path P_{l+1}^{m+1} is also shown. Except for Fig. 2b, the indexs of the participating cuts are represented with respect to index i, i.e., $h < i < j < k$.

In Fig. 2a, P_{l+1}^m makes reflection contacts with both C_i and C_h at their intersection. The adjustment involves moving the contact point of C_i to the left. The next path, P_{l+1}^{m+1}, will make a reflection with C_i but a crossing contact with C_h. Thus, the current fragment of C_i is replaced by the next fragment and the fragment of C_h is deleted from the fragment set. We call this a **(-1)-adjustment** since the number of active fragments is decreased by 1.

In Fig. 2b, P_{l+1}^m makes a reflection contact with C_i and a normal crossing contact with $C_{i'}$. The adjustment involves moving the contact point of C_i to the left, i.e., replacing the current fragment of C_i by the next fragment. The next path P_{l+1}^{m+1} still makes a crossing contact with $C_{i'}$. This is called a **0-adjustment**. Index i' can be smaller or greater than index i.

In Fig. 2c, P_{l+1}^m makes a reflection contact with C_i but a special crossing contact with C_j, i.e., the crossing contact with C_j has degenerated into a reflection or a tangential contact. The adjustment involves substituting the current fragment of C_i with the next fragment and inserting the fragment of C_j next to p_{ij} (the intersection of C_i and C_j) into the fragment set. For the tangential contact case, the next active cut C_k should be also considered(Fig. 2-c2). In order to shorten the path, the incoming angle of P_{l+1}^m with C_k must be greater than the outgoing angle. Thus, the current fragment of C_k should be also substituted by the next fragment. We called it a **(+1)-adjustment** since the number of active fragments is increased by 1. Depending on whether the incoming angle of P_{l+1}^m with C_j is greater (Fig. 2-c1,c2) or smaller (Fig. 2-c3) than the outgoing angle, the next path P_{l+1}^{m+1} will be shorter, or the same as P_{l+1}^m. In the latter case, a (-1)-adjustment on C_j follows. This (-1)-adjustment must be done immediately after the (+1)-adjustment so that the path can still be shortened.

4 Analysis of the algorithm

Lemma 2 *The adjustments on an active cut are all in the same direction at the step of adjusting the path P_{l+1}^m.*

Proof: Omitted in this abstract. □

Lemma 3 *The procedure of constructing P_{l+1} from P_l requires $O(l)$ adjustments.*

Proof: Consider how the active and unactive cuts with respect to P_l change when P_{l+1} is constructed. First, an active cut can become unactive during the procedure of constructing P_{l+1} from P_l, but it can never be active again. Once a cut becomes unactive because of a (-1)-adjustment, the following paths will remain the crossing contact with that cut by virtue of Lemma 2. Next, we show that once a cut becomes active, it can never be unactive again. The proof is by contradiction. After cut C_j becomes active because of a (+1)-adjustment on cut C_i, the adjustments on C_j will be, say, all in the right direction. Suppose that in the following process, there exists another active cut $C_{j'}$ on which a (-1)-adjustment makes C_j unactive. The following paths will move further right to the intersection $p_{j'j}$ along $C_{j'}$. On the other hand, if there exists such a cut $C_{j'}$, index j' must be smaller than index j and the intersection p_{ij} must be left to $C_{j'}$ (see Fig. 3). Then, the path just after C_j becoming active makes a crossing contact with $C_{j'}$. Since $C_{j'}$ is supposed to be active (while C_j remains active), the path can be then adjusted to the situation where it makes a tangential contact with the unactive cut $C_{j'}$ (like Fig. 2-c2), which requires a (+1)-adjustment on C_j to make $C_{j'}$ active. This (+1)-adjustment makes the contact point on $C_{j'}$ move to the left of the intersection $p_{j'j}$. The following paths will go further left along $C_{j'}$ according to Lemma 2. Therefore, the adjustments on $C_{j'}$ can not make C_j

active. This contradicts with our assumption. In summary, there are $O(l)$ (+1)-adjustments and (-1)-adjustments during the procedure of constructing P_l from P_{l+1}. Consider 0-adjustments now. Let us charge the cost of an 0-adjustment to the participating unactive cut. From Lemma 2 and convexity of the watchman paths in cut corners, an unactive cut can take part in at most two 0-adjustments. During the procedure of constructing P_{l+1} from P_l, there is a total number of $O(l)$ unactive cuts. This completes the proof. □

Since each adjustment requires linear time to construct the new optimum path, we obtain:

Theorem 1 *The time complexity of the incremental algorithm is* $O(n^3)$.

Fig. 4 shows an example of incremental construction of the shortest watchman route. The bold segments stand for the active fragments. In Fig. 4a, P_3, P_4 and P_5^0 are shown. P_3 is the watchman path with the greatest index which is just a line segment. In order to obtain P_4, C_1 becomes active. P_5^0 is the initial path and is not optimal with respect to the current fragment set. Fig. 4b shows P_5^1. A (+1)-adjustment on C_1 produces the next path P_5^2 or, exactly, P_5. See Fig. 4c. The shortest watchman route P_6 is shown in Fig. 4d.

References

[1] B. Chazelle, Triangulating a simple polygon in linear time, Technical Report CS-TR-264-90, Dept. of Computer Science, Princeton University, May, 1990.

[2] W.P.Chin and S.Ntafos, Optimum watchman routes, *Inform. Process. Lett.* **28**, 39-44, 1988.

[3] W.P.Chin and S.Ntafos, Shortest watchman routes in a simple polygon, *Discrete Comput. Geometry* **6**, 9-31, 1991.

[4] L.Guibas, J.Hershberger, D.Leven, M.Sharir and R.Tarjan, Linear time algorithms for visibility and shortest path problems inside simple polygons, in *Proc. 2nd ACM Symp. Comput. Geometry*(1987), pp.1-13.

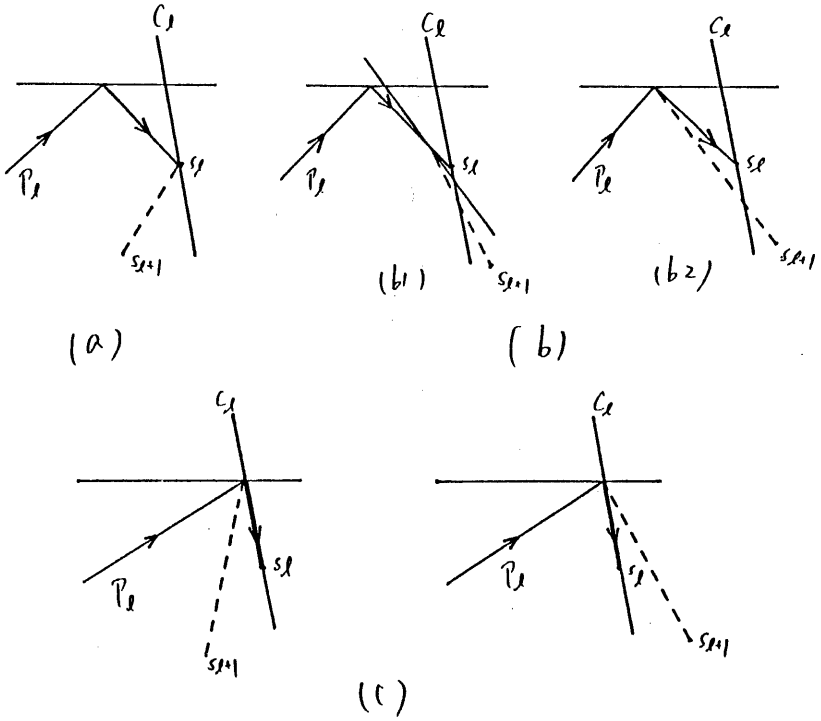


Fig. 1. The three possible configurations between P_l and s_{l+1}.

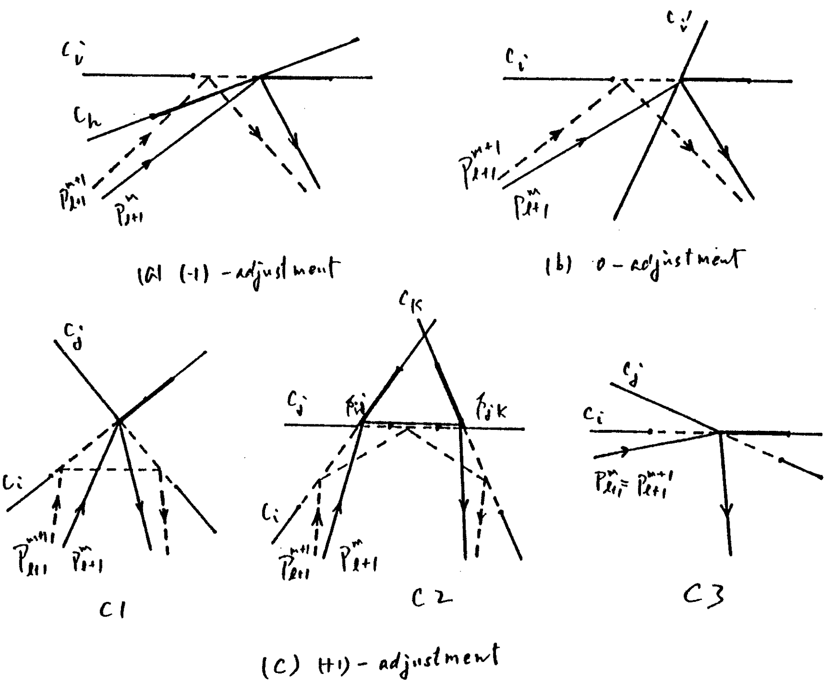


Fig. 2. Types of adjustments on an active cut C_i.

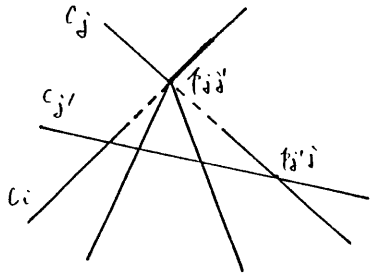
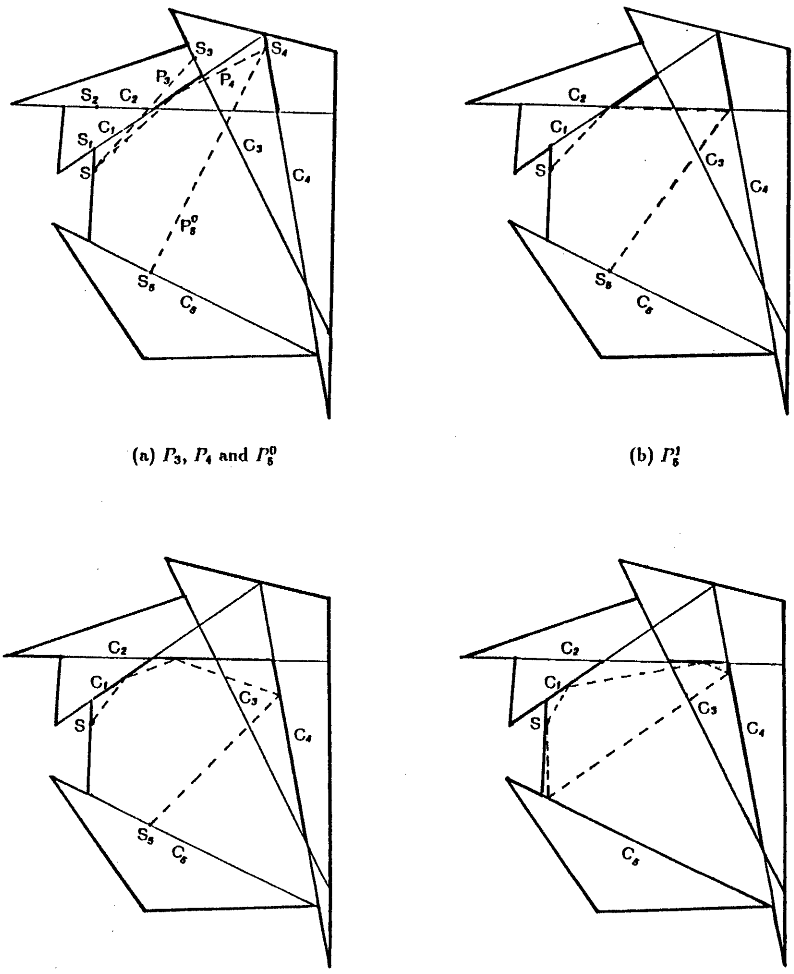


Fig. 3. Illustration for the proof Lemma 3.



(a) P_3, P_4 and P_5^0

(b) P_5^1

(c) $P_5^2 = P_5$

(d) P_6

Fig. 4. Main steps of the incremental algorithm for an instance of the watchman route problem.