Maxmin Location of an Anchored Ray in 3-Space and Related Problems

Frank Follert *

(Extended Abstract)

Abstract

We consider the problem of locating a ray emanating from the origin of 3-space such as to maximize the minimum weighted Euclidean distance to a set of weighted obstacles (points, lines or line segments). We present algorithms based on the parametric search paradigm which run in $O(n \log^4 n)$ time in the case of point obstacles, and in $O(n^2 \log^2 n)$ ($O(n^2 \log^2 n\, 2^{\alpha(n)})$) time in the case of line (segment) obstacles. We also show that for practically interesting restricted settings of the line obstacle problem, subquadratic algorithms can be obtained. Furthermore we discuss some related problems.

1 Introduction

Facility location problems, such as the well known largest empty circle problem, are one of the major topics in geometric optimization. They are mostly motivated by layout problems in operations research and are thus generally formulated in a planar setting. In this abstract we discuss several location problems in 3-space which arise from an entirely different area: neurosurgery. In stereotactical operations, a surgeon removes a tissue sample from a specified point in the patient's brain for diagnostic purposes by intruding a line-shaped instrument. In order to minimize the risk of brain damage, the objective is to choose a position of the instrument which maximizes the clearance to critical brain areas such as the visual cortex or blood vessels.

This task is adequately represented by computing a ray emanating from the origin of 3-space (an *anchored ray*) which maximizes the minimal (weighted) Euclidean distance to a set of (weighted) obstacles representing the critical areas. In the following, let d denote the Euclidean distance and $||.||$ denote the Euclidean norm. We formally state the

Problem Definition: Given a set $O = \{o_1, \ldots, o_n\}$ of obstacles (points, lines, resp. line segments) in 3-space with positive real weights w_i, determine an anchored ray R which maximizes $\min_{1 \leq i \leq n} w_i d(o_i, R)$.

We will refer to problem version A,B or C when dealing with point, line, resp. line segment obstacles.

1.1 Outline of the technique

The algorithms described in this abstract are based on the *parametric search* paradigm, an ingenious optimization technique which was introduced in [13] and has found numerous applications since (cf. e.g. [3]). The basic step in employing parametric search is to give efficient sequential and parallel algorithms for the decision version of the optimization problem. In our case the corresponding decision problem reads "Given a fixed ρ, does there exist a ray R which keeps minimal safety distance ρ to all obstacles?". This decision problem can be transformed into the question whether a set of 'forbidden regions' covers a two-dimensional manifold which represents all possible orientations of an anchored ray. By explicitly computing the union of the forbidden regions, this can be decided efficiently provided that the union size is small.

1.2 Previous results

Problem A and its planar counterpart were introduced in [7] and further studied in [8]. The planar version can be solved in optimal time $O(n \log n)$, whereas the best previous algorithm for the three-dimenional case

*Universität des Saarlandes, Fachbereich 14, Informatik, Lehrstuhl Prof. G. Hotz, Postfach 151150, 66041 Saarbrücken, Germany. Email: follert@cs.uni-sb.de, Fax: +49 681 302 4421.
The author is supported by a Graduiertenkolleg fellowship from DFG, Germany

runs in $O(n \log^5 n)$ time (cf. [8]). The extension to line (segment) obstacles is investigated for the first time in this abstract.

1.3 Overview

The abstract is organized as follows: In section 2 we discuss problem A and present an algorithm which computes a maxmin anchored ray amidst point obstacles in time $O(n \log^4 n)$. Furthermore we prove similar time bounds for some related problems: the dual minmax anchored ray problem and the maxmin anchored line problem. In section 3 we consider line (segment) obstacles and give close to quadratic algorithms for problems B and C. Finally we show that a subquadratic algorithm can be obtained if problem B is modified to meet a certain 'width' criterion.

2 Point Obstacles

2.1 The parametric search technique

Parametric search is an optimization technique which is amenable when we deal with a monotone decision problem $\mathcal{P}(\rho)$ depending on a single real parameter ρ. Monotone means that $\mathcal{P}(\rho)$ is true for all ρ less or equal to a threshold value ρ^* and false for all $\rho > \rho^*$. Parametric search determines ρ^* by performing an implicit binary search on some critical values of ρ with the help of a sequential and parallel version of a decision procedure which answers queries like "Given a fixed ρ, is ρ larger, smaller or equal to ρ^*?". Let T_s denote the running time of the sequential decision algorithm, and T_p, resp. , P the time and number of processors of the parallel version, then ρ^* can be computed in sequential time $O(PT_p + T_s T_p \log P)$. For a detailed description of parametric search, the reader is referred to [3].

2.2 Problem A

First we note that problem A is invariant with regard to transforming the point obstacle set $P = \{p_i = (x_i, y_i, z_i)\}_{1 \le i \le n}$ with weights $w_i > 0$ to $P' = \{p'_i = (w_i x_i, w_i y_i, w_i z_i)\}_{1 \le i \le n}$ with unit weights. In the following we will therefore assume w.l.o.g. that all weights are equal to 1. Consider the following decision problem $\mathcal{P}(\rho)$: Given a positive real ρ, is there an anchored ray R with $\min_{1 \le i \le n} d(p_i, R) \ge \rho$? $\mathcal{P}(\rho)$ is monotone and solving problem A amounts to computing the maximal ρ^* for which $\mathcal{P}(\rho)$ is true, and an anchored ray R^* satisfying $\min_{1 \le i \le n} d(p_i, R^*) = \rho^*$. Let $d_{\min}$ denote $\min_{1 \le i \le n} d(p_i, 0)$. Clearly $\mathcal{P}(\rho)$ is false for all $\rho > d_{\min}$ and consequently $\rho^* \le d_{\min}$. In order to apply the parametric search paradigm, we have to give sequential and parallel algorithms to decide $\mathcal{P}(\rho)$ for $0 \le \rho \le d_{\min}$. An anchored ray $R_{\mathbf{x}} = \{\alpha \mathbf{x}; \alpha \in \mathbf{R}_0^+\}$ can be represented by its intersection $\mathbf{x}/||\mathbf{x}||$ with the unit sphere $\mathcal{S}^2$. The points on $\mathcal{S}^2$ representing rays too close to $p_i \in P$ form an open spherical disc $F_i^\rho = \{\mathbf{x} \in \mathcal{S}^2; d(R_{\mathbf{x}}, p_i) < \rho\}$ which we name a *forbidden region*. $\mathcal{P}(\rho)$ is equivalent to deciding whether the union of the forbidden regions $\cup_i F_i^\rho$ covers the unit sphere $\mathcal{S}^2$. Let Pl_i^ρ denote the plane satisfying $Pl_i^\rho \cap \mathcal{S}^2 = \partial F_i^\rho$ and H_i^ρ denote the closed halfspace bounded by Pl_i^ρ and containing the origin. The intersection $Poly = \cap_i H_i^\rho$ of these halfspaces is a possibly unbounded convex polyhedron. $\mathcal{P}(\rho)$ is false iff $Poly$ is bounded and all vertices $\mathbf{v}$ of $Poly$ lie in the interior of $\mathcal{S}^2$, i.e., they satisfy $||\mathbf{v}|| < 1$. As $Poly$ contains the origin in its interior, its construction can be reduced to a 3D convex hull problem by a standard dual transformation. The convex hull of n points in 3-space can be constructed in $O(n \log n)$ sequential and $O(\log^2 n)$ parallel time using n processors on a CREW PRAM (cf. [14],[1]). Plugging this into the parametric search paradigm, we obtain

Theorem 1 *Given a set $P = \{p_1, \ldots, p_n\}$ of points in 3-space with positive real weights w_i, an anchored ray maximizing the minimum weighted Euclidean distance to P can be computed in time $O(n \log^4 n)$.*

2.3 Related problems

2.3.1 Computing a maxmin anchored line

A straightforward modification is to replace the anchored ray in problem A by a line passing through the origin, an *anchored line*. In the case of unit weights, the resulting problem is equivalent to computing a largest empty cylinder with an anchored axis. It can be tackled using basically the same approach as in the preceding paragraph. An anchored line is represented by its two points of intersection with $\mathcal{S}^2$. The forbidden region F_i^ρ induced by a single obstacle point is the union of two open diametrically opposed spherical discs and the decision problem whether the union of all forbidden regions covers the sphere can again be reduced to a halfspace intersection problem. This observation leads to

Theorem 2 *Given a set $P = \{p_1, \ldots, p_n\}$ of points in 3-space with positive real weights w_i, an anchored line maximizing the minimum weighted Euclidean distance to P can be computed in time $O(n \log^4 n)$.*

2.3.2 The dual minmax problem

The dual version of problem A asks for an anchored ray which minimizes the maximal weighted Euclidean distance to a set P of weighted points in 3-space. Given a positive real ρ, every point $p_i \in P$ induces an *admissible region* $A_i^\rho = \{\mathbf{x} \in S^2; d(p_i, R_\mathbf{x}) < \rho\}$, which is an open spherical disc. The corresponding decision problem asks, whether the intersection of the admissible regions $Int = \cap_i A_i^\rho$ is empty. Int is a convex spherical region and can be computed by divide & conquer. We assume that for each of the two subregions to be intersected in a conquer step, two lists of its boundary vertices sorted along the upper resp. lower chain are given. The essential part of the conquer step is to merge these lists efficiently. This can be done in linear sequential time and in $O(\log\log n)$ parallel time using $\frac{n}{\log\log n}$ processors of a CREW PRAM [10]. Omitting further details, we state

Theorem 3 *Given a set $P = \{p_1, \ldots, p_n\}$ of points in 3-space with positive real weights w_i, an anchored ray minimizing the maximum weighted Euclidean distance to P can be computed in $O(n \log^3 n \log\log n)$ time.*

3 Line (Segment) Obstacles

3.1 Problems B and C

We first deal with problem B, the maxmin ray amidst a set of line obstacles $L = \{l_1, \ldots, l_n\}$. As in the preceding section, we can eliminate weights by a straightforward transformation. It is convenient to parameterize an anchored ray $R_\mathbf{x}$ by its point of intersection $\mathbf{x}$ with the planes $\overline{P} : z = 1$ resp. $\underline{P} : z = -1$. (We omit the treatment of horizontal rays for simplicity in this abstract.) We seek the minimum ρ^* such that the union of all forbidden regions $\overline{F_i^\rho} = \{\mathbf{x} \in \overline{P}; d(R_\mathbf{x}, l_i) < \rho\}$ and $\underline{F_i^\rho} = \{\mathbf{x} \in \underline{P}; d(R_\mathbf{x}, l_i) < \rho\}$ cover both planes $\overline{P}$ and $\underline{P}$. Each F_i^ρ has the shape of an open wedge; it can degenerate into an open halfplane, an open stripe or it may be empty. Following the parametric search paradigm, we show how to solve the decision problem "Given a fixed ρ, do the $\overline{F_i^\rho}$ cover $\overline{P}$?" with sequential and parallel algorithms. ($\underline{P}$ is treated analogously). First we construct the arrangement induced by the lines containing the bounding rays of all forbidden regions. This can be done in time $O(n^2)$ by the standard incremental algorithm. Then we decide by traversing every feature (face, edge, vertex) of the arrangement, whether there is a feature not covered by any forbidden region. This can be accomplished within the same time bounds. In order to parallelize the algorithm we adapt a technique of [2]. The arrangement can be constructed by n^2 processors in $O(\log n)$ parallel time on a CRCW PRAM. This is done by computing all vertices of the arrangement, sorting the edges incident to a vertex around this vertex and sorting the vertices incident to an edge along this edge. For every feature of the arrangement we need to compute the number of forbidden regions it is covered by. For any pair of adjacent faces, this quantity differs at most by one. We construct the dual graph of the arrangement, build a spanning tree of the dual graph and convert it into an Eulerian path. We can then calculate the respective quantities for each face of the arrangement by a parallel prefix algorithm, all in $O(\log n)$ parallel time using n^2 processors. The same can be done for the edges and vertices of the arrangement. Applying the parametric search technique yields the following

Theorem 4 *Given a set $L = \{l_1, \ldots, l_n\}$ of lines in 3-space with positive real weights w_i, an anchored ray maximizing the minimum weighted Euclidean distance to L can be computed in time $O(n^2 \log^2 n)$.*

In the remainder of this section we will briefly discuss how to solve problem C, the case of line segment obstacles $S = \{s_1, \ldots, s_n\}$. The forbidden region F_i^ρ induced by a line segment $s_i \in S$ in the plane $\overline{P}$ (resp. $\underline{P}$) is a convex planar region bounded by at most two line segments and two arcs of conic sections or may be empty. The arrangement of all boundaries in both planes can be constructed by the incremental algorithm of [5] in $O(n^2 2^{\alpha(n)})$ sequential time. ($\alpha(n)$ denotes the extremely slowly growing functional inverse of Ackermann's function.) By proceeding in much the same way as above, we get

Theorem 5 *Given a set $S = \{s_1, \ldots, s_n\}$ of line segments in 3-space with positive real weights w_i,* *an anchored ray maximizing the minimum weighted Euclidean distance to S can be computed in time $O(n^2 \log^2 n\, 2^{\alpha(n)})$.*

Note that the algorithms can easily be modified to compute a maxmin anchored line amidst line (segment) obstacles.

3.2 Subquadratic algorithms for restricted problem B

3.2.1 Restricted problem setting

There is evidence that problem B generally cannot be solved in subquadratic time. In fact, the corresponding decision problem is closely related to the class of n^2-*hard* problems as introduced in [9], and it seems improbable that subquadratic algorithms for problems of this class exist.

The difficulty of problem B is mainly caused by the fact that the complement of the union of all forbidden regions (wedges) may have quadratic complexity. Recent results indicate that the union size of a collection of geometric figures such as triangles, wedges and polygons is close to linear provided that they are *fat*, i.e., do not contain long, skinny parts [12, 6, 11]. We give restrictions on the problem setting that guarantee the forbidden regions to be fat.

- We are only interested in anchored rays keeping a fixed minimal safety distance ρ_0 to all obstacles.
- The minimal distance of a line obstacle from the origin is bounded from above by d_0.

Both restrictions can be justified in view of the intended application of the algorithm. An instance of the restricted problem B, which we will call problem B', is characterized by its *width* $w = \rho_0/d_0$. An algorithm for problem B' with width $w = \rho_0/d_0$ and line obstacles $L = \{l_1, \ldots, l_n\}$ is expected to return an anchored ray maximizing the minimal distance to L if it is $\geq \rho_0$, or to indicate that no such ray exists.

3.2.2 The subquadratic algorithm

Let us first address the decision problem. We cannot apply the bounds for the union size of fat wedges stated in [6], because the opening angles of the forbidden wedges may be arbitrary small, even in the restricted problem setting. By choosing a different representation of the rays, we can nevertheless show that the forbidden regions of problem B' have small union size. An anchored ray is represented by its intersection with the faces of the axis-parallel cube $\mathcal{C}$ of sidelength 2 whose center is the origin. Given a fixed ρ, there is no ray R satisfying $d(R, l_i) \geq \rho\ \forall i$ iff the forbidden regions cover all six faces of $\mathcal{C}$. Let f be an arbitrary face of the cube and Pl be the plane containing it. The forbidden regions $\mathcal{F} = \{F_i^\rho\}_{1 \leq i \leq n}$ of f are a collection of convex, polygonally bounded regions; each F_i^ρ can be described as the intersection of an open wedge W_i^ρ in Pl with f (cf. 3.1). We shall show how to extend each F_i^ρ to a fat polygon whose interior coincides with F_i^ρ inside $int(f)$, the interior of face f. For a formal definition of fatness we use the notion of δ-wideness as introduced in [11].

Definition 1 *For any $0 < \delta \leq 1$, a δ-corridor is an isosceles trapezoid T with vertices p_1, p_2, p_3, p_4 such that $|p_1p_4| = |p_2p_3| = \max\{|p_1p_2|, |p_3p_4|\}/\delta$. For any $0 < \delta \leq 1$, a simple polygon is δ-wide, if for any two edges e, e' of P, and any four points $p_1, p_4 \in e$ and $p_2, p_3 \in e'$ that are the vertices of a γ-corridor Q such that $int(Q) \subseteq int(P)$, it follows that $\gamma \geq \delta$.*

It is easy to see that a triangle with minimal angle α is $2\sin(\frac{\alpha}{2})$-wide.

Lemma 1 *Consider an instance of problem B' with width $w = \rho_0/d_0$ and line obstacles $L = \{l_1, \ldots, l_n\}$. Let $\mathcal{F} = \{F_i^\rho\}_{1 \leq i \leq n}$ be the collection of forbidden regions of a face f of the cube C, i.e. $F_i^\rho = \{\mathbf{x} \in f; d(R_\mathbf{x}, l_i) < \rho\}$. Then there exists a collection $\mathcal{F}'$ of n δ-wide polygons with the property*

$$(\cup_{F' \in \mathcal{F}'} int(F')) \cap int(f) = \cup_i F_i^\rho \cap int(f)$$

and $\delta \geq \min\left\{2\sin(\frac{\pi}{8}), \sqrt{\frac{w^2}{2-2w^2}}, \sqrt{\frac{4w^2}{5-4w^2}}\right\}$.

Proof: (Sketch)
We explicitly show how to construct the element of $\mathcal{F}'$ whose interior covers $F_i^\rho \in \mathcal{F}$ inside $int(f)$. Let C_1 denote the circumcircle of square f in Pl, and C_2 the concentrical circle of radius 2. We distinguish three cases according to the position of the forbidden wedge W_i^ρ (whose intersection with f is F_i^ρ) in the plane Pl.
Case 1: Only one or none of the bounding rays of W_i^ρ intersects C_1. We construct an arbitrary equilateral triangle T containing the intersection of W_i^ρ and C_1 (Figure 1). This triangle is δ-wide for $\delta = 1$.

Case 2: Both bounding rays intersect C_1 and the apex a of W_i^ρ is contained in C_2. The wedge W_i^ρ intersects one of the two tangents to C_1 perpendicular to its medial axis in points p_1 and p_2 such that the triangle ap_1p_2 coincides with W_i^ρ inside C_1. We erect an isosceles right-angled triangle on $\overline{p_1p_2}$, unite it with triangle ap_1p_2 and denote the resulting kite by K (Figure 2). Analytic calculations prove that K is δ-wide for $\delta \geq \min\left\{2\sin(\pi/8), \sqrt{\frac{4w^2}{5-4w^2}}\right\}$.

Case 3: Both bounding rays intersect C_1 and the apex of W_i^ρ is not contained in C_2. The wedge W_i^ρ intersects the two tangents to C_1 perpendicular to its medial axis in four points p_1, p_2, p_3, p_4 which are the vertices of an isosceles trapezoid T (Figure 3). It can be shown that T is δ-wide for $\delta \geq \min\left\{2\sin(\pi/8), \sqrt{\frac{w^2}{2-2w^2}}\right\}$

□

A close to linear bound for the union complexity of the forbidden regions now follows directly from recent bounds for δ-wide polygons. (For a simpler statement of the results we will assume $1/n \leq \delta, w < 1$ in the following.)

Theorem 6 ([11]) *Let $\mathcal{P}$ be a set of δ-wide polygons with n vertices in total. The maximum complexity of the contour of the union for $\mathcal{P}$ is $O((n \log\log n)/\delta)$.*

We compute the boundary of the union of all forbidden regions of face f following the divide & conquer paradigm. The conquer step can be performed by an optimal line segment intersection algorithm, e.g. [4], in time $O(N \log N + k)$, where N denotes the complexity of the two superimposed polygonal regions which is $O(n \log\log n/w)$ and k denotes the number of intersections found which also is $O(n \log\log n/w)$. Thus the merging can be done in time $O(n \log n \log\log n/w)$. The overall running time of the sequential algorithm is $O(n \log^2 n \log\log n/w)$. A parallel version of the algorithm uses the red-blue line intersection algorithm of [16], which computes all points of intersection in optimal parallel time $O(\log N)$ using $N+k/\log N$ processors of a CREW PRAM.

Now we can formulate the overall algorithm. First the algorithm runs the decision procedure with $\rho = \rho_0$. If the result is 'No', the algorithm indicates failure. Otherwise it runs the usual parametric search algorithm and exits with the resulting ray.

Theorem 7 *Given an instance of problem B' with line obstacles $L = \{l_1, \ldots, l_n\}$ and width $w = \rho_0/d_0$, there is an algorithm which computes an anchored ray maximizing the minimum Euclidean distance to L if this minimum distance is $\geq \rho_0$ or indicates that no such ray exists, all in time $O(n \log^5 n \log\log n/w)$.*

Acknowledgement

The author thanks Günter Hotz for pointing out this interesting area of research and Jürgen Sellen and Elmar Schömer for many helpful discussions.

References

[1] N. Amato, F. Preparata. The parallel 3D convex hull problem revisited. *Comp. Geom. & Applications* 2 (1992), pp. 163-173.

[2] P. Agarwal, B. Aranov, M. Sharir, S. Suri. Selecting distances in the plane, *Algorithmica*, 9 (1993) pp. 495-514.

[3] P. Agarwal, M. Sharir, S. Toledo. Applications of parametric searching in geometric optimization, *Journal of Algorithms*, 17 (1994), pp. 292-318.

[4] B. Chazelle, H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. ACM 39 (1992), pp. 1-54.*

[5] H. Edelsbrunner, L. Guibas, J. Pach, R. Pollack, R. Seidel, M. Sharir. Arrangements of curves in the plane - topology, combinatorics and algorithms. *Theor. Comp. Science* 92 (1992), pp. 319-336.

[6] A. Efrat, G. Rote, M. Sharir. On the union of fat wedges and separating a collection of segments by a line, *Comput. Geom.: Theory and Appl.*, 3 (1993), pp. 277-288.

[7] F. Follert. Lageoptimierung nach dem Maximin-Kriterium, Master's Thesis, Univ. d. Saarlandes, Saarbrücken, (1994).

[8] F. Follert, E. Schömer, J. Sellen, M. Smid, Ch. Thiel. Computing a largest empty anchored cylinder and related problems. *Tech. Report MPI-*

I-95-1-001, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

[9] A. Gajentaan, M. Overmars. n^2-hard problems in computational geometry, *Tech. Report RUU-CS-93-15*, Dept. Comput. Science, Utrecht Univ., Utrecht, Netherlands, (1993).

[10] J. JaJa. *An introduction to parallel algorithms*, Addison-Wesley (1992).

[11] M. van Krefeld. On fat partitioning, fat covering and the union size of polygons. *3rd Workshop on Algorithms and Data Structures*, (1993), LNCS 709, pp. 452-463.

[12] J. Matousek, J. Pach, M. Sharir, S. Sifrony, E. Welzl. Fat triangles determine linearly many holes. *SIAM J. Comput.* 23 (1994), pp. 154-169.

[13] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms, *Journal of the ACM*, 30 (1983), pp. 852-865.

[14] F. Preparata, M. Shamos. *Computational geometry: an introduction*, Springer, (1988).

[15] T. Ottmann, P. Widmayer, D. Wood. A fast algorithm for Boolean mask operations. *Comput. Vision Graph. Image Process.* 30 (1985), pp. 249-286.

[16] C. Rüb. Computing intersections and arrangements for red-blue curve segments in parallel. *Proc. 4th Canadian Conf. on Comp. Geom.* (1992), pp. 115-120.

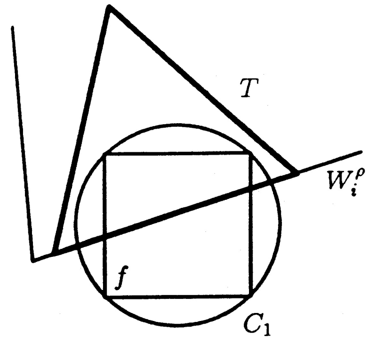

Figure 1: Case 1

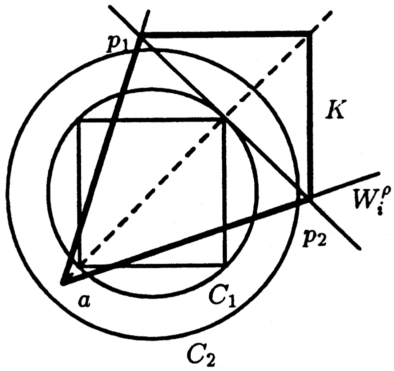

Figure 2: Case 2

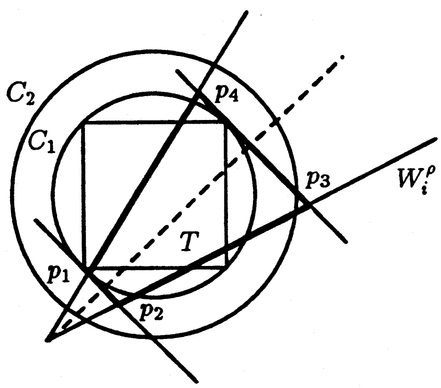

Figure 3: Case 3