# Sampling-based Motion Planning for Optimal Probability of Collision under Environment Uncertainty

Hao Lu and Hanna Kurniawati and Rahul Shome

*Abstract*— **Motion planning is a fundamental capability in robotics applications. Real-world scenarios can introduce uncertainty to the motion planning problem. In this work we study environment uncertainty in general high-dimensional problems wherein the choice of appropriate metrics and formulations are shown to have significant effect on the probability of collision of the solution path. Several practically motivated cost functions have been proposed in literature to model and solve the problem but are shown in this work to suffer from higher probabilities of collision. The current work presents a theoretically sound formulation that was first mentioned in previous work on minimum constraint removal. In this work, approximating the optimal problem is shown to be better in achieving lower probability of collision. To demonstrate the formulation in a sampling-based setting, a mixed integer linear program seeded by greedy search over a roadmap with sampled environments is used to report paths with low probability of collision. Compared against minimizing the sum and minimizing max probability cost functions on a seven degree-of-freedom robotic arm in uncertain environments, we show clear benefits and promise towards motion planning for optimal probability of collision.**

## I. Introduction

Collision checking is a critical primitive computation in motion planning [1], [2]. This computation remains important for motion planning under uncertainty, but such a computation is generally expanded to probabilistic collision checking, which computes the probability that a configuration or a path is collision free, rather than a binary result of whether collision takes place or not. Several lines of work have proposed distinct definitions to approximate the true collision probability [3]–[8]. Often the modeling choice and solution strategy get codesigned, making the diversity of differing definitions problematic for characterizing probabilistic collision checking. In this paper, we focus on probabilistic collision checking against environment uncertainty, i.e., the occupied obstacle subset of the workspace is uncertain. We propose a formulation and a sampling-based algorithm that can approximate the true collision probability for the purpose of planning and exploring the differences between this new approach and some of the commonly used simplified definitions of collision probability.

Collision probability computation is a basic building block for many approaches to planning under uncertainty. Such approaches include sampling-based methods [10] with chance constraints [11], [12] and theoretical guarantees [12], bounded uncertainty [13], [14], reachability regions [15], with weighted cost functions on roadmaps [10], [13], or using a reformulation of the infeasible configuration space subset [6]. Collision probability computation is also a building block of planning under uncertainty approaches that are based on trajectory optimisation with chance constraints [3], [4], [7], [16], combined sampling-based and trajectory-optimzation [16], as well as partially-observable Markov Decision Processes [17]. *Correctly modeling and computing collision probability can be critical to robotics applications*. Optimizing such a metric is also of significance towards considerations of robustness. While asymptotic optimality [18] has been described for motion planning with (typically) path lengths, *guarantees for optimal collision probability is of both theoretical and practical interest*. In this work we both model collision probability, ascribe asymptotic optimality guarantees, as well as demonstrate the benefits of careful (optimal) modeling in solution collision probability.

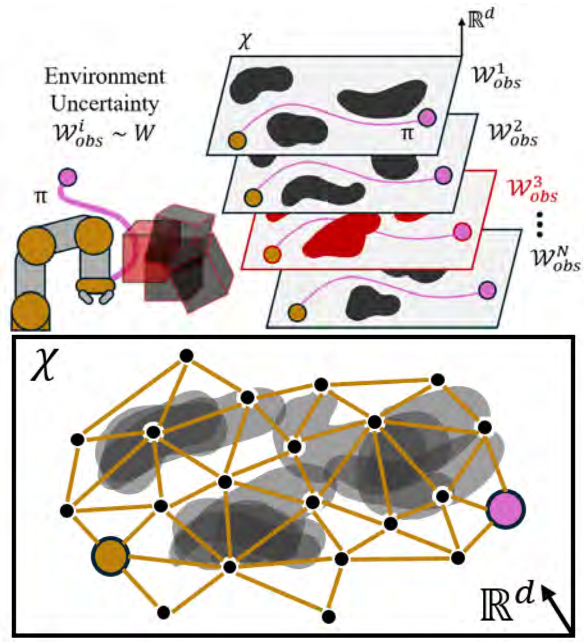



Fig. 1: Probability of collision under environment uncertainty. The top row illustrates what environment uncertainty might look like in the workspace. For workspace obstacle samples $\mathcal{W}^i_{\text{obs}}$, the high-dimensional configuration space $\mathcal{X}$ depends on a corresponding workspace sample and induces potential collisions with trajectories $\pi$. The bottom row shows a visualization of how all the configuration space obstacles might look like, alongside their interactions with an overlaid graph, motivating the connections to Minimum Constraint Removal [9] over the bottom roadmap.

We focus on motion planning under uncertainty problems

The authors are affiliated to the School of Computing at the Australian National University, Canberra, AU. Email `rahul.shome@anu.edu.au`.

where the uncertainty is only in the environment. Specifically, we focus on a robot operation in an environment populated by obstacles with uncertain position and geometry (illustrated in Fig 1). Despite previous work on planning under uncertainty, this class of problem remains relatively open due to the complexity of representing environment uncertainty. In this paper, *we do not aim to explore a suitable environment or uncertainty representation* when environment uncertainty is present. Rather, *we assume that we have access to a sampler* that can sample instantiations of the uncertain environment and directly explore the computation of collision probability for the purpose of motion planning. At all times, the robot position is known exactly and its motion is deterministic. The goal of the motion planner is to find a path with the lowest collision probability from a given start to goal configuration. Although our proposed collision probability computation can be applied to various planning approaches, in this paper, we focus on its application for sampling-based planning under uncertainty.

Many methods have been proposed to compute collision probability for planning purposes. Some approaches make assumptions on the underlying distributions [5] wherein the worst collision probability along the path is minimized. Such assumptions are often practically motivated and convenient but can limit their applicability. Other approaches [3], [4], [7], [16] discretize the trajectory and compute the collision probability of the trajectory as the union bounded sum of collision probability of the discrete configurations along the trajectory. Although they are often extensible and computationally efficient, they are also known to results in diverging collision probability measures (e.g., Sec 4, [4]). While other measures like min-max [5], [8] are robust of trajectory discretization, it remains to be seen how accurate the calculated collision probability can be. Recent work [19] has tried to characterize some of the complexity in defining measures of collision probability over continuous trajectories but much of its analysis relies on normally distributed measures.

In contrast to the above approaches, we formulate the problem to be based on collision probability over continuous trajectories and frame this as a Minimum Constraint Removal (MCR) problem (Fig 1 bottom). MCR was first proposed in a seminal work [9] with several extensions [20], [21]. The potential application of MCR for collision probability computation in motion planning under environment uncertainty has been discussed as early as its seminal paper [9]. However, how they perform in comparison to other approaches for computing collision probability have never been discussed. Some recent work [22] applied MCR in a related formulation to robust manipulation in clutter. Here the focus was on object-level uncertainty measures and with only up to 7 samples of the underlying uncertainty. In this work we focus on studying the significantly more samples (up to 200) and its effect on a principled formulation. *The crucial question to critically evaluate is whether the MCR formulation indeed has benefits or whether alternative or approximate models of uncertainty are practically good enough?* We show that the difference in collision probabilities can be large if we are not careful about choosing the correct (MCR-based) formulation.

**Contributions:** Major building blocks for this work have already been laid out in previous literature. The novelty of the proposed work is primarily in a) formalizing the probability of collision for the purpose of motion planning under environment uncertainty that can handle high-dimensional configuration space and any type of distributions of environment uncertainty as long as it can be sampled, b) concretely connecting our formalism to the minimum constraint removal problem, c) demonstrating theoretical guarantees of the proposed formulation (asymptotic optimality in a sampling-based regime), d) proposing a seeded mixed-integer linear program for the optimal discrete problem over a graph, e) reporting strong empirical evidence to support the careful consideration of the choice of problem formulation vis-a-vis optimizing with alternative models of cost functions.

## II. Problem Formulation

A robot with $d$ degrees of freedom is represented as a $d$-dimensional configuration $x \in \mathcal{X} \subset \mathbb{R}^d$. The robot is situated in a workspace $\mathcal{W} \subset \mathbb{R}^3$. The subset of the workspace occupied by the obstacles is $\mathcal{W}_{\text{obs}} \subset \mathcal{W}$. The robot geometries themselves occupy $vol(x) \subset \mathcal{W}$ that for the configuration $x$. The invalid subset of the configuration space is defined as $\mathcal{X}_{\text{obs}} = \{x \;\; s.t. \; vol(x) \cap \mathcal{W}_{\text{obs}} \neq \emptyset, \; x \in \mathcal{X}\}$. The feasible subset is $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$.

A motion planning problem is defined by a start configuration $x_s \in \mathcal{X}$, a goal configuration[1] $x_g \in \mathcal{X}$. The solution to the motion planning problem is a feasible path $\pi : \mathcal{X}_{\text{free}} \rightarrow [0,1], \; \pi(0) = x_s, \pi(1) = x_g$. Some cost function is defined for $c : \pi \rightarrow \mathbb{R}^{\geq 0}$. Given all possible feasible trajectories $\Pi$, solution to the optimal motion planning problem entails finding a $\pi^*$ that is

$$\pi^* \in \underset{\pi \in \Pi}{\texttt{argmin}} \; c(\pi) \tag{1}$$

Now that the classical motion planning problem has been presented, we shall defined the optimal motion planning problem for minimizing probability of collision under environment uncertainty.

*Definition 1 (Environment Uncertainty):* A realization of the environment corresponds to a specific occupied obstacle subset in the workspace, $\mathcal{W}_{\text{obs}} \in \mathfrak{W}$, where $\mathfrak{W}$ denotes possible subsets of $\mathcal{W}$. Now we can define a probability space of $(\mathfrak{W}, \mathcal{F}^{\mathfrak{W}}, P)$. $\mathfrak{W}$ is a $\sigma$-algebra described over possible sets of world realizations. A probability function is also defined $P : \mathcal{F}^{\mathfrak{W}} \rightarrow [0,1]$.
Next, we will define more carefully collisions between the known robot configurations and uncertain environment.

*Definition 2 (Robot-Environment Collision):* A robot at a configuration $x$ defines an occupied volume of the workspace $vol(x) \subset \mathcal{W}$. A collision indicator is defined as an indicator function $\mathbb{1}_{\text{col}} : \mathcal{X} \times \mathfrak{W} \rightarrow \{0,1\}$ as follows:

$$\mathbb{1}_{\text{col}}(x, \mathcal{W}_{\text{obs}}) = \begin{cases} 1, & \text{when } vol(x) \cap \mathcal{W}_{\text{obs}} \neq \emptyset \\ 0, & \text{when } vol(x) \cap \mathcal{W}_{\text{obs}} = \emptyset \end{cases} \tag{2}$$

[1]Note that the goal is typically a set but for simplicity of the discussion we frame the single goal setting. All insights apply to goal sets.

*Definition 3 (Constraining Environment Set):* Given a robot configuration $x$, all realizations of the environment, describing the set of possible workspaces that collide with the robot at $x$ define the constraining environment set $\mathcal{F}(x)$

$$\mathcal{F}(x) = \{\mathcal{W}_{\text{obs}} \ s.t. \ \mathbb{1}_{\text{col}}(x, \mathcal{W}_{\text{obs}}) = 1, \ \mathcal{W}_{\text{obs}} \subset \mathcal{W}\} \quad (3)$$

It should be that $\mathcal{F}(x) \subset \mathcal{F}^{\mathfrak{W}}$. Using environment uncertainty, we can therefore characterize its probability.

*Definition 4 (Probability of Collision of Configuration):* Given a robot configuration $x$ and environment uncertainty, from Def 1, following the probability space $(\mathfrak{W}, \mathcal{F}^{\mathfrak{W}}, P)$, the probability of collision $\text{Pr}_{\text{col}}$ is defined as

$$\text{Pr}_{\text{col}}(x) = P(\mathcal{F}(x)) \quad (4)$$

*Definition 5 (Probability of Collision of Path):* Given a path $\pi$ that passes through a set of configurations (for the clarity of notation we will allow referring to this membership by $x \in \pi$), the probability of collision of the path can be defined as

$$\text{Pr}_{\text{col}}(\pi) = P(\mathcal{F}(\pi)) \text{ where } \mathcal{F}(\pi) = \cup_{x \in \pi} \mathcal{F}(x) \quad (5)$$

Here let us make the set $\Pi$ to be all possible paths connecting $x_s$ to the $x_g$. We can now define the problem of finding a path that minimizes the collision probability which can be the $c$ in the problem.

*Definition 6 (Planning for Optimal Collision Probability):* Given a configuration space $\mathcal{X}$, environment uncertainty $(\mathfrak{W}, \mathcal{F}^{\mathfrak{W}}, P)$, a start configuration $x_s$ and a goal configuration $x_g$, the motion planning of optimal collision probability can be defined to discover an optimal solution path $\pi^*$ as

$$\pi^* \in \underset{\pi \in \Pi}{\texttt{argmin}} \ \text{Pr}_{\text{col}}(\pi) \quad (6)$$

### A. Foundations

*Theorem 1 (Asymptotic Optimality (AO) in Path Cost):* In sampling-based algorithms, as the number of samples $n$ increases, an asymptotic optimal motion planner requires the discovered solution $\pi_n$ to approach arbitrarily close to the optimal solution cost.

$$\lim_{n \to \infty} \text{Pr}(\{c(\pi_n) \leq (1+\epsilon)c(\pi^*)\}) \to 1, \quad \forall \epsilon > 0 \quad (7)$$

Typically, the environment uncertainty is not available, as stipulated in Def 1. However, we claim that it should be possible to make do with solely the ability to sample environments (or instances of workspace obstacles).

*Assumption 1 (Environment Sampling):* The environment uncertainty can be sampled from some corresponding random variable $W$ to generate realizations of $\mathcal{W}_{\text{obs}} \sim W$ that will follow an unknown underlying probability distribution.

*Theorem 2 (Convergence of Sampling Environments):* Collision checking a sequence of $N$ of world realizations that can be drawn is denoted by $\mathcal{W}^1_{\text{obs}}, \mathcal{W}^2_{\text{obs}} \cdots \mathcal{W}^N_{\text{obs}} \sim W$ can yield an arbitrarily accurate estimate of the probability of collision $\text{Pr}_{\text{col}}$.

*Proof:* Using the weak law of large numbers

$$\mathbb{1}_{\text{col}}(x, \mathcal{W}^i_{\text{obs}}) = \mathbb{1}(\{\mathcal{W}_{\text{obs}} \in \mathcal{F}(x)\}) \quad (8)$$

$$\implies \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\text{col}}(x, \mathcal{W}^i_{\text{obs}}) \xrightarrow{P} P(\mathcal{F}(x)) \quad (9)$$

$$= \text{Pr}_{\text{col}}(x) \quad (10)$$

The same arguments extend to a path $\pi$ which needs to check $\mathbb{1}(\{\mathcal{W}_{\text{obs}} \in \mathcal{F}(\pi)\}) = \mathbb{1}(\{\mathcal{W}_{\text{obs}} \in \cup_{x \in \pi} \mathcal{F}(x)\})$. ■

*Definition 7 (AO in Probability of Collision):* In sampling-based algorithms, as the number of samples $n$ increases and the number of environment samples $N$ increases, asymptotic optimality requires the discovered solution $\pi_{n,N}$ approaches arbitrarily close to the optimal probability of collision.

$$\lim_{n,N \to \infty} |\text{Pr}_{\text{col}}(\pi_{n,N}) - \text{Pr}_{\text{col}}(\pi^*)| \leq \epsilon \quad \forall \epsilon > 0 \quad (11)$$

### B. Connection to Minimum Constraint Removal

Motion planning with environment uncertainty has been pointed out [9] as an application of Minimum Constraint Removal. Here, we present the MCR reduction and correspondingly highlight that our problem inherits the NP-Completeness in the discrete case. *Intuitively, the straightforward analogy to MCR can follow. Considering a regime where worlds $\mathcal{W}^i_{\text{obs}} \sim W$ are sampled and collision checked, each colliding world counts as a constraint. Minimizing the cardinality of the set of colliding worlds correspondingly minimizes the collision probability.*

*Theorem 3 (Discrete Problem Variant is NP-Complete):* A discrete variant of the optimal collision probability over $N$ environments on a graph with $n$ vertices is NP-Complete.

*Proof:* A Discrete MCR over a graph problem instance can be converted into a discrete instance of minimizing collision probability over $N$ world samples in polynomial steps on a graph. Beginning with a graph $\mathcal{G}_n(V, E)$, vertex constraint function $C(v)$ that can report up to $N$ constraints, and start-goal vertices $s, t \in V$, the decision version of MCR certifies whether a solution exists with $k$ constraints. The decision version of discrete MCR is NP-Complete.

Augment $\mathcal{G}_n$ with two unconstrained edges $e_s = (s', s)$ and $e_g = (g, g')$. Construct, through polynomial steps, an edge-to-vertex dual graph $\bar{\mathcal{G}_n}(\bar{V}, \bar{E})$ which corresponds vertices in $\mathcal{G}_n$ to edges in $\bar{\mathcal{G}_n}$. Edges that share a vertex in $\mathcal{G}_n$ are connected by an edge in $\bar{\mathcal{G}_n}$. Here the constraint function $C(v)$ now applies a subset of $N$ worlds to the corresponding edges in $\bar{\mathcal{G}_n}$. The MCR problem on $\mathcal{G}_n$ from $s$ to $t$ will return a solution that maps to a path $\pi_{\text{MCR}}$ from $e_s$ to $e_g$ in $\bar{\mathcal{G}_n}$, with some set of constraints out of $N$ constraints.

Let the domain of constraints in $C(v)$ correspond to $N$ world samples $\mathcal{W}^i_{\text{obs}} \sim W$ the MCR path $\pi_{\text{MCR}}$ corresponds to the count of $|\mathcal{F}(\pi_{\text{MCR}})|$, and thereby the discretized probability of collision $\text{Pr}_{\text{col}}(\pi_{\text{MCR}}) = \frac{|\mathcal{F}(\pi_{\text{MCR}})|}{N}$. The decision version to certify a solution existing with $k$ constraints represents the decision version of $\text{Pr}_{\text{col}}(\pi_{\text{MCR}}) = \frac{k}{N}$. This is shown in previous work [9] to be NP-Complete. ■

**Algorithm 1:** SEARCH+MILP

**Input:** Number of configuration samples $n$, Number of world samples $N$, start $x_s$, goal $x_g$

**Output:** Path $\pi_{n,N}$

// Preprocess

$\mathcal{F}_N \leftarrow$SAMPLEWORLDS($N$);

$\mathcal{G}_n \leftarrow$SAMPLEROADMAP($n$);

**for** $e \in \mathcal{G}_n.E$ **do**

  $\mathcal{G}_n \leftarrow$UPDATEEDGE($\mathcal{G}_n, e, \mathcal{F}_N$);

// Query

$\pi_{n,N} \leftarrow$GREEDYSEARCH($\mathcal{G}_n, x_s, x_g$);

// Optimal MILP

$\pi_{n,N} \leftarrow$SEEDEDOPTMILP($\mathcal{G}_n, \mathcal{F}_N, x_s, x_g, \pi_{n,N}$);

**return** $\pi_{n,N}$

## III. METHOD

To study collision probability computation for the purpose of motion planning under environment uncertainty, we need to devise a method that can solve the problem presented in Def 7. In this section, we introduce the algorithm to solve the motion planning problem in uncertain environments. The input of the algorithm is a start configuration $x_s$ and goal configuration $x_g$, as well as the number of configuration samples $n$ we would like to construct in the roadmap, and the number of world samples $N$ which is to model the uncertainty of the environment. The output of the algorithms is a path from the start configuration to the goal configuration which has optimal collision probability.

Algo 1 starts by sampling a set of worlds based on the number of world samples and constructing a roadmap based on the input number of configuration samples. Each edge in the roadmap is updated with the collision probability in the sampled worlds. The explanation of collision probability is from Definition 4. After preprocessing, a greedy search algorithm (similar to the best first variant in MCR [9]) is employed to find a good enough approximated path from the start state to the goal state using the updated roadmap. This path serves as a seed for refinement. The algorithm then utilizes a Mixed Integer Linear Programming (MILP) model to optimize the path to find the optimal collision probability. Finally, the optimized path is returned as the solution.

### A. MILP Model

A Mixed Integer Linear Program model is designed to find the optimal solution to the minimum collision probability in the discrete case over a roadmap with $n$ vertices and $N$ sampled world realizations. The graph $\mathcal{G}_n$ is augmented with a loop edge $(x_s, x_g)$ connecting the start and goal vertex. The model is described as follows.

$$\min \sum_{w \in \mathcal{F}_N} x_w \cdot M + \sum_{e \in \mathcal{G}_n.E} x_e \qquad \text{[A]}$$

subject to

$$x_e \in \{0,1\} \qquad \forall e \in \mathcal{G}_n.E \quad \text{[B]}$$

$$x_w \in \{0,1\} \qquad \forall w \in \mathcal{F}_N \quad \text{[C]}$$

$$i_v \in \{0,1\} \qquad \forall v \in \mathcal{G}_n.V \quad \text{[D]}$$

$$2 \cdot (1 - i_v) + \sum_{e \in \texttt{adj}(v)} x_e = 2 \qquad \forall v \in \mathcal{G}_n.V \quad \text{[E]}$$

$$M \cdot (1 - x_w) + \sum_{e \in \texttt{wTOe}(w)} x_e \leq M \qquad \forall w \in \mathcal{F}_N \quad \text{[F]}$$

$$x_{(x_s, x_g)} = 1 \qquad \text{[G]}$$

Three binary variables are defined in the above equations [B] [C] [D]. Variable $x_e$ represents if the edge $e$ in graph $\mathcal{G}_n$ is selected, $x_w$ represents if the world $w$ collides with any of the selected edges, $i_v$ indicates if the vertex $v$ in graph $\mathcal{G}_n$ is selected. The constraint [E] indicates that each node $v$ can either have 0 or 2 edges and when it has 0 edges, the node $v$ is not selected, and when it is selected it must have 2 edges. Disconnected loops are avoided through an edge count residual in the optimization objective. The constraint [F] connects the variable $x_w$ and $x_e$ and indicates that if the world $w$ collides with any edges, then for the corresponding $x_e$ for each edge $e$ within the collision set must be active. If the world $w$ has no collision with any of the edges in the graph $\mathcal{G}_n$, then no $x_e$ be selected. The number $M$ is assigned to some large constant. The constraint [G] is to connect the start node $x_s$ with $x_g$ and activate the synthetic loopback edge $(x_s, x_g)$. The objective of the optimization process is given by the equation [A]. It is to minimize the total number of colliding worlds (as well as as secondary objective of minimizing the number of edges). The secondary objective[2] serves the purpose of avoiding disconnected loops from the path continuity constraint [E].

### B. Analysis

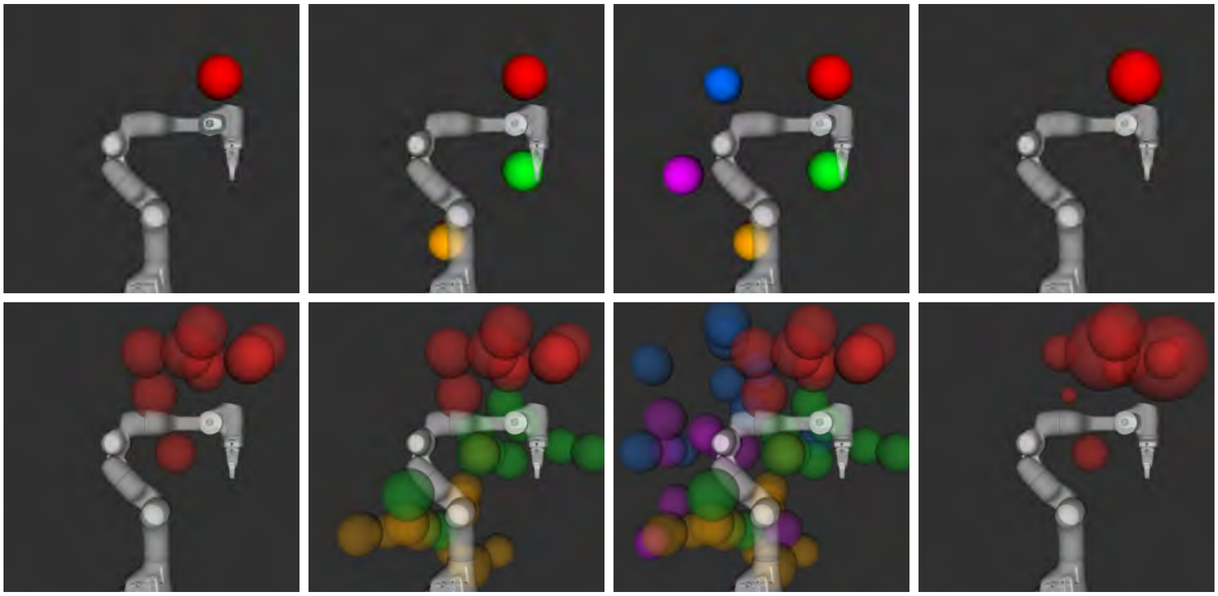
*Assumption 2:* For all small enough $\epsilon_\delta > 0$, there exists some $\delta > 0$ such that an optimal path that minimizes the probability of collision $\pi^*$ is strongly $\delta$-clear. It is assumed that any near-optimal path contained in the clearance region $\pi^\delta$ has arbitrarily low error in collision probability, $|\mathrm{Pr}_{\mathrm{col}}(\pi^\delta) - \mathrm{Pr}_{\mathrm{col}}(\pi^*)| \leq \epsilon_\delta$.
The assumption describes some necessary smoothness for sampling to work. There must exist nearby smooth paths around an optimal path that will be arbitrarily close to the optimal collision probability.

*Theorem 4 (Algo 1 is asymptotically optimal):* As $n$ and $N$ both asymptotically increase, Algo 1 will return paths $\pi_{n,N}^{\mathrm{ALG}}$ that will differ with arbitrarily small error from the optimal collision probability.

$$\lim_{n,N \to \infty} |\mathrm{Pr}_{\mathrm{col}}(\pi_{n,N}^{\mathrm{ALG}}) - \mathrm{Pr}_{\mathrm{col}}(\pi^*)| \leq \epsilon \;\forall \epsilon > 0 \qquad (12)$$

*Proof:* [Sketch] Thm. 1 from asymptotic optimality of roadmaps constructed from random geometric graphs guarantee that all smooth paths will be part of $\mathcal{G}_n$ asymptotically.

[2]It is a straightforward extension to make the secondary objective the sum of edge path lengths. This makes potentially makes the model harder and we choose the simpler version for the purposes of this study.



**1 obstacle** **3 obstacle** **5 obstacle** **1 obs + size**

Fig. 2: The figure shows the four benchmark settings, (*from left to right*) with 1, 3, and 5 obstacles, whose positions are uncertain, and the fourth benchmark has 1 obstacle whose position and size are uncertain. The first row visualizes one possible world sample. The second row shows the environment uncertainty with 10 world samples differing in position (left three), position and size (rightmost), with each color representing the same obstacle from different world samples.

Assm 2 guarantees that there exists a clearance region around an optimal path with an arbitrarily near-optimal solution. Asymptotically optimal sampling-based roadmaps will approximate the smooth near-optimal paths with arbitrarily small error asymptotically in $n$. These paths are guaranteed to possess collision probabilities with arbitrarily low error.

It still needs to be shown that approximation necessary from sampling $\mathcal{W}^1_{\text{obs}} \cdots \mathcal{W}^N_{\text{obs}} \sim W$ guarantees that optimizing for the approximate collision probability will discover the near-optimal solutions described above. This follows from Thm 4. As $N \to \infty$ the error in the measure of probability will be arbitrarily small. Then, as $n, N \to \infty$, the graph $\mathcal{G}_n$ will contain a near-optimal path with arbitrarily low error to the optimal collision probability and the environment sampling results in an arbitrarily small error in the approximation of the probability of collision. Since Algo 1 reports the optimal solution on $\mathcal{G}_n$, asymptotic optimality follows. ∎

### C. Implementation Details and Incremental Variant

Once we have the sampled worlds and the roadmap, the algorithm goes through all the edges and calculates the collision probability of each edge by counting *the set* of sampled worlds colliding with the edge. The computation of the path needs to account for the sets (their cardinality) and this is a primary reason why graph search by itself cannot overcome the lack of optimal substructure. While the MILP model by itself can compute the optimal solution with the current $n, N$-discretization, seeding it with a suboptimal solution is more effective.

Algo 1 involves a preprocessing stage which needs to collision check all edges of the roadmap versus world samples, before queries can be resolved. An incremental variant is implemented to search optimization in contrast to the preprocessing-intensive nature of the first version. It begins with a roadmap collision-checked against one sample of the environment and then incrementally adds information from $N$ worlds only on the solutions obtained from search. The search can be repeated till there is no change in the optimal path. The discretization of collision checking is initialized to be sparse but can be increased to the finest resolution. The suboptimal solution can be seeded to MILP as in Algo 1.

## IV. Results

This section will present a few empirical indications of *whether the choice of problem formulation in Sec II and solutions discovered out of sampling-based solvers 1 has consequences on measured probability of collision* in high-DoF motion planning problems in uncertain environments.

All experiments were performed on a $13^{th}$ Gen Intel i9 machine with 128GB RAM on single-threaded processes for each algorithmic step. Our implementation utilizes the OMPL (Open Motion Planning Library) [23] framework, as well as the ROS2 (Robot Operating System 2) [24] and MoveIt2 [25]. The MILP model has been implemented in Gurobi [26]. The roadmap is based on a PRM* [18].

***Benchmarks:*** As shown in Fig 2, all experiments are conducted in simulation using a 7-DoF Franka Panda arm. (**1/3/5 obstacles**) benchmarks sets up 1, 3, or 5 spherical obstacles of radius 0.1m. Here the position of the center of each sphere(s) is uncertain. For the (**1 obs + size**) benchmark we vary both the position of the center as well as the radius of the spheres.

Next we need a way to generate uncertain environments in the benchmarks to run experiments. To generate the

| | max | sum | search + milp | search + milp | search + milp | search + milp |
|---|---|---|---|---|---|---|
| | | | 10 worlds | 50 worlds | 100 worlds | 200 worlds |
| **1 obstacle** | 0.3280 (±0.1952) | 0.2130 (±0.1710) | 0.2803 (±0.1753) | 0.2109 (±0.1470) | 0.1955 (±0.1386) | **0.1704** (±0.1250) |
| **3 obstacle** | 0.7397 (±0.1767) | 0.5669 (±0.2022) | 0.5740 (±0.1726) | 0.5190 (±0.1746) | 0.4920 (±0.1606) | **0.4345** (±0.1512) |
| **5 obstacle** | 0.8476 (±0.1401) | 0.7172 (±0.1806) | 0.7631 (±0.1422) | 0.6646 (±0.1560) | 0.6527 (±0.1553) | **0.5895** (±0.1567) |
| **1 obs + size** | 0.4498 (±0.2131) | 0.3132 (±0.2007) | 0.3839 (±0.2236) | 0.2989 (±0.1694) | 0.2799 (±0.1715) | **0.2666** (±0.1690) |

TABLE I: Probability of collision (± std) averaged over 50 experiments each for benchmarks with 1, 3, 5 obstacles, and 1 obstacle+size using the max, sum optimization baselines. Algo 1 (search + milp) uses information from 10, 50, 100, and 200 world samples. The best performance is highlighted in **bold**.

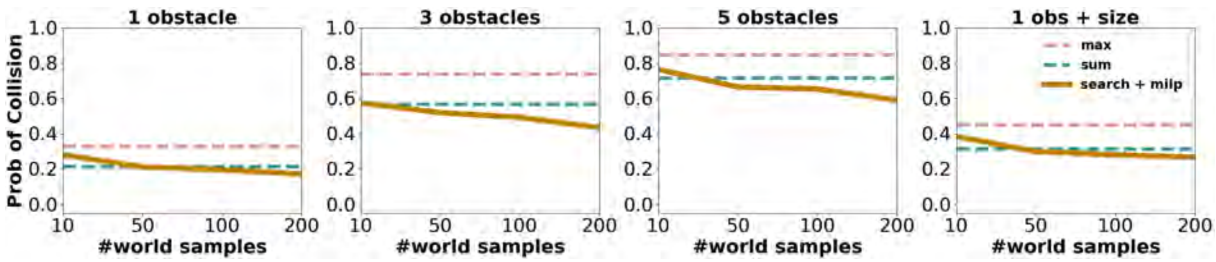



Fig. 3: This figure visually illustrates the trend of Algo 1 (search + milp) improving in the probability of collision as the number of samples increases. The dotted lines are the baselines (max and sum) using 200 world samples.

| | 1 obs | 3 obs | 5 obs | 1 obs + size |
|---|---|---|---|---|
| greedy search | 0.1712 (±0.1268) | 0.4367 (±0.1522) | 0.5931 (±0.1584) | 0.2670 (±0.1694) |
| search + milp | 0.1704 (±0.1250) | 0.4345 (±0.1512) | 0.5895 (±0.1567) | 0.2666 (±0.1690) |
| incr search + milp | 0.1719 (±0.1270) | 0.4449 (±0.1588) | 0.5980 (±0.1589) | 0.2725 (±0.1685) |

TABLE II: Probability of collision (±std) averaged over 50 experiments each for benchmarks with 1, 3, 5 obstacles, and 1 obstacle+size to evaluate variants of Algo 1, only running greedy search, running search and MILP using 200 worlds, and an incremental version.

experiments, we use Gaussians (one stddev is visualized in Fig 2) to model the position (and size) uncertainty. Note that across the benchmarks, none of the evaluated algorithms will be informed about the underlying distribution of individual obstacles, beyond having access to sampling fully defined worlds. Note that the consideration of Gaussians is only relevant for informing the world sampler, which is treated as a black box within all the compared methods in all the experiments. For each benchmark we create 50 different planning problems under environment uncertainty.

- For each of the benchmarks we randomly sample the mean(s) of the position uncertainty Gaussians for each of the obstacles. We generate 10 different environment uncertainty distributions per benchmark type. The standard deviation of these Gaussians is 0.2m for the position of the centers. We sample the radius by using Gamma distribution with $r \sim \frac{\Gamma(6,1)}{40}$ (this has a mean radius of 0.15m). This is to demonstrate that the algorithm we propose does not depend on the type of the underlying distribution of environment uncertainty.
- For each environment uncertainty condition we randomly generate one 5000-node PRM* roadmap. There are a total of 40 roadmaps with each of **1/3/5 obstacles**,**1 obs + size** having 10 roadmaps.
- For each environment uncertainty distribution (and roadmap) we randomly sample five motion planning problems (start-goal configurations).

We then run the total of 50 motion planning problems described in Fig 2 and report the data.

***Metrics:*** The key reported metric is the probability of collision of solution paths to motion planning problems under environment uncertainty. The reported numbers are calculated by sampling 200 worlds from the environment uncertainty distribution, collision checking the path against each world sample, and reporting the fraction of worlds in collision out of 200.

***Comparisons:*** For each experiment, we primarily evaluate Algo 1 against comparison points — a best first search over the roadmap minimizing two other cost functions.

- **search+milp**: This corresponds to Algo 1 with a time budget of 30s. It first executes the search then runs the MILP solver for the remainder of 30s.
- **sum**: inspired by union bounds that are applied to probabilities in literature minimizes the sum of collision probabilities of configurations along the solution path. Per configuration collision probability values are calculated wrt 200 world samples though notably this cost function within the search only uses the probability value and no information about which worlds collided.
- **max**: focuses on the minimization of the maximum collision probability of configurations along the solution path. Like the other baseline, per configuration collision probability values are calculated wrt 200 world samples and only the value is utilized during the search.

We evaluate **search+milp** versions that use different number of world samples (10, 50, 100, and 200). Note that Algo 1 discriminates between the worlds that collide and attempts to minimize that set, and our claim is that the quality of our solution gets better with more information (more world samples). As an ablation we report the performance of **search+milp** after the search step. We report an incremental variant giving 15s each to search and MILP.

### A. Takeaways

The tables are shown in Tab I, II. The numbers report an average and one standard deviation over 50 experiments. Tab I and Fig 3 clearly show that solving the optimal formulation of the problem using Algo 1 (search+milp) **significantly outperforms** using alternative cost functions (max or sum). Also notably, Algo 1 also demonstrates the convergence claims with the increasing number of samples as the **probability of collision improves**. The environment clutter (number of obstacles) and the additional source of size uncertainty increases the probability of collision in the problems, but the gap in the performance between search+milp versus max or sum persists with increasing environment complexity. Fig 3 also visually indicates that the performance of Algo 1 (search+milp) overcomes both max and sum with **relatively few world samples** (less than 50).

We also report a short ablation for Algo 1 (search+milp) using information from 200 worlds. Only running the greedy search step already achieves solution qualities close to the final reported solution probability of collision. Note that the second row (search + milp) is seeded with the solution obtained from the greedy search step and always finds improvements within the allotted 30s. The third row is an incremental variant of the search that also shows promising performance. All of these support that our proposed formulation of reasoning over sets of sampled worlds in a principled way seems to be the key ingredient behind performance.

Building on traditional approaches that characterize asymptotic optimality of roadmaps in terms of their size, here we choose to focus on the novel element in this problem – the properties of the solution as the number of worlds increases. It is of interest to explore the effect of both roadmap size and world samples in future work.

Note that before running the search over the roadmaps for experiments in Tab I and Fig 3, the roadmap has to be preprocessed with collision information for the corresponding number of worlds. This is expensive and takes over one and a half hours per 5000-node roadmap for 200 worlds. This significant precomputation in the multi-query framework forms the proposed initial implementation but notably each query is capped at 30s with Gurobi capable of reporting anytime solutions. Future work will look into improving the efficiency of the roadmap construction running times, and scalability. Nonetheless, in this work the experiments have conclusively demonstrated that *a principled choice of modeling motion planning under environment uncertainty yield lower probabilities of collision*.

## V. Discussion

This work makes a principled attempt at removing the shroud of uncertainty that has encumbered one of the fundamental subroutines in robotic motion planning — collision checking, which, in the presence of environment uncertainty, has caused dissonance in models and solvers in robotics literature. In this work, based on connections drawn in previous work on minimum constraint removal, we highlight that it is important to be considerate in modeling collision probability for motion planning under environment uncertainty and demonstrate that this choice, when done carefully has attractive theoretical properties along with concrete benefits in achieving solutions with better probabilities of collision in high-dimensional problems.

Several open questions remain. Richer forms of uncertainty, including state, action, and observation will be investigated in connection to the current problem. Practical solvers are also of keen interest to the community, especially with achievable finite-time bounds on suboptimality or risk. A set of practical applications like manipulation also introduce scenarios (like contact) where probabilities of collision between specific parts of the scene might become degenerate at 1. This work provides a stepping stone towards recommending a concrete standard for collision probability, both on what is appropriate for the current problem as well as what might be promising for the problems of the future.

## References


[1] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[2] S. M. LaValle, J. J. Kuffner, B. Donald, *et al.*, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and computational robotics: new directions*, vol. 5, pp. 293–308, 2001.

[3] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[4] L. Janson, E. Schmerling, and M. Pavone, "Monte carlo motion planning for robot trajectory optimization under uncertainty," in *Robotics Research: Volume 2*. Springer, 2017, pp. 343–361.

[5] C. Park, J. S. Park, and D. Manocha, "Fast and bounded probabilistic collision detection for high-dof trajectory planning in dynamic environments," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 980–991, 2018.

[6] B. Axelrod, L. P. Kaelbling, and T. Lozano-Pérez, "Provably safe robot navigation with obstacle uncertainty," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1760–1774, 2018.

[7] T. Lew, R. Bonalli, and M. Pavone, "Chance-constrained sequential convex programming for robust trajectory optimization," in *2020 European Control Conference (ECC)*. IEEE, 2020, pp. 1871–1878.

[8] C. Quintero-Pena, A. Kyrillidis, and L. E. Kavraki, "Robust optimization-based motion planning for high-dof robots under sensing uncertainty," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9724–9730.

[9] K. Hauser, "The minimum constraint removal problem with three robotics applications," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 5–17, 2014.

[10] B. Burns and O. Brock, "Sampling-based motion planning with sensing uncertainty," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3313–3318.

[11] T. Summers, "Distributionally robust sampling-based motion planning under uncertainty," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6518–6523.

[12] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 5097.

[13] L. J. Guibas, D. Hsu, H. Kurniawati, and E. Rehman, "Bounded uncertainty roadmaps for path planning," in *Algorithmic Foundation of Robotics VIII: Selected Contributions of the Eight International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2010, pp. 199–215.

[14] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 723–730.

[15] A. Wu, T. Lew, K. Solovey, E. Schmerling, and M. Pavone, "Robust-RRT: Probabilistically-complete motion planning for uncertain nonlinear systems," in *The International Symposium of Robotics Research*. Springer, 2022, pp. 538–554.

[16] S. Dai, S. Schaffert, A. Jasour, A. Hofmann, and B. Williams, "Chance constrained motion planning for high-dimensional robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8805–8811.

[17] H. Kurniawati, "Partially observable markov decision processes and robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 253–277, 2022.

[18] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[19] L. Paiola, G. Grioli, and A. Bicchi, "On the evaluation of collision probability along a path," *arXiv preprint arXiv:2311.08204*, 2023.

[20] A. Krontiris and K. Bekris, "Computational tradeoffs of search methods for minimum constraint removal paths," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 6, no. 1, 2015, pp. 181–185.

[21] E. Eiben, J. Gemmell, I. Kanj, and A. Youngdahl, "Improved results for minimum constraint removal," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[22] R. Wang, C. Mitash, S. Lu, D. Boehm, and K. E. Bekris, "Safe and effective picking paths in clutter given discrete distributions of object poses," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5715–5721.

[23] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[24] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074

[25] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a MoveIT! case study," *arXiv preprint arXiv:1404.3785*, 2014.

[26] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com