31

SHORTEST PATHS IN 3-SPACE, VORONOI DIAGRAMS WITH BARRIERS, AND RELATED COMPLEXITY AND ALGEBRAIC ISSUES

Wm. Randolph Franklin and Varol Akman

Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12180-3590, USA
Phone: (518) 266-6077, telex 646542

ABSTRACT

We consider the problem of computing the shortest path under the Euclidean metric between source and goal points in 3-space while avoiding clashes with polyhedral obstacles. This can be thought as the ultimate version of the notorious TRAVELING SALESMAN problem in terms of generality and is known as the FINDPATH problem in artificial intelligence and robotics. We show that this problem is solved using algebraic elimination techniques in a straightforward yet very inefficient manner. We then introduce a Voronoi-based strategy for solving the subproblem of determining the sequence of obstacle edges through which the shortest path passes. This is based upon a natural extension of Franklin's "Partitioning the plane to calculate minimal paths to any goal around obstructions" [Tech. Rep., ECSE Dept., Rensselaer Polytechnic Inst., Troy, NY, Nov. 1982] to 3-space. In 3-space, a very desirable feature of the plane partitions disappears making the space partitions complicated. For this case, we suggest an approximation technique.

KEYWORDS: robotics, artificial intelligence, computational geometry, algebraic computing, TRAVELING SALESMAN problem, FINDPATH problem, Voronoi diagrams.

This material is based upon work supported by the National Science Foundation under grants ECS 80-21504 and ECS 83-51942. The second author is also supported in part by a Fulbright award.

NATO ASI Series, Vol. F17
Fundamental Algorithms for Computer Graphics
Edited by R. A. Earnshaw

1. INTRODUCTION

Let P_1, P_2,..., P_n be solid polyhedral objects in 3-space. A very important problem in computational geometry (which is known as FINDPATH in artificial intelligence and robotics, and has wide applications) is to find a shortest path between two points S and G (commonly termed as the source and the goal points) avoiding intersections with P_i, $i=1,\dots,n$. Touching the boundaries of P_i is allowed. Throughout this paper, we shall use the L_2 (Euclidean) metric to measure distances.

In 2-space where the obstacles are polygons whose interiors are forbidden, the problem is easy to solve. Since the shortest path can only be a polygonal path whose vertices bend at the vertices of the given polygons the problem is reduced to the following subproblems: (i) Construct a "visibility graph" whose nodes consist of $\{S,G\} \cup \{T: T \text{ is a vertex of } P_i, i=1,\dots,n\}$. A link in this graph connects a pair of vertices visible from each other and carries a weight equal to the distance between these vertices. (ii) Search through this graph to find the shortest path from S to G using an algorithm such as Dijkstra [1959]. Lee-Preparata [1984] mentions an algorithm to accomplish steps (i) and (ii) in $O(m^2 \log m)$ total time where $m=\Sigma_i |P_i|$. ($|P|$ denotes the number of vertices of polygon P.)

In 3-space the problem is much more difficult. In this case, the shortest path is also a polygonal path but the only thing we can say about its vertices is that they bend on the edges of the given polyhedra. The characterization of these bend points is a formidable task.

There have been various developments in the area of path planning in the last two decades. For brevity, we shall mention only a certain section of it. (Akman [1984] contains a long list of references.) Lozano-Perez [1981, 1983], Brooks [1983], Donald [1983], and Nguyen [1984] report many applications oriented toward robotics. Reif [1979], Schwartz-Sharir [1983a, 1983b, 1983c, 1984], Sharir-Arielsheffi [1984], Hopcroft-Schwartz-Sharir [1984], O'Dunlaing-Yap [1983], Spirakis-Yap [1983, 1984], and O'Dunlaing-Sharir-Yap [1983] report work mostly on the computational complexity of several special cases of path planning. Finally, shortest path computation has also been treated in recent papers such as Franklin [1982], Franklin-Akman [1984], Franklin-Akman-Verrilli [1985], Sharir-Schorr [1984], Lee-Preparata [1984], and O'Rourke-Suri-Booth [1984]. Sharir and Schorr's work is especially interesting in that it mentions many results on the nature of shortest paths on a convex polyhedron. For example, they prove that "A shortest path cannot pass through a vertex or a ridge point of the polyhedron" where a ridge point is defined as a goal point on the polyhedron for which there are at

31-2

least two shortest paths from a given S on the polyhedron. The crux of their paper nevertheless is the following result which is arrived after employing complex data structures and algorithms:

"Given a convex polyhedron P and a point S on it, P can be preprocessed in $O(|P|^2 \log |P|)$ time to produce a data structure (taking $O(|P|^2)$ space) with the help of which one can find in $O(|P|)$ time the shortest path along the surface of P from S to any G."

The shortest path problem is in some ways may be considered as an extension of the NP-complete (in the strong sense) TRAVELING SALESMAN problem (TSP) where we wish to determine the shortest path (or tour) that traverses the nodes of a given graph in any order, cf. Johnson-Papadimitriou [1981] and Papadimitriou-Steiglitz [1982]. Although there are some technical difficulties arising from the distance metric, the Euclidean version of TSP (ΔTSP) is also NP-complete as shown by Papadimitriou [1977].

Rest of this paper is organized as follows. Section 2 treats the problem of finding the shortest paths in 3-space using an algebraic approach. Section 3 deals with the subproblem of specifying which sequence of edges a shortest path should follow. Finally, Section 4 mentions some complexity issues and algebraic problems created by shortest path determination.

2. SHORTEST PATHS IN 3-SPACE: AN ALGEBRAIC APPROACH

If we want to find the shortest path from S to G in the presence of obstacles P_i, $i=1,\ldots,n$, the first thing is to check whether G can be reached from S directly. Note that, since we allow a shortest path to touch an obstacle, this would entail checking line segment SG against each P_i for at most one intersection. This can be done using standard methods. Chazelle-Dobkin [1979] gives a fast $(O(\log^2 |P|))$ algorithm for line-polyhedron intersection detection for convex polyhedra. Thus, in the sequel, we shall assume that such a check has already been made and SG is not the shortest path.

It is intuitively clear that the shortest path from S to G will be a polygonal path which bends on some edges of some obstacles, i.e., it cannot touch the interior of a face of an obstacle. (A formal proof is quite involved; Chein-Steinberg [1983] gives a proof in 2-space.) This observation immediately gives an algorithm to compute the shortest path. First, list all permutations of $\{e\colon e$ is an edge of P_i, $i=1,\ldots,n\}$ of positive length. Second, for each permutation in this list, compute the shortest of the polygonal paths which visits each line of this permutation exactly once in the

31-3

given order. Thus, at the end of this step, we have a list of permutations and the shortest polygonal paths associated with each permutation. (A polygonal path is specified by its consecutive vertices: S, bend points on the lines belonging to a permutation in that order, and G.) Now start at the top of this list. Test the shortest polygonal path associated with this permutation against each P_i for intersection. The only intersection points reported by this process must be the ones that we already know, i.e., the vertices of the polygonal path at hand. Otherwise, we discard this polygonal path (because it passes through one or more obstacle(s)) and continue with the next permutation. We note in passing that in Sharir-Schorr [1984] this last step is missing; thus their algorithm is incomplete.

It is emphasized that when there are more than one shortest paths, with a slight modification of the above algorithm one can obtain all of them. The number of shortest paths is an interesting problem in itself. Figure 1 shows a particular arrangement of a workspace in 2-space which clearly demonstrates that there may be an exponential number of shortest paths between S and G. A few things need some explanation in this figure. It is assumed that P and all the even-numbered obstacles are semi-infinite or large enough so that a shortest path cannot tour around them. All P_i, $i=1,\ldots,n$, and the "teeth" of P are aligned along the line connecting S to G. In this specific case there exist $2^{0.5n}$ shortest paths. It is trivial to extend this workspace to 3-space by simply erecting prisms for each polygon.

Given a permutation, the problem of finding the bend points of a shortest polygonal path on these lines can be solved using algebraic means. Before we proceed to show this, we shall state a problem and two useful lemmas regarding shortest polygonal paths through a set of lines. (For proofs of the lemmas, cf. Sharir-Schorr [1984].)

> LINE VISITATION problem (LVP): Given a sequence $l_1, l_2, \ldots, l_n$ of lines in 3-space, what is the shortest path from S to G constrained to pass through each of the lines $l_1, l_2, \ldots, l_n$ in this order?

Let $C_1, C_2, \ldots, C_n$ be the bend points of the shortest path on the given lines. For notational ease, we shall denote S (resp. G) by C_0 (resp. C_{n+1}).

LEMMA 2.1. For each $i=1,\ldots,n$, the angle between $C_{i-1}C_i$ and l_i is equal to the angle between C_iC_{i+1} and l_i.

LEMMA 2.2. The shortest path from S to G passing through the sequence of lines $l_1, l_2, \ldots, l_n$ in this order is unique.

We now give some algebraic preliminaries that will be necessary

31-4

for the upcoming presentation. Let A and B be polynomials of positive degree with coefficients in a commutative ring with an identity element. If $A(x)=\sum a_i x^i$ and $B(x)=\sum b_i x^i$ where degree(A)=m and degree(B)=n, the "Sylvester matrix" of A and B is the m+n by m+n matrix:

$$\begin{bmatrix}
a_m & a_{m-1} & \cdots & a_0 & & & \\
 & a_m & a_{m-1} & \cdots & a_0 & & \\
 & & & \vdots & & & \\
 & & a_m & a_{m-1} & & \cdots & a_0 \\
b_n & b_{n-1} & \cdots & b_0 & & & \\
 & b_n & b_{n-1} & \cdots & b_0 & & \\
 & & & \vdots & & & \\
 & & b_n & b_{n-1} & & \cdots & b_0
\end{bmatrix}$$

in which there are n rows of A coefficients, m rows of B coefficients, and all elements not printed are 0. The "resultant" of A and B, denoted by resultant(A,B), is the determinant of the Sylvester matrix.

THEOREM 2.1 (Collins [1971]). If A and B are polynomials of positive degrees over a unique factorization domain then resultant(A,B)=0 if and only if A and B have a common divisor of positive degree.

In this paper, we shall be dealing with multivariate polynomials. In this case, the following interpretation of the resultant becomes crucial. The resultant of two multivariate polynomials A and B (both given in variables $x_1, x_2, \ldots, x_r$) with respect to x_s, $1 \leq s \leq r$ is obtained as follows: (i) Write both A and B in terms of single variable x_s treating the other variables as constants. (ii) Compute the resultant of these new polynomials using the original definition above. The outcome of this is another polynomial with one less variable, i.e., x_s has been eliminated. We shall denote it by resultant(A,B,x_s).

THEOREM 2.2 (Collins [1971]). Let A and B be multivariate polynomials in variables $x_1, x_2, \ldots, x_r$ with positive degrees m and n respectively. Write both A and B in terms of the single variable x_r as explained above. Let C be resultant(A,B,x_r). If $(a_1, a_2, \ldots, a_r)$ is a common zero of A and B then $C(a_1, a_2, \ldots, a_{r-1})=0$.

31-5

Conversely, if $C(a_1,a_2,\ldots,a_{r-1})=0$, then at least one of the following is true:

(a) All coefficients of A are 0.
(b) All coefficients of B are 0.
(c) The constant coefficients of A and B are both 0.
(d) For some a_r, $(a_1,a_2,\ldots,a_r)$ is a common zero of A and B.

This theorem immediately suggests a way to solve multivariate polynomial equations simultaneously.

EXAMPLE 2.1 (adapted from Collins [1971]). Let [A=0,B=0,C=0] be a system of three equations in variables x,y,z with integer coefficients. Compute f(x)=resultant(resultant(A,C,z),resultant(B,C,z),y). By Theorem 2.2, if (a,b,c) is a solution to the given system then f(a)=0. Similarly we can compute polynomials g(y)=resultant(resultant(A,C,z),resultant(B,C,z),x) and h(z)=resultant(resultant(A,C,y),resultant(B,C,y),x) such that g(b)=0 and h(c)=0 whenever (a,b,c) is a zero of the system. Thus, one can solve f, g, and h individually to find their roots to arbitrary accuracy and then decide which triples (a,b,c) are solutions of the system.

Now we proceed to outline the algebraic solution to the LVP. In the following we refer the reader to Figure 2. Assume that each line is given by its two distinct points and assign different coordinate systems to each line, i.e., let line l_i be parametrized by x_i. Also, for each line compute N_i which is a unit vector along l_i in any direction. ($||V||$ denotes the length of vector V.) From Lemma 2.1, it is seen that:

$$\frac{C_{i-1}C_i \cdot N_i}{||C_{i-1}C_i||}=\frac{C_i C_{i+1} \cdot N_i}{||C_i C_{i+1}||}$$

If we rewrite the above equation after inserting values of C_{i-1}, C_i, C_{i+1} in terms of x_{i-1}, x_i, x_{i+1}, respectively, and remove the square root signs, then we obtain a quartic in three variables, x_{i-1}, x_i, and x_{i+1}. Repeating this for all lines, we end up with the following system of n quartics:

$$\begin{aligned}
&Q_1(x_1,x_2)=0\\
&Q_2(x_1,x_2,x_3)=0\\
&\quad\ldots\\
&Q_i(x_{i-1},x_i,x_{i+1})=0\\
&\quad\ldots\\
&Q_n(x_{n-1},x_n)=0
\end{aligned}$$

Theoretically, the above system of equations can be solved

31-6

using resultants as demonstrated in Example 2.1. This is a classical method known as the "elimination theory", cf. Van der Waerden [1970]. Alternatively, we can use a numerical technique such as the Newton-Raphson method for solving a system of nonlinear equations.

If l_1, $l_2, \ldots,$ l_n are but line segments then the shortest path may be bending at points located outside these line segments. In this case, Sharir-Schorr [1984] states that the shortest path will have to pass through some endpoints of these segments at which it will subtend different entry and exit angles contrary to Lemma 2.1. Thus the problem is reduced to a collection of subproblems where a shortest path passes through the interior points of a subsequence of line segments.

3. PARTITIONING 3-SPACE AROUND POLYHEDRA

A common specialization of the shortest path problem occurs when S and the obstacles are fixed, and new paths should be calculated as G moves around the workspace. For example, a manipulator arm may pick up a part from a pile of parts in a fixed location, and then move somewhere in the scene to work with it.

In Franklin [1982], an important construction based on an extension of Voronoi diagrams in the plane is given which, for a given S in 2-space, partitions the plane into a set of regions such that all the G within any given region have the same list of bend points. (For another extension of Voronoi diagrams, see Lee-Drysdale [1981].) This reduces the problem of finding a shortest path to the preprocessing step (finding the regions), plus the task of determining which region contains G (searching or querying). The last step is easy since the borders of the regions are either straight line segments or portions of hyperbolae. Thus, existing point location algorithms can be used after some slight modifications. In the common case where G varies while S and the obstacles are fixed, the shortest path can be found by merely repeating the search (point location) phase.

In this section, we shall try to emulate Franklin's approach in 3-space. Here, the regions will have the following property: All the points in a given region are reached from S after visiting the same sequence of edges of the obstacles. We first work on a very simple case, namely, a solid triangle.

Let W_1, W_2, and W_3 be points in 3-space. These points describe a triangle $W_1W_2W_3$ if they are not colinear. Let S be any point in 3-space not in the plane E of $W_1W_2W_3$. Assuming that $W_1W_2W_3$ is a solid triangle we want to partition the space into regions such that

31-7

if a new point G is specified we would be able to tell whether G can be directly reached from S, and if not, which edge of the triangle (W_1W_2, W_2W_3, or W_3W_1) the shortest path must touch.

Obviously, if G is outside the semi-infinite prism (frustum) obtained by subtracting the pyramid described by base $W_1W_2W_3$ and apex S from the infinite pyramid described similarly then the shortest path is SG. Thus we found one of the regions, R_0. Note that R_0 has all the points of the space that are not obstructed by $W_1W_2W_3$. See Figure 3.

Otherwise, G may belong to one of three regions R_1, R_2, or R_3. R_1 is the region such that if $G \varepsilon R_1$ then the shortest path is via edge W_2W_3. R_2 is the region such that if $G \varepsilon R_2$ then the shortest path is via edge W_3W_1. Finally, R_3 is the region such that if $G \varepsilon R_3$ then the shortest path is via edge W_1W_2. When G is on the boundary of two regions there may be two or three shortest paths.

Now, we shall compute the boundaries between the pairs R_1 and R_2, R_2 and R_3, and R_3 and R_1. In the sequel, S is assumed to be the origin. (This is easy to achieve by translating everything in the workspace by -S.) We shall first compute the boundary between R_1 and R_3. Take G such that SG $\cap$ $W_1W_2W_3$ is not empty. If $G \varepsilon R_1 \cap R_3$ then there exists a path to G either via W_1W_2 or via W_2W_3 and rendering equal lengths. Labeling the bend points of these paths with the triangle by C_{12} and C_{23}, we get:

$$SC_{12}+C_{12}G=SC_{23}+C_{23}G$$

The left hand-side is equal to $||G_{12}||$ where G_{12} is the point obtained by rotating G about W_1W_2 until it is coincident to the plane of SW_1W_2 and on the opposite side of S with respect to W_1W_2. Similarly, the right hand-side is equal to $||G_{23}||$ where G_{23} is the rotated image of G about W_2W_3.

Before we continue with our analysis, we give a list of useful vector and trigonometric identities that we shall employ frequently. (x and . denote vector cross and dot products.)

(I1) $||A||^2=1$ if A is a unit vector.
(I2) $A\times(B\times C)=(A.C)B-(A.B)C$
(I3) $(A\times B).(C\times D)=\begin{vmatrix} A.C & A.D \\ B.C & B.D \end{vmatrix}=(A.C)(B.D)-(A.D)(B.C)$
(I4) $||A+B||^2=||A||^2+||B||^2||+2||A||\ ||B||\cos\theta$
where θ is the angle between A and B.
(I5) $\sin 2\theta=2\sin\theta\cos\theta$
(I6) $\cos(\theta+\Omega)=\cos\theta\cos\Omega-\sin\theta\sin\Omega$

It is known that if P is a point and P' is its rotated version by an angle θ about an axis U (a unit length vector) passing through the origin then:

31-8

$$P' = (P.U)U + (P-(P.U)U)\cos\theta + (U\times P)\sin\theta$$

Using the last formula, it is easy to see that:

$$G_{12} = W_1 + fN_{12} + (W_1G - fN_{12})\cos\alpha + (N_{12}\times W_1G)\sin\alpha$$

where α is the dihedral angle between the planes of triangles SW_1W_2 and GW_1W_2, $N_{12}=W_1W_2/||W_1W_2||$, and $f=W_1G.N_{12}$. Using (I5):

$||G_{12}||^2=$

(a) $(W_1+fN_{12})^2$
(b) $+(W_1G-fN_{12})^2\cos^2\alpha$
(c) $+(N_{12}\times W_1G)^2\sin^2\alpha$
(d) $+2(W_1+fN_{12}).(W_1G-fN_{12})\cos\alpha$
(e) $+2(W_1+fN_{12}).(N_{12}\times W_1G)\sin\alpha$
(f) $+(W_1G-fN_{12}).(N_{12}\times W_1G)\sin 2\alpha$

The following are the simplifications:

Using (I1), (a) is simplified to $||W_1||^2+2f(W_1.N_{12})+f^2$.
(b) is simplified to $(||W_1G||^2-f^2)\cos^2\alpha$.
Using (I3), (c) is simplified to $(||W_1G||^2-f^2)\sin^2\alpha$.
Using (I1), (d) becomes $2(W_1.W_1G-f(W_1.N_{12}))\cos\alpha$.
(e) is simplified to $2(W_1.(N_{12}\times W_1G))$.
(f) is identically 0.

It is possible to simplify (d) and (e) further. Noting that the normal of the plane of the triangle SW_1W_2 is:

$$M_{S12}=\frac{N_{12}\times W_1}{||N_{12}\times W_1||}$$

and the normal of the plane of the triangle GW_1W_2 is:

$$M_{G12}=\frac{N_{12}\times W_1G}{||N_{12}\times W_1G||}$$

one obtains:

$$\cos\alpha=M_{S12}.M_{G12}$$

After some routine calculations, one arrives at:

$$\cos\alpha=\frac{W_1.(W_1G-fN_{12})}{||N_{12}\times W_1||\ ||W_1G\times N_{12}||}$$

In a similar manner, but using the cross product:

31-9

$$\sin \alpha = ||M_{S12} x M_{G12}||, \text{ or equivalently}$$

$$\sin \alpha = \frac{||W_1.(N_{12} x W_1 G)||}{||N_{12} x W_1||\ ||W_1 G x N_{12}||}$$

Thus, we showed that:

$$2\cos \alpha(W_1.(W_1G-fN_{12}))=$$
$$2\cos^2 \alpha ||N_{12}xW_1||\ ||N_{12}xW_1G|| \text{ and}$$

$$2\sin \alpha(W_1.(N_{12}xW_1G))=$$
$$2\sin^2 \alpha ||N_{12}xW_1||\ ||N_{12}xW_1G||$$

Returning to our original equation, we obtain a more symmetric equation:

$$(*)\ ||G_{12}||^2=||W_1||^2+||W_1G||^2 +2((W_1G.N_{12})(W_1.N_{12})+ ||W_1GxN_{12}||\ ||W_1xN_{12}||)$$

Now, we shall give a geometric interpretation of this equation. Expanding the dot and cross products in (*), we obtain:

$$||G_{12}||^2=||W_1||^2+||W_1G||^2 +2(||W_1G||\cos a_1||W_1||\cos(\pi-a_2) +||W_1G||\sin a_1||W_1||\sin(\pi-a_2)), \text{ or}$$

$$||G_{12}||^2=||W_1||^2+||W_1G||^2 -2||W_1G||\ ||W_1||\cos(a_1+a_2), \text{ using (I6).}$$

Above, a_1 and a_2 are the angles of GW_1W_2 and SW_1W_2, cf. Figure 4a. Finally, it is emphasized that this last formula is simply a statement of (I4) on triangle SW_1G_{12} as can be seen from Figure 4b.

Up to this point, we found a formula which gives $||G_{12}||^2$ in terms of known quantities (W_1 and N_{12}) and the unknown G (with coordinates x,y,z). The formula for $||G_{23}||^2$ (resp. $||G_{31}||^2$) is analogous to (*); just change W_1 to W_2 (resp. W_3) and N_{12} to N_{23} (resp. N_{31}). In terms of degree, the following example shows that the surfaces between regions R_i are in general ternary quartics although they may degenerate to planes in some cases.

EXAMPLE 3.1. Given the triangle with coordinates $W_1(1,2,1)$, $W_2(0,0,1)$, and $W_3(2,0,1)$, we shall compute the boundaries of regions R_1 and R_2, R_2 and R_3, and R_3 and R_1.

Using (*), the surface between regions R_1 and R_2 is found as $||G_{12}||^2-||G_{23}||^2=0$, or:

31-10

$$-2x-4y+10+2(x+2y-5 + \sqrt{(z-1)^2+(1/5)(2x-y)^2} - \sqrt{(z-1)^2+y^2})=0$$

which is further simplified to:

$$(2x-y)^2=5y^2$$

The surface between R_2 and R_3 is given by $||G_{23}||^2-||G_{31}||^2=0$, or:

$$4x-8+2(-(2/5)(x-2y-2) + \sqrt{(z-1)^2+y^2} - \sqrt{(21/5)(z-1)^2+(21/25)(2x+y-4)^2})=0$$

or, after some operations to remove the square root signs:

$$\begin{aligned}&256z^4-1024z^3+(228y^2+(256-128x)y-128x^2+512x+1024)z^2\\&+(-456y^2+(256x-512)y+256x^2-1024x)z-48y^4\\&+(576-288)y^3+(-272x^2+1088x-860)y^2\\&+(32x^3-192x^2+256x)y+16x^4-128x^3+256x^2=0\end{aligned}$$

Finally, the surface between R_3 and R_1 is found to be:

$$-8x-4y+16+\sqrt{21}\sqrt{5(z-1)^2+(2x+y-4)^2}-\sqrt{5}\sqrt{5(z-1)^2+(2x-y)^2}=0$$

which is also transformed to a quartic omitted here for brevity.

THEOREM 3.1. Let P be a convex polygon with vertices V_1, $V_2, \ldots, V_n$ and S a point outside the plane of P. It is possible to partition P into at most n convex regions (each completely containing an edge of P) such that if G is later specified inside one of these regions then the shortest path between S and G is via the associated edge of this region.

Proof. We give a constructive proof. Rotate S about the lines defined by edges V_1V_2, V_2V_3, etc. until it is coincident to the plane of P and always on the opposite side of a particular edge compared to the interior of P. This is basically an unfolding of the pyramid with apex S and base P to the base plane. Thus, n image points are obtained which will be denoted by S_{12}, S_{23}, etc. Draw the Voronoi diagram of these points and clip it against the window P. This partitions P into at most n convex regions since each Voronoi polygon is convex. Figure 5 demonstrates this operation. Let us denote the regions by R_{12}, R_{23}, etc. It is seen that there is a border line passing through each vertex of P. It is obvious that if G is inside a region R_{ij} then we just connect it to the associated image point of this region, namely S_{ij}. The intersection of this line segment with the associated edge V_iV_j of this region is the bend point X of the shortest path from S to G. X may be placed into 3-space by folding again.

Theorem 3.1 hints an important property of the regions R_1, R_2, R_3, namely, their intersections with the plane of $W_1W_2W_3$ must be

31-11

straight lines. Returning to Example 3.1:

EXAMPLE 3.1 (continued). To see the intersection of the boundary between R_1 and R_2 with E (the plane of the triangle) put z=1 in their boundary equation. This gives:

$$y=\frac{2}{1+\sqrt{5}}x$$

Continuing with the boundary of R_2 and R_3 we obtain the line:

$$y=\frac{-(8+2\sqrt{21})x+16+4\sqrt{21}}{9+\sqrt{21}}$$

Finally, for the boundary of R_3 and R_1, we have:

$$y=\frac{(8+2\sqrt{21}+2\sqrt{5})x-16-4\sqrt{21}}{-4+\sqrt{5}-\sqrt{21}}$$

These three lines intersect at:

$$X=(\sqrt{5}+1)Y/2,\quad Y=\frac{16+4\sqrt{21}}{13+2\sqrt{21}+4\sqrt{5}+\sqrt{105}}$$

in the z=1 plane. Point (X,Y) has the property that it is possible to go from S (origin) to (X,Y) on a shortest path touching any edge of $W_1W_2W_3$. Furthermore, it is the only such point on E. As a final note, (X,Y) can also be obtained as the intersection of the three Voronoi polygons constructed by the images of S on E as described in Theorem 3.1.

It is emphasized that the method exemplified up to now can be applied in the presence of a solid polygon too. In this case we are required to compute all the potential boundaries between pairs of regions. Although conceptually easy, this would be a difficult when it comes to intersect boundaries to compute their intersection curves.

The extension of the method to more than one obstacle (polygon) seems difficult. In this case, a very desirable property of plane partitions around polygons as discovered by Franklin disappears. We shall depict this with the aid of Figure 6a. First, a brief account of Franklin's approach is in order. (The reader is referred to Franklin [1982] and Franklin-Akman-Verrilli [1985] for a detailed description.) Note that in the plane, once a subdivision is formed there is only one sequence of bend points for it (provided that it is not on a boundary curve in which case there may be more). In

31-12

Figure 6a, there are two obstacles (line segments) A_1B_1 and A_2B_2 in the plane and a source S is given as shown. Franklin's algorithm partitions the plane into 5 regions in this case. R_0 holds goal points directly reachable from S. R_1 holds points which cause a shortest path to bend at B_1. R_2 holds points which cause a shortest path to bend at A_1. The boundary between R_1 and R_2 is a portion of a hyperbola. R_3 holds points which give rise to a shortest path bending at B_2. Finally, R_4 describes shortest paths bending first at A_1 and then at A_2. The boundary of R_3 and R_4 is also a portion of a hyperbola. All other boundary curves are linear. A crucial property of this diagram is as follows: "A bend point acts as a source point for a later region." For instance, A_1 acts as a souce point for the points of R_4. Similarly, B_1 acts as a source for the points of R_3. Thus, the source point is continuosly "pushed back" and this is the underlying reason for the fact that all curves are either line segments or hyperbolic sections.

In 3-space, we cannot immediately see an analogue of this property. When we place another triangle $V_1V_2V_3$ in Example 3.1, the new regions induced by this obstacle will be separated by surfaces of order higher than four (Figure 6b). Thus, whereas the boundary curves remain as hyperbolae in 2-space, in 3-space they would grow with every new polygonal obstacle placed into the workspace. One practical way to get around this problem is to approximate the boundaries with more manageable surfaces (such as quadrics) and to keep them as such even when new obstacles are introduced. This, we think, is possible since the boundary surfaces are generally smooth. The reader is referred to Figure 7 where we plotted the intersection curves of the boundaries computed in Example 3.1 with the z=2 plane.

4. COMPLEXITY AND ALGEBRAIC ISSUES

The method outlined in Section 2 to find the shortest paths is a brute force approach. However, this may be the only available approach in the light of striking similarities of our problem and the TSP. It would be interesting to determine whether there is a heuristic for this problem like Christofides' 50-percent heuristic for TSP (Garey-Johnson [1979]). We now give a partial complexity analysis for Section 2.

The enumeration of the permutations of positive length as required by the algorithm takes time proportional to the factorial of the total number of edges of the given polyhedra. Given a permutation, finding the bend points using resultants is also a costly process. If A and B are polynomials in variables x_1, $x_2, \ldots, x_r$ and $C = \text{resultant}(A, B, x_r)$ then C is the sum of at most $(m_r+n_r)!$ terms, each of which being a product of n_r A coefficients and m_r B coefficients. (A and B have degrees m_r and n_r in variable

31-13

x_r.) It can be shown that the degree of C in variable x_{r-1} is bounded above by $m_r n_{r-1} + n_r m_{r-1}$ if A and B have degrees m_{r-1} and n_{r-1} in x_{r-1}. Therefore, 2M N is seen to be an upper bound on the degree of C^r if $M = \max_i m_i$ and $N = \max_i n_i$.

In Collins [1971], the computing time of a resultant algorithm is analyzed as a function of the degrees and the coefficient sizes of its inputs. As a special case it is proved that when all degrees are equal and the coefficient size is fixed, the computing time is $O(d^{cr})$ where d is the common degree, r is the number of variables, and c is a constant.

It can be seen that the detection of the intersections of a polygonal path with the obstacles as required by our algorithm will be subsumed by the previous computations. The following is a crude argument when all P_i are convex. Take a polygonal path made of k line segments. Testing this against all polyhedra takes $O(n^2\ v \log^2 v)$ time in the worst case of $k=O(n\ v)$ where $v=\max_i |P_i|$.

There are technical problems with the Euclidean FINDPATH as stated by Papadimitriou [1977] and Garey-Graham-Johnson [1976] in the context of ΔTSP. It is known that (Grunbaum [1967] and Franklin [1983]) there exist configurations in the real projective plane which are not realizable in the rational projective plane. In the light of this, we must require infinite precision in the input (polyhedral vertex coordinates), i.e., a symbolic rather than numeric approach in inputting the coordinates. Even when one imposes the restriction that only points with rational coordinates be allowed as input, it is easy to end up with irrational distances under the L_2 metric. This can be dealt with as long as one keeps such distances merely as square roots and employs algebraic manipulation algorithms. However, if we state FINDPATH as a decision problem, i.e., "Does there exist a shortest path with length λ or less?" we suspect that FINDPATH becomes NP-complete. This originates from the difficulty of comparing numbers symbolically, or in other words, the identification of algebraic numbers (Mignotte [1982]). The symbolic expression for the length of a given shortest path on n lines may involve n+1 square roots. An attempt to compare this expression to an integer λ by repeated squaring to eliminate the square roots can take exponential time. An alternate way would be to evaluate each square root with sufficient accuracy so that their sum can be compared to λ. There is a best-known upper bound on the number of operations required to achieve that, namely, $O(m\ 2^n)$, cf. Garey-Graham-Johnson [1976]. Here m is the number of digits with guaranteed correctness. Unfortunately, there is no known polynomial way to reach this accuracy.

The drawbacks that we mentioned can be avoided if we replace the Euclidean metric by another which closely approximates it. Define $d'(x,y)=\lceil d(x,y) \rceil$ using the regular ceiling function. It is trivial to show that this still is a metric satisfying the triangle inequality. The loss of precision can be tolerated if in the beginning everything is scaled by an appropriately large number.

Regarding the Voronoi approach outlined in Section 3, there are many unanswered questions. To our best knowledge, point location in 3-space in the presence of curved surfaces of arbitrary complexity is an area with not many results. Kalay [1982] considers point location in the presence of polyhedra. Recent work reported in Chazelle [1983], drawing inspiration from a method published by Arnon-Collins-McCallum [1984a, 1984b] is at least conceptually applicable to our problem. In general, Chazelle proves that given n fixed-degree r-variate polynomials with rational coefficients, after $O(n^{c(r)})$ preprocessing time and spending polynomial space, it is possible to determine the region including a given point in $O(2^r \log n)$ time. (Note, however, that c(r) is an exponential function of r.) In fact, the mentioned work of Arnon-Collins-McCallum has many other far-reaching applications in algebraic geometry, one of them being the problem of intersecting high-order surfaces as required by our Voronoi approach in Section 3.

31-15

REFERENCES

V. AKMAN, Findminpath algorithms for task-level (model-based) robot programming, Tech. Rep., ECSE Dept., Rensselaer Polytechnic Inst., Troy, NY, March 1984.

D. ARNON, G. E. COLLINS AND S. MCCALLUM, Cylindrical algebraic decomposition I: the basic algorithm, <u>SIAM Journal on Computing</u> 13, 4 (Nov. 1984), 865-877.

D. ARNON, G. E. COLLINS AND S. MCCALLUM, Cylindrical algebraic decomposition II: an adjacency algorithm for the plane, <u>SIAM Journal on Computing</u> 13, 4 (Nov. 1984), 878-889.

R. A. BROOKS, Solving the Find-Path problem by good representation of free space, <u>IEEE Transactions on Systems, Man, and Cybernetics</u> 13, 3 (March/April 1983), 190-197.

B. M. CHAZELLE, Fast searching in a real algebraic manifold with applications to geometric complexity, Tech. Rep., Computer Science Dept., Brown Univ., Providence, RI, June 1984.

B. M. CHAZELLE AND D. DOBKIN, Detection is easier than computation, <u>Proc. 12th ACM SIGACT Conf.</u> (May 1979), 146-153.

O. CHEIN AND L. STEINBERG, Routing past unions of disjoint linear barriers, <u>Networks</u> 13 (1983), 389-398.

G. E. COLLINS, The calculation of multivariate polynomial resultants, <u>Journal of the ACM</u> 18, 4 (Oct. 1971), 515-532.

E. W. DIJKSTRA, A note on two problems in connexion with graphs, <u>Numerische Mathematik</u> 1 (1959), 1, 269-271.

B. R. DONALD, The Mover's problem in automated structural design, Tech. Rep., Artificial Intelligence Lab, Massachusetts Inst. of Technology, Cambridge, MA, June 1983.

W. R. FRANKLIN, Partitioning the plane to calculate minimal paths to any goal around obstructions, Tech. Rep., ECSE Dept., Rensselaer Polytechnic Inst., Troy, NY, Nov. 1982.

W. R. FRANKLIN AND V. AKMAN, Minimal paths between source and

31-16

goal points located on/around a convex polyhedron, Proc. 22nd Allerton Conf. on Communication, Control, and Computing (Sep. 1984).

W. R. FRANKLIN, V. AKMAN AND C. VERRILLI, Voronoi diagrams with barriers and on polyhedra for minimal path planning, The Visual Computer - An International Journal on Computer Graphics (1985).

W. R. FRANKLIN, Algebra problems in CAD computations, Tech. Rep., ECSE Dept., Rensselaer Polytechnic Inst., Troy, NY, Oct. 1983.

M. R. GAREY, R. L. GRAHAM AND D. JOHNSON, Some NP-complete geometric problems, Proc. 8th ACM Symp. on Theory of Computing (May 1976), 10-22.

M. R. GAREY AND D. S. JOHNSON, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman, San Francisco, CA, 1979.

B. GRUNBAUM, Convex Polytopes, Wiley, New York, 1967.

J. E. HOPCROFT, J. T. SCHWARTZ AND M. SHARIR, On the complexity of motion planning for multiple independent objects; PSPACE hardness of the "Warehouseman's problem", Tech. Rep., Computer Science Div., New York Univ., Courant Inst. of Mathematical Sciences, New York, Feb. 1984.

D. S. JOHNSON AND C. H. PAPADIMITRIOU, Computational complexity and the Traveling Salesman problem, Chap. 3 of The Traveling Salesman Problem, eds. E. L. Lawler, J. K. Lenstra and A. G. Rinooy Kan, Wiley, New York, 1981.

Y. E. KALAY, Determining the spatial containment of a point in general polyhedra, Computer Graphics and Image Processing 19 (1982), 303-334.

D. T. LEE AND R. L. DRYSDALE, Generalizations of Voronoi diagrams in the plane, SIAM Journal on Computing 10, 1 (Feb. 1981), 73-87.

D. T. LEE AND F. P. PREPARATA, Euclidean shortest paths in the presence of rectilinear barriers, Networks 14 (1984), 393-410.

T. LOZANO-PEREZ, Automatic planning of manipulator transfer movements, IEEE Transactions on Systems, Man, and Cybernetics 11, 10 (Oct. 1981), 681-698.

31-17

T. LOZANO-PEREZ, Spatial planning: a Configuration Space approach, IEEE Transactions on Computers 32, 2 (Feb. 1983), 108-120.

M. MIGNOTTE, Identification of algebraic numbers, Journal of Algorithms 3 (1982), 197-204.

V-D. NGUYEN, The Find-Path problem in the plane, Tech. Rep., Artificial Intelligence Lab, Massachusetts Inst. of Technology, Cambridge, Mass., Feb. 1984.

C. O'DUNLAING, M. SHARIR AND C. K. YAP, Retraction: a new approach to motion-planning, Proc. 15th ACM Symp. on Theory of Computing (April 1983), 207-220.

C. O'DUNLAING AND C. K. YAP, The Voronoi method for motion-planning: I. The case of a disc, Tech. Rep., Computer Science Div., New York Univ., Courant Inst. of Mathematical Sciences, New York, March 1983.

J. O'ROURKE, S. SURI AND H. BOOTH, Shortest paths on polyhedral surfaces, Proc. 2nd Annual Symp. on Theoretical Aspects of Computer Science (Jan. 1985).

C. H. PAPADIMITRIOU, The Euclidean Traveling Salesman problem is NP-complete, Theoretical Computer Science 4 (1977), 237-244.

C. H. PAPADIMITRIOU AND K. STEIGLITZ, Combinatorial Optimization: Algorithms and Complexity, Addison-Wesley, Reading, MA, 1982.

J. H. REIF, Complexity of the Mover's problem and generalizations, Proc. 20th IEEE Conf. on Foundations of Computer Science (1979), 421-427.

J. T. SCHWARTZ AND M. SHARIR, On the "Piano Movers" problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers, Communications on Pure and Applied Mathematics XXXVI (1983), 345-398.

J. T. SCHWARTZ AND M. SHARIR, On the "Piano Movers" problem: II. General techniques for computing topological properties of real algebraic manifolds, Advances in Applied Mathematics 4 (1983), 298-351.

J. T. SCHWARTZ AND M. SHARIR, On the "Piano Movers" problem: III. Coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers, International Journal of Robotics Research 2 (Fall 1983), 3, 46-75.

31-18

J. T. SCHWARTZ AND M. SHARIR, On the "Piano Movers" problem: V. The case of a rod moving in three-dimensional space amidst polyhedral obstacles, Communications on Pure and Applied Mathematics XXXVII (1984), 815-848.

M. SHARIR AND E. ARIEL-SHEFFI, On the "Piano Movers" problem: IV. Various decomposable two-dimensional motion-planning problems, Communications on Pure and Applied Mathematics XXXVII (1984), 479-493.

M. SHARIR AND A. SCHORR, On shortest paths in polyhedral spaces, Proc. 16th ACM Symp. on Theory of Computing (1984), 144-153.

P. SPIRAKIS AND C. K. YAP, On the combinatorial complexity of motion coordination, Tech. Rep., Computer Science Div., New York Univ., Courant Inst. of Mathematical Sciences, New York, April 1983.

P. SPIRAKIS AND C. K. YAP, Strong NP-hardness of moving many discs, Information Processing Letters 19 (1984), 55-59.

B. L. VAN DER WAERDEN, Algebra, 2 vols., Ungar, New York, 1970.

31-19

FIGURES

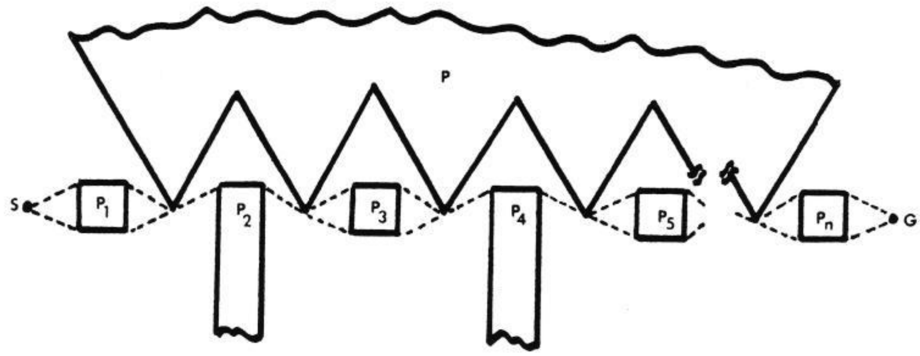


FIGURE_1

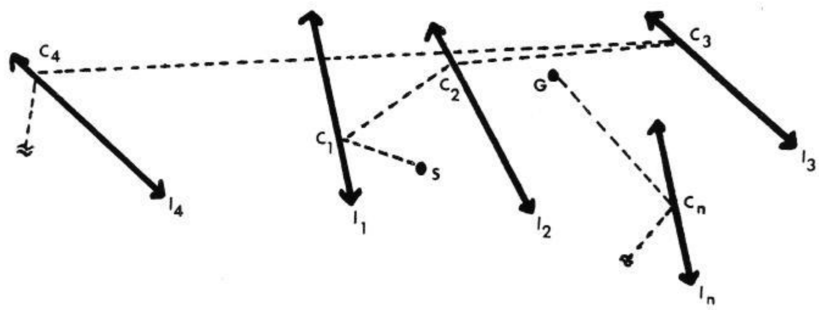
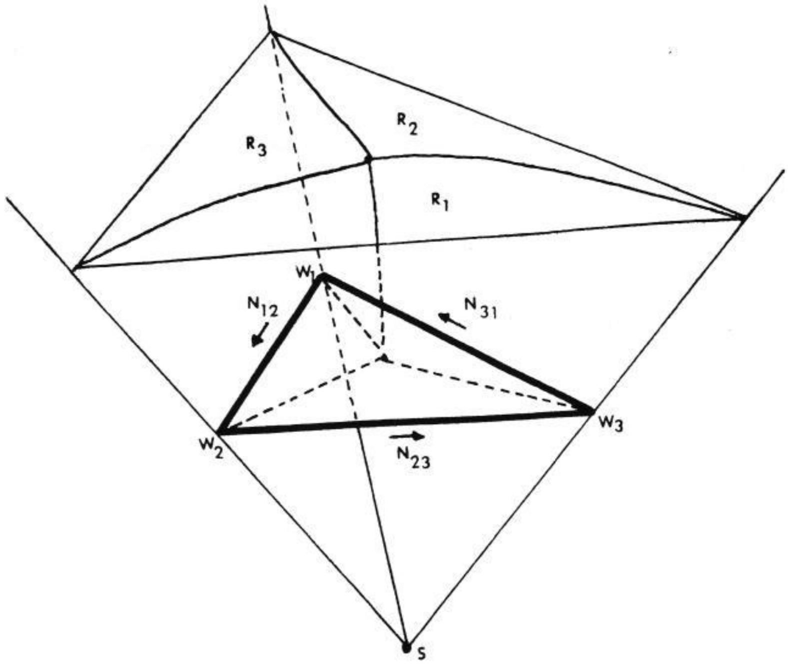
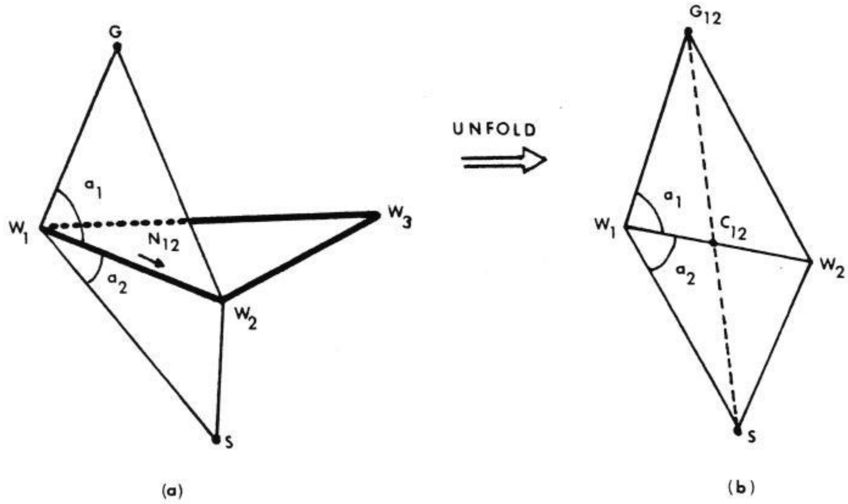


FIGURE_2

31-20



F I G U R E - 3



F I G U R E - 4

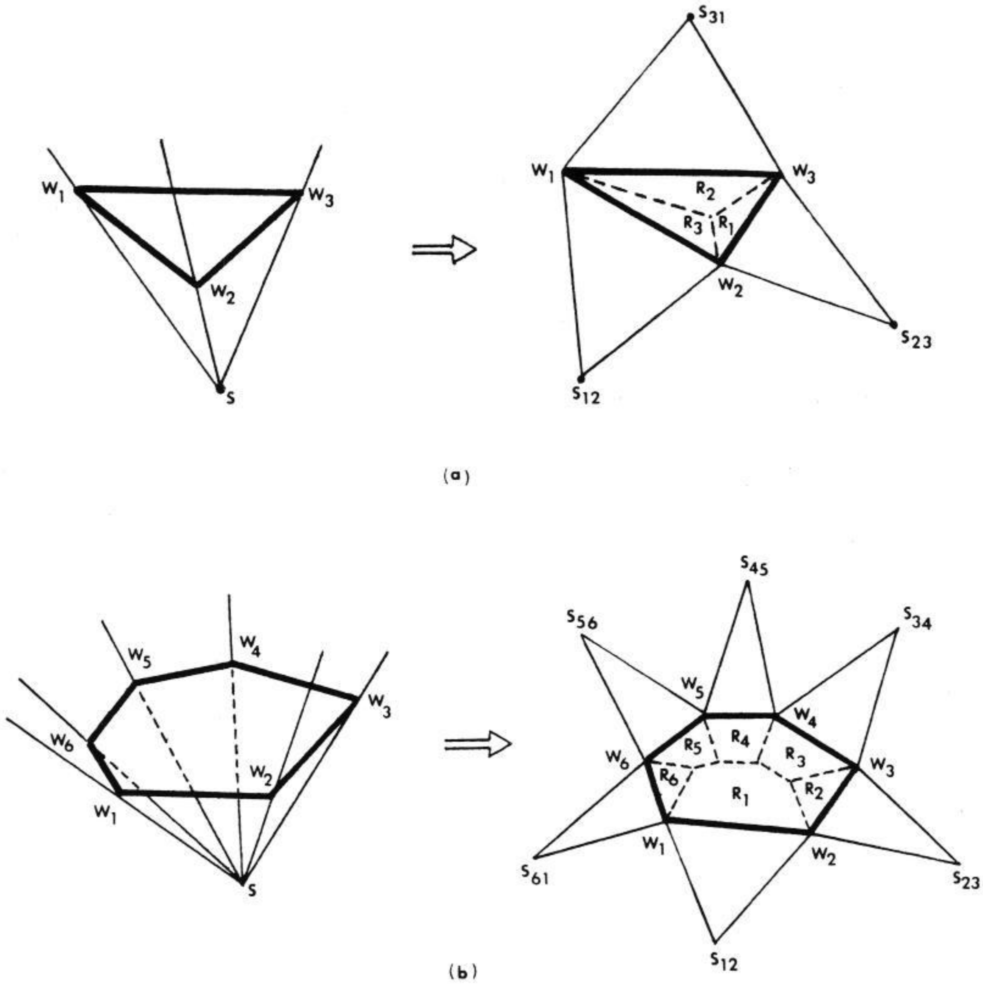


FIGURE_5

31-22

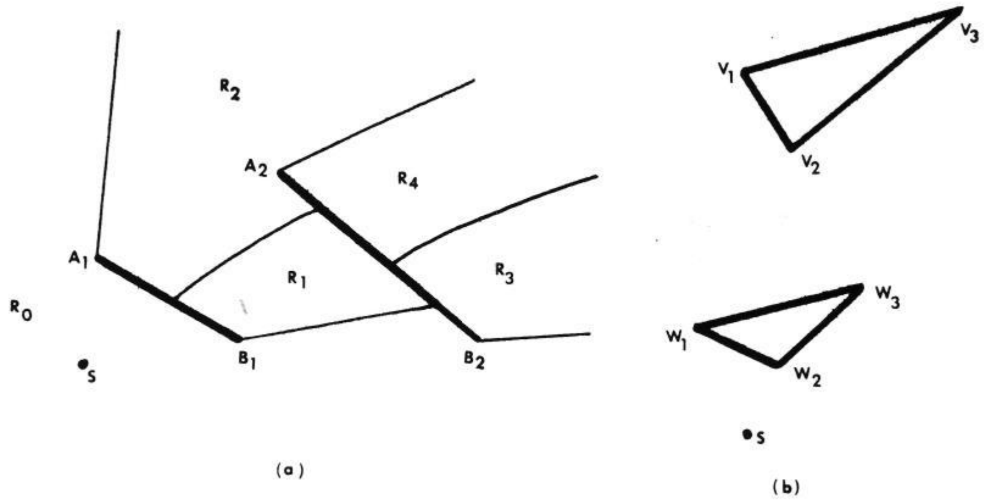


FIGURE_6

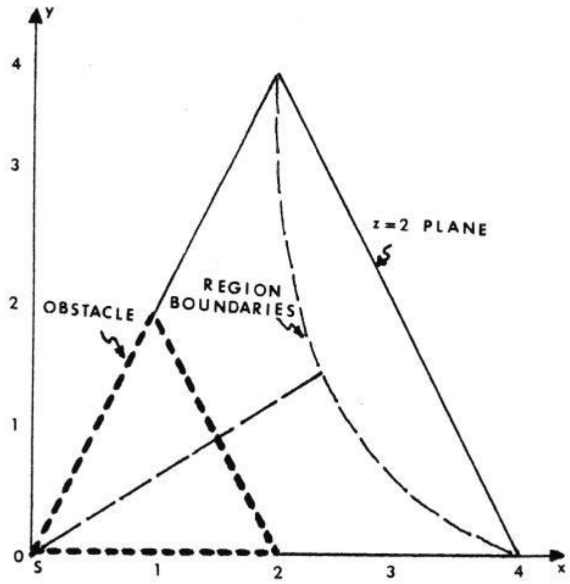


FIGURE_7