Path Planning with Objectives Minimizing Length and Maximizing Clearance

Mansoor Davoodi* Maryam Sanisales*

Abstract

In this paper, we study the problem of bi-objective path planning with the objectives minimizing the length and maximizing the *clearance* of the path, that is, maximizing the minimum distance between the path and obstacles. We consider a set of vertical segments as the obstacles and propose an efficient algorithm for finding all intervals of Pareto optimal solutions when the fist objective is evaluated with Euclidean metric and the second one is evaluated by Manhattan metric. Finally, we show that the algorithm results in finding $(\sqrt{2}, 1)$-approximation Pareto optimal solutions when both objectives are evaluated with Euclidean metric.

1 Introduction

Path Planning (PP) is one of the challenging problems in the field of robotics. The goal is to find optimal path(s) for two given start and destination points among a set of obstacles. However, usually minimizing the length of the the path is considered as the optimality criterion, regarding the application of the problem other objectives such as smoothness and clearance, that is maximizing the distance between the path and obstacles, have been also considered in the literature [2]. For example, in many applications, the robot needs to move around in order to perform its task properly. The need to move around the environment led to the question of what path a robot can take to accomplish its task, in addition to being safe. In this paper, we define the *optimal path* regarding to two objectives minimizing the length of the path and maximizing the minimum distance between the path and obstacles.

A classical approach to compute the minimum length path is computing the visibility graph of obstacles and convert the problem to a graph search problem. For a set of n obstacle vertices, the visibility graph can be computed in $O(n^2 \log n)$ time using a tree structure and ray technique [4]. This approach is one of the best-known algorithms to obtain the shortest path where the distance between the path and the obstacle is equal to zero– a path with clearance zero. Also, Hershberger et al [5] proposed an efficient planar structure for PP problem in $O(n \log n)$ time.

*Department of Computer Sciences and Information Technology, Institute for Advances Studies in Basic Sciences, Zanjan, Iran. mdmonfared@iasbs.ac.ir ; maryamsan@iasbs.ac.ir

Wein [7], by using a combination of the Voronoi diagram and the visibility graph introduced a new type of visibility structure called *Voronoi Visibility Diagram* to find the shortest path for a predefined value λ of clearance. He considered the PP problem in the setting of single objective optimization problem with the objective minimizing the length subject to minimum clearance λ. Geraerts [3] proposed a new data structure called *Explicit Road Map* that creates the shortest possible path with the maximum possible clearance. The introduced structure is useful for computing the path in the corridor spaces. Davoodi [1] studied the problem of bi-objective PP in grid with the two objectives of minimizing path length and maximizing *clearance* and then showed that Pareto optimal solutions to the two-objective problem in the grid workspace. He also studied the problem in the continuous space under Manhattan metric, and proposed an $O(n^3)$ time algorithm where the obstacles are n vertical segments. The problem under Euclidean metric was remain as an open problem.

We study the problem of bi-objectives PP in continuous space with the objectives minimizing the length of the path and maximizing its minimum distance from obstacles. The goal is computing Pareto optimal solutions, that are the solutions which cannot be shortened if and only if their clearance is minimized– Since this problem is a bi-objective optimization problem in the continuous space, there an infinite number of Pareto optimal solutions. So, it is impossible to provide a polynomial algorithm to compute all the solutions. To smooth this issue, we focus on different Pareto optimal solutions, the optimal paths with different middle points –called *Distinct Pareto Optimal* paths. We consider a PP search space with n vertical segments as obstacles and propose an $O(n^3)$ time algorithm to compute all distinct Pareto optimal solutions where the length of the paths is evaluated with Euclidean and the clearance is evaluated with Manhattan Metric. Then, we show that the solutions are efficient approximation solutions of the problem where the both objectives are evaluated under Euclidean metric.

The next section briefly introduces bi-objective path planning problem. Section 3 proposes exact algorithm for computing distinct Pareto optimal solutions when the length and clearance objectives are evaluated under Euclidean and Manhattan metric, respectively. Section 4 extends the results to the case both the objectives are evaluated under Euclidean distance.

2 Bi-objective Path Planning Problem

Let $O = \{s_1, s_2, ..., s_n\}$ be a set of left to right sorted vertical segments as the obstacles, and s and t be the start and destination points in the plane. Assume, w.l.o.g., that s and t lie on the left and right side of the obstacles, respectively. Let the shortest path from s to t denote by *s-t-path*.

For a collision-free path $P = \{s = v_0, v_1, \cdots, v_m, v_{m+1} = t\}$ with m *breakpoints* $v_1, v_2, \cdots, v_m$, define $L(P)$ as the length of P under Euclidean metric. Also, define $C_1(P)$ and $C_2(P)$ as the *clearance* of P under Manhattan and Euclidean metrics, respectively, that is minimum distance between P and the obstacles. We denote the clearance of P with $C(P)$ in general, when the metric is not the case. The objectives are minimizing $L(P)$ and maximizing $C(P)$ in the bi-objective path planning problem.

For two collision-free paths P and P', we call P *dominates* P' and denote it by $P \preceq P'$, if $L(P) < L(P')$ and $L(P) \geq L(P')$, or if $L(P) \leq L(P')$ and $L(P) > L(P')$. For any pair of paths P and P', three cases may happen, (i) $P \preceq P'$, (ii) $P' \preceq P$ and (iii) none of them dominates. in the third case we call P and P' are *non-dominated*. That means, there is no strict preference between P and P'.

Given the definition of dominate, paths such as P which is not dominated by other collision-free path, are called *Pareto optimal paths*. Indeed, any improvement of P in its length or clearance comes from scarifying it in the other objective. Let Π^* be the set of all Pareto optimal path. There is a Mapping from the path planning workspace to the objective space $L - C$. We also call the projection of Π^* in $L - C$ space *Pareto Front*. For any small clearance value λ, there is some shortest path in the workspace [1]. Therefore, the projection of Pareto front of the problem on *C-space* is one component, while it is possible the projection of Pareto front of the problem on *L-space* are several components.

Since the workspace of bi-objective path planning is continuous, Pareto front is infinite set in general. Thus, there is no algorithm to construct path is Π^* one-by-one. Two approaches are proposed to handle this issue in this paper:

- Finding all different (or *Distinct*) Pareto optimal paths, the path with different breakpoints in the workspace. In other words, the Pareto front is a set of discrete components. We can compute the *extreme* solution of each component.
- Finding a set of finite and polynomial size of solutions which are an approximation set of all Pareto fronts.

The first approach will result in finding different intervals $I_1, I_2, ..., I_k$, for some k, of clearance. We will compute the lower and upper bound of the intervals and map it with a set of shortest paths with *almost* same breakpoints. We use this approach and solve the problem of bi-objective path planning for solving the problem when clearance of the path is measured with Manhattan metric. Also, we show these computed paths are approximation solutions for the case length and clearance of the paths are measured with Euclidean metric.

3 Bi-Objective Path Planning with Euclidean Length and Manhattan Clearance

Let us explain our approach to solve the bi-objective path planning algorithm roughly at first. We construct a general map for computing a tree of shortest path – called *SPM*– with clearance $\lambda = 0$ by using the idea proposed by Lee and Preparata [6]. The idea is using a *sweepline* from the start point s to the destination point t algorithm based on monotonicity of the shortest. While the sweep-line moves from left to right, the left side vertices of the segments construct a tree with the root s and the parent of each vertex p is the vertex p' such that p' is the last breakpoint in the shortest path between s and p. Also, in each step the right side of the sweep line is decomposed to a set of regions with the property that te points lie inside a region has the same parent in the left side of the sweepline. When the sweepline reaches the destination point t, the shortest path (with clearance $C_1(P) = \lambda = 0$) can be easily computed from s to t by a simple backward manner. After computing such a general map –called SPM(0)– we increase value of λ from 0 to ∞ to compute the Pareto front intervals. To this end, we compute all critical λ may change the shortest path tree, particularly the breakpoints of *s-t-paths*. Since, the obstacles of the workspace, $O = \{s_1, s_2, ..., s_n\}$, are vertical segments, the following observation is clear.

Observation 1 *The shortest path from s to the endpoints of the obstacles (and also to t) is an $x-$monotone path.*

For finding the shortest path with clearance λ, we need to expand (or fat) the obstacles with size λ. Define $O(\lambda) = \{s_1(\lambda), s_2(\lambda), \cdots, s_n(\lambda)\}$, where $s_i(\lambda)$ is the segment s_i after fatting with size λ under Manhattan distance. That is, if $\oplus$ shows the Minkowski sum of two objects, $s_i(\lambda) = s_i \oplus Sq(\lambda)$, where $Sq(\lambda)$ is a square with the diameter size 2λ which rotates $\pi/4$. Indeed, the boundary of $Sq(\lambda)$ is the set of all points in the plane whose Manhattan distance from the center of $Sq(\lambda)$ is exactly λ.

Observation 2 *The breakpoints of any s-t-path with clearance λ belong to the set $O(\lambda)$.*

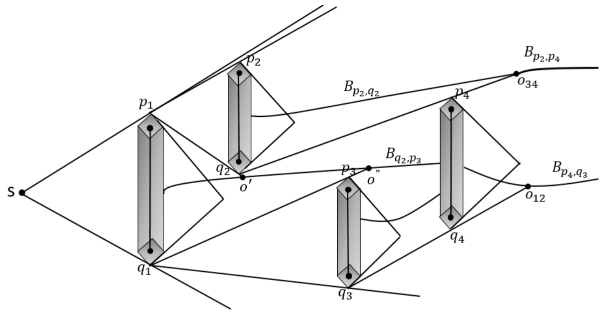

Figure 1: The SPM map for four obstacles with visibility edges, SPM-point and bisectors.

Based on Observation 2, our strategy in finding the Pareto optimal intervals is first computing SPM for $O(\lambda = 0)$. Then by increasing λ, we handle the events may change SPM and the shortest path from s to t. To this end, we construct a data structure that can handle the events and create the shortest possible path and report the paths that are distinct. The tree defines the shortest path on the set $O(\lambda) \cup \{s, t\}$ with root s. We will explain below important features of the tree [6].

3.1 SPM(0), the Shortest Path Map for $O(\lambda = 0)$

SPM(λ =0) is an incremental constructed tree at root s which is obtained by a sweepline strategy and contains the shortest path from s to any obstacle's vertices. Suppose SPM(0) is available for the obstacles $s_1 = \overline{p_1q_1}, \cdots, s_{i-1} = \overline{p_{i-1}q_{i-1}}$. Each node p_j (or q_j), for $j < i$ has a particular $weight(p_j)$ that shows the length of s-p_j-path. The right halfplane of s_{i-1} is decomposed to a set of regions corresponding with a node in SPM(0) tree as its parent.

When the sweepline meets obstacle $s_i = \overline{p_iq_i}$, first, the regions which p_i and q_i lies are founded, and then they inserted as new leaves into SPM(0) with the parents corresponding with the regions. Also, their weights is computed using the weights of their parents. Finally, the decomposition of the halfplane of s_i is updated using bisector of p_i and q_i and the new visibility edges. The bisector of p_i and q_i, denoted by B_{p_i,q_i}, is the intersection of regions corresponding with p_i and q_i. That is, any points p on the right side of s_i which length of *s*-p_i-*p-path* and *s*-q_i-*p-path* are the same (see Fig.1). The points of SPM are generally of three types; the intersection points between a pair of bisectors, between bisectors, and obstacles and between bisectors and the visibility edges of the obstacles.

Theorem 1 *For a set of n vertical segments, the size of the SPM containing all points and bisectors is linear and the SPM can be constructed in $O(n \log n)$ time [6].*

3.2 SPM(λ), the Shortest Path Map for $O(\lambda > 0)$

After Computing SPM(0), by fatting the obstacles with size λ, we able to compute and report a Pareto optimal solution with clearance λ as shown in Fig.1. To find all Pareto optimal intervals, we need to consider all the distinct paths that are the endpoints of the fronts. When an increase in changes the path, some breakpoints may change, in which case an event occurs.

$s_i(\lambda) = s_i \oplus Sq(\lambda)$ has six vertices and they can be easily computed as linear functions respect to parameter λ, e.g., if p_i and q_i are the top and bottom endpoints of s_i, the highest and the lowest vertices of $s_i(\lambda)$ can be shown $p_i(\lambda) = y_{pi} + \lambda$ and $q_i(\lambda) = y_{qi} - \lambda$, where y_{pi} and y_{qi} denotes the y coordination of p_i and q_i, respectively. Note that, the other four vertices of s_i play a local changes in the space decomposition and can be taking to account without any increasing in the time or space of the algorithms' order. We explain the details of this issue in the Appendix. When λ increases, it is possible some shortest path change. We call such values of λ the *critical* λs that can be obtained by considering all events may change the structure of the SPM. Three types of events may occur:

1) A function $p_i(\lambda)$ (or $q_i(\lambda)$) intersects with $B_{p_i,p_k}(\lambda)$ for some i and k.

2) A function $p_i(\lambda)$ (or $q_i(\lambda)$) intersects with some visibility edges of the obstacles. When visibility edges change. The equations of these edges may also not be constant. There are two types. The first type, the visibility edges between the functions $p_i(\lambda)$ (or $q_i(\lambda)$) and $p_j(\lambda)$ That slope of edge is constant. the second type, the visibility edges between the functions $p_i(\lambda)$ and $q_j(\lambda)$ That slope of edge is linear function.

3) Two obstacles are joined while they are fatting.

To handle the last type, we simply consider the two joined neighbor obstacles as one obstacle with the topmost and bottommost functions they have. Similar to the explanation in the Appendix, it is possible to handle the differences between their x-coordinates. The size of such type of events is $O(n)$.
The first type of events occurs when a function $p_i(\lambda)$ (or $q_i(\lambda)$) intersects with the bisector $B_{p_j,p_k}(\lambda)$ for some j and k. We need to handle this case if $p_i(\lambda)$ lies on the corresponding region of one of $p_i(\lambda)$ or $q_i(\lambda)$, otherwise no change is need. So, we update SPM by inserting an edge (q_k, p_i) and update the weights of the sub tree with root q_k (See Fig. 2).
The second type of events occurs when a functions $p_i(\lambda)$ (or $q_i(\lambda)$) intersects with some $O(n^2)$ visibility edges. So, we update SPM by removing the edge (q_j, p_k) and inserting an edge (p_i, p_k) (See Fig. 3).

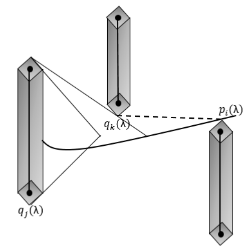

Figure 2: Illustration of the first type of events.

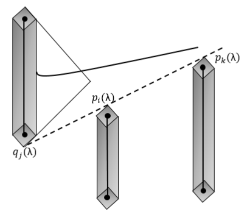

Figure 3: Illustration of the second type of events.

After each iteration of the above processes, we update the length of all shortest path from s to the endpoints of the obstacles and update the bisectors one-by-one from left to right as well as updating the critical λa in the heap to extract the minimum critical λ.

Using a heap structure, the mentioned events and critical λs can be handled efficiently. Let m be the number of breakpoints of *s-t-path*. So, using SPM(λ) *s-t-path* can be reported in $O(m)$ time for any value of λ. Also, all the *s-t-path* are represented with at most $O(n^2+R)$ breakpoints in $O(mn^2+nR)$ time, where R is the number of insertion and deletions to the heap structure, which is $O(n^2)$ in the worst case. Since $m = O(n)$, the time complexity of the algorithm is $O(n^3)$.

4 Bi-Objective Path Planning with Euclidean Length and Clearance

To solve the problem of bi-objective path planning when the both objectives are evaluated with Euclidean distance, we need to fat the obstacles using a disk with radius λ, denoted by $D(\lambda)$, instead of $Sq(\lambda)$. However, the structure of the proposed algorithm remain unchanged for computing the expanded obstacles, it difficult to update the length of the shortest path. In fact, we need to present the shortest paths as a function respect to parameter λ. When $D(\lambda)$ is used to Minkowski sum, the shortest path is obtained by tangent lines of some growing disks which is a high order function based on λ. So, in the following, we show that the proposed algorithm in the previous section provides approximation solutions under the following definition when the both objectives are evaluated under Euclidean distance.

Definition. Let Π be a bi-objective minimization problem with the objectives f_1 and f_2. A solution X is an (α, β)-*approximation* Pareto optimal solution for Π, if there is no solution Y such that $f_1(X) \geq \alpha f_1(Y)$ and $f_2(X) > \beta f_2(Y)$, or $f_1(X) > \alpha f_1(Y)$ and $f_2(X) \geq \beta f_2(Y)$.

Theorem 2 . *For the problem of bi-objective path planning under Euclidean metric, there is a* $(\sqrt{2}, 1)$*- approximation of Pareto optimal solutions.*

Proof. The proof is a straightforward result of the fact that $D(\lambda)$ can be approximated using the cocentric square $Sq(\lambda)$. □

5 Conclusion

In this paper, we considered a biobjective path planning problem with objectives minimizing the length and maximizing the clearance– That is maximizing the minimum distance between the path and objectives. We assumed the workspace contains a set of n vertical segments, and propose an efficient $O(n^3)$ time algorithm for finding Pareto optimal solutions of the problem where the length and clearance are evaluated by Euclidean and Manhattan metrics, respectively. Also, we show that such path are good approximation solutions when the both objectives are evaluated by Euclidean metric. So, an open problem remains here is proposing an efficient algorithm to find the Pareto optimal solutions for this case of the problem.

References

[1] M. Davoodi, Bi-objective path planning using deterministic algorithms. *journal of Robotics and Autonomous Systems*, 93:105-115, 2017.

[2] M. Davoodi, F. Panahi, A. Mohades, S. N. Hashemi Clear and smooth path planning, Applies Soft Computing, 32, pp. 568–579, 2015.

[3] R. Geraerts, Planning short paths with clearance using explicit corridors, *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 19972004, 2010.

[4] S. K. Ghosh and D. M. Mount, An output sensitive algorithm for computing visibility graphs, *in 28th Annual Symp. on Found. of Computer Science*, 1987.

[5] J. Hershberger and S. Suri, an Optimal Algorithm for Euclidean Shortest Paths, 28(6), pp. 2215-2256, 1999.

[6] D. T. Lee and F. P. Preparata, Euclidean shortest paths in the presence of rectilinear barriers, Networks, 14(3), pp. 393-410, 1984.

[7] R. Wein, J. P. van den Berg, and D. Halperin, The Visibility Voronoi Complex and Its Applications, Proc. twenty-first Annu. Symp. Comput. Geom. - SCG 05, 39250, pp. 63, 2005.

Appendix

Consider Fig.4 and let explain constructing the map of the shortest path for clearance λ (SPM(λ)) when we have an obstacle $l = (a, p, c, d, q, e)$.

Let $w(p)$ and $w(q)$ be the length of *s-p-path* and *s-q-path*, respectively. For any arbitrary point x lies on the boundary of the regions $R(1), R(2), R(3)$ and $R(4)$, the length of *s-x-path* is same as *s-p-x-path* or *s-q-x-path*, as defined below:

$$d(x, p) + w(p) = d(q, q) + w(q)$$

This bisector can be defined as follows:
$B_{p,q} = \{x | d(x, d) + w(q) + \sqrt{2}\lambda = d(x, p) + w(b) + \sqrt{2}\lambda\}$.

That means, in $SPM(\lambda)$ the parent of any point x lies on $B_{p,q}$ can be either p or q in $SPM(\lambda)$. However, if x lies in $R(2)$ or $R(3)$, its parent should be consider vertex c or b. Note that q is parent of c, and p is parent of b.

Also, If the point s lies in the region $R(5)$, the shortest path from s to x passes through a or d. In this case a is the parent of p, and d is the parent of q in $SPM(\lambda)$.

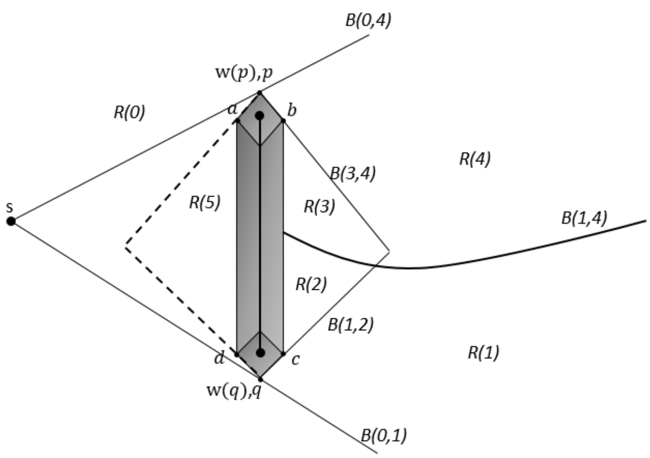

Figure 4: SPM-map when the obstacle set consists of one segment.