Complete Path Planning That Simultaneously Optimizes Length and Clearance

Basak Sakcak[1] and Steven M. LaValle[1]

Abstract— This paper considers a fundamental, optimal path planning problem that requires simultaneously minimizing path length and maximizing obstacle clearance. We show that in even simple planar settings with point and disc obstacles, the set of alternative solutions such that no one is clearly better than another (the set of Pareto-optimal solutions) is uncountably infinite. In spite of this difficulty, we introduce a complete, efficient algorithm that computes the Pareto front and a data structure that finitely represents the complete set of all Pareto-optimal paths. Particular optimal paths can then be selected from the computed data structure during execution, based on any additional conditions or considerations.

I. INTRODUCTION

Path planning is one of the fundamental problems in robotics. It refers to computing a feasible path from an initial state of the robot to a goal that avoids obstacles and satisfies the constraints introduced by the robot motion model or by the task specifications. It is also expected that the path optimizes some problem-dependent criteria.

Path length and clearance are the two most common criteria to optimize, yet they seem to be in conflict. Consider the problem of finding the shortest path for a point robot in plane, which can be computed using e.g., the visibility graph [1] or the shortest-path map [2], [3] of the obstacles. Such a length-optimal path usually grazes the obstacles, i.e., it has zero clearance (Fig. 1). This can be dangerous if the environment representation or the robot state estimation introduces uncertainties, especially for robots that move in close proximity to humans [4]. It is also shown that paths with low clearance induce discomfort to the passenger of an autonomous personal mobility vehicle [5]. Paths with maximum clearance can be obtained by computing the generalized Voronoi diagram of the obstacles [6]. However, maximizing path clearance can lead to significantly longer paths while providing only a slightly better clearance for a small portion (Fig.2).

Fig. 1. The shortest path usually grazes obstacles.

Optimizing solely path length or clearance results in paths that can be otherwise undesirable. Therefore, a significant research effort is towards finding the best tradeoff between

these two [7]. Based on the generalized Voronoi diagram of the obstacles or the skeleton of the environment, algorithms that compute short paths, while favoring paths with higher clearance are proposed (e.g., [4], [8], [9], [10], [11]). A desired clearance value can be ensured by inflating the obstacles (i.e., computing the Minkowski sum of the obstacles and a disc with corresponding radius) and computing the shortest path [12], [13]. Visibility-Voronoi complex (VV-complex) [14] implicitly contains the visibility graph and parts of the generalized Voronoi diagram for all clearance values. A weighted length metric is also proposed to query VV-complex, which minimizes path length while penalizing the vicinity to obstacles [15]. These approaches assume that the tradeoff between path length and the clearance can be made a priory. However, this is not true in general as it depends on the task specifications and on the environment. We argue that the conflicting nature of these two objectives fits under the formulation of multi-objective motion planning.
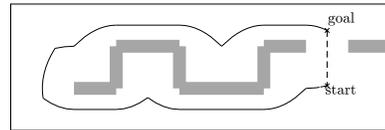

Fig. 2. A path with slightly less clearance (dashed path) can be significantly shorter than a path achieving maximum clearance for a small portion (solid path) that connects the same start and goal positions.

Multi-objective motion planning arises often in robotics; instances of this problem are tackled in the context of multiple robots with independent goals [16], legged robots [17], and human perception-optimized planning [18]. If scalarization of objectives is possible, single-objective path planning methods can be used to find a path with unique optimal cost [19]. However, for the general case, multiple paths associated with objective vectors such that not one objective could be improved without deteriorating another need to be computed. These paths are called *Pareto-optimal*. Considering a grid representation, solution strategies can be formulated as extensions to graph search algorithms [20], [21] which renders the computed Pareto-optimal paths sensitive to the underlying grid representation. In [22], a sampling-based algorithm incorporated with an evolutionary algorithm [23] is proposed to handle continuous configuration spaces. Path planning that considers path length and clearance as optimization objectives is also addressed within the multi-objective optimization framework, and algorithms based on evolutionary methods are proposed [24], [25]. Although evolutionary algorithms correspond to a standard way

of handling generic multi-objective optimization problems, they lack the formal guarantee in achieving Pareto-optimal solutions. Related to our approach, a deterministic algorithm that computes the set of all Pareto-optimal solutions, considering Manhattan distance, appears in [26].

This paper addresses path planning in plane that simultaneously optimizes path length and clearance. Rather than adopting arbitrary weights for the objectives, we propose an algorithm that characterizes the complete set of Pareto-optimal solutions. Our approach relies on the introduced Pareto-Optimal Paths Graph (POP-graph), which contains edges on the visibility graph of the inflated obstacles that can take part in the shortest path for all clearance values. Having computed all Pareto-optimal combinations of path length and clearance, one can then choose the best compromise between the two conflicting objectives, according to the application at hand, and retrieve the corresponding path from POP-graph. To the best of our knowledge, this is the first method that computes the entire set of Pareto-optimal paths considering Euclidean length and clearance.

II. Multi-objective path planning

Multi-objective path planning is the problem of finding a path that optimizes the multiple objectives while satisfying the constraints of standard path planning problems. A path, π, is a continuous mapping to the robot configuration space X, i.e., $\pi : [0,1] \rightarrow X$. There are obstacles, which are open sets and subsets of $\mathbb{R}^{2,3}$, that prohibit the robot to have certain configurations due to collisions. The set of configurations that the robot is not in collision with obstacles is defined as X_{free}. Furthermore, depending on the problem and the robot motion model, additional constraints might exist that depend on robot configuration or its derivatives or both. Then, we can define the set of valid paths Π_{valid} such that $\forall \pi \in \Pi_{valid}$, $\pi(0) = x_0$, $\pi(1) = x_1$, in which x_0 is the initial configuration of the robot and x_1 is the goal, $\pi(s) \in X_{free}$ and constraints are satisfied $\forall s \in [0,1]$.

If Π_{valid} is an empty set, there is no feasible solution to the problem, if not, the goal is to find paths in Π_{valid} that simultaneously optimize the criteria given by the vector $J \in \mathbb{R}^{d_{obj}}$ of objective functions where $J(\pi) = [J_1(\pi), J_2(\pi), \dots J_{d_{obj}}(\pi)]^T$. The vector of objective functions induce a partial ordering, denoted by $\prec$, called *dominance*. Considering the case where all the objective functions are to be minimized, this relation is expressed $\forall \pi, \pi' \in \Pi_{valid}$, as

$$\pi \prec \pi' \iff \forall i, \ J_i(\pi) \leq J_i(\pi') \land \exists k \mid J_k(\pi) < J_k(\pi')$$

in which $i, k \in \{1, \dots, d_{obj}\}$.

If there are conflicting objectives, improving the solution with respect to one objective function could result in deteriorating it with respect to another. Therefore, given two d_{obj}-dimensional vectors $J(\pi)$ and $J(\pi')$, we can not always establish a dominance relation between π and π'. A path $\pi^\star \in \Pi_{valid}$ is *non-dominated*, i.e., *Pareto-optimal*, if there is no other path in Π_{valid} that dominates $\pi^\star$. Pareto set refers to the set of all non-dominated solutions and computing this

set is the essence of multi-objective optimization. We reserve the term *Pareto front* to describe the Pareto set of solutions in the objective space.

In general, conflicting objectives result in an infinite number of Pareto optimal solutions. Therefore, a standard approach is to use evolutionary algorithms such as NSGA-II [27] and find as many diverse solutions as possible to compute an approximation of the Pareto front. However, in certain problems, it is possible to represent infinitely many solutions in terms of distinct families where each family has a representation that is invariant to the solutions that belong to it. This way, we can represent infinitely many Pareto optimal solutions using a finite number of families. In the following, we demonstrate this idea using a simple example.
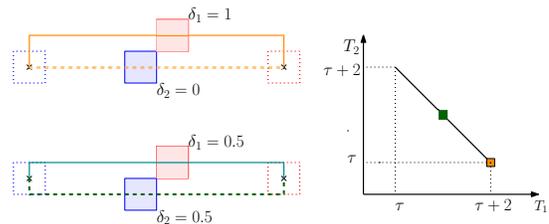

Fig. 3. Initial positions of the robots are marked with dashed lines with respective colors. Red robot follows the solid line and blue robots follows the dashed line. (left) Paths corresponding to different sacrifice values. (right) Pareto front in the objective space. Each point (green and orange squares) maps to the same colored path in the workspace.

Example 1: Consider the case where there are two unit square robots that need to exchange places (Fig. 3). Both robots can translate in $\mathbb{R}^2$ in four directions, i.e. right, left, forward and backward, in constant unit speed, and can change their direction of motion instantaneously. They have their own independent objectives: minimize arrival time to the destination. We can define a notion of *sacrifice* as the distance each robot slides sideways in the opposite direction of the other to make space. Let $\delta_{1,2} \in [0,1]$ denote the amount of sacrifice each robot makes. For instance, if robot 2 commits to going straight, then, the best solution for robot 1 is to slide 1 unit either to the left or to the right, thus, $\delta_1 = 1$ and $\delta_2 = 0$ (orange paths in Fig. 3). The arrival time of each robot is then computed as $T_{1,2} = \tau + 2\delta_{1,2}$, in which τ is the time it takes to reach the goal by going straight at constant speed. Therefore, we can represent the Pareto front in the objective space as a single family of solutions such that $\delta_2 = 1 - \delta_1$, $\forall \delta_1 \in [0,1]$.

III. Minimizing length and maximizing clearance path planning problem

We address a specific instance of multi-objective path planning, a bi-objective problem, which corresponds to finding all the non-dominated paths that simultaneously minimize the path length and maximize the path clearance. We consider a robot that is capable of translating in any direction in plane, whose position with respect to an absolute reference frame is denoted by $q \in \mathbb{R}^2$. A path π_{q_0, q_1} is a continuous curve connecting an initial position, q_0, to a goal, q_1, i.e.,

$\pi_{q_0,q_1} : [0,1] \to \mathbb{R}^2$, such that $0 \mapsto q_0$ and $1 \mapsto q_1$. The obstacles $\mathcal{O} := \{O_1, O_2, \ldots, O_n\}$, $\mathcal{O} \subset \mathbb{R}^2$, correspond to a set of points (discs) or interior-disjoint polygons. Given the distance metric $d(q_1, q_2) = \|q_1 - q_2\|$, in which $\|.\|$ is the Euclidean norm on $\mathbb{R}^2$, $J_L(\pi_{q_0,q_1})$ is the path length. Clearance of a path measures the minimum distance between the path and the obstacles, i.e.,

$$J_C(\pi_{q_0,q_1}) = \min_{s \in [0,1]} \min_{p \in \bigcup \mathcal{O}} (d(\pi_{q_0,q_1}(s), p))$$

in which p denotes a point in the obstacle set.

Let Π_{q_0,q_1} be the set of all paths that connect the initial position q_0 to goal q_1 without intersecting with the obstacles. To each path $\pi_{q_0,q_1} \in \Pi_{q_0,q_1}$, we can attribute a 2-dimensional vector $J = [J_L(\pi_{q_0,q_1}), J_C(\pi_{q_0,q_1})]^T$ that contains the path length and the clearance. As we are minimizing the path length and maximizing the clearance, path π_{q_0,q_1} dominates path π'_{q_0,q_1}, i.e., $\pi_{q_0,q_1} \prec \pi'_{q_0,q_1}$, if $J_L(\pi_{q_0,q_1}) < J_L(\pi'_{q_0,q_1})$ and $J_C(\pi_{q_0,q_1}) \geq J_C(\pi'_{q_0,q_1})$ or if $J_C(\pi_{q_0,q_1}) > J_C(\pi'_{q_0,q_1})$ and $J_L(\pi_{q_0,q_1}) \leq J_L(\pi'_{q_0,q_1})$. Based on these definitions, the problem is formulated as follows.

Minimizing length and maximizing clearance path planning problem (MinLMaxC): Find the set of all non-dominated paths $\Pi^\star_{q_0,q_1} \subseteq \Pi_{q_0,q_1}$ such that

$$\Pi^\star_{q_0,q_1} = \{\pi^\star_{q_0,q_1} \in \Pi_{q0,q1} \mid \nexists\, \pi_{q_0,q_1} \text{ for which}$$
$$\pi_{q_0,q_1} \prec \pi^\star_{q_0,q_1}\}. \quad (1)$$

In the case that the goal q_1 is visible from q_0 for all admissible clearance values such that both q_1 and q_0 have enough clearance or all the paths in Π_{q_0,q_1} have zero clearance, the set $\Pi^\star_{q_0,q_1}$ contains a single element. However, these are trivial instances and for the remaining settings, $\Pi^\star_{q_0,q_1}$ has the following property.

Proposition 1: There exists an interval $[c_0, c_1] \subset \mathbb{R}$, in which $c_0 < c_1$, such that $\Pi^\star_{q_0,q_1}$ contains a path with clearance c, $\forall c \in [c_0, c_1]$.

Proof: Let π_{q_0,q_1} be the shortest path with clearance $c = J_C(\pi_{q_0,q_1})$ that is not the line segment connecting q_0 and q_1. Similarly, let π'_{q_0,q_1} be the shortest path with clearance $c' = J_C(\pi'_{q_0,q_1})$ in which $c_0 \leq c < c' = c_1$. Let $\mathcal{M}^c$ and $\mathcal{M}^{c'}$ be the sets of points such that for each point in the set, the distance between the point and the closest obstacle is less than c and c', respectively. Since the clearances of π_{q_0,q_1} and π'_{q_0,q_1} are c and c' respectively, $\pi_{q_0,q_1} \cap \mathcal{M}^c = \emptyset$ and $\pi'_{q_0,q_1} \cap \mathcal{M}^{c'} = \emptyset$. If we allow π'_{q_0,q_1} to have a clearance $c < c'$, we can shorten it thanks to the triangular inequality, since now it can intersect with $\mathcal{M}^{c'} \setminus \mathcal{M}^c$. Then, it is true that the length $J_L(\tilde\pi_{q_0,q_1})$ of this shortened path $\tilde\pi_{q_0,q_1}$ is less than $J_L(\pi'_{q_0,q_1})$, i.e., $J_L(\tilde\pi_{q_0,q_1}) < J_L(\pi'_{q_0,q_1})$. Furthermore, since π_{q_0,q_1} is the shortest path with clearance c, it is also true that $J_L(\pi_{q_0,q_1}) \leq J_L(\tilde\pi_{q_0,q_1}) < J_L(\pi'_{q_0,q_1})$. Since $c' > c$ and $J_L(\pi_{q_0,q_1}) < J_L(\pi'_{q_0,q_1})$ both solutions take part in $\Pi^\star_{q0,q1}$, $\forall c < c'$. ∎

Since $[c_0, c_1]$ is an uncountably infinite set, the following is true for $\Pi^\star_{q_0,q_1}$.

Corollary 1: Set of Pareto optimal paths $\Pi^\star_{q_0,q_1}$ defined in (1) is an uncountably infinite set.

We can formulate **MinLMaxC** as a graph search on a grid and apply a suitable multi-objective graph search algorithm, such as Multi-objective A* (MOA*) [20], to find an approximation of the Pareto set. However, the level of approximation will be tightly related to the selected grid and increasing the grid resolution would not necessarily improve the diversity of the solutions as this depends on the environment. Thanks to Proposition 1, we can also take a different approach and augment the search space with one of the objectives. To this end, we introduce the notion of c-feasibility; a path π_{q_0,q_1} is called c-feasible if $J_C(\pi_{q_0,q_1}) \geq c$. Consequently, we can define the **shortest c-feasible path planning problem (MinL$_\mathbf{c}$)** as the problem of finding the minimum length c-feasible path. Therefore, **MinLMaxC** can be reformulated such that it translates to the set of **MinL$_\mathbf{c}$** for all admissible clearance values (c-values), i.e., $c \in [c_{min}, c_{max}]$. The endpoints of the admissible clearance interval are determined such that c_{min} is the c-value when the goal is no longer reachable by a straight line incident to[1] q_0 and c_{max} is the maximum attainable clearance, i.e., the clearance of the path with maximum clearance on the Voronoi diagram connecting the initial and goal positions. To solve a single **MinL$_\mathbf{c}$**, we can rely on the visibility graph of the obstacles. To this end, the shortest c-feasible path can be found by inflating all the obstacles by c, i.e., computing the Minkowski sum of the obstacles and the open disc centered at the origin with radius c, and computing the shortest path on the visibility graph formed by the bitangents of the inflated obstacles.

Since $\Pi^\star_{q_0,q_1}$ is an uncountably infinite set (Corollary 1), solving all the corresponding **MinL$_\mathbf{c}$** is not possible. We can obtain an approximation of the Pareto set of **MinLMaxC** by sampling the interval $[c_{min}, c_{max}]$ (either discretizing using a fixed step-size or adopting a more tailored sampling strategy) to obtain a finite number of clearance values and the corresponding **MinL$_\mathbf{c}$**. It is clear that, the level of approximation is related to the particular discretization strategy that is selected and the number of **MinL$_\mathbf{c}$** to solve. To overcome the discrepancies introduced by an arbitrary discretization, we propose to rely on an event-based discretization and compute a finite number of families that collectively represent all the Pareto-optimal paths for **MinLMaxC**. To achieve this, we first compute a set of critical events which might have an impact on the Pareto front defined in the objective space.

IV. PROPOSED STRATEGY

This section introduces our strategy to solve the **MinLMaxC** by means of creating a data structure that carries the solution to **MinL$_\mathbf{c}$** for all admissible c-values, i.e., $[c_{min}, c_{max}]$. We first introduce the POP-graph, which contains the potential edges that can take part in the shortest path from a start position q_0, to a goal q_1, for all $c \in [c_{min}, c_{max}]$. In the following, we explain how to construct

---

[1] If the goal, q_1, is visible from q_0 for $c = 0$, the path length equals $d(q_0, q_1)$ for all $0 < c \leq c_{min}$. Thus, the objective vector $[d(q_0, q_1), c_{min}]^T$ will dominate $[d(q_0, q_1), c']^T$, $\forall c' < c_{min}$.

and query this data structure to compute the Pareto front, considering point (disc) obstacles.

A. Implicit Graph Representation

While constructing the POP-graph, we rely on an implicit representation for the edges, that we adopt from VV-complex [14]. Graph nodes do not explicitly map to a point in $\mathbb{R}^2$, instead, they are supported by the circles centered at the obstacles. Each circle is the boundary of the disc corresponding to an obstacle inflated by c. Note that the inflated obstacle is an open disc, thus, the path can intersect with the circle. Consequently, the edges are the bitangents to the circles.
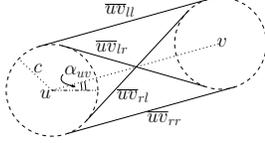
Fig. 4. The four bitangents to the circles $\mathcal{C}_c(u)$ and $\mathcal{C}_c(v)$ of radius c, centered at u and v, respectively.

For two distinct point obstacles u and v, $\vec{uv}$ represents the ray starting from u and passing through v. Each pair of circles with radius $c > 0$ centered at u and v, corresponding to the inflated obstacles, $\mathcal{C}_c(u)$ and $\mathcal{C}_c(v)$, respectively, can have at most four bitangents. They are denoted as $\overline{uv}_{ll}$, $\overline{uv}_{rr}$, $\overline{uv}_{lr}$, and $\overline{uv}_{rl}$, in which the subscripts denote the tangency side with respect to $\vec{uv}$ (Fig. 4). In the following, we will use $\overline{uv}$ to denote a generic edge without specifying the tangency sides. Let α_{uv} be the angle between the absolute x-axis (aligned with $\vec{q_0 q_1}$) and the ray $\vec{uv}$. The outer bitangents $\overline{uv}_{rr}$ and $\overline{uv}_{rr}$ retain the same slope, α_{uv} for all $c > 0$. The inner bitangents $\overline{uv}_{lr}$ and $\overline{uv}_{rl}$ rotate about the midpoint between u and v, i.e., $\frac{1}{2}(u + v)$, clockwise and counterclockwise, respectively. Hence, the slope of $\overline{uv}_{lr}$ is equal to $\alpha_{uv} - \arcsin \frac{2c}{d(u,v)}$, consequently, the slope of $\overline{uv}_{rl}$ is equal to $\alpha_{uv} + \arcsin \frac{2c}{d(u,v)}$ for varying c. Note that, for $c > \frac{1}{2} d(u, v)$, $\overline{uv}_{lr}$ and $\overline{uv}_{rl}$ disappear since two circles intersect.

Each edge is associated with a validity range, i.e., a set of disjoint clearance intervals, in which it is potentially on the shortest path from a start position to goal. Therefore, the graph topology varies with the clearance. The interval boundaries, i.e., the endpoints of the intervals forming the validity range of an edge, are determined by critical events.

The resulting POP-graph $\langle \mathcal{L}_E, \mathcal{R} \rangle$ is a list of implicit edges, together with their validity ranges $\mathcal{R}$, such that each edge $e \in \mathcal{L}_E$ is valid only for a range of c-values, i.e., for $c \in \mathcal{R}(e)$.

B. Critical events and event handling

We construct POP-graph by progressively adding edges as they interact with the existing ones, i.e., as they get involved in critical events. Assuming general position for the point obstacles, in particular, assuming that no three points are collinear, the edge list starts with $\mathcal{L}_E = \{\overline{q_0 q_1}\}$. Every time we add a new edge to the list $\mathcal{L}_E$ we compute the

corresponding critical events that it is involved in. Each event, $\langle c, e_1, e_2, \texttt{type} \rangle$, in which $\texttt{type}$ refers to the type of the critical event, is added to the event list $\mathcal{Q}$. While $\mathcal{Q}$ is not empty we pop the event with the smallest c-value and proceed according to its type. To this end, we identified three critical events: *split event*, *blocking event* and *unblocking event*. In the following, we explain what each event corresponds to and how we handle the changes that it induces to the graph topology.
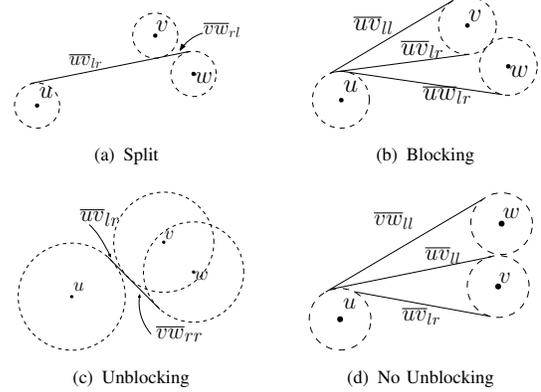
Fig. 5. Critical events. (a) $\overline{uw}_{ll}$ is split by the obstacle centered at v, $\overline{uv}_{lr}$ and $\overline{vw}_{lr}$ are created. (b) $\overline{uv}_{lr}$ is blocked since it can not take part in a shortest path to goal. (c) $\overline{uv}_{lr}$ can get unblocked once it gets equally sloped with $\overline{uw}_l r$, i.e., once obstacle at v splits $\overline{uw}_{lr}$. (d) If an outer tangent gets blocked, it will not get unblocked since its slope is constant.

Split Event: When an edge, $\overline{uv}$, becomes tangent to the circle $\mathcal{C}_c(w)$ corresponding to the obstacle centered at w at c, then $\overline{uv}$ is split into two by $\mathcal{C}_c(w)$. This can also be seen as the event when the bitangents $\overline{uv}$, $\overline{uw}$ induced by the same obstacle vertex (same tangency side) are equally sloped. This event closes the clearance interval of $\overline{uv}$ definitely at c (Theorem 2 in [14]) and it is replaced by two edges, $\overline{uw}$ and $\overline{wv}$. If $\overline{uw}$ and $\overline{wv}$ are not part of $\mathcal{L}_E$ they are included. Furthermore, if it is the first time that we considered bitangents to obstacle w, we also add two more edges: $\overline{q_0 w}$ and $\overline{wq_1}$. The validity ranges of newly added edges start at $\max\{c, \min \mathcal{R}(\overline{uv})\}$. This ensures that the newly included edges will not take part in POP-graph before the edge that they interacted with.

Blocking Event: An edge can get temporarily blocked, which means that for the interval that it is blocked it can not be a part of the shortest path. This happens if two edges $\overline{uw}$ and $\overline{uv}$ get equally sloped at c such that $\mathcal{C}_c(v)$ splits the edge $\overline{uw}$, and $\overline{vw}$ is an inner tangent (Fig. 5a). It is possible that $\overline{uv}$ is not a part of POP-graph yet. Then, the shortest path to any point that falls the opposite side of the ray aligned with $\overline{uv}$ with respect to $\mathcal{C}_c(v)$ will not contain $\overline{uv}$. Any edge that proceeds $\overline{uv}$ in the shortest path will be contained in the area limited by the circular arc on $\mathcal{C}_c(v)$ and $\overline{vw}$. Therefore, when the circles $\mathcal{C}_c(v)$ and $\mathcal{C}_c(w)$ intersect at $c = \frac{1}{2} d(v, w)$, $\overline{uv}$ gets blocked since the corresponding discs create a discontinuity in the free space for inflation radius $c > \frac{1}{2} d(v, w)$ (Fig. 5b). Note that, same is true also for the case when $\overline{uv}$ is an inner tangent. In this case, $\overline{vw}$ gets blocked when $\mathcal{C}_c(u)$ and $\mathcal{C}_c(v)$

intersect at $c = \frac{1}{2}d(u,v)$. This situation can be reverted, i.e., $\overline{uv}$ (or similarly $\overline{vw}$) can regain its valid status, by an unblocking event. When an edge gets blocked, we close its validity interval at c. Similar to split event, we add the edges connecting the blocking obstacle to start and goal, if they are not already in $\mathcal{L}_E$.

Unblocking Event: Unblocking event happens when an edge that lost its validity due to a blocking event, regains its validity. Each unblocking event has a blocking pair, yet the reverse is not always true. In other words, a blocked edge may never regain its validity. An edge $\overline{uv}$ that got blocked can get unblocked once it becomes outer bitangent to the circles $\mathcal{C}_c(v)$ and $\mathcal{C}_c(w)$ (Fig.5c). However, if the blocked edge is already an outer bitangent, then it will not get unblocked since the slope of outer bitangents are constant (Fig. 5d). Note that, for a blocking event associated with the clearance value c_{block}, it is also possible that unblocking event happens at $c < c_{block}$, in this case, $\overline{uv}$ will not get blocked. In the case of an unblocking event, we are not creating new edges but updating only the validity range of $\overline{uv}$ by including a new interval starting at c.

C. Computing the Pareto front

Once the POP-graph is created, we can query it to solve **MinLMaxC**. This corresponds to finding a set of intervals such that within each interval, solution to **MinL$_\mathbf{c}$** is represented as a sequence of implicit edges. We start with computing all the branches from q_0 to q_1 represented on POP-graph such that the intersection of the validity ranges of the edges forming the branch is not empty. For each branch $\mathcal{S} = \{e_1, e_2, \ldots e_K\}$, in which K is the number of implicit edges in the sequence, together with the corresponding validity ranges $\mathcal{R}(e_j)$, $j = 1, \ldots, K$, we can compute the interval that this branch is valid, i.e.,

$$\mathcal{R}(\mathcal{S}) = \left[\max_{e_j \in \mathcal{S}}\{\min \mathcal{R}(e_j)\}, \min_{e_j \in \mathcal{S}}\{\max \mathcal{R}(e_j)\} \right].$$

Therefore, $\mathcal{S}$ is capable of representing a family of infinitely many paths, denoted as $\Pi_\mathcal{S}$, in terms of a sequence of bitangents $\forall c \in \mathcal{R}(\mathcal{S})$. Furthermore, there is a bijective map $\mu_\mathcal{S} : \mathcal{R}(S) \to \Pi_\mathcal{S}$ such that each $c \in \mathcal{R}(\mathcal{S})$ maps to a unique c-feasible path $\pi^c_{q_0,q_1} \in \Pi_\mathcal{S}$. Note that all paths in $\Pi_\mathcal{S}$ belong to the same homotopy class. If $\mathcal{S}$ contains edges with validity ranges composed of disjoint intervals, then, we can replicate this sequence and compute the range of each replica accordingly until each sequence has a continuous validity range.

A path $\pi_{q_0,q_1} \in \Pi_\mathcal{S}$ is constructed as a concatenation (denoted by ;) of M sub-paths, i.e., $\pi_{q_0,q_1} = \pi_1; \pi_2; \ldots; \pi_M$, such that $\pi_1(0) = q_0$, $\pi_M(1) = q_1$ and $\pi_m(1) = \pi_{m+1}(0), \forall m \in \{1, \ldots, M-1\}$. In our case, each sub-path π_m, $m = 1, \ldots, M$, is either a line segment corresponding to a bitangent or a circular arc and follow the rule that if π_m is a bitangent then π_{m+1} is a circular arc, moreover, π_1 and π_M are tangents to the corresponding circles from q_0 and q_1 respectively. Therefore, for a path $\pi^c_{q_0,q_1} = \mu_\mathcal{S}(c)$, the path length is given as the sum of the lengths of these sub-paths,

i.e., $J_L(\pi^c_{q_0,q_1}) = \sum_{m=1}^{M} l_{\pi_m}(c)$. Length of each sub-path, l_{π_m} can be computed as a function of c such that

$$l_\pi(c) = \begin{cases} \sqrt{d(u,v)^2 - 4c^2} & \text{if } \pi \text{ is a bitangent} \\ \theta \cdot c & \text{if } \pi \text{ is a circular arc} \\ \sqrt{d(u,v)^2 - c^2} & \text{if } \pi \in \{\pi_1, \pi_M\} \end{cases}$$

in which u and v are the centers of the two circles supporting the bitangent and θ corresponds to the angle difference between the slopes of two consecutive bitangents π_{m-1} and π_{m+1}, in which π_m is the circular arc. Recall that the slope of an outer bitangent is constant, while an inner bitangent rotates about the midpoint between the centers of the two supporting circles.

Once all the edge sequences from initial position to goal are found, we can divide $[c_{min}, c_{max}]$, the interval of admissible c-values, into a set of disjoint sub-intervals such that $\bigcup_{i=1}^{N} I_i = [c_{min}, c_{max}]$, in which N is the total number of sub-intervals and $I_i = (c^i_{min}, c^i_{max})$. The endpoints of the sub-intervals, c^i_{min} and c^i_{max}, correspond to the endpoints of the validity ranges of the computed sequences, which are determined by the critical events. A sequence $\mathcal{S}$, hence the path family $\Pi_\mathcal{S}$, is valid in an interval I_i iff $I_i \subseteq \mathcal{R}(\mathcal{S})$. By the way we construct the sub-intervals, it is not possible to have the endpoints of the validity range of a sequence to be in a sub-interval (excluding the boundaries). Then, the Pareto front in the objective space can be expressed through the bijective map $l^\star : [c_{min}, c_{max}] \to \mathbb{R}_{>0}$ which is the length of the non-dominated path as a function of clearance, i.e.,

$$l^\star(c) = \begin{cases} \min_{\mathcal{S} \in \Psi}\{J_L(\mu_\mathcal{S}(c)) \mid I_1 \in \mathcal{R}(\mathcal{S})\} & \text{if } c \in I_1 \\ \vdots \\ \min_{\mathcal{S} \in \Psi}\{J_L(\mu_\mathcal{S}(c)) \mid I_N \in \mathcal{R}(\mathcal{S})\} & \text{if } c \in I_N \end{cases}$$

in which $\mu_\mathcal{S}$ maps c to $\pi^c_{q_0,q_1}$ which is the shortest c-feasible path represented by the sequence $\mathcal{S}$ that is valid in the corresponding interval and Ψ is the set of all sequences. Note that even though the map $l^\star$ is bijective, we can not guarantee that the corresponding shortest path for each clearance value is unique as this depends on the environment topology.

We can easily compute the Pareto front in the objective space, i.e., compute the minimum of two or more functions. However, finding at which c-value a path dominates another (if that is the case) within an interval is more involved since it requires finding the zeros of the length difference of these two paths. This is not an easy task in general and it might call for numerical methods. Furthermore, due to the circular arcs, path length may not be algebraic for certain c-values [13], which can prohibit finding the roots exactly. However, we can easily make a tradeoff between the conflicting objectives using the Pareto front in the objective space and select the shortest path with desired clearance among the paths in the corresponding interval. Recall that this operation does not require us to construct or search the visibility graph for the corresponding clearance value since we already know the representation of each family (corresponding branch of

POP-graph as a sequence of edges) that corresponds to the selected clearance.

D. Numerical example

Here, we present a numerical example to show the effectiveness of the proposed approach. The considered environment is populated with 10 point (disc) obstacles. Fig. 6 shows the computed Pareto front in the objective space. Non-dominated paths, i.e., elements of the Pareto set, with different c-values are shown in Fig. 7. In total, the Pareto-optimal paths correspond to 6 families. As expected due to Proposition 1, there is a non-dominated path for each c-value that is admissible. However, the inverse is not true. As clearance increases, certain homotopy classes reach a limit clearance. This happens when a path represented on the visibility graph of the inflated obstacles intersects with the one on the Voronoi diagram for the same homotopy class. Thus, it can no longer dominate a longer path in a different homotopy class if the latter one has higher clearance, i.e. if that class continues to be valid. This is the reason for the discontinuities in Fig. 6, since the families corresponding to the paths shown in Fig. 7 for c-values of 28.25 and 35, respectively, reach the end of their validity range. For 100 trials of 10 randomly generated obstacles in the environment shown in Fig. 7, it took on average 0.06 seconds to construct the POP-graph and 0.0314 seconds to compute the Pareto front on a laptop equipped with an Intel Core i7-9750H CPU and 32 GB of RAM. There were at most 13 Pareto-optimal families and the median was 7.
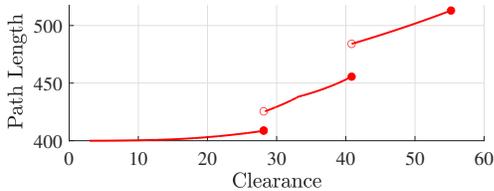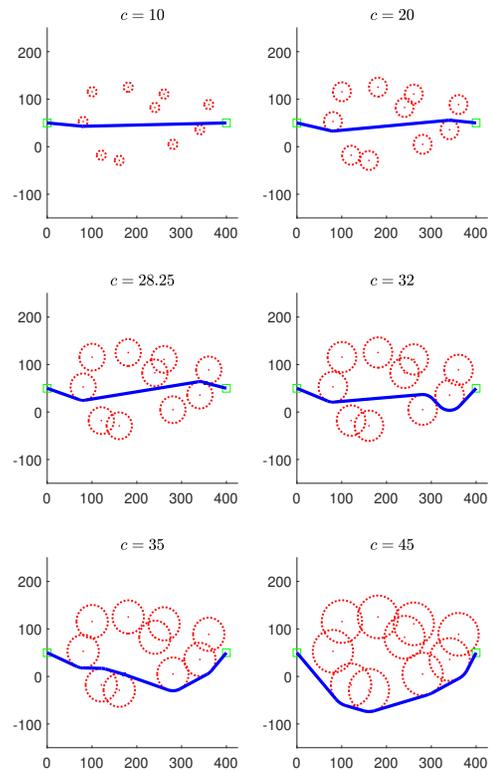

Fig. 7. Non-dominated paths with different clearance values. Initial and final points are marked as green squares, the obstacles are red dots, and the dashed circles correspond to inflated obstacles.

Fig. 6. Pareto front represented in the objective space.

E. Computational efficiency

Since two edges, of which there are are at most $O(n^2)$ in the POP-graph, can get involved in an event only once, there are at most $O(n^2)$ events in which n is the number of obstacles (vertices). In the worst case, each event requires us to add new edges and check for their corresponding events. Adding new edges and updating the validity range of the corresponding edge require a constant number of operations to be done, resulting in $O(1)$ time. For each edge, checking for new events have time complexity $O(n)$, resulting in a total time complexity of $O(n^3)$ to build the complete POP-graph. Extracting the representation of the set of Pareto-optimal families is sensitive to the total number of classes for a given problem. At present, we know only that the set of families is finite, but do not have an upper bound on the worst case over all possible planar environments. In practice, we have observed that very few families arise, even from fairly complex environments. It remains an open problem

to determine tight bounds on the number of Pareto-optimal families in terms of the input representation.

V. CONCLUSION

In this paper, we proposed a method to compute the Pareto front for path planning that simultaneously optimizes path length and clearance. The core of our approach is to compute critical events that determine the clearance intervals in which Pareto optimal paths belong to the same family and share a common implicit representation. Event-based nature of our method enables a discretization of the Pareto front that does not suffer from the resolution limitations of standard methods. Having computed all non-dominated combinations of path length and clearance, one can make the most suitable tradeoff given the problem at hand.

Extension to polygonal obstacles is a subject of ongoing research and requires us to compute the events when each vertex (incident edges) takes part in POP-graph. Another possible extension is to compute a clearance varying decomposition of the plane, similar to continuous Dijkstra [2], [3], in which the critical events would correspond to changes in the partitions. Note that a geometrically exact decomposition may not be possible due to circular arcs. At the moment, the POP-graph needs to be reconstructed if there is a change in the environment. Although this seems to be amenable for a real-time implementation, "repairing" relevant parts of the graph can be interesting.

References

[1] S. K. Ghosh, *Visibility algorithms in the plane*. Cambridge university press, 2007.

[2] J. Hershberger and S. Suri, "Efficient computation of euclidean shortest paths in the plane," in *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*. IEEE, 1993, pp. 508–517.

[3] J. Hershberger, S. Suri, and H. Yıldız, "A near-optimal algorithm for shortest paths among curved obstacles in the plane," in *Proceedings of the twenty-ninth annual symposium on Computational geometry*, 2013, pp. 359–368.

[4] J. Denny, E. Greco, S. Thomas, and N. M. Amato, "Marrt: Medial axis biased rapidly-exploring random trees," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 90–97.

[5] Y. Morales, A. Watanabe, F. Ferreri, J. Even, K. Shinozawa, and N. Hagita, "Passenger discomfort map for autonomous navigation in a robotic wheelchair," *Robotics and Autonomous Systems*, vol. 103, pp. 13–26, 2018.

[6] C. Ó'Dúnlaing and C. K. Yap, "A "retraction" method for planning the motion of a disc," *Journal of Algorithms*, vol. 6, no. 1, pp. 104–111, 1985.

[7] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.

[8] R. Geraerts and M. H. Overmars, "Clearance based path optimization for motion planning," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, 2004, pp. 2386–2392 Vol.3.

[9] R. Geraerts, "Planning short paths with clearance using explicit corridors," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 1997–2004.

[10] M. Candeloro, A. M. Lekkas, and A. J. Sørensen, "A voronoi-diagram-based dynamic path-planning system for underactuated marine vessels," *Control Engineering Practice*, vol. 61, pp. 41–54, 2017.

[11] Y. Dong, E. Camci, and E. Kayacan, "Faster rrt-based nonholonomic path planning in 2d building environments using skeleton-constrained path biasing," *Journal of Intelligent & Robotic Systems*, vol. 89, no. 3, pp. 387–401, 2018.

[12] J. Van Den Berg and M. Overmars, "Planning the shortest safe path amidst unpredictably moving obstacles," in *Algorithmic Foundation of Robotics VII*. Springer, 2008, pp. 103–118.

[13] E.-C. Chang, S. W. Choi, D. Y. Kwon, H. Park, and C. K. Yap, "Shortest path amidst disc obstacles is computable," *International Journal of Computational Geometry & Applications*, vol. 16, no. 05n06, pp. 567–590, 2006.

[14] R. Wein, J. P. Van den Berg, and D. Halperin, "The visibility-voronoi complex and its applications," *Computational Geometry*, vol. 36, no. 1, pp. 66–87, 2007.

[15] R. Wein, J. Van Den Berg, and D. Halperin, "Planning near-optimal corridors amidst obstacles," in *Algorithmic Foundation of Robotics VII*. Springer, 2008, pp. 491–506.

[16] S. M. LaValle and S. A. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 912–925, 1998.

[17] M. Brandao, M. Fallon, and I. Havoutis, "Multi-controller multi-objective locomotion planning for legged robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4714–4721.

[18] I. Becerra, M. Suomalainen, E. Lozano, K. J. Mimnaugh, R. Murrieta-Cid, and S. M. LaValle, "Human perception-optimized planning for comfortable vr-based telepresence," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6489–6496, 2020.

[19] B. Sakcak, L. Bascetta, G. Ferretti, and M. Prandini, "An admissible heuristic to improve convergence in kinodynamic planners using motion primitives," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 175–180, 2019.

[20] B. S. Stewart and C. C. White, "Multiobjective a," *Journal of the Association for Computing Machinery*, vol. 38, no. 4, pp. 775–814, 1991.

[21] L. Mandow and J. P. e. De la Cruz, "A new approach to multiobjective a* search," in *Proceedings of the 19th international joint conference on Artificial intelligence*, 2005, pp. 218–223.

[22] D. Yi, M. A. Goodrich, and K. D. Seppi, "Morrf: sampling-based multi-objective motion planning," in *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015, pp. 1733–1739.

[23] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[24] O. Castillo, L. Trujillo, and P. Melin, "Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots," *Soft Computing*, vol. 11, no. 3, pp. 269–279, 2007.

[25] F. Ahmed and K. Deb, "Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms," *Soft Computing*, vol. 17, no. 7, pp. 1283–1299, 2013.

[26] M. Davoodi, "Bi-objective path planning using deterministic algorithms," *Robotics and Autonomous Systems*, vol. 93, pp. 105–115, 2017.

[27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.