Shortest paths in a simple polygon in the presence of "forbidden" vertices

Extended Abstract

Evanthia Papadopoulou and D. T. Lee
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60208
e-mail: papadopo@eecs.nwu.edu, dtlee@eecs.nwu.edu

April 1994

1 Introduction

We propose a new variant of the shortest path problem among obstacles in the plane.

Shortest path problem in a polygonal domain in the presence of forbidden vertices: Given a polygonal domain P, a source point s, a destination point t, and a subset F of the polygonal domain vertices, find the shortest path from s to t such that it

1. Does not cross the polygonal domain
2. Turns only on boundary vertices
3. Never turns on a vertex of the set F.

More generally, find the collection of such paths from s to all vertices of the polygonal domain.

For $F = \emptyset$, the problem is the ordinary shortest path problem in the plane. For a survey see [7]. In that case, ($F = \emptyset$) restriction 2 is immaterial since it is well known that shortest paths turn only on obstacle vertices.

This problem is motivated by a communication problem in which stations are located at obstacle vertices. A source point s is to broadcast messages to all (or some) stations via straight-line paths. Suppose some of the stations are faulty and thus, can not be used to receive or relay messages. The goal is to establish shortest communication paths from s to all non-faulty stations such that no path from s passes through a faulty station.

The shortest path problem in the presence of forbidden vertices can be solved using the visibility graph of the vertices in the polygonal domain. Since paths are restricted to turn only at polygonal vertices the visibility graph contains all feasible paths. Note that unlike the ordinary shortest path problem, a feasible path from s to t or some other vertex may not exist. An algorithm to solve the problem is the following.

- Construct the visibility graph VG of the polygonal domain including point s (see [2] for an output sensitive visibility graph algorithm). Assign the Euclidean distance as weights on edges.
- Apply *e.g.,* Dijkstra's shortest path algorithm on *VG* starting at node s with the following modification: When a node in F is to be scanned (has the minimum label) delete all edges incident to it.
- Report the collection of paths found by the algorithm.

The time complexity of the above algorithm depends on the size and the time to construct the visibility graph, and therefore is quadratic in the number of obstacle vertices. For the ordinary case ($F = \emptyset$), a recent result by Mitchell [6] gives a subquadratic ($O(n^{5/3+\epsilon})$) time and space algorithm using the continuous Dijkstra paradigm (n is the number of vertices of the polygonal domain). For $F \neq \emptyset$, it is an open question whether a solution more efficient than the above can be found.

Consider the case where the polygonal domain is a simple polygon P of n vertices. The ordinary shortest path problem can be solved in $O(n)$ time [5, 3, 4]. In the presence of "forbidden" vertices the above algorithm will give a solution in quadratic time. In this abstract we study the geometric properties of the problem in order to give a geometric solution which avoids constructing the whole visibility graph and to further explore the connection between the visibility graph and path planning. We briefly give the main ideas for an $O(kn \log n)$ time algorithm where k is the number of forbidden vertices.

2 Definitions.

Let P be a simple polygon, s be a source point in the interior of P and $F = \{f_1, f_2, \ldots, f_k\}$ be the set of forbidden vertices of P. Let t be a point in P. The shortest path from s to t which satisfies conditions 1,2 and 3 above is referred to as the *shortest alternative path* from s to t and is denoted by $\alpha(s,t)$. The length of $\alpha(s,t)$ is called the *alternative distance.* The *shortest alternative path tree* is the collection of shortest alternative paths from s to all vertices of P, and is denoted by $Q(s,P)$. For two points x, y in P, the last vertex (or x if there is none) before y on $\alpha(x,y)$ is called the *alternative anchor* of y with respect to x.

A problem arising in the presence of forbidden vertices is that $\alpha(s,t)$ need not be monotone. It may even self intersect (figure 1(a)). Moreover, unlike the ordinary case, given a polygon diagonal $\overline{v_1v_2}$ $\alpha(s,v_1)$ and $\alpha(s,v_2)$, do not provide enough information to find $\alpha(s,x)$ for a point x on $\overline{v_1v_2}$ or to find $\alpha(s,y)$ for a vertex y in $P(\overline{v_1v_2})$ (figure 1(b)).

We denote the ordinary shortest path from s to t by $\pi(s,t)$, and the ordinary shortest path tree by $T(s,P)$. The set of all descendants of a vertex v in $T(s,P)$ is denoted by $D(v)$. The length of $\pi(s,t)$ is the *geodesic distance* of t from s. The ordinary *anchor* of a point y with respect to a point x is the last vertex (or x if there is none) before y on $\pi(x,y)$. It is well known (see for example [3]) that $T(s,P)$ is a planar tree rooted at s with straight line edges. Each of its edges is either a side or an interior chord of P. $Q(s,P)$ on the other hand, is not necessarily planar. It consists of the part of $T(s,P)$ derived by deleting the subtrees rooted at $f_i \in F$ union the collection of the shortest alternative paths from s to $D(f_i), f_i \in F$.

The *extension segment* of diagonal $\overline{xy}$ is the maximal initial section of a half-line co-linear with $\overline{xy}$ extending from y in the direction of increasing distance from x until it hits the boundary of P. For two vertices x, y we define the *extension segment* emanating from y with respect to x, to be the extension segment of $\overline{vy}$ where v is the anchor of y with respect to x. See figure 2.

Polygon vertices, unless otherwise noted, are ordered according to the clockwise order induced by the polygon boundary. For a subset X of polygon vertices the polygon induced by X, $P(X)$, is the polygon formed by sequentially connecting the vertices in X in the order they appear on the polygon boundary. Note that $P(X)$ is not necessarily a subpolygon of P. A polygon diagonal $d = \overline{xy}$ partitions P into two parts, one of which contains s. We denote the part of P not containing s by $P(\overline{xy})$ (see figure 2). Two polygon vertices u, v, are f_i*-visible* if their visibility is not blocked by the forbidden vertex f_i. Note that vertices u, v are not necessarily visible. For example, in figure 2 u_5 and v_2 are *f-visible*.

3 The case of a simple polygon with exactly one "forbidden" vertex.

Let $F = \{f\}$. We assume, without loss of generality, that $\pi(s,u)$ turns left on f for $u \in D(f)$.

Let U_f be the set of vertices visible from f in $D(f)$. Let V_f be the set of vertices visible from f between the extension segments of $\overline{fy}$ and $\overline{yf}$, where y is the anchor of f with respect to s. Unless otherwise noted, we denote vertices in U_f by u and vertices in V_f by v. The two sets are ordered clockwise around the boundary of P starting at f. We consider v_1 as the vertex following the last vertex u_{last} of U_f *i.e.* we consider v_1 as u_{last+1}. $P(U_f, V_f, f)$, the polygon induced by $U_f \cup V_f \cup f$, is denoted by P'. Let $V_f(u_i)$ denote the vertices in V_f *f-visible* from u_i *i.e.*, the vertices in V_f to the left of the extension segment of $\overline{u_i f}$. Let $U_f(u_i) = \{u_k \in U_f, k > i\}$. See figure 2.

Lemma 3.1 *For any $u_i \in U_f$, $\alpha(s,u_i)$ must either turn at a vertex in $V_f(u_i)$, or otherwise turn at a visible pair of vertices v, u with $u \in U_f(u_i)$ and $v \in V_f(u) - V_f(u_i)$. In other words, $\alpha(s,u_i) = \pi(s,v) \cup \pi(v,u_i)$ for $v \in V_f(u_i)$ or otherwise $\alpha(s,u_i) = \pi(s,v) \cup \overline{vu} \cup \pi(u,u_i)$ for $u \in U_f(u_i), v \in V_f(u) - V_f(u_i)$.*

Lemma 3.2 *For any vertex x in $P(\overline{u_i u_{i+1}})$, $\alpha(s,x)$ must turn at some vertex in $V_f(u_i) \cup U_f(u_i)$ i.e., $\alpha(s,x) = \alpha(s,y) \cup \pi(y,x)$, $y \in V_f(u_i) \cup U_f(u_i)$.*

Vertices in V_f are weighted with their geodesic distance from s. Suppose vertices in $U_f(u_i)$ are weighted with their alternative distance from s. Consider the weighted geodesic Voronoi diagram of $V_f(u_i) \cup U_f(u_i)$, $Vor_{P'}(V_f(u_i) \cup U_f(u_i))$, in polygon P'. For details on geodesic Voronoi diagrams see [1].

Lemma 3.3 *The set of vertices which are possible alternative anchors for vertices in $P(\overline{u_i u_{i+1}})$ are those whose regions in $Vor_{P'}(V_f(u_i) \cup U_f(u_i))$ intersect diagonal $\overline{u_i u_{i+1}}$, $u_i \in U_f$.*

We first find the list of regions of $Vor_{P'}(V_f(u_i))$ crossing $\overline{u_i u_{i+1}}$, $L^*(\overline{u_i u_{i+1}}, V_f)$, then the list of regions of $Vor_{P'}(U_f(u_i))$ crossing $\overline{u_i u_{i+1}}$, $L^*(\overline{u_i u_{i+1}}, U_f)$, and we merge the two lists.

The difficulty is how to find $L^*(\overline{u_i u_{i+1}}, V_f)$ efficiently. For this purpose we define the following. Let *chain*(u_i) denote the convex chain of the ordinary shortest path between v_1 and u_i. *chain*(u_i), $\forall u_i \in U_f$ implies a unique triangulation of the part of P delimited by *chain*(u_1) and $U_f \cup \{v_1\}$ (see figure 3). Let $L(V_f(u_i))$ denote the list of regions of $Vor_{P'}(V_f(u_i))$ intersecting with *chain*(u_i).

To find $L^*(\overline{u_i u_{i+1}}, V_f)$ we use the triangulation of P' induced by *chain*(u_i), $u_i \in U_f$. We consider vertices in U_f and V_f in clockwise order starting at u_1 and v_1 respectively. At all times we consider only *f-visible* vertices. Suppose we have constructed $L(V_f(u_i))$. We extend $L(V_f(u_i))$ from *chain*(u_i) to *chain*(u_{i+1}) using Aronov's extension procedure [1], and split the list at u_{i+1}. Part one of the list gives $L^*(\overline{u_i u_{i+1}}, V_f)$ and part 2 is used to continue. Then we add to part 2 any vertices *f-visible* from u_{i+1} one by one, and update to get $L(V_f(u_{i+1}))$. The algorithm takes $O(r \log r)$ time, $r = |U_f| + |V_f|$, *i.e.*, $O(n \log n)$. To find $L^*(\overline{u_i u_{i+1}}, U_f)$, we use the triangulation of P' induced by diagonals $\overline{u_i f}$, $u_i \in U_f$ and use a variation of Aronov's extension procedure. The time is $O(|U_f| \log |U_f|)$. Merging $L^*(\overline{u_i u_{i+1}}, U_f)$ and $L^*(\overline{u_i u_{i+1}}, V_f)$ can be easily done using a simplification of the usual Euclidean Voronoi diagram merge procedure [8].

Let $L^*(\overline{u_i u_{i+1}}, V_f \cup U_f)$ denote the obtained list for each diagonal . To derive the shortest alternative paths for vertices in $D(f)$ we extend $L^*(\overline{u_i u_{i+1}}, V_f \cup U_f)$ in $P(\overline{u_i u_{i+1}})$ using Aronov's extension procedure [1]. $P(\overline{u_i u_{i+1}})$, $\forall u_i \in U_f$ is arbitrarily triangulated. For each vertex $w \in P(\overline{u_i u_{i+1}}) \cup \{u_i\}$ encountered while extending, let y be the anchor of the region in $L^*(d, U_f)$ to which w belongs (d is the current diagonal). Then edge $\overline{yw}$ is in $\alpha(s,w)$. See figure 4. When $L^*(d, V_f \cup U_f)$ is left with the region of only one $v \in V_f \cup U_f$, we are back in the ordinary shortest path case.

The total time complexity is $O(n \log n)$.

4 The case of more than one "forbidden" vertex.

Suppose $F = \{f_1, f_2\}$ and without loss of generality $f_1 \notin D(f_2)$. Let V_{f_1}, U_{f_1} and V_{f_2}, U_{f_2} if $f_2 \notin D(f_1)$. be defined as in the previous section. If $f_2 \notin D(f_1)$ and $f_2 \notin V_{f_1}$, f_2 does not interact with the alternative paths to $D(f_1)$ and thus the algorithm of the previous section can be used for $D(f_1)$. Similarly, f_1 does not interact with $D(f_2)$ if $f_2 \notin D(f_1)$ and $f_1 \notin V_{f_2}$. Suppose $f_2 \in D(f_1) - U_{f_1}$. To find the alternative paths to $D(f_1)$ we can use the algorithm of the previous section for $f = f_1$. While extending the obtained list of regions in $D(f_1)$, f_2 will be encountered. We continue extending until the list is left with only the region of f_2. Any vertices in the region of f_2 are descendants of f_2 and form the new $D(f_2)$. For the shortest alternative paths to the new $D(f_2)$ we can repeat the algorithm of the previous section for $f = f_2$. Here, we will only discuss the case where $f_2 \in V_{f_1}$. $f_2 \in U_{f_1}$ can be handled in a similar manner.

Assume $f_2 \in V_{f_1}$ and $\pi(s,u), u \in D(f_1)$ turns left at f_1. f_2 partitions V_{f_1} in 3 parts: $V^1_{f_1}$, the part of V_f to the right of $\overline{f_1 f_2}$, $V^2_{f_1}$, the part of $V_{f_1} - D(f_2)$ to the left of $\overline{f_1 f_2}$, and $V^3_{f_1}$, $V_{f_1} \cap D(f_2)$. See figure 5. There are two problems to resolve here. One is that the alternative weights of $V^3_{f_1}$ are not known and the other that both f_1 and f_2 may block visibility between U_{f_1} and $V^1_{f_1}$. To resolve the second problem, in addition to U_{f_1} we need to consider W_{f_2}, the set of vertices in $D(f_1)$ visible from f_2. In general $U_{f_1} \cap W_{f_2} \neq \emptyset$.

Let $Z_{f_1} = U_{f_1} \cup W_{f_2}$. Z_{f_1} is ordered clockwise according to the polygon boundary starting at f_1. Let $\pi(z_i, z_{i+1})$ be the shortest path between z_i and z_{i+1}. If $\pi(z_i, z_{i+1})$ is the polygon diagonal $\overline{z_i z_{i+1}}$, then $P(\pi(z_i, z_{i+1}))$ is $P(\overline{z_i z_{i+1}})$. Otherwise, $P(\pi(z_i, z_{i+1}))$ is the union of all $P(\overline{q_r q_{r+1}})$ where $\pi(z_i z_{i+1})$ is the convex chain $q_1 q_2, \ldots q_t, q_1 = z_i, q_t = z_{i+1}, t > 2$. Let Z'_{f_1} denote the set of all vertices on $\pi(z_i z_{i+1}), z_i \in Z_{f_1}$.

Let $L^*(\pi(z_i, z_{i+1}), V^k_{f_1}), k = 1,2,3$ denote the list of Voronoi regions in polygon $P(Z'_{f_1}, V^k_{f_1}, f_1, f_2)$ of the "appropriate" vertices in $V^k_{f_1}$ crossing $\pi(z_i, z_{i+1})$. For $V^2_{f_1}$ and $V^3_{f_1}$ the "appropriate" vertices are $V^2_{f_1}(u_i)$ and $V^3_{f_1}(u_i)$. For $V^1_{f_1}$ the "appropriate" vertices are those specified by lemma 4.1. See for example figure 6. Let $L^*(\pi(z_i, z_{i+1}), Z_{f_1})$ denote the list of Voronoi regions of the "appropriate" vertices in Z_{f_1} crossing $\pi(z_i, z_{i+1})$. In this case the "appropriate" vertices are given by lemma 4.2.

Lemma 4.1 *Consider $z_i, z_{i+1} \in Z_{f_1}$. Let $z_i = u_k$ if $z_i \in U_{f_1}$, or $z_i \in P(\overline{u_k u_{k+1}})$ otherwise. Let $z_{i+1} = w_{j+1}$ if $z_{i+1} \in W_{f_2}$, or $z_{i+1} \in P(\overline{w_j w_{j+1}})$ otherwise. The $V^1_{f_1}$ candidate alternative anchors for vertices in $P(\pi(z_i, z_{i+1}))$ are the vertices both f_1-visible from u_k and f_2-visible from w_{j+1}.*

Lemma 4.2 *For $z_i, z_{i+1} \in Z_{f_1}$, the candidate alternative anchors from Z_{f_1} for vertices in $P(\pi(z_i, z_{i+1}))$ are $\{z_k \in U_{f_1}, k > i\}$ and $\{z_k \in W_{f_2}, k \leq i\}$.*

Assuming the alternative weights of $V^3_{f_1}$ are known, we find $L^*(\pi(z_i z_{i+1}), V^k_{f_1})$ for $k = 1,2,3$ and merge to get $L^*(\pi(z_i, z_{i+1}), V_{f_1})$, the list of Voronoi regions in P of the "appropriate vertices in V_{f_1} crossing $\pi(z_i, z_{i+1})$. It can be shown that the result of merging is correct although the three lists are defined on different polygons not necessarily subpolygons of P. We also find $L^*(\pi(z_i, z_{i+1}), Z_{f_1})$ and merge with $L^*(\pi(z_i, z_{i+1}), V_{f_1})$ to get $L^*(\pi(z_i, z_{i+1}), V_{f_1} \cup Z_{f_1})$. Finally, we extend as in the previous section. The difficulty is how to find $L^*(\pi(z_i z_{i+1}), V^1_{f_1})$ without increasing the time complexity. Modifying the previous algorithm we are able to give an $O(|Z'_{f_1}| + |V^1_{f_1}|)\log(|Z'_{f_1}| + |V^1_{f_1}|))$ time algorithm.

To find $L^*(\pi(z_i, z_{i+1}), V_{f_1} \cup Z_{f_1})$ we do not need the actual alternative weights of $V^3_{f_1}$. As lemma 4.3 indicates, it is enough to find the alternative weights from $V^1_{f_2}$, the part of V_{f_2} to the left of $\overline{f_2 f_1}$ (see figure 5). We can find these alternative weights in the same way as we find the alternative weights of Z_{f_1} from $V^1_{f_1}$.

Lemma 4.3 *If a vertex $v \in V^1_{f_1}$ is the alternative anchor of a vertex in $D(f_1)$ then the alternative anchor of v is a vertex in $V^1_{f_2}$.*

After the alternative weights of Z_{f_1} are found we can find the alternative weights of $D(f_2)$.

In general, for an arbitrary number k of forbidden vertices, instead of finding the whole visibility graph we find only the visibility from the forbidden vertices. For $f_i \in F$, Z_{f_i} is the set of vertices in $D(f_i)$ visible from any forbidden vertex of $V_{f_i} \cup U_{f_i}$. We break V_{f_i} and U_{f_i} into groups of normal vertices separated by forbidden ones and work with only two groups of vertices at a time in a similar way with the case of only two vertices. This approach will give an $O(kn \log n)$ algorithm where k is the number of forbidden vertices.

5 Conclusion and open problems

We presented the shortest path problem in a simple polygon in the presence of forbidden vertices and discussed a geometric solution. The first open problem to consider is whether or not the above briefly presented algorithms are optimal. Even in the presence of only one forbidden vertex can we do better than $O(n \log n)$? Is linear time achievable?

As the number of forbidden vertices increases, the problem "loses" in geometric flavor. The main reason is the lack of path monotonicity. If the number of forbidden vertices is large, constructing the whole visibility graph is more efficient than the above mentioned method which constructs the subgraph of the visibility graph from the forbidden vertices only. Whether a subquadratic geometric solution in the presence of an arbitrary number of forbidden vertices can be derived is an open question.

References

[1] B. Aronov. " On the geodesic Voronoi diagram of point sites in a simple polygon". *Algorithmica*. 4. 109-140, 1989.

[2] Ghosh, S. K. and D. M. Mount, "An Output Sensitive Algorithm for Computing Visibility Graphs." *SIAM J. Computing*, 20, 888-910, 1991

[3] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan, "Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons". *Algorithmica*. 2. 209-233. 1987.

[4] J. Hershberger and J. Snoeyink, "Computing minimum length paths of a given homotopy class", *In Proc. 2nd Workshop Algorithms Data Struct.*, vol 519 of *Lecture notes in Computer Science*. 1991. 331-342.

[5] Lee, D.T. and F.P. Preparata, "Euclidean Shortest Paths in the Presence of Rectilinear Barriers." *Networks*, 14 1984, 393-410.

[6] Mitchell, J. S. B., "Shortest paths among obstacles in the plane", *Proc. 9th ACM Symp. on Comput. Geometry,*, May 1993, 308-317.

[7] Mitchell, J. S. B., and S. Suri, "A Survey of Computational Geometry," manuscript.

[8] Preparata, F. P. and M. I. Shamos, *Computational Geometry: an Introduction*, Springer-Verlag. New York, NY 1985.

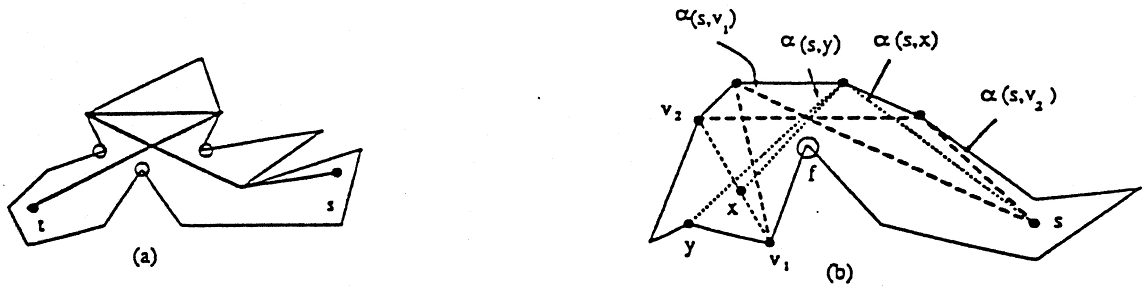

Figure 1: Problems arising because of forbidden vertices. Vertices in circles are "forbidden".

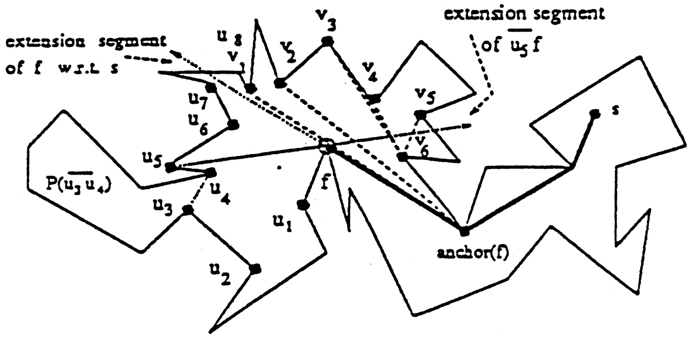

Figure 2: $V_f = \{v_1 \ldots, v_6\}$, $U_f = \{u_1, \ldots, u_7\}$, $v_1 = u_8$, $V_f(u_5) = \{v_1, v_2, v_3, v_4, v_5\}$, $U_f(u_5) = \{u_6, u_7\}$

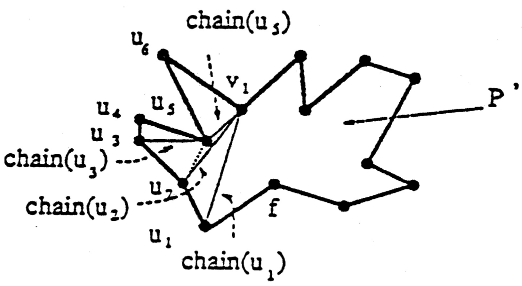

Figure 3: Definition of $chain(u_i)$.

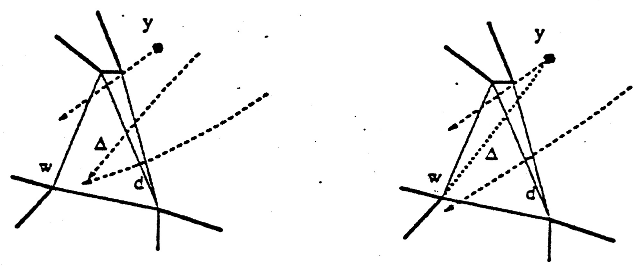

Figure 4: w lies in the region of y thus, edge $\overline{yw} \in \in$ $\alpha(s, w)$. Note the list of regions before and after the he extension in triangle Δ.

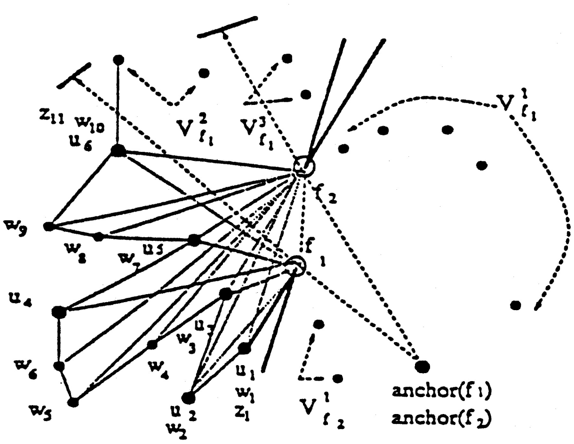

Figure 5: Definitions for $f_2 \in V_{f_1}$.

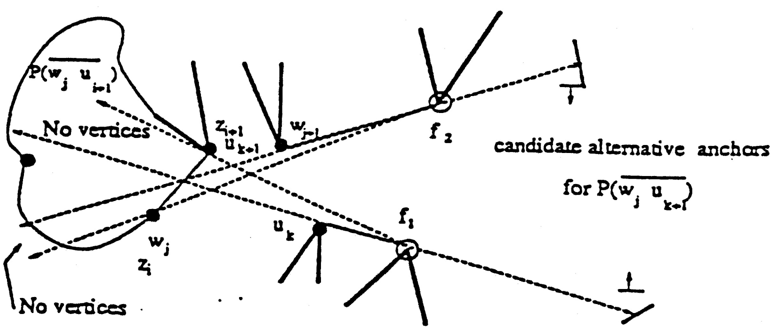

Figure 6: The $V_{f_1}^1$ candidate alternative anchors for vertices in $P(\pi(w_j, u_{k+1}))$ are those f_1-*visible* from u_k and f_2-*visible* from w_{j+1}. Here $\pi(w_j u_{k+1}) = \overline{w_j u_{k+1}}$.