Path Planning in a Weighted Planar Subdivision Under the Manhattan Metric

Mansoor Davoodi* † Hosein Enamzadeh* ‡ Ashkan Safari* §

Abstract

In this paper, we consider the problem of path planning in a weighted polygonal planar subdivision. Each polygon has an associated positive weight which shows the cost of path per unit distance of movement in that polygon. The goal is finding a minimum cost path under the Manhattan metric for two given start and destination points. We propose an $O(n^2)$ time and space algorithm to solve this problem, where n is the total number of vertices in the subdivision. We also study the case of rectilinear regions in three dimensions, and generalize the proposed algorithm to find a minimum cost path under the Manhattan metric in $O(n^3 \log n)$ time and $O(n^3)$ space.

1 Introduction

Path planning (PP) problem is one of the fundamental problems in motion planning whose objective is to find an optimal path with minimum length between two start and destination points s and t in a work space. In the classical version of PP, the work space contains some obstacles, and the path must avoid these obstacles [7, 14]. However, in a general formulation of PP – called *Weighted Region Problem* (WRP) – which was first introduced by Mitchell and Papadimitriou [17], each obstacle has an associated weight and a path is allowed to enter them at extra costs. In fact, these weights represent the cost per unit distance of movement in the obstacles (or say *weighted regions*). This generalization of PP has a lot of applications, e.g., it can be used in self-driving cars navigation, robot motion planning [6], military purposes [16], crowd simulation [13], and gaming applications [13]. An important theoretical result on WRP [9] has shown that this problem cannot be solved in the algebraic computation model over the rational numbers under the Euclidean metric. Motivated by this result, we investigate WRP under the Manhattan metric and show that it can be solved efficiently in polynomial time.

*Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran

†mdmonfared@iasbs.ac.ir

‡hosein.enamzadeh@iasbs.ac.ir

§ashkan.safari@iasbs.ac.ir

Mitchell and Papadimitriou [17] introduced an ϵ-optimal algorithm with running time of $O(n^8 L)$, where n is the total number of vertices of polygonal regions and L is the precision of problem's instance. Precisely, $L = O(\log(nNW/\epsilon w))$, where N is the maximum integer coordinate of any vertex of the subdivision, W and w are the maximum non-infinite and minimum non-zero integer weights assigned to the faces of the subdivision, and $\epsilon > 0$ is a user-specified error tolerance. The output is the shortest path from the starting point s to all vertices of the polygons with an error tolerance ϵ under the Euclidean metric. Mata and Mitchell [16] have proposed an algorithm based on constructing a relatively sparse graph – called *pathnet* – that can search for paths that are close to optimal. They have proved that a pathnet of size $O(nk)$ can be constructed in $O(kn^3)$ time. As a matter of fact, the pathnet limits the paths that can extend from vertices with k cones at each vertex. Searching for a path on the constructed pathnet yields a path whose weighted length is at most $(1 + \epsilon)$ of optimal path. Precisely, $\epsilon = \frac{W/w}{k\Theta_{min}}$, where W/w is the ratio of the maximum non-infinite weight to the minimum non-zero weight, and θ_{min} is the minimum internal face angle of the subdivision. One of the common techniques for obtaining approximate shortest paths is to positioning Steiner points for discretizing the edges of the triangular regions and then constructing a graph by connecting them. Finally, by using graph search algorithms such as Dijkstra, an approximate minimum cost path can be computed [1, 2, 18].

There are several variants of WRP due to the metric and the shape of weighted regions. Lee et al. [15] have solved the problem in the presence of isothetic obstacles (the boundary edges of obstacles are either vertical or horizontal line segments). They have presented two algorithms for finding the shortest path under the Manhattan metric. The first algorithm runs in $O(n \log^2 n)$ time and $O(n \log n)$ space, and the second one runs in $O(n \log^{3/2} n)$ time and space. Gewali et al. [10] have considered a special case of this problem in which there are only three types of regions: regions with weight of ∞, regions with weight of 0, and regions with weight of 1. They have presented an algorithm in $O(m + n \log n)$ time, where $m \in O(n^2)$ is the number of visibility edges. Furthermore, they have presented an algorithm for the case that linear feathers are added. Precisely, edges of the subdivision are allowed to have arbitrary weights. Their algorithm for this case takes $O(n^2)$ time for constructing a graph of size $O(n^2)$ for searching the shortest path. In fact, it takes $O(n^2 \log n)$ time for finding the shortest path. Gheibi et al. [11] have discussed the problem in an arrangement of lines. Due to the fact that this special case of the problem has unbounded regions, they have presented a minimal region – called *SP-Hull* – to bound the regions. This minimal region contains the minimum cost path from s to t. They construct *SP-Hull* in $O(n \log n)$ time, where n is the number of lines in the arrangement. After constructing *SP-Hull*, an approximate minimum cost path can be obtained by applying the existing approximation algorithms within bounded regions. Jaklin et al. [13] have analyzed the problem when the weighted regions are cells of a grid. They have also presented a new hybrid method – called *vertex-based pruning* – which is able to compute paths that are ϵ-optimal inside a pruned subset of the scene.

In this paper, we consider a planar subdivision with arbitrary positive weights. We present an algorithm which constructs a planar graph in $O(n^2)$ time with $O(n^2)$ vertices and edges, where n is the total number of vertices of the subdivision. The constructed graph contains the minimum cost path between two points s and t in the plane, where the distances are measured under the weighted Manhattan metric – the length of a path is the weighted sum of Manhattan lengths of the sub-paths within each region. It has been shown that this problem is unsolvable over the rational numbers when the distances are measured under the weighted Euclidean metric [9]. To the best of our knowledge, this is the first result that presents an exact algorithm for solving WRP under the Manhattan metric in a case where the regions are arbitrary simple polygons with positive weights. We propose an exact algorithm for finding the minimum cost path under the weighted Manhattan metric in $O(n^2)$ time which is also a $\sqrt{2}$−approximation for the Euclidean metric. Also, we show that the proposed algorithm can be used for WRP with rectilinear subdivision in three dimensions in $O(n^3 \log n)$ time and $O(n^3)$ space.

This paper is organized in five sections. In section 2, we give some preliminaries and definitions. In section 3, we present our algorithm for constructing a graph which contains the minimum cost path in a two dimensional work space, and prove that the shortest path is within the constructed graph. In section 4, we generalize the algorithm for the case of rectilinear regions in three dimensions, and in section 5, we draw a conclusion.

2 Preliminaries and Definitions

The problem of weighted region path planning, WRP, considered in this paper is defined as follows: let $\mathcal{S}$ be

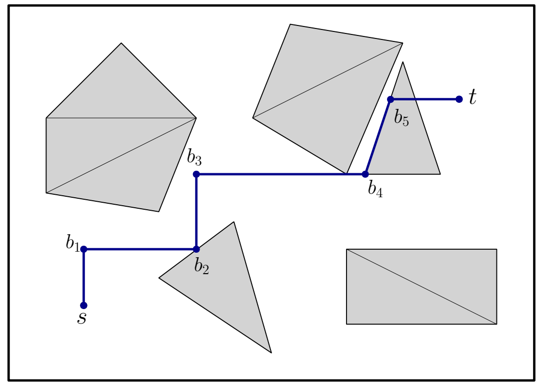

Figure 1: A path from s to t with seven breakpoints.

a subdivision of the plane into polygonal regions with n vertices, and $s, t \in \mathcal{S}$ be two start and destination points in the plane. Each region of $\mathcal{S}$ has an associated positive weight. The weight of an edge $e \in \mathcal{S}$ (boundary of regions) is assumed to be $\min\{w_r, w_{r'}\}$, where w_r and $w_{r'}$ are the weights of regions incident to e. The goal is to find a minimum cost path between s and t, where the distances are measured under the weighted Manhattan metric – the length of a path is the weighted sum of Manhattan lengths of the sub-paths within each region.

Let π_{st} denote a path between s and t which consists of some sub-paths between consecutive *breakpoints*. A breakpoint is a point on the path in which the path turns. We also consider s and t as breakpoints (see Fig. 1). Let $\rho_1, \rho_2, ..., \rho_k$ be sub-paths between consecutive breakpoints of a path π_{st} in which each ρ_i, for $i = 1, 2, \ldots, k$ lies completely within one region. If a part of a path π_{st} does not lie totally in one of the regions, we decompose it to some sub-paths. We denote $d(\rho_i)$ as the Manhattan distance between two endpoints of ρ_i. The weighted length of a path π_{st} under the Manhattan metric, denoted by $d_w(\pi_{st})$, is defined as:

$$d_w(\pi_{st}) = \sum_{i=1}^{k} d(\rho_i) \times w_i,$$

where w_i is the weight of the region in which ρ_i lies.

A path π_{st} is called a horizontal (resp., vertical) path if it consists of a horizontal (resp., vertical) sub-path between only two consecutive breakpoints. Also, we say two horizontal (resp., vertical) paths are consecutive if and only if they have the same starting and termination points. This definition is used in Lemma 1.

The basic idea behind the proposed algorithm is reducing the problem to a graph searching problem. Therefore, we provide an algorithm for constructing a graph that contains the minimum cost path under the weighted Manhattan metric. The constructed graph is a planar graph with $O(n^2)$ vertices and edges, where n is the total number of vertices of the subdivision. For planar graphs with positive edge weights, Henzinger et al. [12] have given a linear-time algorithm to compute single-source shortest paths. By running this algorithm on the constructed graph, we obtain the minimum cost path between s and t under the Manhattan metric in $O(n^2)$ time. Since a simple polygon with n vertices can be triangulated in $O(n \log n)$ time and $O(n)$ space [8], w.l.o.g. we assume all the regions to be triangular regions in all parts of the paper.

3 The Graph Construction Algorithm

3.1 The Algorithm

Let $\mathcal{G} = (V, E)$ be a graph. First, we initialize $V = \emptyset$ and $E = \emptyset$. Let $HL(\alpha_i)$ and $VL(\alpha_i)$ be horizontal and vertical lines passing through point α_i, for $i = 1, 2, \ldots, n$. Precisely, α_i, for $i = 1, 2, \ldots, n$ are the vertices of the subdivision which contain s, t, and the vertices of the triangles. We add s, t, vertices of the triangles, and the intersection points among $HL(\alpha_i)$ and $VL(\alpha_j)$, for $i, j = 1, 2, \ldots, n$ to V. We also add the intersection points among $HL(\alpha_i)$ (resp., $VL(\alpha_i)$), for $i = 1, 2, \ldots, n$ and the edges of the triangles to V. Next, we add the line segments between two consecutive vertices in V that lie on the considered horizontal lines, vertical lines or the edges of the triangles as edges of $\mathcal{G}$ to E. For an edge $(u, v) \in E$ where lies in a region with wight w_i, let $d(u, v)$ denote the Manhattan distance between two endpoints of the edge. The weight of the edge is equal to the product of $d(u, v)$ and w_i. Note that each edge lies completely within one region.

The basic idea of our algorithm is to extend four rays to the up, down, right and left directions (horizontal and vertical lines) at every vertex of the subdivision. This idea has similarity to vertical cell decomposition (VCD) method [14]. In this method, the free space is partitioned into a finite collection of one-dimensional and two-dimensional cells by extending rays upward and downward through free space. In this method, the rays are not allowed to enter obstacles, however, in our algorithm the rays are extended to all parts of the subdivision since the paths are allowed to enter weighted regions at extra costs. Also, we extend rays to the four directions at every vertex, however, in the VCD method the rays are extended only upward and downward. In both methods, the motion planning problem is reduced to a graph search problem. In VCD method, a roadmap is constructed by selecting sample points from the cell centroids, however, in our algorithm the graph is constructed by intersecting the rays with each other and also by the edges of the triangles.

Some of the edges of $\mathcal{G}$ which lie on an edge of a triangle are oblique. These edges are useful when two triangular regions are close to each other and the region among them has a lower weight than these triangles. A path which passes between these two triangles cannot be completely horizontal or vertical since it will enter the triangles. So it will be oblique and lie on one of the edges of the triangles (see the sub-path between b_4 and b_5 on Fig. 1).

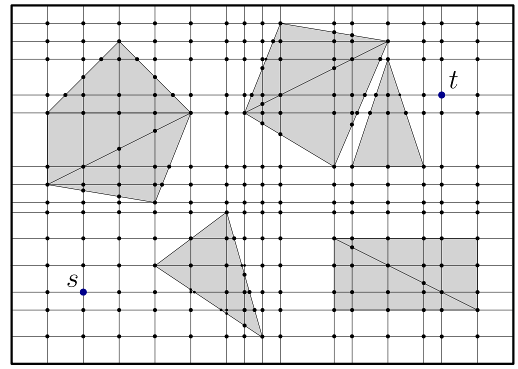

Figure 2: The constructed graph of Fig. 1.

According to the construction of the graph, some vertices and edges are added to the graph by vertical and horizontal lines passing through vertices of the subdivision. We call the part of the work space which lies between two consecutive horizontal (resp., vertical) lines, a *horizontal lane* (resp., *vertical lane*) denoted by LH (resp., LV). So each LH (resp., LV) is surrounded by two consecutive horizontal (resp., vertical) lines. Therefore, when we say the lines of an LH (resp., an LV), we mean these consecutive lines.

For constructing the graph, we can use one of the line segments intersections algorithms [3, 5] which computes all k intersections among n line segments in the plane in $O(n \log n + k)$ time. These intersection points are vertices of $\mathcal{G}$. After specifying the set of vertices of $\mathcal{G}$, the set of edges of $\mathcal{G}$ can be specified. It takes $O(n^2)$ time to construct $\mathcal{G}$ since the graph has $O(n^2)$ vertices and edges. The constructed graph of the work space on Fig. 1 is shown on Fig. 2. For simplicity, we do not triangulate the white regions with weight 1 in these figures. Precisely, we can apply the proposed algorithm in a polygonal subdivision in which the regions are not triangular. The triangulation of the regions just helps us for showing that $\mathcal{G}$ contains the minimum cost path between s and t.

For computing the minimum cost path under the Manhattan metric between s and t, we can apply Dijkstra's algorithm to $\mathcal{G}$. In this case, the minimum cost path is obtained in $O(n^2 \log n)$ time. However, since $\mathcal{G}$ is a planar graph with positive edge weights, we can apply the algorithm presented by Henzinger et al. [12], which is a linear-time algorithm, to $\mathcal{G}$. Therefore, the minimum cost path is obtained in $O(n^2)$ time.

3.2 Correctness Proof

Now, we show that the constructed graph contains the minimum cost path between s and t under the Manhattan metric. Since our metric for measuring the distance is Manhattan, we can convert any path between s and t to a path which consists of vertical and horizontal line segments. In other words, when a sub-path between two consecutive breakpoints is oblique, we can replace it by two horizontal and vertical line segments where the cost of movement on these horizontal and vertical line segments is equal to the cost of movement along the oblique line segment. In a case where a sub-path lies between two close triangular regions and the region between these two triangular regions has lower weight than these triangles, by applying this conversion, some parts of the horizontal and vertical line segments may lie in the triangular region with higher weight. In this case, we can replace the part which lies in a triangular region with higher cost with a line segment which lies on an edge of the triangles (see the sub-path between b_4 and b_5 on Fig. 1). Since the weight of each of the edges of the work space is equal to the minimum weight of the regions that are incident to that edge, the cost of movement between two breakpoints on the replaced line segments is equal to the cost of movement along the oblique line segment. Therefore, a path between s and t can only consist of horizontal, vertical, and oblique line segments, the latter of which are located on the edges of the triangles. As a result, all the paths that we consider in the following lemmas consist of the above mentioned line segments. Our first objective is to prove the following lemma.

Lemma 1 *Let π_1, π_2, and π_3 be three consecutive horizontal (or vertical) sub-paths from s' to t' which lie inside an LH (resp., an LV) and pass through $k > 0$ triangular regions. If $d_w(\pi_2) < d_w(\pi_1)$, then $d_w(\pi_3) < d_w(\pi_2)$.*

Proof. We consider the case $k = 2$, the proof is similar for any $k > 0$. For simple comparison among the sub-paths, let the points s' and t' lie on the same horizontal line segment. Assume w.l.o.g. that both triangles have vertical edges (see Fig. 3). The weighted lengths of π_1, π_2 and π_3 are defined as follows (refer to Fig. 3 for the notations):

$$d_w(\pi_1) = (w_1 \times a_1) + (w_2 \times a_2) + z_2 + x_2 + L,$$

$$d_w(\pi_2) = (2 \times h) + x_1 + (w_1 \times b_1) + (w_2 \times b_2) + x_2 + L,$$

$$d_w(\pi_3) = (2 \times h) + x_1 + (2 \times h') + z_1 + (w_1 \times c_1) + (w_2 \times c_2) + L.$$

According to Fig. 3, $a_1 = b_1 + x_1$ and $a_2 = b_2 - z_2$. Due to the assumption that $d_w(\pi_2) < d_w(\pi_1)$, we have the following inequality:

$$(2 \times h) < x_1 \times (w_1 - 1) + z_2 \times (1 - w_2),$$

and due to the triangle similarity theorems we have the following equations:

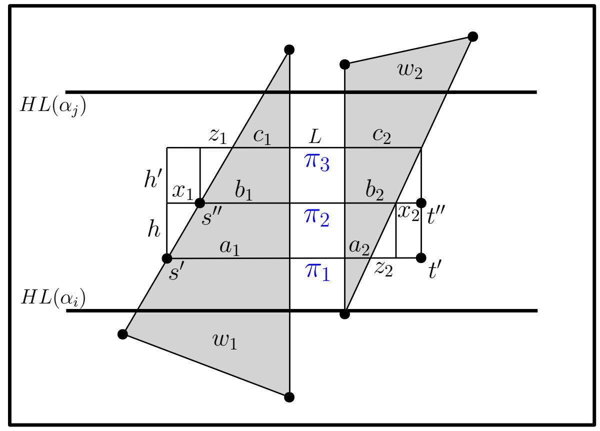

Figure 3: Three consecutive horizontal sub-paths from s' to t' through two triangular regions.

$$\frac{x_1}{h} = \frac{z_1}{h'}, \frac{z_2}{h} = \frac{x_2}{h'}.$$

By applying the triangle similarity equations in the mentioned inequality and adding $(w_1 \times c_1) + (w_2 \times b_2)$ to both sides of the inequality we get:

$$(2 \times h') + z_1 + (w_1 \times c_1) + (w_2 \times c_2) < (w_1 \times b_1) + (w_2 \times b_2) + x_2 \Longrightarrow d_w(\pi_3) < d_w(\pi_2).$$

Thus, the weighted length of π_3 is less than π_2. In fact, the proof is based on the following equation:

$$\frac{h}{h'} = \frac{x_1}{z_1} = \frac{z_2}{x_2},$$

and since $\frac{h}{h'}$ is constant, we can generalize the proof for any $k > 0$ triangular regions between s' and t'. Therefore, the lemma holds. □

Note that inside an LH (resp., an LV), we can consider all the triangles to have vertical (resp., horizontal) edges since vertical (resp., horizontal) lines are considered passing through vertices of the subdivision. The result of this lemma helps us to show that there exists a shortest path between s and t under the Manhattan metric such that all the horizontal (resp., vertical) sub-paths between consecutive breakpoints in *LHs* (resp., *LVs*) lie on the lines of the *LHs* (resp., *LVs*). We call such a path, a *perfect shortest path* between s and t, denoted by π^p_{st}. Note that according to the principle of optimality, since π^p_{st} is optimal in length, all of its sub-paths in *LHs* and *LVs* are also optimal in length.

Lemma 2 *There exists a shortest path between s and t under the Manhattan metric such that, for any sub-path of the shortest path in an LH (resp., an LV), all the horizontal (resp., vertical) sub-paths between consecutive breakpoints lie on the lines of the LH (resp., LV).*

According to Lemma 2, a path between the entrance (s') and exit point (t') of an LH (resp., an LV) is not optimal in length, unless there exists an optimal path in length such that all the horizontal (resp., vertical) sub-paths between consecutive breakpoints lie on the lines of the LH (resp., LV). Precisely, there is always a path $\pi^p_{s't'}$ in an LH (resp., an LV). According to the construction of the graph, lines of an LH (resp., an LV) are edges of $\mathcal{G}$ and a horizontal (resp., vertical) sub-path of a path $\pi^p_{s't'}$ between two consecutive breakpoints in an LH (resp., an LV) lies on the edges of $\mathcal{G}$.

Corollary 3 *For any path $\pi^p_{s't'}$ in an LH (resp., an LV), the sub-paths between consecutive breakpoints cannot be simultaneously horizontal (resp., vertical) and lie between two lines of the LH (resp., LV).*

Lemma 4 *A breakpoint of a path $\pi^p_{s't'}$ in an LH (resp., an LV) is located on a line of an LH or an LV or possibly both.*

Proof. We assume that b is a breakpoint in an LH which is not located on a line of the LH or a LV. According to Corollary 3, the line segment that is incident to b cannot be horizontal. Therefore, one of the line segments is vertical and the other one is located on an edge of a triangle. Since b is also located in an LV and is not located on one of the lines of the LV, the vertical line segment incident to b lies between the left and right lines of the LV, which contradicts Corollary 3. Thus, the lemma holds. □

Lemma 4 shows that the breakpoints of the perfect shortest paths in *LHs* (resp., *LVs*) must lie on the lines of the *LHs* and *LVs*, meaning that they lie on the edges of $\mathcal{G}$ (since the lines of *LHs* and *LVs* are edges of G). The next step is to show that these breakpoints are located on the vertices of $\mathcal{G}$.

Lemma 5 *For a path $\pi^p_{s't'}$ in an LH (resp., an LV), the breakpoints of the path are located on the vertices of $\mathcal{G}$.*

Proof. According to Lemma 4, a breakpoint of a path $\pi^p_{s't'}$ in an LH (resp., an LV) is located on a line of an LH or an LV or possibly both. If a breakpoint is located on both a line of an LV and a line of an LH, it is on the intersection point of these two lines. Thus, it is on a vertex of $\mathcal{G}$. If it is only located on a line of an LH or an LV, and one of the incident line segments lies on a triangle edge, then the breakpoint is located on a vertex of $\mathcal{G}$ (since the intersection of an LH or LV line with a triangle edge is a vertex of $\mathcal{G}$). Therefore, the breakpoints of a path $\pi^p_{s't'}$ are on the vertices of $\mathcal{G}$. □

Lemma 5 shows that the breakpoints of a path $\pi^p_{s't'}$ in an LH (resp., an LV) are located on the vertices of $\mathcal{G}$. The next step is to show that a path $\pi^p_{s't'}$ under the Manhattan metric in an LH (resp., an LV) is on $\mathcal{G}$. To this end, we need to show that the edges of the path $\pi^p_{s't'}$ are on the edges of $\mathcal{G}$.

Lemma 6 *A path $\pi^p_{s't'}$ in an LH (resp., an LV) is on $\mathcal{G}$.*

Proof. According to Lemma 5, the breakpoints of a path $\pi^p_{s't'}$ in an LH (resp., an LV) are on the vertices of $\mathcal{G}$. Let e be an edge between two consecutive breakpoints. If e is on an edge of a triangle, it is on $\mathcal{G}$. Now we assume that e is in an LH and is not on $\mathcal{G}$. According to Corollary 3, e cannot be horizontal since it must lie on one of the lines of the LH and the lines of LHs are edges of $\mathcal{G}$. Therefore, it is a vertical edge. Since it is also located in an LV and is not on $\mathcal{G}$, it is not on a line of the LV. Therefore, it contradicts Corollary 3. Thus, e is on $\mathcal{G}$. □

According to Lemma 6, perfect shortest paths in *LHs* and *LVs* which are sub-paths of a path π^p_{st} are on the constructed graph. Note that in all the lemmas, a path between s and t only consists of horizontal, vertical, and oblique line segments, the latter of which are located on the edges of the triangles. In the continuous work space, an arbitrary path between s and t consists of line segments which are not in the form of the mentioned line segments. Finally, we prove that there exists a shortest path between s and t on $\mathcal{G}$.

Theorem 7 *For a shortest path π_1 under the weighted Manhattan metric in the continuous work space from s to t, there exists a path π_2 from s to t on $\mathcal{G}$ such that $d_w(\pi_2) \leq d_w(\pi_1)$.*

Proof. It is obvious that when the metric for measuring the distance is Manhattan, any arbitrary path in the continuous work space, can be converted to a path which consists of the three mentioned line segments without increment in the cost of the path. Thus, we convert π_1 to π'_1 such that the line segments in π'_1 are in the form of the mentioned line segments. Obviously, $d_w(\pi'_1) = d_w(\pi_1)$. According to the principle of optimality, each sub-path of an optimal path in length is also optimal. Therefore, π'_1 consists of optimal sub-paths in length in *LHs* and *LVs*. According to Lemma 2, for any shortest path in an LH (resp., an LV), there exists a path $\pi^p_{s't'}$ and due to the Lemma 6, perfect shortest paths in *LHs* and *LVs* are on $\mathcal{G}$. Thus, π'_1 can be converted to a perfect shortest path (π_2) without increment in the cost of the path. Therefore, a path from s to t on $\mathcal{G}$ exists (π_2) whose weighted length is not greater than π_1. □

According to Theorem 7, $\mathcal{G}$ contains a shortest path from s to t under the weighted Manhattan metric. Since simple polygons can be triangulated in $O(n \log n)$ time and $O(n)$ space [8], work spaces with simple polygonal regions can be discretized by using the mentioned graph construction algorithm. Thus, the proposed algorithm solves WRP under the Manhattan metric.

Theorem 8 *The weighted region problem in a planar polygonal subdivision with positive weights under the Manhattan metric can be solved in $O(n^2)$ time and space, where n is the total number of vertices of the subdivision.*

By using the triangular inequality, it is easy to see that the length of a path under the Manhattan metric is at most $\sqrt{2}$ times of the length of the path under the Euclidean metric. Thus, the proposed algorithm is also a $\sqrt{2}$-approximation algorithm for solving WRP under the Euclidean metric.

4 The Three-Dimensional Case

In this section, we consider WRP in three dimensions. It has been shown that the problem of finding a shortest path under any L^P metric in a three-dimensional polyhedral environment is NP-hard [4]. Here, we consider a specific variation where the regions are rectilinear.

Since the metric for measuring the distance is Manhattan, any oblique path between two consecutive breakpoints in three-dimensional space can be converted to three parallel line segments to x, y and z axes without increment in the cost of the path. Thus, we consider all the paths to be rectilinear.

Let n be the total number of vertices of the subdivision and let (x_i, y_i, z_i), for $i = 1, 2, \dots, n$ be the coordinates of the vertices of the regions (and of s and t). Let $\mathcal{P}$ be the set of planes $x = x_i$, $y = y_i$, $z = z_i$, for $i = 1, 2, \dots, n$. The set of vertices of the graph consists of the intersection points among at least three planes in $\mathcal{P}$, and the set of edges of the graph consists of the line segments between two consecutive vertices of the graph which lie on the intersection lines between at least two planes in $\mathcal{P}$. The constructed graph has $O(n^3)$ vertices and edges, and by applying Dijkstra's algorithm to it, the minimum cost path under the Manhattan metric can be obtained in $O(n^3 \log n)$ time.

Similar to the definitions of LH and LV in the planar case, we define similar notations for the three-dimensional case. Let XYC denote a part of the work space which is surrounded by two consecutive planes orthogonal to the x-axis and two consecutive planes orthogonal to the y-axis in $\mathcal{P}$ which is called an $XY-container$. Precisely, an XYC is not surrounded along the z-axis. XZC and YZC notations are defined similarly. Since all the paths are considered to be rectilinear, for any path in an XYC, there exists an equivalent path in length such that all the sub-paths between consecutive breakpoints along the z-axis are located on the planes surrounding XYC. Precisely, according to the graph construction algorithm, each XYC consists of some cuboids where the cost of movement in every part of a cuboid is equal. Therefore, the sub-paths along the z-axis in a cuboid have the same cost when they are located either on the planes surrounding XYC or inside the cuboid. Similar results hold for an XZC and a YZC. Thus, an equivalent path in length between s and t exists where all the sub-paths between consecutive breakpoints are located on the considered planes in $\mathcal{P}$. Arguments similar to the ones used in Theorem 7 show that the constructed graph contains the minimum cost path between s and t under the Manhattan metric.

Theorem 9 *The weighted region problem in a three-dimensional work space among rectilinear regions with positive weights under the Manhattan metric can be solved in $O(n^3 \log n)$ time and $O(n^3)$ space, where n is the total number of vertices of the subdivision.*

5 Conclusion

In this paper, we have considered a generalization of path planning problem – called *weighted region problem* (WRP). While unsolvability of WRP over the rational numbers under the Euclidean metric has been proved [9], we proposed an algorithm for solving WRP under the Manhattan metric which is also a $\sqrt{2}$-approximation solution for the Euclidean case. We also considered the case of rectilinear regions in three dimensions, and generalized our algorithm for it. Improving the time complexity of the algorithm and providing a better approximation factor for the Euclidean metric remain open.

References

[1] L. Aleksandrov, M. Lanthier, A. Maheshwari, and J.-R. Sack. An ε—approximation algorithm for weighted shortest paths on polyhedral surfaces. In *Scandinavian Workshop on Algorithm Theory*, pages 11–22. Springer, 1998.

[2] L. Aleksandrov, A. Maheshwari, and J.-R. Sack. Determining approximate shortest paths on weighted polyhedral surfaces. *Journal of the ACM (JACM)*, 52(1):25–53, 2005.

[3] I. J. Balaban. An optimal algorithm for finding segments intersections. In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 211–219, 1995.

[4] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 49–60. IEEE, 1987.

[5] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *Journal of the ACM (JACM)*, 39(1):1–54, 1992.

[6] J. Chestnutt, K. Nishiwaki, J. Kuffner, and S. Kagami. An adaptive action model for legged navigation planning. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 196–202. IEEE, 2007.

[7] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of robot motion: theory, algorithms, and implementation.* MIT press, 2005.

[8] M. de Berg, M. van Kreveld, M. Overmars, and O. C. Schwarzkopf. Polygon triangulation. In *Computational Geometry*, pages 45–61. Springer, 2000.

[9] J.-L. De Carufel, C. Grimm, A. Maheshwari, M. Owen, and M. Smid. A note on the unsolvability of the weighted region shortest path problem. *Computational Geometry*, 47(7):724–727, 2014.

[10] L. P. Gewali, A. C. Meng, J. S. Mitchell, and S. Ntafos. Path planning in $0/1/\infty$ weighted regions with applications. *ORSA Journal on Computing*, 2(3):253–272, 1990.

[11] A. Gheibi, A. Maheshwari, and J.-R. Sack. Weighted region problem in arrangement of lines. In *CCCG*, 2013.

[12] M. R. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *journal of computer and system sciences*, 55(1):3–23, 1997.

[13] N. Jaklin, M. Tibboel, and R. Geraerts. Computing high-quality paths in weighted regions. In *Proceedings of the Seventh International Conference on Motion in Games*, pages 77–86, 2014.

[14] S. M. LaValle. *Planning algorithms.* Cambridge university press, 2006.

[15] D. Lee, C.-D. Yang, and T. Chen. Shortest rectilinear paths among weighted obstacle. *International Journal of Computational Geometry & Applications*, 1(02):109–124, 1991.

[16] C. S. Mata and J. S. Mitchell. A new algorithm for computing shortest paths in weighted planar subdivisions. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 264–273, 1997.

[17] J. S. Mitchell and C. H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *Journal of the ACM (JACM)*, 38(1):18–73, 1991.

[18] Z. Sun and J. H. Reif. On finding approximate optimal paths in weighted regions. *Journal of Algorithms*, 58(1):1–32, 2006.

Appendix

Proof of Lemma 2

Lemma 2 *There exists a shortest path between s and t under the Manhattan metric such that, for any sub-path of the shortest path in an LH (resp., an LV), all the horizontal (resp., vertical) sub-paths between consecutive breakpoints lie on the lines of the LH (resp., LV).*

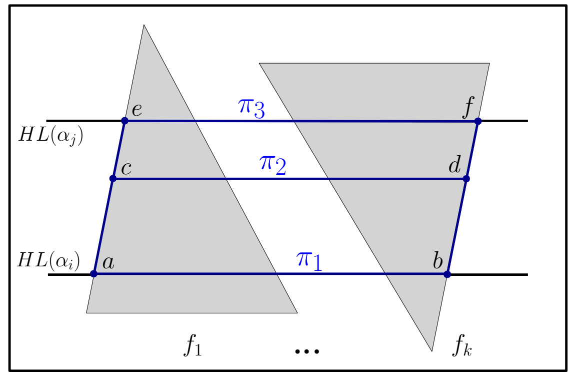

Figure 4: Three horizontal paths passing through k triangular regions.

Proof. Suppose the lemma for the case of a horizontal lane. Similarly, the lemma holds for a vertical lane. We consider s' as the entrance point to the LH and t' as the exit point. W.l.o.g. we consider that s' is on the left side of t'. Due to the assumption that the path between s and t is optimal in length, any sub-path of this path is also optimal in length. Thus, the path between s' and t' is optimal in length. We consider a path between s' and t' where a horizontal sub-path between two consecutive breakpoints does not lie on the lines of the LH. We show that there exists an equivalent path in length between s' and t' such that all the horizontal sub-paths between consecutive breakpoints lie on the lines of the LH. We assume c and d as two consecutive breakpoints such that the horizontal sub-path between them does not lie on the lines of the LH (see Fig. 4). There are k triangular regions between c and d and the sub-path between these two breakpoints must pass all k triangular regions (w.l.o.g. assume c and d are located on the edges of the triangles). We also assume that the path between s' and t' contains other two breakpoints – we call them a and b – which are on the lower line of the LH (these two breakpoints are also located on the edges of the triangles). For passing these triangles, a path can directly go from a to b. Since the path between s' and t' is optimal in length, the path which contains c and d (π_2) has less than or equal length to the case in which it goes directly from a to b (π_1). If $d_w(\pi_1) = d_w(\pi_2)$, an equivalent path in length which does not contain the horizontal path between c and d exists. If $d_w(\pi_1) < d_w(\pi_2)$, it contradicts our assumption that the path between s and t is optimal in length. For the other case where $d_w(\pi_2) < d_w(\pi_1)$, we consider another path which goes from a to e (a breakpoint on the upper line of the LH and on the edge of the left most triangle) and then from e to f (a breakpoint on the upper line of the LH and on the edge of the right most triangle) and then to b (π_3). According to Lemma 1, since $d_w(\pi_2) < d_w(\pi_1)$, therefore, $d_w(\pi_3) < d_w(\pi_2)$ and this contradicts our assumption that the path between s and t is optimal in length. Thus, the lemma holds. □