TR-91-008

On the Cruising Guard Problem[1,2]

On the Cruising Guard Problem[1,2]

Y. T. Ching[3,4], M. T. Ko[5] and H. Y. Tu[6]

Abstract

Let P be a *weakly visibility polygon* from e which is an edge of P. The k-*cruising guard problem* is to find a set of k disjoint segments, s_i, $i = 1, ..., k$, on e, such that P is weakly visible from the union of these k segments and the longest $|s_i|$, $i = 1, ..., k$, is minimized. In this paper we present a linear time algorithm for the case of k=1, and an $O(c \cdot n)$ time algorithm for the case of k=2, where c is bounded by the number of the reflex vertices in P. For the general case $k > 2$, we solved a variation of the previous defined problem. We leave the general solution as an open problem.

Key words: Cruising guard problem, Weakly visible, Computational Geometry.

---

[1]The authors are in lexicographical order.

[2]This work was done while the third author was with Institute of Information Science, Academia Sinica.

[3]Institute of Information Science, Academia Sinica, Taipei, Taiwan, Republic of China

[4]Department of Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China

[5]Institute of Information Science, Academia Sinica, Taipei, Taiwan, Republic of China

[6]Department of Computer Science, Purdue University, West Lafayette, IN 47907

1. Introduction

Let $P = \{v_0, v_1, ..., v_{n-1}\}$, the sequence of the vertices in clockwise direction, denote a simple polygon. Let $e_0 = \overline{v_{n-1}, v_0}$ and $e_i = \overline{v_{i-1}, v_i}$ for $i = 1, 2, ..., n-1$ be the edges of the polygon connecting the corresponding vertices. Since the boundary of P is assumed to be directed clockwise, the interior of P lies to the right of each edge.

Definition 1: P is a *weakly visibility polygon* from an edge e, if and only if for every point p in the interior of P or on the boundary of P, there is a point p_e on e such that $\overline{p, p_e}$ lies in the interior of P[13]. The edge e is refered to as an *anchor* of P.

Given an weakly visibility polygon P and its anchor e, consider the placement of k mobile robots on e and each robot cruises along a segment on e so that the entire polygon can be seen from the k robots. If $k = 1$, Fig. 1 shows that the guard has to patrols along almost the whole edge so that the entire polygon can be seen from the guard. Consider the case of more than one guard on e and assume that each guard patrols along a portion of e in constant speed. In order to monitor the entire polygon so that each point can be seen as often as possible, we should minimize the longest distance that one of the guards patrols. This observation defines the *k-cruising guard problem.*

Definition 2: Assume that P is a weakly visibility polygon from a given edge e. Given a constant k, the *k-cruising guard problem* is to find k segment s_j, $j = 1, 2, ..., k$ on e such that P is visible from $S = \cup s_j$ and the maximum of $|s_j|, j = 1, ..., n)$ is minimized where $|s|$ denote the length in Euclidean distance of the line segment s.

For simplicity, we assume without loss of generality that P is in a *standard form*[1]. We state briefly the standard form in the following. Let P be a weakly visibility polygon with anchor e_0 where $e_0 = \overline{v_{n-1}, v_0}$. As shown in Fig. 2, let v'_{n-1} and v'_0 be the intersections, if any, of the line $\overleftarrow{v_{n-1}, v_0}$ and P. It is clear that the points in region A (B) can only be seen by v_{n-1} (v_0 respectively) on e_0. Hence, v_{n-1} and v_0 must be visited by some cruising guards in order to see A and B. This implies that v_{n-1} and v_0 must be included in the optimal solution of the cruising guard problem. Let P' be a simple polygon obtained by replacing v'_0 and v'_{n-1} with v_0 and v_{n-1}

respectively, and deleting regions A and B. P' has the property that all of its vertices lie on the same side of the line $\overleftrightarrow{v_{n-1}, v_0}$. We say that P' is in standard form. It is easy to see that the standard form of a weakly visibility polygon P with anchor e_0 can be obtained in $O(n)$ time. In the following. we assume that the input polygon P with anchor e_0 is in standard form. We also assume that the anchor e_0 is on positive x-axis and v_{i-1} is at the origin in the Cartesian coordinate system.

2. One Cruising Guard Problem

In this section, we consider the case when $k = 1$. Recall that P is weakly visible from the anchor e_0. If $k = 1$, the cruising guard problem is simply to find a shortest segment on e_0 denoted S, from which P is weakly visible.

Definition 3: For each point p in polygon P, we define r_p (and l_p) to be the right most (left most respectively) point on e_0 such that p is visible from r_p (and l_p). For vertex v_i on P, we use r_i and l_i to denote r_{v_i} and l_{v_i} for short.

It is easy to see that p is visible from every point on $\overline{r_p, l_p}$.

Definition 4: For an edge $e_i = \overline{v_{i-1}, v_i}$, let $t(i)$ be a segment on e_0 such that $t(i)$ is weakly visible from $\overline{v_{i-1}, v_i}$. The line segment $t(i)$ is said to be a *type 2 segment* if $r_{i-1} < l_i$, and is denoted $t_2(i)$. Similarly, if $r_{i-1} > l_i$, then $t(i)$ is said to be a *type 1 segment* and denoted $t_1(i)$ (see Fig. 3).

Notice that, if $t(i)$ is a type 2 segment, then the entire $t_2(i)$ is required so that e_i can be visible. That is, if there is only one cruising guard, the guard must patrol the whole $t_2(i)$. If $t(i)$ is a type 1 segment, e_i is visible from every point on $t_1(i)$. In this case, the guard can be placed at any point on $t_1(i)$ and the whole e_i can be seen from the guard.

From the above observation, the necessary and sufficient conditions for a segment S' on e which can see e_i, $\forall i$, are as the following:

$\forall\, t_1(i)$, $t_1(i) \cap S' \neq \phi$ and $\forall\, t_2(i)$, $t_2(i) \cap S' = t_2(i)$,

We state without proof the following lemma.

Lemma 1: The shortest S', denoted S, is the optimal solution for the 1-cruising guard problem.

We have the algorithm for the 1-cruising guard problem as the following.

Algorithm (1-cruising guard problem):
(1) For each v_i, compute r_i and l_i;
(2) For each e_i, determine that $\overline{l_i, r_{i-1}}$ is of $t_2(i)$ or $t_1(i)$;
(3) Let $R_2 = \phi$ and $L_2 = \phi$ if there is no type 2 edge otherwise let R_2 be the rightmost l_i of $t_2(i)$ and L_2 be the leftmost r_i of $t_2(i)$.
(4) Let $R_1 = \phi$ and $L_1 = \phi$ if there is no type 1 edges otherwise let R_1 be the rightmost l_i and L_1 be the leftmost r_{i-1}.
(5) $R = \max\{R_1, R_2\}$ and $L = \min\{L_1, L_2\}$;
(6) If $L < R$ then output $\overline{L, R}$ otherwise output any point on $\overline{R, L}$.

Step (1) and step (2) of the algorithm can be done in $O(n)$ due to the result by Avis and Toussaint[1] in which a linear algorithm for determining the visibility of a polygon from an edge was presented. For steps from (3) to (6), the operations required are to select the rightmost and the leftmost points which can also be accomplished in $O(n)$. Thus we have the following theorem.

Theorem 1: The 1-cruising guard problem can be solved in $O(n)$ time.

3. Two Cruising Guard Problem

Let $S = \overline{L, R}$ be the segment obtained for the 1-cruising guard problem. As mentioned in the previous section, S can be a point. If S is a point, then one guard is sufficient. Thus, in the following, we assume that S is not a point and two guards are required to patrol along segments s_1 and s_2 respectively where s_1 and s_2 are on S.

Lemma 2: Let s_1 and s_2 be the optimal solution for the two-cruising guard problem. The intersection of s_1 and s_2 must be empty.
Proof We first show that the intersection of s_1 and s_2 cannot be a segment. Assume the intersection $s' = s_1 \cap s_2$, each guard can patrol till the midpoint of s' such that the entire polygon still can be observed by the two guards but the longest distance cruised by one of the guards is reduced. Secondly, we show that the intersection cannot be a point neither. Assume that $|s_1| \geq |s_2|$ and point $p = s_1 \cap s_2$. Let all the points on s_1 except p are to the left of p. If all $t(i)$ contain p are of type 1, since one point on $t_1(i)$ is sufficient to observe the entire corresponding edge, it is obvious that $|s_1|$ can be reduced so that $s_1 \cap s_2$ is empty but the entire polygon is weakly visible by the two guards. If there are $t_2(i)$ that contain p. Let p'_i be the right most point on

e_i that can see p. The point p'_i defines $l_{p'_i}$ and $r_{p'_i}$. Note that $r_{p'_i}$ is p and $l_{p'_i}$ is always to the left of p. Furthermore, points on e_0 which are to the left of $l_{p'_i}$ can see all the points to the right of p'_i on e_i and points on e which are to the right of $r_{p'_i}$ can see all points to the left of p'_i on e_i. Thus $|s_1|$ can be reduced. The lemma follows.

Corollary 1: $|S|/2$ is the upper bound of $\max(\{|s_1|, |s_2|\})$, i.e. neither s_1 nor s_2 crosses p_m where p_m is the midpoint of S.

In the following, we assume that s_1 is on segment $\overline{L, p_m}$ and s_2 is on segment $\overline{p_m, R}$.

The main idea of our algorithm consists of two steps.

(1) For each edge e_i, find a point $p_i \in e_i$ such that $\min(\{|\overline{l_{p_i}, p_m}|, |\overline{p_m, r_{p_i}}|\})$ is maximized.
(2) Let R' be the leftmost $r_{p_i}, i = 1, ..., n-1$ and L' be the rightmost $l_{p_i}, i = 1, ..., n-1$. We have $\{s_1, s_2\} = \{\overline{L, L'}, \overline{R', R}\}$.

With a little difference from Section 2, we define the visibility relation diagram as follows.

Definition 5: The *Visibility Relation Diagram* of e_0 and e_i, denoted VRD_{e_i}, on domain $e_0 \times e_i$ is defined as the visibility relationship between pair of points on e_0 and e_i respectively. Let $u \in e_0$ and $u' \in e_i$. The region R_{e_i} in domain $e_0 \times e_i$ is the set of points (u, u') that u and u' are visible to each other in P. The rest of points in domain $e_0 \times e_i$ are in the set R_ϕ (see Fig. 4).

In order to investigate the properties of the boundary of VRD_{e_i}, we parameterized the point on e_i as $\mathrm{u_i} = \mathrm{v_{i-1}} + \alpha_i \cdot \mathrm{e_i}$, where $0 \leq \alpha_i \leq 1$ and $\mathrm{v_{i-1}}$ is the location vecter of vertex v_{i-1} and $\mathrm{e_i} = \mathrm{v_i} - \mathrm{v_{i-1}}$. We also use u_i to denote the point of $\mathrm{u_i}$.

If u_0 and u_i are visible to each other, then $\mathrm{u_0} - \mathrm{u_i}$ divides the vertices on P into two subsets $\{v_0, v_1, ..., v_{i-1}\}$ and $\{v_i, v_{i+1}, ..., v_{n-1}\}$. Let CH_R and CH_L be the convex hulls of these two sets of vertices respectively. The space between these two convex hulls is the set of points which is the union of all $\mathrm{u_0} - \mathrm{u_i}$ where u_0 and u_i are visible to each other. Recall that, each pair of (u_0, u_i) determines a point in $VRDe_i$. It is easy to see that $VRDe_i$ is a connected component.

The channel is defined by a pair of convex chains on CH_R and CH_L. If $u_0 - u_i$ intersects one of the convex chains at a vertex, then (u_0, u_i) is a point on the boundary of $VRDe_i$. Assume that $u_0 - u_i$ intersects a convex cahin on CH_R at a vertex v_k and we tune α_i to make $u_0 - u_i$ rotates about v_k. Note that u_0, v_k, and u_i are collinear in this case so that $(u_0 - v_k) \times (u_i - v_k) = (v_{n-1} + \alpha_0 \cdot e_0 - v_k) \times (v_i + \alpha_i \cdot e_i - v_k) = 0$. Thus, the locus of (u_0, u_i) in domain $e_0 \times e_i$, which is the boundary of $VRDe_i$, is a part of a hyperbolic function. We use $f_{i,k}(\cdot)$ to denote the hyperbolic function. If $u_0 - u_i$ intersects a convex chain on CH_R at an edge $\overline{v_k, v_{k'}}$, then (u_0, u_i) is at an intersection of two hyperbolic functions $f_{i,k}(\cdot)$ and $f_{i,k'}(\cdot)$. Therefore the convex chain on CH_R determines a sequence of hyperbolic functions in $e_0 \times e_i$ denoted $F_i(\alpha_i)$. Similarly the convex chain on CH_L define a sequence of hyperbolic functions $g_{i,k}(\cdot)$ as well and denoted $G_i(\alpha_i)$. Since R_{e_i} is a connected component, $F_i(\alpha_i)$ and $G_i(\alpha_i)$ does not intersect or they can intersect at two end points of $F_i(\alpha_i)$ and $G_i(\alpha_i)$, i.e., $\alpha_i = 0$ or (and) $\alpha_i = 1$. Assume that $u_0 = r_i$ and $u_0 - u_i$ intersects CH_R at v_i $(\alpha_i = 0)$. As we increasing α_i and maintaining $u_0 - u_i$ to intersect CH_R at a point, it is not difficult to see that $F_i(\alpha_i)$ is monotone decreasing. Similar observation shows that $G_i(\alpha_i)$ is also monotone decreasing.

Note that, $F_i(1)$ is r_{i-1} and $G_i(0)$ is l_i. Hence, if $F_i(1) \geq G_i(0)$, then e_i is a type 1 segment. If $F_i(1) \geq G_i(0)$ then e_i is a type 2 segment. For a point p_j on e_i, $F_i(p_j)$ is r_{p_j} of p_j while $G_i(p_j)$ is l_j of p_j.

Let $\overline{a, b}$ be a line segment on edge e_i, such that in domain $e_0 \times [a, b]$, $F_i(\alpha_i)$ and $G_i(\alpha_i)$ are respectively defined by a single $f(\cdot)$ and a single $g(\cdot)$ (see Fig. 5). We first concentrate on the problem that $\overline{a, b}$ must be seen from the two guards and the maximum distance patroled by one of the guards is minimized. We call this the local optimal solution for the 2-cruising guards problem. Recall that, if there are two guards, each guard patrols along a segment on $\overline{L, p_m}$ and $\overline{p_m, R}$ where p_m is the mid-point of $\overline{L, R}$ (the optimal solution of 1-cruising guard problem). Note that, two guards must visit the points L and R respectively. Assume that the two paths cruised by two guards are s_{i_L} and s_{i_R}, $L \in s_{i_L}$ and $R \in s_{i_R}$. In order to achieve the local optimal solution. We consider the following cases.

1. p_m is to the left of $(f(b) + g(b))/2$ or to the right of $(f(a) + g(a))/2$. We look at the first case. If p_m is to the left of $(f(b) + g(b))/2$ then $\overline{a, b}$ can be

observed by the guard who patrols along $\overline{f(b), R}$. Therefore, $s_{i_L} = L$ and $s_{i_R} = \overline{f(b), R}$. The later one is similar.

2. $p_m \in \overline{(f(b) + g(b))/2, (f(a) + g(a))/2}$. In this case, we search for a point $p_i \in e_i$ such that $|\overline{f(p_i), p_m}| = |\overline{g(p_i), p_m}|$. That means the distance required to be patroled by each guard reduced by equal amount, i.e., two guards patrol along segments $s_{i_L} = \overline{L, f(p_i)}$ and $s_{i_R} = \overline{g(p_i), R}$ respectively. Only under this condition, the longer distance patroled by one of the guards is minimized.

Note that these conditions can be obtained by solving the following equations:
$[\mathrm{v_{n-1}} + \alpha_0 \cdot \mathrm{e_0} - \mathrm{v_j}] \times [\mathrm{v_i} + \alpha_i \cdot \mathrm{e_i} - \mathrm{v_j}] = 0$ (vertex v_j and points on e_0 and e_i are collinear)
$[\mathrm{v_{n-1}} + \alpha'_0 \cdot \mathrm{e_0} - \mathrm{v_k}] \times [\mathrm{v_i} + \alpha_i \cdot \mathrm{e_i} - \mathrm{v_k}] = 0$ (vertex v_k and points on e_0 and e_i are collinear)
$[\mathrm{v_{n-1}} + \alpha_0 \cdot \mathrm{e_0} - \mathrm{p_m}] = [\mathrm{v_{n-1}} + \alpha'_0 \cdot \mathrm{e_0} - \mathrm{p_m}]$ (optimal solution).

According to above observations, the optimal solution should satisfy the following lemma.

Lemma 3: The optimal solution of 2-cruising guard problem is a pair of segments (s_1, s_2), such that $\forall\, s_{i_1}$, $s_{i_1} \subset s_1$ and $\forall\, s_{i_2}$, $s_{i_2} \subset s_2$.

Now we formally outline the algorithm.

Algorithm (2-cruising guard problem):
(1) Compute L and R by Algorithm for the 1-cruising guard problem;
(2) If $L > R$ then we need only one guard and can put him on any point of $\overline{R, L}$;
(3) For each e_i, compute its Visibility Relation Diagram VRD_{e_i};
(4) For each VRD_{e_i}, partition it to $O(n)$ subdomains as described in Fact 3, and compute their (s_{i_1}, s_{i_2});
(5) Let s_1 = the longest s_{i_1} obtained from step (4); s_2 = the longest s_{i_2} obtained from step (4);
(6) Output (s_1, s_2).

As mentioned in 1-cruising guard problem, step (1) needs only $O(n)$. Steps (3) and (4) can be accomplished in $c \cdot O(n) + (n - k) \cdot C$ time, where c is the number of pockets in polygon P, and C is a constant. For step (5), the pair of longest segments, s_1 and s_2, can be found in linear time. Hence, the time complexity of Algorithm (2-cruising guard problem) is bounded by $O(c \cdot n)$.

Theorem 2: The 2-cruising guard problem can be solved in $O(c \cdot n)$ time.

4. K - Cruising Guard Problem

In this section, we first define a variation of the k-cruising guard problem and then give an algorithm to the variation version. The general k-cruising guard problem for $k > 2$ is not solved. We shall discus this problem later.

The variation version of the k-cruising guard problem is defined as the following. Consider the case that if the maximum distance patrol by each guard is no greater than r, then does there exist a placement of k guards that each guard patrols along a segment s_i, $|s_i| < r$, so that P is weakly visible from the guards.

Definition 6: Given $G_i(x)$ and $F_i(x)$ as defined in the previous section, the *transition mappings* are defined as follows.
$\tau_i(r) = G_i(F_i^{-1}(R-r))$ and $\psi_i(r) = F_i(G_i^{-1}(L+r))$ where r is a real variable ranged in $[0, R-L]$.

As mentioned in the previous section, $\overline{L, R}$ is the solution for the 1-cruising guard problem and the points L and R must be visited by one of the guards. $G_i^{-1}(L+r)$ is a point p_i on e_i such that the points $L+r$, p_i, and a vertex v_k are collinear. Furthermore, if there is a guard who patrol along the segment $\overline{L, L+r}$, the segment $\overline{v_{i-1}, p_i}$ on e_i can be seen by the guard. The rest of the points on e_i have to be taken care by the next guard who patrols on the segment $\overline{F_i(G_i^{-1}(L+r), F_i(G_i^{-1}(L+r)+r} = \overline{\tau_i(r), \tau_i(r)+r}$ The observation gives the algorithm as the following.

For each e_i, we compute the transition mapping function $\tau_i(r) = F_i(G_i^{-1}(L+r)$. The leftmost $\tau_i(r)$, for all $i = 1, \ldots, n$, is the point that the next guard should start with. Apply this method for at most k times and the variation version is solved. Note that each iteration takes $O(n)$ time.

The general solution for the k-cruising guard problem involves the inversion of the transition mapping function stated above. It is know that a polynomial with order 4 or less can be solved in O(1) time. The inverse of transition mapping function has order higher that 4.

5. Conclusion :

We have presented a linear time algorithm for the 1-cruising guard problem and an $O(c \cdot n)$ time algorithm for the 2-cruising guard problem. The general solution for k-cruising guard problem is an open problem. We strongly believe that the time bound for the 2-cruising guard problem can be improved.

In this paper, we minimize the longest distance that one of the guards patrols. Another interesting problem will be to minimize the total length that all guards patrol. For the case of $k = 2$, it seems our approach doesn't help to attack this problem. We pose this as another open problem.

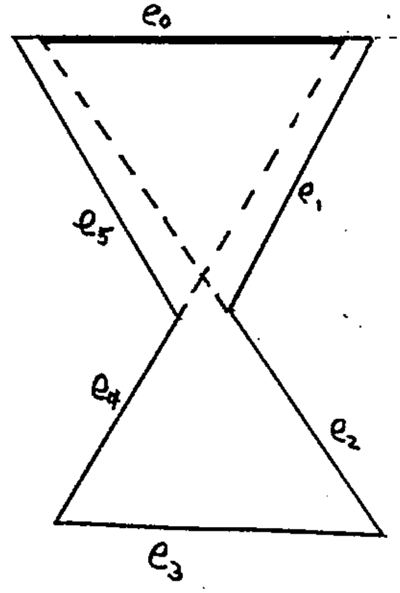

Fig. 1 Almost the whole e_0 is required to see e_3

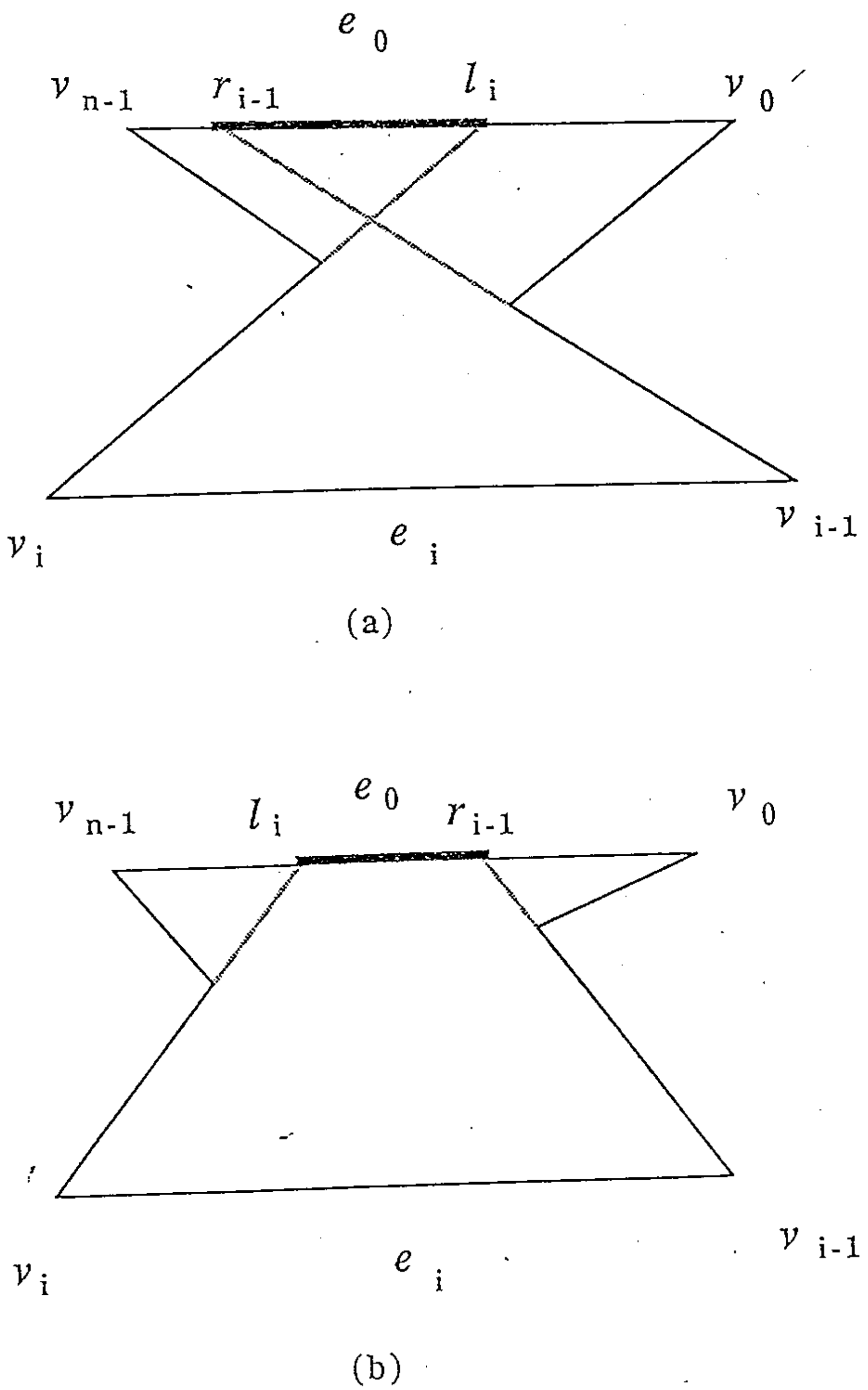

Fig. 3 (a) Type 2 segment (b) Type 1 segment

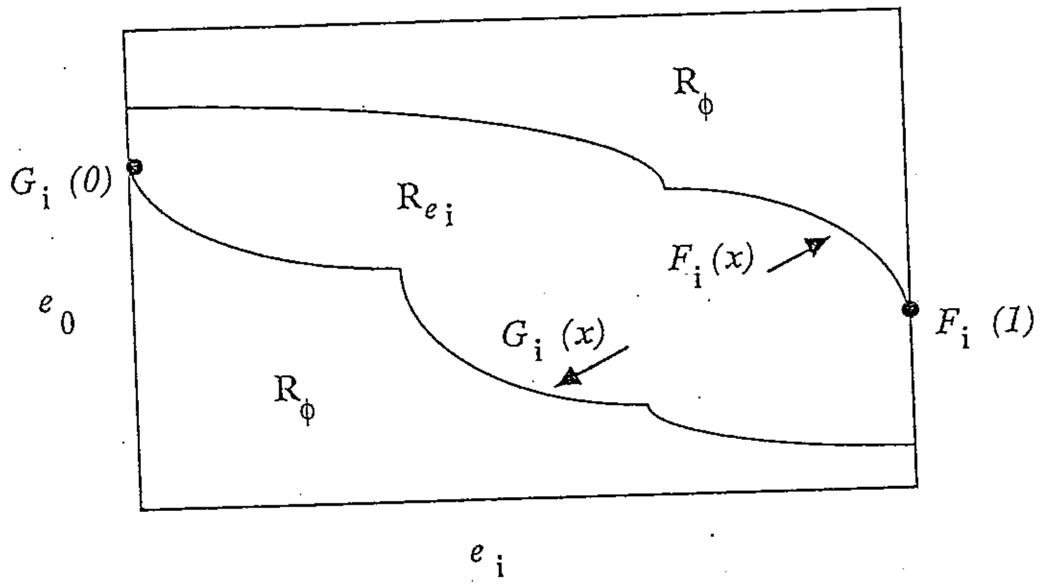

Fig. 4 Visbility Relation Diagram of $e_0 X e_i$.

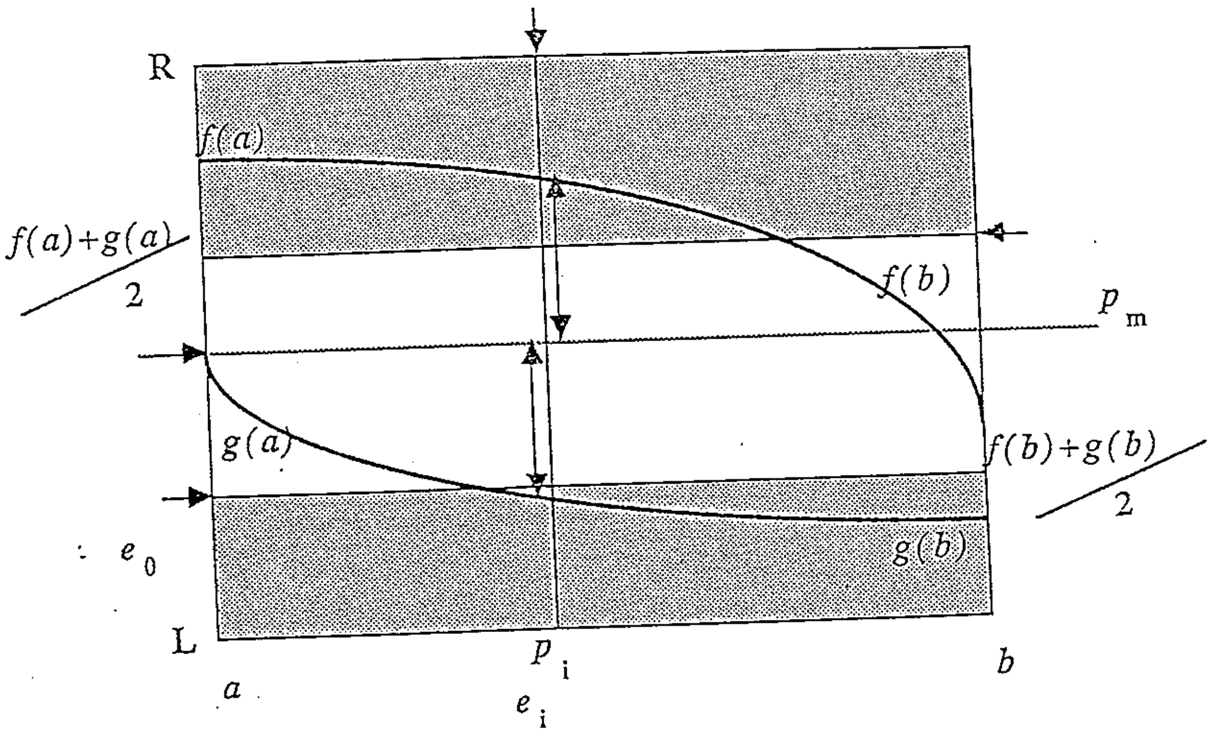

Fig. 5 Illustration for the local optimal solution.